Computer Vision and Image Understanding 201 (2020) 103061

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Learning instance-aware object detection using determinantal point processes

Nuri Kim, Donghoon Lee, Songhwai Oh *

Department of Electrical and Computer Engineering, ASRI, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 08826, Republic of Korea

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
41A05
41A10
65D05
65D17

Keywords:
Determinantal Point Processes
Object Detection
Crowd Detection

ABSTRACT

Recent object detectors localize instances and classify candidate regions simultaneously. The number of candidate regions is typically larger than the number of objects and each region is evaluated independently. To assign a single detection bounding box for each object, heuristic algorithms, such as non-maximum suppression (NMS), have been used widely. While simple heuristic algorithms are effective for stand-alone objects, they often fail to detect overlapped objects. In this paper, we address this issue by training a network to distinguish different objects using the relationship between candidate boxes. We propose an instance-aware detection network (IDNet), which can learn to extract features from candidate regions and measure their similarities. Based on pairwise similarities and detection qualities, the IDNet selects a subset of candidate bounding boxes using instance-aware determinantal point process inference (IDPP). Extensive experiments demonstrate that the proposed algorithm achieves significant improvements for detecting overlapped objects compared to existing state-of-the-art detection methods on CrowdHuman, Pascal VOC, and MS COCO datasets.

1. Introduction

Object detection is one of the fundamental problems in computer vision. Its goal is to detect objects by classifying and regressing bounding boxes in an image (Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Redmon and Farhadi, 2017; Liu et al., 2016; Zhang et al., 2018; Choi et al., 2019; Peng et al., 2019; Zhao et al., 2019). It has received much attention because of its wide range of applications, such as object tracking (Wang et al., 2019), surveillance (Gawande et al., 2020), and face detection (Ranjan et al., 2017). Despite the advances in object detection, it is still difficult to assign correct detections when objects are overlapped. Such detection failures happen frequently on 2D images as shown in Fig. 1(a), i.e., a person in the middle and a person on the right are not detected. We attribute this to the fact that existing object detectors focus on predicting object class labels without learning to distinguish different object instances in the same class.

In order to address this issue, we propose an instance-aware detection network (IDNet), which learns to differentiate representations for different object instances. The representation is trained to describe the similarity of object instances in terms of their appearance in detection bounding boxes and spatial arrangements. During inference, the algorithm automatically selects a subset of initial detection bounding boxes as the final detection result based on the mutual similarity. In this way, we can assign detection results for overlapping objects, i.e., if representations of two overlapping bounding boxes have low similarity, then they are more likely representing different object instances.

This approach can be developed on top of existing detectors. In particular, we use a two-stage detector as a baseline detector to obtain candidate bounding boxes. Then, IDNet extracts features of all candidates using a CNN branch. We named this part of IDNet as a region identification network (RIN) since its output feature is learned to be distinguishable for different object instances. Its training signal comes from an instance-aware detection (ID) loss. We formulate the loss based on determinantal point processes (DPPs) (Kulesza and Taskar, 2012) which aims to select the optimal subset from a ground set. It is suitable for object detection as we want to detect objects by selecting correct bounding boxes from all candidate boxes. In order to use DPPs, we need to define two terms: the quality of each detection candidate and similarity between candidates. We let RIN and the ID loss model the similarity between detections. In the following, we discuss how to design the quality of each detection candidate effectively. We observe that detectors frequently report multiple bounding boxes around a single object. For example, as shown in Fig. 1(b), there are two bounding boxes assigned to a dog, which are categorized as a dog and a horse with high confidence scores. Since existing bounding box suppression algorithms, such as NMS, can only remove boxes in the same object class, the dog bounding box cannot eliminate the horse bounding box once its confidence exceeds the threshold. Therefore, it motivated us to refine the detection score. We propose a sparse score (SS) loss to

* Corresponding author.
E-mail addresses: nuri.kim@rllab.snu.ac.kr (N. Kim), donghoon.lee@rllab.snu.ac.kr (D. Lee), songhwai@snu.ac.kr (S. Oh).

https://doi.org/10.1016/j.cviu.2020.103061
Received 5 September 2019; Received in revised form 22 May 2020; Accepted 5 August 2020
Available online 17 August 2020

(a) Missing detection due to overlapped bounding boxes. (b) False positive detection for overlapped objects.

Fig. 1. Detection errors. Detection results from an object detector on Pascal VOC dataset. Incorrect or missing detections are shown in dashed boxes. **(a)** There are two people who are not detected. **(b)** An image with duplicated detections for a single object, where a *horse* is a false positive for a *dog*.

address this problem. It aims to reevaluate detection scores of top-m categories to assign a high quality to the correct bounding boxes and low quality to wrong bounding boxes.

We conduct extensive experiments on three object detection benchmark datasets (CrowdHuman Shao et al., 2018, Pascal VOC Everingham et al., 2010, and MS COCO Lin et al., 2014) with two strong baseline detectors such as Faster R-CNN (Ren et al., 2015) and LDDP (Azadi et al., 2017). The proposed IDNet significantly improves detection accuracy for overlapped objects, e.g., 10% AP improvement on CrowdHuman, as well as overall objects on Pascal VOC (improved by 5.7% mAP) and COCO crowd set (improved by 1.3% mAP).

2. Related work

Class-aware detection algorithms. The goal of class-aware or multi-class object detection is to localize objects in an image while predicting the category of each object. These systems are usually composed of region proposal networks and region classification networks (Girshick, 2015; Ren et al., 2015; Liu et al., 2016). To improve detection accuracy, a number of different optimization formulations and network architectures have been proposed (Ren et al., 2015; Kong et al., 2016; Azadi et al., 2017; Redmon et al., 2016; Liu et al., 2016; Redmon and Farhadi, 2017; Dai et al., 2016b). Ren et al. (2015) use convolutional networks, called region proposal networks, to get region proposals and combine it with Fast R-CNN. Kong et al. (2016) utilizes each layer's feature for detecting small objects in an image. A real-time multi-class object detector is proposed by combining region proposal networks and classification networks in Redmon et al. (2016). Liu et al. (2016) improve the performance of Redmon et al. (2016) using multiple detectors for each convolutional layer. To increase network efficiency, fully connected layers are replaced by convolution layers in Dai et al. (2016b). Redmon and Farhadi (2017) extend (Redmon et al., 2016) by classifying thousands of categories using the hierarchical structure of categories in the dataset.

DPPs have been used to improve detection qualities before. Azadi et al. (2017) propose to suppress background bounding boxes, while trying to select correct detections. However, this method focuses on adjusting detection scores and uses a fixed visual similarity matrix based on WordNet (Miller, 1995), while our algorithm learns the similarity matrix from data.

Instance-aware algorithms. Instance-aware algorithms have been developed to provide finer solutions in different problem domains. Instance-aware segmentation aims to label instances at the pixel level (Dai et al., 2016a; Ren and Zemel, 2017). Dai et al. (2016a) propose a cascade network which finds each instance stage by stage. Similar to RIN, a network in Dai et al. (2016a) finds features of each instance. Note that instance segmentation requires expensive pixel-level annotations. On the other hand, the proposed method improves object detectors based on cheaper bounding box annotations. Ren and Zemel (2017) use a recurrent neural network to sequentially find each instance. A face detector which takes keypoints of faces as input is suggested in Li et al. (2016). The dataset for this application contains face labels for identifying different faces, while the standard object detection datasets only have a small number of categories.

In object detection, Wang et al. (2018) introduce a repulsion loss (RepLoss) to improve localization of instances. However, their approach is limited to a single-class detection problem and uses NMS (Felzenszwalb et al., 2010) as a post-processing method. Lee (Lee et al., 2016) provide an inference method to find an optimal subset of detection candidates for pedestrian detection considering the individualness of each detection candidate. However, this approach tackles a single-class detection problem and uses features computed from a network pre-trained on the ImageNet dataset (Deng et al., 2009), instead of training the network for the desired purpose. Our method tackles a challenging multi-class detection task by learning distinctive features of object instances from data.

Recently, a detector which learns the structural relationship between objects is proposed in Liu et al. (2018), where the detection score of an object is scaled by considering scene context and relationship between objects. Liu et al. (2018) show that training with a structural relationship can implicitly reduce redundant detection boxes, while our method explicitly suppresses the scores of duplicated detection boxes. Hu et al. (2018) utilize a modified attention module from Vaswani et al. (2017) for learning a relationship between bounding boxes. The module scales the scores using the instance relationship similar to ours. However, this method uses the standard cross-entropy loss and smooth L1 loss, while our IDNet tackles this problem by training a detector with novel losses.

3. Backgrounds

3.1. Determinantal point processes

A Determinantal Point Process (DPP) is a point process that defines a probability of a subset proportional to a determinant of a kernel matrix measuring the quality and similarity between a pair of elements. Thus, DPP has a high probability for subsets with qualitative and diverse elements. Since off-diagonal terms are subtracted in the calculation of the probability, the off-diagonal terms mean negative correlations between elements. If two elements are perfectly the same, a negative correlation of two elements is high and two elements might not co-occur. On the other hand, if two elements are independent, a negative correlation between two elements is zero and two elements tend to co-occur (see Kulesza and Taskar, 2012 for details). Because of this property of a DPP, it is used in various machine learning fields, such as document and video summarization (Chao et al., 2015; Zhang et al., 2016; Lin and Bilmes, 2012), sensor placement (Krause et al., 2008), recommendation systems (Zhou et al., 2010) and multi-label classification (Xie et al., 2017), to select a desirable subset from a set of candidates.

3.2. Basic losses in two-stage detectors

A two-stage detector has four loss functions. The two are for detecting regional boxes on the first stage, and the other two are for dividing each region into each class to further refine classification and regression on the second stage. In the first stage, a cross-entropy loss is used to detect whether a region is foreground or background and the localization loss is used to make the box closer to an ground truth object. In the second stage, there are a cross-entropy loss to determine which category the region box belongs to and a localization loss to learn how to further refine the regression of objects in each class.

Suppose the first stage predicts the objectness probability p_i and location shifts l_i, where i is an index of a region. the second stage predicts c_j of a category and a location shift t_j, where j is an index of a

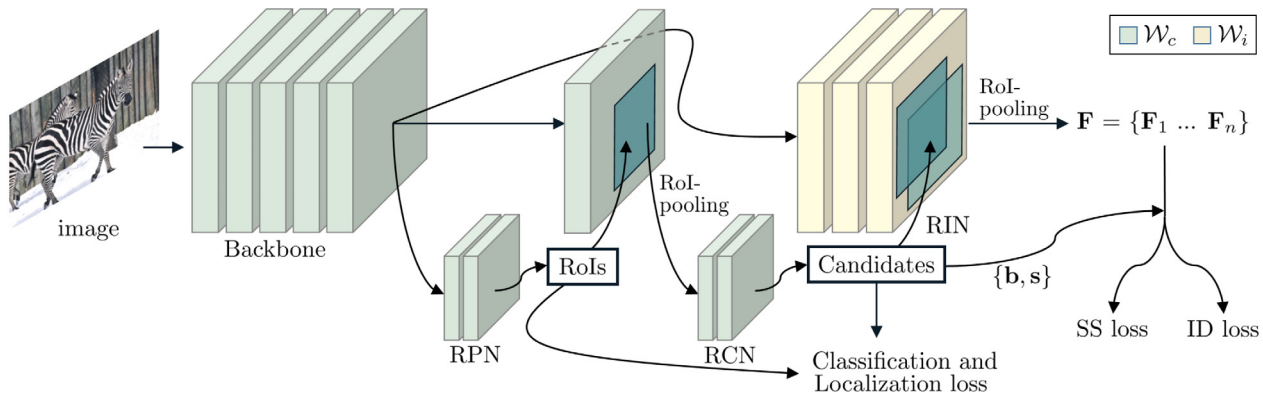


Fig. 2. Pipeline of the instance-aware detection network (IDNet). Let $\mathcal{W}_c$ denote weights for the backbone network, region proposal network (RPN), and region classification network (RCN). We also represent $\mathcal{W}_i$ as weights for the region identification network (RIN). Using the features extracted from RIN (**F**), IoUs calculated from bounding boxes (**b**), and the detection quality (**q**), a probability of a subset of bounding boxes to be selected can be calculated. IDNet is trained with the proposed SS loss and ID loss, as well as a basic classification and localization loss for a detector.

bounding box. Then, a classification and localization loss is expressed as follows:

$$\begin{aligned}&\mathcal{L}_{CL}(\{p_i\},\{l_i\},\{c_j\},\{t_j\}) \\ &= \sum_i \mathcal{L}_b(p_i, p_i^*) + \sum_i \mathbf{1}_{p_i^*>0}\mathcal{L}_r(l_i, l_i^*) \\ &+ \sum_j \mathcal{L}_m(c_j, c_j^*) + \sum_j \mathbf{1}_{c_j^*>0}\mathcal{L}_r(t_j, t_j^*),\end{aligned} \tag{1}$$

where l_i^*, t_j^*, p_i^*, c_j^* are ground truths. Here, $\mathbf{1}_{c_j^*>0}$ is an indicator function, which outputs 1 when the jth bounding box has a foreground label.

4. Proposed method

4.1. Network architecture

An overview of the proposed IDNet is shown in Fig. 2. IDNet is composed of a region proposal network (RPN), a region classification network (RCN) and a region identification network (RIN). Based on image feature maps from the backbone network, RPN predicts region proposals, i.e., the region of interests (RoIs). Then, an RoI pooling layer pools regional features from feature maps for each RoI. Using regional features, RCN classifies the regions into multiple categories while localizing the regions. RIN calculates features to distinguish object instances.

As shown in Table 1, RIN consists of seven convolutional layers, three fully connected layers, three max-pooling layers, and a RoI-pooling layer. Since RIN utilizes parameters of a backbone network, the size of input channel (c_{in}) is chosen according to the backbone network, e.g, 64 for VGG-16 and ResNet-101. The parameters $c_1 = 64, c_2 = 128,$ and $c_3 = 128$ are used for training with CrowdHuman and VOC. For COCO, $c_1 = 128, c_2 = 256,$ and $c_3 = 256$ are used. At the end of each convolutional and fully-connected layer except the last layer has a batch normalization (Ioffe and Szegedy, 2015) and a rectified linear unit (ReLU) in order. We set all convolutional layers to have filters with a size of 3×3 pixels and a stride of one.

4.2. Determinantal point processes for detection

Given a set $\mathcal{Y}$, a DPP aims to increase the probability of sampling a subset $Y \subseteq \mathcal{Y}$ that has high quality and diverse items. It is a useful property for object detection. Let $\mathcal{Y} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, where n is the number of candidate bounding boxes in an image. If $\boldsymbol{Y}$ is a DPP, then the probability of sampling a subset Y is defined as follows:

$$\mathcal{P}_{\mathbf{L}}(\boldsymbol{Y} = Y) = \frac{\det(\mathbf{L}_Y)}{\sum_{Y' \subseteq \mathcal{Y}} \det(\mathbf{L}_{Y'})} = \frac{\det(\mathbf{L}_Y)}{\det(\mathbf{L}+\mathbf{I})}, \tag{2}$$

Table 1
RIN architecture.

Layer	Type	Parameter	Remark
0	Convolution	$c_{in} \times 3 \times 3 \times c_1$	stride 1
1	Convolution	$c_1 \times 3 \times 3 \times c_1$	stride 1
2	Convolution	$c_1 \times 3 \times 3 \times c_2$	stride 1
3	Convolution	$c_2 \times 3 \times 3 \times c_2$	stride 1
4	Max pooling	–	size 2 × 2, stride 2
5	Convolution	$c_2 \times 3 \times 3 \times c_3$	stride 1
6	Convolution	$c_3 \times 3 \times 3 \times c_3$	stride 1
7	Convolution	$c_3 \times 3 \times 3 \times c_3$	stride 1
8	RoI-pooling	–	size 15 × 15
9	Fully connected	$(15^2 \cdot c_3) \times 1000$	–
10	Fully connected	1000 × 1000	–
11	Fully connected	1000 × 256	–

where $Y \subseteq \mathcal{Y}$, a kernel matrix $\mathbf{L} \in \mathbb{R}_+^{n \times n}$ is a real symmetric positive semi-definite matrix, an indexed kernel matrix $\mathbf{L}_Y \in \mathbb{R}_+^{|Y| \times |Y|}$ is a submatrix of $\mathbf{L}$ indexed by the elements of Y, and $\mathbf{I} \in \mathbb{R}^{n \times n}$ is an identity matrix. The kernel matrix is defined as $\mathbf{L} = \mathbf{S} \odot \mathbf{q}\mathbf{q}^T$, where $\mathbf{S} \in \mathbb{R}^{n \times n}$ is a similarity matrix, $\mathbf{q} \in \mathbb{R}^n$ is a quality vector, and $\odot$ is the element-wise multiplication. IDNet is trained to maximize Eq. (1) when Y is ground truth bounding boxes.

For calculating DPP probability, a quality vector of bounding boxes and similarity matrix between bounding boxes are required. We define the quality vector as $\mathbf{q} = (1 - \alpha)\ \mathbf{s} + \alpha$, where $\mathbf{s}$ is a detection score and $\alpha = 0.25$. It changes the range of $\mathbf{s}$ from $[0, 1]$ to $[\alpha, 1/\alpha]$, since a determinant of a large kernel matrix can be too big or too small, which causing learning error. For $\mathbf{S}$, we use the appearance of the candidate boxes and the spatial arrangement between boxes. The appearance can be represented by the feature $\mathbf{F}$ extracted by RIN. For the spatial arrangement, we use $\mathbf{IoU}_{ij}$ between two boxes $\mathbf{b}_i$ and $\mathbf{b}_j$. Then, the similarity matrix can be defined as $\mathbf{S} = \lambda \bar{\mathbf{F}}\bar{\mathbf{F}}^T + (1 - \lambda)\mathbf{IoU}$, where $\lambda \in [0, 1]$, $\bar{\mathbf{F}}$ is a normalized feature and $\mathbf{IoU}$ is a matrix defined as $[\mathbf{IoU}]_{ij} = \mathrm{IoU}_{ij}$. This way of defining the similarity matrix is inspired by Lee et al. (2016). We summarize above notations in Table 2.

4.3. Learning detection quality

As region classification network (RCN) classifies each RoI independently, multiple detections with different categories often have high detection scores. For example, as shown in Fig. 1(b), a detector might report a *horse* nearby a *dog* as they are visually similar. In this case, we want to suppress the horse detection as it is wrong. However, it is difficult to do this for existing bounding box suppression methods, such as NMS, since they do not suppress boxes in different categories. To alleviate this issue, we propose a sparse score (SS) loss to detect

Table 2
Notations in this paper..

Notation	Definition	Description
RoIs	–	Region of interest boxes which are proposed from RPN.
$\mathbf{b}$	–	Candidate bounding boxes which are proposed from RCN.
$\mathbf{IoU}_{ij}$	$\#(\mathbf{b}_i \cap \mathbf{b}_j)/\#(\mathbf{b}_i \cup \mathbf{b}_j)$	Intersection over union (IoU) of two bounding boxes.
$\mathbf{s}$	–	Detection score corresponding to the candidate bounding boxes.
$\mathbf{q}$	$\mathbf{q} = (1-\alpha)\,\mathbf{s} + \alpha$	Quality vector of candidate bounding boxes.
$\bar{\mathbf{F}}_i$	$\mathbf{F}_i/\|\mathbf{F}_i\|_2$	Normalized feature of a bounding box i.
$\mathbf{S}_{ij}$	$\lambda \mathbf{V}_i \mathbf{V}_j^T + (1-\lambda)\mathbf{IoU}_{ij}$	Similarity between box i and j. $0 < \lambda < 1$.
$\mathbf{L}$	$\mathbf{S} \odot \mathbf{q}\mathbf{q}^T$	Kernel matrix of DPPs.

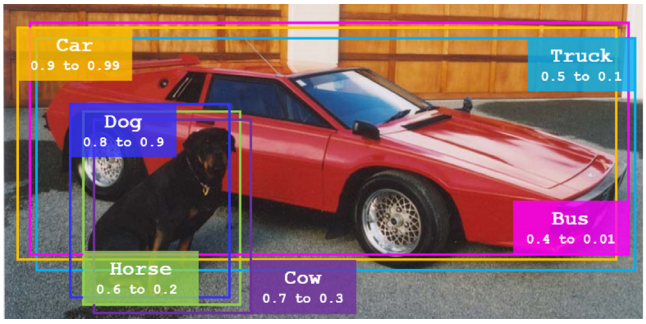


Fig. 3. An example image when the **sparse score (SS)** loss is applied, when $m = 3$. Object detectors output multiple candidate boxes with different category labels around a single object. By applying the SS loss, we can increase the score of correct categories, such as dog and car, while decreasing that of wrong categories, such as horse, cow, truck, and bus.

an object with the correct class label by removing the other candidate boxes with incorrect categories.

We first select categories with top-m detection scores among n_c categories from each RoI. Let $\mathcal{Y}_m$ be all bounding boxes of top-m categories from all RoIs and Y_{pos} be a set of positive boxes, i.e., bounding boxes with a top-1 category in each RoI. For example, as shown in Fig. 3, when there are three bounding boxes ($m = 3$) with different category labels for a single instance, we can put a collection of bounding boxes with the correct category label to a positive set, which contains a car and a dog. The SS loss increases the detection scores of a car and a dog, while decreasing detection scores of a horse, a cow, a truck and a bus. To this end, the SS loss is defined as the negative log-likelihood of the probability choosing the correct bounding boxes among top-m bounding boxes using eq. (2):

$$\begin{aligned}&\mathcal{L}_{SS}(Y_{pos}, \mathcal{Y}_m) \\ &= -\log\left(\sum_{Y \subseteq Y_{pos}} \mathcal{P}_{\mathbf{L}_{\mathcal{Y}_m}}(Y)\right) = -\log\left(\sum_{Y \subseteq Y_{pos}} \frac{\det(\mathbf{L}_Y)}{\det(\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})}\right) \\ &= -\log\det(\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}}) + \log\det(\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m}),\end{aligned} \tag{3}$$

where $\sum_{Y \subseteq Y_{pos}} \det(\mathbf{L}_Y) = \det(\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}})$. Since we need a similarity matrix of elements to calculate a DPP kernel matrix, $\mathbf{L}$, we extract features from RIN, which is fixed while learning detection quality.

With all losses defined above, the weights for a backbone, a region proposal network (RPN), and a region classification network (RCN), which are denoted by $\mathcal{W}_c$ in Fig. 2, can be learned by optimizing[1]

$$\min_{\mathcal{W}_c} \lambda_{ss}\mathcal{L}_{SS}(Y_{pos}, \mathcal{Y}_m) + \mathcal{L}_{\text{CL}}(\{p_i\}, \{l_i\}, \{c_i\}, \{t_i\}), \tag{4}$$

where λ_{ss} is used to balance the SS loss with the classification and localization loss.

[1] The gradients of the sparse score (SS) loss are derived in Appendix A.2.

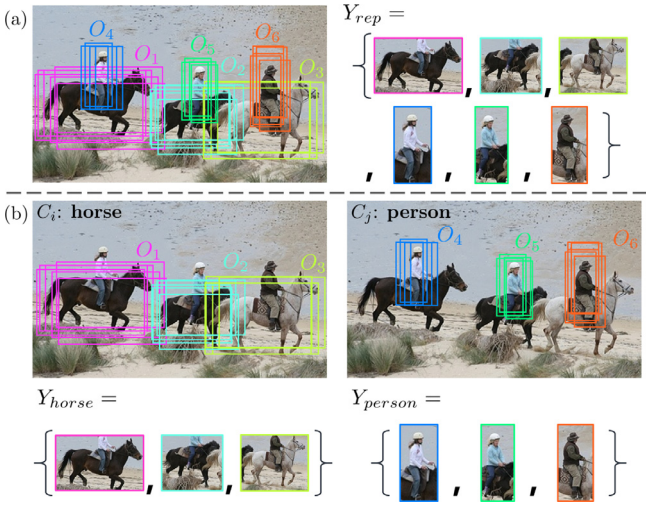


Fig. 4. Example images when the **instance-aware detection (ID)** loss is applied. **(a)** The all-object ID loss have all objects for the representative boxes and increases the DPP probability of the representative set. This makes the determinant of the subset higher, which makes the feature of each instance further in the feature space. **(b)** In the same way, the category-specific ID loss makes the feature for each instance in a category different.

4.4. Learning to distinguish object instances

An instance-agnostic detector solely based on object category information often fails to detect objects in proximity. For accurate detections from real-world images with frequent overlapping objects, it is crucial to distinguish different object instances. To address this problem, we propose the instance-aware detection (ID) loss. The objective of this loss function is to obtain similar features from the same instance and different features from different instances. This is done by maximizing the probability of a subset of the most distinctive bounding boxes.

Let $\mathcal{Y}_s$ be a set of all candidate bounding boxes which intersect with the ground truth bounding boxes. Let $Y_{rep} \subseteq \mathcal{Y}_s$ be a set of the most representative boxes, i.e., candidate boxes which are closest to the ground truth boxes obtained by the Hungarian algorithm (Kuhn, 1955). For example, in Fig. 4(a), there are six instances consisting of three people and three horses. Increasing the DPP probability of the representative set, consisting of a bounding box for each instance, makes the instances in the image to be repulsive in a feature space, which makes a distinct feature for each instance. Then, the ID loss for all objects is defined as follows:

$$\begin{aligned}\mathcal{L}_{ID}^{all}(Y_{rep}, \mathcal{Y}_s) &= -\log(\mathcal{P}_{\mathbf{L}_{\mathcal{Y}_s}}(Y_{rep})) \\ &= -\log\det(\mathbf{L}_{Y_{rep}}) + \log\det(\mathbf{L}_{\mathcal{Y}_s} + \mathbf{I}_{\mathcal{Y}_s}).\end{aligned} \tag{5}$$

Due to the determinant, it decreases the cosine similarity between V_i and V_j if i and j are from different instances. As we select boxes nearby the ground truth bounding boxes to construct $\mathcal{Y}_s$, the network can learn what bounding boxes are similar or different.

In addition to , we set an objective which focuses on differentiating instances from the same category. For category C_k, $\mathcal{Y}_{C_k}$ is candidate

boxes in the kth category and $Y_{C_k} \subseteq \mathcal{Y}_{C_k}$ is a set of candidate boxes which are closest to the ground truth boxes. Y_{C_k} is also obtained by the Hungarian algorithm (Kuhn, 1955). The category-specific ID loss is defined as follows:

$$\begin{aligned}&\mathcal{L}_{ID}^{cs}(Y_{C_k}, \mathcal{Y}_{C_k}) \\ &= -\log(\mathcal{P}_{\mathbf{L}_{\mathcal{Y}_{C_k}}}(Y_{C_k})) \\ &= -\log\det(\mathbf{L}_{Y_{C_k}}) + \log\det(\mathbf{L}_{\mathcal{Y}_{C_k}} + \mathbf{I}_{\mathcal{Y}_{C_k}}).\end{aligned} \tag{6}$$

It provides an additional guidance signal to train the network since it is more difficult to distinguish similar instances from the same category than instances from different categories. After applying category-specific ID loss, objects in each category, Y_{C_k}, have different features because of the repulsiveness of DPPs (see Fig. 4(b)). We find an improvement when we use both of all-object ID loss and category-specific ID loss, compared to cases when only one of them is used. Finally, the ID loss is defined by combining the all-object ID loss and the category-specific ID loss:

$$\begin{aligned}&\mathcal{L}_{ID}(Y_{rep}, \mathcal{Y}_s, Y_{C_k}, \mathcal{Y}_{C_k}) \\ &= \mathcal{L}_{ID}^{all}(Y_{rep}, \mathcal{Y}_s) + \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{ID}^{cs}(Y_{C_k}, \mathcal{Y}_{C_k}).\end{aligned} \tag{7}$$

The goal of the ID loss is to find all instances while discriminating different instances. While the ID loss aims to distinguish instances, the classification and localization loss tries to classify categories. The difference between their goals makes a network perform worse when both losses are used simultaneously. To alleviate the problem, we trained weights of RIN ($\mathcal{W}_i$ in Fig. 2) separate from $\mathcal{W}_c$. Therefore, the detection quality ($\mathbf{q}$) is fixed while training RIN. Given a set of candidate bounding boxes and subsets of them, weights of RIN can be learned by optimizing[2]

$$\min_{\mathcal{W}_i} \mathcal{L}_{ID}(Y_{rep}, \mathcal{Y}_s, Y_{C_k}, \mathcal{Y}_{C_k}). \tag{8}$$

4.5. Inference

Given a set $\mathcal{Y}$ of candidate bounding boxes, the similarity matrix $\mathbf{S}$ and the detection quality $\mathbf{q}$, Algorithm 1 (IDPP) finds the most representative subset of bounding boxes. The problem of finding a subset that maximizes the probability is NP-hard (Kulesza and Taskar, 2012). Fortunately, due to the log-submodular property of DPPs (Kulesza and Taskar, 2012), we can approximately solve the problem using a greedy algorithm, such as Algorithm 1, which iteratively adds an index of a detection candidate until it cannot make the cost of a new subset higher than that of the current subset (Azadi et al., 2017), where the cost of a set Y is $\log(\prod_{i\in Y}\mathbf{q}_i^2 \cdot \det(\mathbf{S}_Y))$.

Algorithm 1: Instance-Aware DPP Inference (IDPP).

$Y^* = \emptyset$
while $\mathcal{Y} \neq \emptyset$ **do**
  $j^* = \text{argmax}_{j\in\mathcal{Y}} \log(\prod_{i\in Y^*\cup\{j\}}\mathbf{q}_i^2 \cdot \det(\mathbf{S}_{Y^*\cup\{j\}}))$
  $Y = Y^* \cup \{j^*\}$
  if Cost(Y) > Cost(Y^*) **then**
    $Y^* \leftarrow Y$
    delete j^* from $\mathcal{Y}$
  else
    return Y^*
  end if
end while
return Y^*

5. Experiments

Datasets and baseline methods. We comprehensively evaluated IDNet on CrowdHuman (Shao et al., 2018), Pascal VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014), which include a single, 20 and 80 categories, respectively.

To demonstrate that our IDNet is effective for detecting overlapped objects, we carry out experiments on CrowdHuman dataset (Shao et al., 2018), which is a recent benchmark dataset for detecting humans in a crowd. It is a larger and more crowded dataset than KITTI and COCO Persons datasets. For example, the number of persons per image is 0.63 for KITTI, 4.01 for COCO Persons, and 22.64 for CrowdHuman dataset (Shao et al., 2018). Since the number of occluded people is significantly higher than other datasets, making it a suitable dataset for evaluating detection systems in crowd situations. For training, CrowdHuman contains 15,000 images and 4,370 images for validation. Since the annotation file for test images is not publicly available, we choose to use validation set to test our method.

Since CrowdHuman has a single class, for showing that the proposed method can apply on multi-class object detection, we have constructed crowd sets for VOC and COCO dataset, VOC `crowd` from the VOC 07 `test` and COCO `crowd` from COCO `val`, respectively. The crowd sets contain at least one overlapped object in an image. Unless otherwise specified, we define overlapped objects as those who overlap with another object over 0.3 IoU in all experiments. The number of images in VOC `crowd` is 283 and COCO `crowd` consists of 5,471 images. The indices of crowd sets will be made publicly available. Since the goal of our algorithm is to discriminate instances with given candidate bounding boxes, we adopt Faster R-CNN as a proposal network to get candidate detections, but other proposal networks can be used in our framework. We implement baseline methods, Faster R-CNN (Ren et al., 2015), RepLoss (Wang et al., 2018), and LDDP (Azadi et al., 2017) to compare with our algorithm. Since there are few methods tested on the crowd sets, we choose the baselines for a fair comparison. Note that our baseline implementation achieves a reasonable performance of 71.0% mAP when trained with VOC 07 using VGG-16 as a backbone, considering that the performance in the original paper, Ren et al. (2015), is 69.9% mAP.

We use different inference algorithms for each method. Unless otherwise stated, Faster R-CNN and RepLoss use NMS with overlap threshold 0.3, LDDP uses LDPP, and IDNet uses IDPP as an inference algorithm. LDPP is an inference algorithm proposed in LDDP (Azadi et al., 2017), which uses a fixed class-wise similarity matrix while our IDPP uses the instance-aware features extracted from RIN.

Implementation details. All baseline methods and our IDNet are implemented based on the Faster R-CNN in Tensorflow (Abadi et al., 2016), where the most parameters, such as a learning rate, optimizer, data augmentation strategy, and batch size, are the same as the original paper, Ren et al. (2015). In our method, we use backbone networks, e.g., VGG-16 and ResNet-101, pre-trained on the ImageNet (Deng et al., 2009) and RIN module is initialized with Xavier initialization (Glorot and Bengio, 2010). RIN shares the parameters in a backbone, such as the layers until the `conv2` of VGG-16 (Simonyan and Zisserman, 2014) and the `conv1` of ResNet-101 (He et al., 2016), to conserve memory. We set m to five for the VOC and ten for COCO, since VOC has around five categories in the super-category and COCO has ten categories in the super-category on average. We set the ratio between the spatial similarity and visual similarity (λ) to 0.6, which is a similar value compared with Zhang et al. (2016) and Lee et al. (2016). Since the performance of a detector is poor during the early stage of training, top-m bounding boxes do not contain similar categories. Thus, we set λ_{ss} to zero during the early stage of training. λ_{ss} is increased to 0.01 after the early stage. The early stages are chosen around 60% of total training iterations. We use 40k iterations for CrowdHuman and VOC 07, 70k for VOC 0712, and 360k for COCO. Additionally, we set the dimension of $\mathbf{F}_i$ to 256 as it performs the best.

[2] The gradients of the instance-aware detection (ID) loss are derived in Appendix A.1.

Table 3
Detection results on CrowdHuman `val`.

Method	Inference	mAP				
		$\texttt{crowd}_3$	$\texttt{crowd}_4$	$\texttt{crowd}_5$	$\texttt{crowd}_6$	$\texttt{crowd}_7$
# of images		4370	3879	3143	2087	1,052
Faster R-CNN (Ren et al., 2015)	NMS	52.0	51.8	51.1	44.4	44.2
RepLoss (Wang et al., 2018)	NMS	52.2	52.0	51.5	48.4	44.2
LDDP (Azadi et al., 2017)	LDPP	52.9	52.8	52.5	52.0	51.4
IDNet	IDPP	**58.9**	**56.3**	**55.8**	**54.9**	**54.2**

Evaluation metrics. For evaluation, we use the mean average precision (mAP). We report mAP which considers detection candidates over IoU 0.5 as correct objects for CrowdHuman and VOC. For COCO, we evaluate performance with three types of mAPs in standard MS COCO (Lin et al., 2014) protocols: AP, AP_{50}, and AP_{75}. AP reports the average values of mAP at ten different IoU thresholds from .5 to .95, AP_{50} reports mAP at IoU 0.5, and AP_{75} reports mAP at IoU 0.75. A high score in AP_{75} requires better localization of detection boxes.

5.1. CrowdHuman

We train IDNet with CrowdHuman for showing that our IDNet with the corresponding inference method (IDPP) is effective on hard occlusion cases. Since CrowdHuman dataset only has a single category, there are no detection errors from confusing categories. Therefore, the sparse score loss is always zero and does not affect learning.

For experiments, we train IDNet with VGG-16 backbone and use the SS loss with the classification and localization loss for 70k iterations and then use the ID loss for later 30k iterations. We use the same anchor size/ratio as in the VOC experiment. For CrowdHuman, we additionally compare our method with RepLoss (Wang et al., 2018), which tackles the problem of the occlusion between humans.

In Table 3, we measure mAP on images with different occlusion levels: from mild occlusions (overlap IoU > 0.3) in $\texttt{crowd}_3$ to severe occlusions (overlap IoU > 0.7) in $\texttt{crowd}_7$. Results show that the proposed algorithm significantly improves the detection by 10% mAP for crowded cases compared to the Faster R-CNN baseline. RepLoss shows 0.2% mAP better than Faster R-CNN, but for the severe occlusion, the performance is similar to Faster R-CNN and 10% lower than IDNet. The LDDP shows similar performance to Faster R-CNN and RepLoss when the occlusion is not severe. Although it is relatively high in the case of severe occlusion such as $\texttt{crowd}_7$, IDNet performs better than LDDP for crowd sets in CrowdHuman.

5.2. Pascal VOC

For VOC 07, we train a network with VOC 07 `trainval`, which contains 5k images. For VOC 0712, we train a network with VOC 0712 `trainval`, which includes 16k images. All methods are tested on VOC 07 `test`, which has 5k images. After training IDNet with the SS loss and the classification and localization loss, we train RIN to learn differences of instances with the ID loss for 30k iterations for VOC 07, and 20k iterations for VOC 0712. While training RIN, the parameters in other modules except RIN are frozen. A VGG-16 backbone is used for all tested methods for Pascal VOC.

Since IDNet is effective for overlapped objects, we report recall which is calculated as a ratio of detected objects among the overlapped objects (Fig. 5). For calculating recall, we check that there are detected objects among the objects overlapped with another object above a fixed IoU threshold. After calculating the probability of detecting overlapped objects in each category, the results are averaged over categories. The recall is a better performance measure than mAP for showing the robustness to overlap. This is because the recall is calculated only for overlapped objects, while the mAP is calculated for all objects in an image containing at least a single overlapped object.

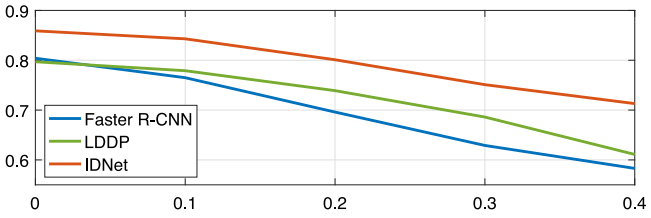


Fig. 5. Recall curves of Faster R-CNN, LDDP, and IDNet on VOC 07. The results are evaluated at different overlap IoU thresholds, from .0 to .4. Our proposed IDNet has a higher recall on crowded cases and effectively detects object with high overlaps.

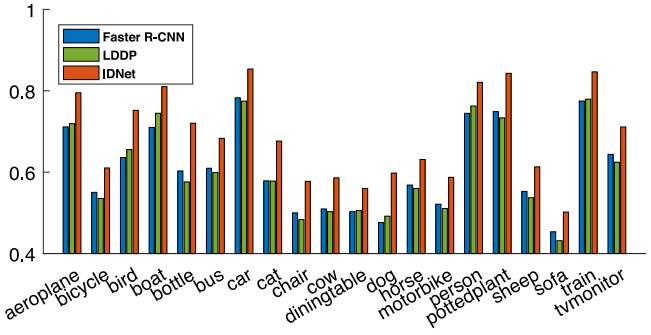


Fig. 6. Probability of finding correct bounding boxes after training IDNet with **SS** loss. For the evaluation, IDNet is trained with VOC. The categories are sampled for the best view.

In Fig. 5, recall for the objects with overlap over 0.4 is increased from 58% (Faster R-CNN) to 71% (IDNet), which is an impressive improvement. For all overlap regions, recall is higher than baseline methods and as the overlap ratio gets higher, the performance gap between Faster R-CNN and IDNet gets bigger. The results show that IDNet is effective for detecting objects in proximity.

To verify that SS loss affected the improvements, we extract candidate boxes having detection scores over a fixed threshold (0.01) in Fig. 6. When a predicted box overlaps with the ground truth box by 0.5 of IoU or more, we consider it as a correct box. We divide the number of correct boxes by the number of bounding boxes to check how many boxes are correctly classified. Fig. 6 shows that IDNet achieves superior performance on this measure for all categories compared to other methods. On average, IDNet achieves 66.7% while Faster R-CNN has 55.2% and LDDP has 54.5% for VOC. The results indicate that the SS loss can successfully remove incorrectly classified bounding boxes.

To demonstrate that our IDNet is effective for detecting overlapped objects on the standard mAP, we tested Faster R-CNN, LDDP and our IDNet*[3] on VOC `crowd`. IDNet shows impressive improvements compared to Faster R-CNN with an improvement of 5.7% mAP for VOC 07 and 2.5% for VOC 0712. We also observe improvements over LDDP: 4.5% improvement in mAP for VOC 07 and 1.4% improvement for VOC 0712. Next, when we evaluated mAP for VOC `test`, the mAP compared

[3] IDNet* is a version of IDNet only using ID loss, not SS loss.

Table 4
Detection results on VOC 07 `test` and VOC `crowd`. Legend: **07:** VOC 07 `trainval`, **07+12:** VOC 0712 `trainval`. All methods are trained using a VGG-16 backbone network. IDNet* is a version of IDNet only using ID loss.

Method	Inference	Train	mAP	
			`test`	`crowd`
Fast R-CNN (Girshick, 2015)	NMS	07	66.9	–
SSD300 (Liu et al., 2016)	NMS	07	68.0	–
Faster R-CNN (Ren et al., 2015)	NMS	07	71.0	56.5
Faster R-CNN (Ren et al., 2015)	NMS(0.4)	07	71.8	61.2
Faster R-CNN (Ren et al., 2015)	NMS(0.5)	07	71.0	61.5
Faster R-CNN (Ren et al., 2015)	NMS(0.6)	07	67.9	61.6
Faster R-CNN (Ren et al., 2015)	NMS(0.7)	07	62.9	58.4
LDDP (Azadi et al., 2017)	LDPP	07	70.9	57.7
IDNet*	IDPP	07	**72.0**	**62.2**
Fast R-CNN (Girshick, 2015)	NMS	07+12	70.0	–
SSD300 (Liu et al., 2016)	NMS	07+12	74.3	–
Faster R-CNN (Ren et al., 2015)	NMS	07+12	75.8	62.0
LDDP (Azadi et al., 2017)	LDPP	07+12	76.4	63.1
IDNet*	IDPP	07+12	**76.6**	**64.5**

Table 5
Detection results on COCO `val` and COCO `crowd`. All methods are trained with COCO `train`.

Method	Inference	Backbone	AP		AP_{50}		AP_{75}	
			`test`	`crowd`	`test`	`crowd`	`test`	`crowd`
Faster R-CNN (Ren et al., 2015)	NMS	VGG-16	26.2	19.2	46.6	36.9	26.9	18.4
LDDP (Azadi et al., 2017)	LDPP	VGG-16	26.4	19.6	46.7	37.9	26.8	18.6
IDNet	IDPP	VGG-16	**27.3**	**20.5**	**47.6**	**38.2**	**28.2**	**20.0**
Faster R-CNN (Ren et al., 2015)	NMS	ResNet-101	31.5	23.5	52.0	42.5	33.5	23.0
LDDP (Azadi et al., 2017)	LDPP	ResNet-101	31.4	23.8	51.7	43.0	33.4	23.4
IDNet	IDPP	ResNet-101	**32.7**	**24.4**	**53.1**	**43.4**	**34.8**	**24.4**

with baseline methods is increased for both VOC 07 and VOC 0712 (Table 4).

We tested different NMS thresholds as shown in Table 4. It shows that the proposed algorithm consistently works favorably against different settings of NMS. Although Faster R-CNN (Ren et al., 2015) used a NMS threshold of 0.3, we found that performance of a detector is better when the threshold is 0.4. However, even though the threshold is 0.4, the mAP on VOC `test` is about 0.2% lower than that of IDNet, and the performance in the VOC `crowd` is about 1% lower than IDNet. As the NMS threshold increased to 0.6, the performance in the VOC `crowd` increases to 61.6%, but the performance decreased from 0.7. When the NMS threshold is set to 0.6 to increase mAP in the VOC `crowd`, the performance in the VOC `test` was 67.9%, which was 4.1% lower than IDNet.

When using NMS, we should test by changing the parameters of the NMS according to the test set. However, in reality, it is difficult to find out the degree of occlusion in the test environment. Since the method we proposed in this paper is robust to the degree of occlusion of objects, it can be applied to more challenging situations regardless of the degree of occlusions.

5.3. MS COCO

MS COCO is composed of 80k images in the `train` set and 40k images in the `val` set. After training a network with the SS loss and the classification and localization loss, we train RIN module with the ID loss for 20k additional iterations.

In Table 5, we report the results using multiple APs for COCO. With respect to COCO `crowd`, Table 5 shows that the performance is improved from 19.2% to 20.5% AP for VGG-16. Since the larger number of categories in COCO makes distinguishing instances harder, the improvement is smaller than the results on VOC `crowd`. To demonstrate the general effectiveness of our IDNet, we also provide the results when the backbone network is replaced by ResNet-101. The performance of IDNet is improved from 23.5% AP to 24.4% AP on the ResNet-101 backbone, compared with Faster R-CNN, which shows the effectiveness of our IDNet on a stronger backbone. We also observe that the improvement on the AP_{75} is bigger than the improvement on the AP_{50}, which means IDNet with the IDPP inference algorithm is effective for the localization accuracy.

Table 6
Ablation study on COCO. All results are from IDNet using VGG-16 as a backbone.

Loss		AP	
SS	ID	`val`	`crowd`
		26.2	19.2
✓		27.0	19.6
	✓	26.5	19.7
✓	✓	**27.3**	**20.5**

For all COCO `val` images, the performance is improved by 1.1% AP for the VGG-16 backbone and 1.2% AP for the ResNet-101 backbone (Table 5). We attribute the reason for the improvements to the fact that there are many similar categories in COCO, which has eight categories for each of 11 super categories on average. Since a number of duplicated candidate boxes can be generated, our SS loss can remove duplicated bounding boxes to increase the final detection performance.

Inference time. We measure the average inference time per image using VGG-16 as a backbone network on COCO `val`, which is a subset of 5k samples from the `val`. All running times are measured on a machine with Intel Core 3.7 GHz CPU and Titan X GPU.

Our algorithm takes 2.14 s to find candidate boxes and extract features of them, and 0.33 s to select bounding boxes using IDPP. Since Faster R-CNN takes 1.61 s and LDDP (Azadi et al., 2017) takes 1.55 s, an extra time of 0.86 s is needed for detecting objects in an image compared with Faster R-CNN and 0.92 s compared with LDDP. Although our algorithm takes more time to inference, it can be used in problems which require exact detections in a crowd.

5.4. Ablation study

We analyze the influence of the ID loss and SS loss in Table 6, where IDNet is trained with COCO `train` using VGG-16 as a backbone. In

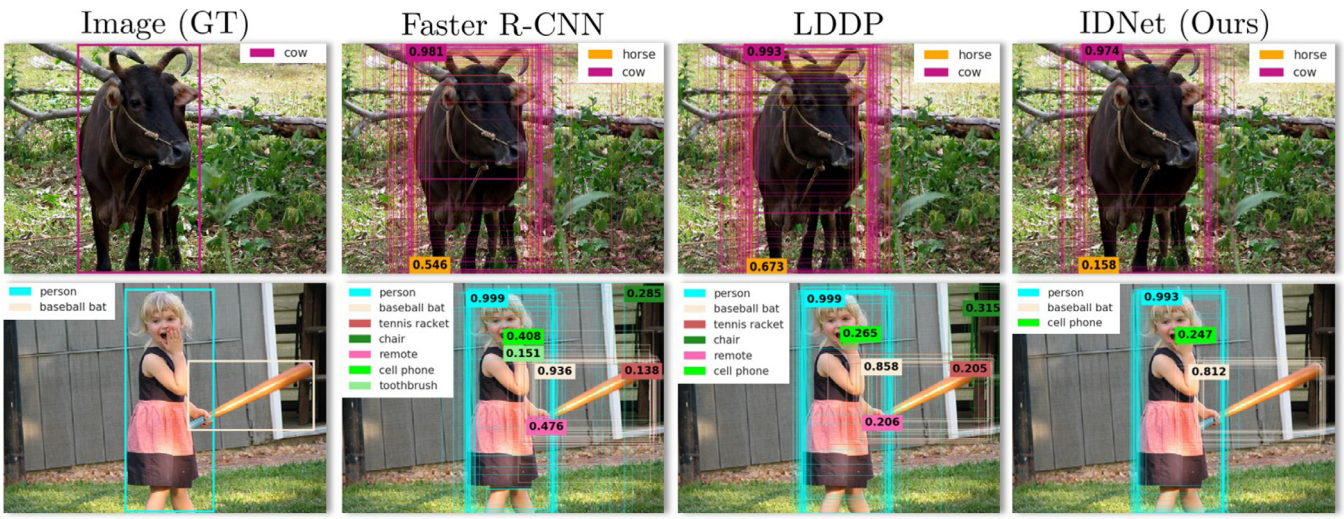
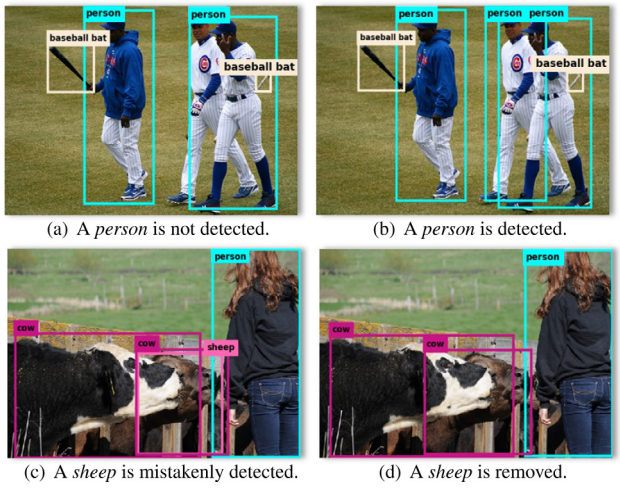


Fig. 7. Scores of candidate boxes after training with each method. The leftmost column shows the ground truth boxes, and the other columns show the results of Faster R-CNN, LDDP, and IDNet from left to right. For each method, candidate boxes with scores over 0.1 and the maximum score of each category are visualized on each image. All methods are trained on COCO `train` using VGG-16 as a backbone.



(a) A *person* is not detected. (b) A *person* is detected.

(c) A *sheep* is mistakenly detected. (d) A *sheep* is removed.

Fig. 8. Qualitative detection results of Faster R-CNN vs. IDNet. (a), (c) are results of Faster R-CNN and (b), (d) are results of IDNet. In (b), IDNet detect a *person*, which is not detected on Faster R-CNN in (a). In (d), IDNet successfully suppresses an incorrect label, *sheep*, while Faster R-CNN reports a *sheep* in (c).

ablation studies, we check our IDNet with two post-processing methods: NMS and IDPP. In the first two rows in Table 6, we use NMS for the experiments that do not use the ID loss, since IDPP uses the trained features with the ID loss. In the last two rows of Table 6, we use IDPP with a trained RIN module.

Instance-aware detection loss. The ID loss is made to be effective for detecting objects in a crowded scene. In the third row of Table 6, the performance is improved from 19.2% to 19.7% AP on COCO `crowd`. Comparing the second row and the last row, the performance is improved by 0.9% AP. In Fig. 8(a), a *person* is not detected in Faster R-CNN, while our IDNet detects the *person* in Fig. 8(b) since IDNet learns to discriminate different objects. This result indicates that ID loss is effective for detecting objects in proximity.

Sparse score loss. Since the SS loss is designed to remove incorrectly classified bounding boxes, the SS loss is effective for all testing images. Thus, we focus on the results on COCO `val` column in Table 6. The results show that as the SS loss is used, the performance is improved by 0.8% AP.

In Fig. 8(c), a *sheep* is erroneously detected for a *cow*, while our IDNet removes this erroneous detection of a *sheep* in Fig. 8(d) as IDNet learns to remove incorrectly classified bounding boxes. It shows that the SS loss can alleviate duplicated bounding box problem in a detector.

Since Fig. 8 only shows the final detections, we visualize images with candidate boxes in Fig. 7 to show the changes in detection scores. The score threshold is fixed to 0.1 and the highest score in each category is written in each image.

We first compare the result with Faster R-CNN. Since Faster R-CNN does not have any loss to decrease the scores of incorrect categories, the highest score of a *horse* in Faster R-CNN is 0.546 while the score in IDNet is 0.158 (see the first row of Fig. 7). For images in the second row of Fig. 7, the maximum score of an incorrect category, *remote*, is 0.476 in Faster R-CNN, while the maximum score of a *remote* is under the threshold (0.1) in IDNet.

We also compare the result with LDDP (Azadi et al., 2017). The LDDP loss (Azadi et al., 2017) is defined to increase the score of a single subset using a category-level relationship, while our SS loss is defined to decrease scores of all possible subsets containing incorrect candidate boxes using an instance-level relationship between candidate boxes. Thus, after softmax is applied to scores, the SS loss can better suppress the detection scores of bounding boxes with incorrect categories. For example, as shown in the third and last columns of Fig. 7, given a *cow* image, the detection score for a *horse* is decreased from 0.673 (LDDP) to 0.158 (IDNet). It shows that the SS loss can successfully suppress scores of duplicated bounding boxes around a correct bounding box as expected.

6. Conclusion

We propose IDNet which tackles two challenges in object detection by introducing two novel losses. First, we propose the ID loss for detecting overlapped objects. Second, the SS loss is introduced to suppress erroneous detections of wrong categories. By introducing these two losses using DPPs, we demonstrate that learning an instance-level relationship is useful for accurate detection. IDNet performs favorably for overall test sets and achieves significant improvements on the crowd sets. Additionally, the ablation studies show that IDNet learns to suppress erroneous detections of wrong categories. While the inference time is moderately slower than other detection methods, our algorithm is useful for real-world situations which require separating objects in proximity.

Table A.7
Detection results on VOC 07 `test` Legend: **07:** VOC 07 `trainval`, **07+12:** VOC 0712 `trainval`. All methods are trained with the classification and localization loss, using a VGG-16 backbone network.

Method	Inference	Train	mAP	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	mbike	Person	Plant	Sheep	Sofa	Train	Tv
Faster R-CNN (Ren et al., 2015)	NMS	07	71.0	69.4	78.5	69.6	55.0	**57.3**	**80.5**	82.5	82.8	52.9	78.5	67.6	79.4	**84.9**	75.4	77.9	**45.2**	68.7	65.7	74.6	**74.0**
LDDP (Azadi et al., 2017)	LDPP	07	70.9	67.7	**79.2**	68.2	**57.9**	53.9	75.2	79.7	84.8	**53.7**	79.2	67.5	**80.9**	84.0	75.7	78.0	44.7	73.3	66.7	73.8	73.1
IDNet*	IDPP	07	**72.0**	**71.4**	79.0	**70.5**	**58.0**	53.7	77.8	**83.9**	**85.8**	52.9	**81.0**	**68.9**	80.7	84.3	**75.3**	**79.7**	43.9	**74.9**	**66.6**	**76.6**	73.9
Faster R-CNN (Ren et al., 2015)	NMS	07+12	75.8	77.2	**84.1**	74.8	**67.3**	65.5	82.0	87.4	**87.9**	58.7	81.5	69.8	85.0	85.1	77.7	79.2	47.2	75.4	71.8	82.3	75.8
LDDP (Azadi et al., 2017)	LDPP	07+12	76.4	76.9	83.0	75.0	66.5	64.3	**83.4**	87.5	87.7	**61.2**	81.5	70.0	**86.0**	84.9	**81.9**	**83.3**	48.6	75.7	**72.3**	82.6	**76.5**
IDNet*	IDPP	07+12	**76.6**	**78.8**	82.8	**75.9**	66.3	**66.6**	82.9	**88.1**	87.2	59.6	**82.4**	**70.6**	85.1	**85.7**	80.7	82.6	**50.0**	**78.3**	70.9	**82.8**	75.5

Table A.8
Detection results on VOC `crowd`. Legend: **07:** VOC 07 `trainval`, **07+12:** VOC 0712 `trainval`. All methods are trained with the classification and localization loss, using a VGG-16 backbone network.

Method	Inference	Train	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster R-CNN (Ren et al., 2015)	NMS	07	56.5	45.5	**63.3**	44.2	41.2	57.6	54.5	69.4	37.3	48.5	68.5	65.8	56.4	62.9	63.1	67.6	29.8	66.5	**53.1**	63.6	**70.1**
LDDP (Azadi et al., 2017)	LDPP	07	57.7	38.2	61.4	47.9	37.7	54.3	54.5	74.6	**48.1**	49.5	**76.1**	**70.3**	60.3	63.3	60.3	**73.7**	31.4	70.5	52.3	63.6	66.3
IDNet*	IDPP	07	**62.2**	**65.5**	62.6	**56.2**	**48.9**	**61.3**	**61.2**	**75.5**	38.6	**50.8**	67.7	65.7	**68.0**	**67.5**	**70.4**	73.5	**35.8**	**74.1**	50.6	**81.8**	68.4
Faster R-CNN (Ren et al., 2015)	NMS	07+12	62.0	**100.0**	59.4	**60.1**	28.5	61.3	53.2	72.0	51.4	51.9	67.0	67.0	55.1	76.9	71.4	69.4	32.6	67.5	**61.1**	**63.6**	70.2
LDDP (Azadi et al., 2017)	LDPP	07+12	63.1	78.5	64.6	55.6	**34.8**	60.3	52.1	76.9	**55.4**	**56.7**	**72.8**	**69.0**	**69.0**	73.2	69.3	**76.3**	**41.4**	73.4	48.2	**63.6**	**70.5**
IDNet*	IDPP	07+12	**64.5**	88.3	**68.8**	59.8	31.9	**64.1**	**61.7**	**79.0**	48.7	54.4	72.3	66.5	64.2	**77.7**	**71.7**	75.6	37.7	**77.0**	57.5	**63.6**	70.0

Table A.9
Detection results on MS COCO `val`. All methods are trained on MS COCO `train` with the classification and localization loss.

Method	Inference	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
Faster R-CNN (Ren et al., 2015)	NMS	VGG-16	26.2	46.6	26.9	10.3	29.3	36.4	25.5	38.1	39.0	17.9	44.0	55.7
LDDP (Azadi et al., 2017)	LDPP	VGG-16	26.4	46.7	26.8	10.5	29.4	36.8	25.0	37.4	38.4	16.0	43.1	55.3
IDNet	IDPP	VGG-16	**27.3**	**47.6**	**28.2**	**10.9**	**30.1**	**38.0**	**25.9**	**39.4**	**40.6**	**18.6**	**45.1**	**58.9**
Faster R-CNN (Ren et al., 2015)	NMS	ResNet-101	31.5	52.0	33.5	12.5	35.2	45.9	29.2	43.2	44.2	20.6	49.9	63.8
LDDP (Azadi et al., 2017)	LDPP	ResNet-101	31.4	51.7	33.4	12.3	35.3	46.0	28.5	41.9	42.9	18.2	48.2	63.4
IDNet	IDPP	ResNet-101	**32.7**	**53.1**	**34.8**	**13.1**	**36.4**	**47.6**	**29.5**	**44.3**	**45.6**	**21.2**	**51.2**	**65.8**

Table A.10
Detection results on MS COCO `crowd`. All methods are trained on MS COCO `train` with the classification and localization loss.

Method	Inference	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
Faster R-CNN (Ren et al., 2015)	NMS	VGG-16	19.2	36.9	18.4	8.5	24.3	31.0	17.0	28.6	29.6	13.4	36.4	47.8
LDDP (Azadi et al., 2017)	LDPP	VGG-16	19.6	37.9	18.6	8.9	24.6	31.6	16.6	28.4	29.6	12.9	36.4	47.7
IDNet	IDPP	VGG-16	**20.5**	**38.2**	**20.0**	**9.1**	**25.7**	**33.0**	17.0	**30.9**	**33.2**	**14.4**	**39.2**	**56.0**
Faster R-CNN (Ren et al., 2015)	NMS	ResNet-101	23.5	42.5	23.0	10.4	29.6	38.5	19.3	32.8	34.0	16.1	41.7	54.6
LDDP (Azadi et al., 2017)	LDPP	ResNet-101	23.8	43.0	23.4	10.5	30.0	39.4	19.2	32.4	33.7	15.0	41.4	55.2
IDNet	IDPP	ResNet-101	**24.4**	**43.4**	**24.4**	**10.9**	**30.6**	**40.0**	**19.6**	**33.7**	**34.8**	**16.5**	**42.4**	**56.4**

CRediT authorship contribution statement

Nuri Kim: Visualization, Conceptualization, Methodology, Software, Writing - original draft, Validation, Formal analysis. **Donghoon Lee:** Conceptualization, Writing - review & editing. **Songhwai Oh:** Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning). This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments).

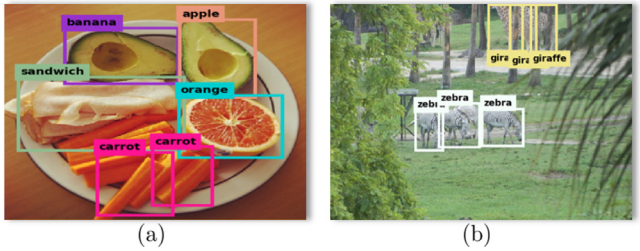


Fig. C.9. Failure cases of IDNet. (a) A detector find an incorrect category; **(b)** A detector cannot distinguish a completely occluded object. The class labels are arranged for the best view.

Appendix A. Gradient of losses

In this section, we derive the gradients of the proposed instance-aware detection (ID) loss and sparse score (SS) loss. For notational convenience, we assume that the matrix $\mathbf{M}_x$ has the same dimension as $\mathbf{M}$ and its entries corresponding to x is copied from $\mathbf{M}$ while remaining entries are filled with zero, for any matrix $\mathbf{M}$ and indices x.

A.1. Gradient of instance-aware detection loss

Here, we show the gradient with respect to the normalized feature ($\bar{\mathbf{F}}$). As the derivative of the log-determinant is $\partial \log\det(\mathbf{L}) =$

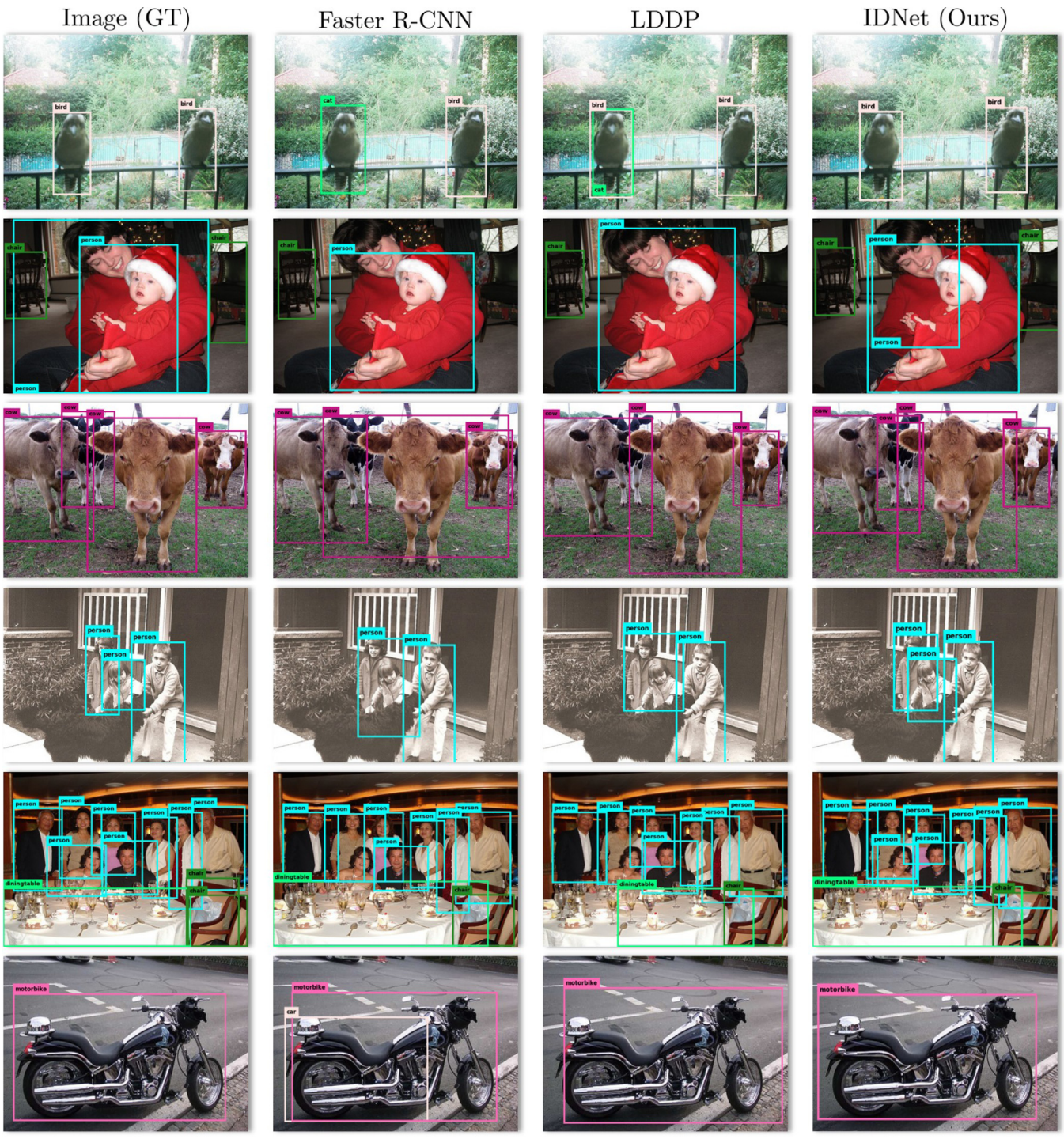


Fig. C.10. Visualization results on Pascal VOC 07 `test`. The leftmost column shows the ground truth boxes, and the other columns show the results of Faster R-CNN, LDDP, and IDNet from left to right. For each method, final boxes with scores over 0.6 are visualized on each image. All methods are trained on VOC 07 `trainval` using VGG-16 as a backbone.

$\partial(\mathrm{Tr}(\log(\mathbf{L}))) = \mathrm{Tr}(\mathbf{L}^{-T}\partial\mathbf{L})$, the derivative of the category-specific ID loss is as follows:

$$
\begin{aligned}
&\{\partial\mathcal{L}_{ID}^{cs}(Y_{C_k}, \mathcal{Y}_{C_k})\}_{C_k} \\
&= -\partial \log\det(\mathbf{L}_{Y_{C_k}}) + \partial \log\det(\mathbf{L}_{\mathcal{Y}_{C_k}} + \mathbf{I}_{\mathcal{Y}_{C_k}}) \\
&= -\partial\,\mathrm{Tr}(\log(\mathbf{L}_{Y_{C_k}})) + \partial\,\mathrm{Tr}(\log(\mathbf{L}_{\mathcal{Y}_{C_k}} + \mathbf{I}_{\mathcal{Y}_{C_k}})) \\
&= -\mathrm{Tr}(\mathbf{L}_{Y_{C_k}}^{-T}\partial\mathbf{L}_{Y_{C_k}}) + \mathrm{Tr}((\mathbf{L}_{\mathcal{Y}_{C_k}} + \mathbf{I}_{\mathcal{Y}_{C_k}})^{-T}\partial(\mathbf{L}_{\mathcal{Y}_{C_k}} + \mathbf{I}_{\mathcal{Y}_{C_k}})) \\
&= -\langle {\mathbf{L}_{Y_{C_k}}}^{-1},\ \partial\mathbf{L}_{Y_{C_k}}\rangle + \langle(\mathbf{L}_{\mathcal{Y}_{C_k}} + \mathbf{I}_{\mathcal{Y}_{C_k}})^{-1},\ \partial\mathbf{L}_{\mathcal{Y}_{C_k}}\rangle, \qquad \text{(A.1)}
\end{aligned}
$$

where $\langle\cdot,\cdot\rangle$ is the Frobenius inner product, $\odot$ is the element-wise multiplication, and $k \in \{1, \ldots, n_c\}$ is the kth category. Note that the n_c is the number of categories. We only calculate the gradient of the ID loss on the similarity feature ($\mathbf{V}$), where $\mathbf{S} = \lambda \cdot \mathbf{V}\mathbf{V}^T + (1-\lambda)\cdot\mathbf{IoU}$. Since $\mathbf{IoU}$ is a constant, the derivative of $\mathbf{L}$ is as follows:

$$\partial\mathbf{L} = \lambda \cdot \mathbf{Q} \odot (\partial\mathbf{V}\mathbf{V}^T + \mathbf{V}\partial\mathbf{V}^T), \qquad \text{(A.2)}$$

where $\mathbf{Q} = \mathbf{q}\mathbf{q}^T$. Note that $\mathbf{Q}$ is fixed while deriving gradient of the ID loss. Using the property that $\langle\mathbf{A},\ \mathbf{B}\odot\mathbf{C}\rangle = \langle\mathbf{A}\odot\mathbf{B},\ \mathbf{C}\rangle$, where $\mathbf{A}, \mathbf{B},$ and $\mathbf{C}$

Fig. C.11. Visualization results on COCO `val`. The leftmost column shows the ground truth boxes, and the other columns show the results of Faster R-CNN, LDDP, and IDNet from left to right. For each method, final boxes with scores over 0.6 are visualized on each image. All methods are trained on COCO `train` using VGG-16 as a backbone.

are arbitrary matrices, we can derive this:

$$\begin{aligned}&\{\partial\mathcal{L}_{ID}^{cs}(Y_{C_k},\mathcal{Y}_{C_k})\}_{C_k}\\&=-2\lambda\cdot\langle(\mathbf{Q}_{Y_{C_k}}\odot\mathbf{L}_{Y_{C_k}})^{-1}\mathbf{V}_{Y_{C_k}},\ \partial\mathbf{V}_{Y_{C_k}}\rangle\\&\quad+2\lambda\cdot\langle\mathbf{Q}_{\mathcal{Y}_{C_k}}\odot(\mathbf{L}_{\mathcal{Y}_{C_k}}+\mathbf{I}_{\mathcal{Y}_{C_k}})^{-1}\mathbf{V}_{\mathcal{Y}_{C_k}},\ \partial\mathbf{V}_{\mathcal{Y}_{C_k}}\rangle.\end{aligned}\tag{A.3}$$

By seeing the matrix in element-wise,

$$\begin{aligned}&\left\{\frac{\partial\mathcal{L}_{ID}^{cs}(Y_{C_k},\mathcal{Y}_{C_k})}{\partial\mathbf{V}}\right\}_{C_k}\\&=-2\lambda\cdot(\mathbf{Q}_{Y_{C_k}}\odot\mathbf{L}_{Y_{C_k}})^{-1}\mathbf{V}_{Y_{C_k}}\\&\quad+2\lambda\cdot\mathbf{Q}_{\mathcal{Y}_{C_k}}\odot(\mathbf{L}_{\mathcal{Y}_{C_k}}+\mathbf{I}_{\mathcal{Y}_{C_k}})^{-1}\mathbf{V}_{\mathcal{Y}_{C_k}}.\end{aligned}\tag{A.4}$$

Since the gradient of $\mathcal{L}_{ID}^{all}$ is similar to a gradient of $\mathcal{L}_{ID}^{cs}$, we omit the derivation of that. Then, we can construct the gradient of the ID loss by summing up (A.4) for all batches and categories:

$$\begin{aligned}&\frac{\partial\mathcal{L}_{ID}(Y_{rep},\mathcal{Y}_s,Y_{C_k},\mathcal{Y}_{C_k})}{\partial\mathbf{V}}\\&=\frac{\partial\mathcal{L}_{ID}^{all}(Y_{rep},\mathcal{Y}_s)}{\partial\mathbf{V}}+\sum_{k=1}^{n_c}\left\{\frac{\partial\mathcal{L}_{ID}^{cs}(Y_{C_k},\mathcal{Y}_{C_k})}{\partial\mathbf{V}}\right\}_{C_k}.\end{aligned}\tag{A.5}$$

A.2. Gradient of sparse score loss

The derivation for calculating the gradient of the SS loss is similar with the derivation of the instance-aware detection loss, while the gradient for the SS loss is derived over the quality ($\mathbf{q}$). Note that $\mathbf{S}$ is fixed while deriving gradient of the SS loss. The derivative of the SS

loss is as follows:

$$\begin{aligned}
&\partial \mathcal{L}_{SS}(Y_{pos}, \mathcal{Y}_m) \\
&= -\partial \log\det(\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}}) + \partial \log\det(\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m}) \\
&= -\partial \operatorname{Tr}(\log(\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}})) + \partial \operatorname{Tr}(\log(\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})) \\
&= -\operatorname{Tr}((\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}})^{-T} \partial \mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}}) \\
&\quad + \operatorname{Tr}((\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})^{-T} \partial (\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})) \\
&= -\langle (\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}})^{-1},\ \partial \mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}} \rangle \\
&\quad + \langle (\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})^{-1},\ \partial \mathbf{L}_{\mathcal{Y}_m} \rangle.
\end{aligned} \tag{A.6}$$

Similar to the derivation of the ID loss, by using the following properties,

$$\begin{aligned}
&\partial \mathbf{L} = \mathbf{S} \odot (\partial \mathbf{q} \mathbf{q}^T + \mathbf{q} \partial \mathbf{q}^T), \\
&\langle \mathbf{A},\ \mathbf{B} \odot \mathbf{C} \rangle = \langle \mathbf{A} \odot \mathbf{B},\ \mathbf{C} \rangle,
\end{aligned} \tag{A.7}$$

we can derive this:

$$\begin{aligned}
&\partial \mathcal{L}_{SS}(Y_{pos}, \mathcal{Y}_m) \\
&= -2 \cdot \langle \mathbf{S}_{Y_{pos}} \odot (\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}})^{-1} \mathbf{q}_{Y_{pos}},\ \partial \mathbf{q}_{Y_{pos}} \rangle \\
&\quad + 2 \cdot \langle \mathbf{S}_{\mathcal{Y}_m} \odot (\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})^{-1} \mathbf{q}_{\mathcal{Y}_m},\ \partial \mathbf{q}_{\mathcal{Y}_m} \rangle.
\end{aligned} \tag{A.8}$$

Thus, the final derivative of the SS loss is as follows:

$$\begin{aligned}
&\frac{\partial \mathcal{L}_{SS}(Y_{pos}, \mathcal{Y}_m)}{\partial \mathbf{q}} \\
&= -2 \cdot \mathbf{S}_{Y_{pos}} \odot (\mathbf{L}_{Y_{pos}} + \mathbf{I}_{Y_{pos}})^{-1} \mathbf{q}_{Y_{pos}} \\
&\quad + 2 \cdot \mathbf{S}_{\mathcal{Y}_m} \odot (\mathbf{L}_{\mathcal{Y}_m} + \mathbf{I}_{\mathcal{Y}_m})^{-1} \mathbf{q}_{\mathcal{Y}_m}.
\end{aligned} \tag{A.9}$$

Appendix B. More experimental results

In this section, we provide full results on Pascal VOC and MS COCO datasets. For the results on all test images are in Tables A.7 and A.9. The results on the crowd sets are in Tables A.8 and A.10.

Appendix C. Example visualization

We visualize qualitative results of IDNet on VOC 07 and MS COCO. For comparison, we also visualize the ground truth bounding boxes in each image, and the results of Faster R-CNN and LDDP. For Faster R-CNN and LDDP, only bounding boxes with a score threshold of 0.6 are visualized. The threshold is designated in their paper, Azadi et al. (2017). For IDNet, we use 0.2 as a score threshold.

Failure cases analysis. The left image of Fig. C.9 shows that the detector detected the bounding box of the wrong category for avocados. This means that the detector has found a class similar to avocado, such as banana and apple because there are no categories in a dataset. This case suggests that there is a need to suppress further scores for pictures in the absence of a detection class, i.e., background category. In the right of Fig. C.9, a giraffe is hidden behind two trees. If there is an occlusion for an object, detectors tend not to notice that it is a single object. Then detectors choose several bounding boxes for the object. Since IDPP tries to find the most representative bounding boxes, it would select all of the created bounding boxes, which increases the number of false detections.

Successful cases. The successful images of IDNet are visualized in Fig. C.10 for VOC 07, and Fig. C.11 for MS COCO. In Fig. C.10, the first and last row images show that incorrect class bounding boxes are suppressed while selecting a correct class, which means that IDNet suppressed bounding boxes with incorrect categories. The results on the other rows show the objects in proximity are detected while other methods fail. The results show that overlapped objects are successfully detected in IDNet. In Fig. C.11, all results show that IDNet can detect overlapped objects.

The results show the proposed IDNet can detect overlapped objects compared to the other algorithms while suppressing bounding boxes with incorrect categories.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. Tensorflow: A system for large-scale machine learning. In: USENIX Symposium on Operating Systems Design and Implementation, OSDI.

Azadi, S., Feng, J., Darrell, T., 2017. Learning detection with diverse proposals. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Chao, W.-L., Gong, B., Grauman, K., Sha, F., 2015. Large-margin determinantal point processes. In: Conference on Uncertainty in Artificial Intelligence, UAI.

Choi, J., Chun, D., Kim, H., Lee, H.-J., 2019. Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving. In: IEEE International Conference on Computer Vision, ICCV.

Dai, J., He, K., Sun, J., 2016a. Instance-aware semantic segmentation via multi-task network cascades. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Dai, J., Li, Y., He, K., Sun, J., 2016b. R-FCN: Object detection via region-based fully convolutional networks. In: Neural Information Processing Systems, NIPS.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (VOC) challenge. Int. J. Comput. Vision 88 (2), 303–338.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32 (9), 1627–1645.

Gawande, U., Hajari, K., Golhar, Y., 2020. Pedestrian detection and tracking in video surveillance system: issues, comprehensive review, and challenges. In: Recent Trends in Computational Intelligence. IntechOpen Publisher.

Girshick, R., 2015. Fast R-CNN. In: IEEE International Conference on Computer Vision, ICCV.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics, AISTATS.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y., 2018. Relation networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, ICML.

Kong, T., Yao, A., Chen, Y., Sun, F., 2016. HyperNet: Towards accurate region proposal generation and joint object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Krause, A., Singh, A., Guestrin, C., 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. J. Mach. Learn. Res. 9, 235–284.

Kuhn, H.W., 1955. The hungarian method for the assignment problem. Naval Res. Logist. Q. 2 (1–2), 83–97.

Kulesza, A., Taskar, B., 2012. Determinantal point processes for machine learning. arXiv preprint arXiv:1207.6083.

Lee, D., Cha, G., Yang, M.-H., Oh, S., 2016. Individualness and determinantal point processes for pedestrian detection. In: European Conference on Computer Vision, ECCV.

Li, Y., Sun, B., Wu, T., Wang, Y., 2016. Face detection with end-to-end integration of a ConvNet and a 3D model. In: European Conference on Computer Vision, ECCV.

Lin, H., Bilmes, J.A., 2012. Learning mixtures of submodular shells with application to document summarization. In: Conference on Uncertainty in Artificial Intelligence, UAI.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common objects in context. In: European Conference on Computer Vision, ECCV.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: European Conference on Computer Vision, ECCV.

Liu, Y., Wang, R., Shan, S., Chen, X., 2018. Structure inference net: Object detection using scene-level context and instance-level relationships. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM 38 (11), 39–41.

Peng, J., Sun, M., Zhang, Z., Tan, T., Yan, J., 2019. Efficient neural architecture transformation search in channel-level for object detection. In: Neural Information Processing Systems, NIPS.

Ranjan, R., Patel, V.M., Chellappa, R., 2017. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans. Pattern Anal. Mach. Intell.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Redmon, J., Farhadi, A., 2017. YOLO9000: Better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Neural Information Processing Systems, NIPS.

Ren, M., Zemel, R.S., 2017. End-to-end instance segmentation with recurrent attention. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J., 2018. CrowdHuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Neural Information Processing Systems, NIPS.

Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., Shen, C., 2018. Repulsion loss: Detecting pedestrians in a crowd. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H., 2019. Fast online object tracking and segmentation: A unifying approach. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1328–1338.

Xie, P., Salakhutdinov, R., Mou, L., Xing, E.P., 2017. Deep determinantal point process for large-scale multi-label classification. In: IEEE International Conference on Computer Vision, ICCV.

Zhang, K., Chao, W.-L., Sha, F., Grauman, K., 2016. Video summarization with long short-term memory. In: European Conference on Computer Vision, ECCV.

Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z., 2018. Single-shot refinement neural network for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR.

Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., Ling, H., 2019. M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network. Association for the Advancement of Artificial Intelligence (AAAI).

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J.R., Zhang, Y.-C., 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. Proc. Natl. Acad. Sci. 107 (10), 4511–4515.

Nuri Kim is a Ph.D. student in the Department of Electrical and Computer Engineering at Seoul National University, Seoul, Korea. Her research interests include object detection and computer vision. She received the B.S. degree (with great honors) in the School of Electrical and Electronics Engineering from Korea University, Seoul, Korea, in 2016.

Donghoon Lee received the BS, MS and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2011, 2013, and 2018 respectively. He is currently a computer vision and machine learning engineer at Apple, Cupertino, CA, USA. His research interests include machine learning and computer vision.

Songhwai Oh received the B.S. (with highest honors), M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1995, 2003, and 2006, respectively.

He is currently a Professor in the Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea. Before his Ph.D. studies, he was a Senior Software Engineer at Synopsys, Inc. and a Microprocessor Design Engineer at Intel Corporation. In 2007, he was a Postdoctoral Researcher in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. From 2007 to 2009, he was an Assistant Professor of electrical engineering and computer science in the School of Engineering, University of California, Merced. He is an Associate Editor for the IEEE Transactions on Robotics and IEEE Robotics and Automation Letters. His research interests include robotics, computer vision, cyber–physical systems, and machine learning.