Expert Systems With Applications 172 (2021) 114602

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A survey and performance evaluation of deep learning methods for small object detection

Yang Liu *, Peng Sun, Nickolas Wergeles, Yi Shang

Department of Electrical Engineering and Computer Science (EECS), University of Missouri, 201 Naka Hall, Columbia, MO, 65201, USA

ARTICLE INFO

Keywords:
Small object detection
Computer vision
Convolutional neural networks
Deep learning

ABSTRACT

In computer vision, significant advances have been made on object detection with the rapid development of deep convolutional neural networks (CNN). This paper provides a comprehensive review of recently developed deep learning methods for small object detection. We summarize challenges and solutions of small object detection, and present major deep learning techniques, including fusing feature maps, adding context information, balancing foreground-background examples, and creating sufficient positive examples. We discuss related techniques developed in four research areas, including generic object detection, face detection, object detection in aerial imagery, and segmentation. In addition, this paper compares the performances of several leading deep learning methods for small object detection, including YOLOv3, Faster R-CNN, and SSD, based on three large benchmark datasets of small objects. Our experimental results show that while the detection accuracy on small objects by these deep learning methods was low, less than 0.4, Faster R-CNN performed the best, while YOLOv3 was a close second.

1. Introduction

Object detection is one of the fundamental tasks in computer vision. Typically, object detection and recognition involve two steps: first, the potential location of each target object is localized; then, the objects are classified into different categories. Before the bloom of deep learning methods, object detection methods relied on manually designed features and designed classifiers based on how humans understand objects. In recent years, the field of object detection has dramatically advanced due to the success of deep learning, especially deep convolutional neural networks (CNN). Object detection has been widely used in many applications, such as autonomous driving, visual search, virtual reality (VR), and augmented reality (AR), etc.

Even though accurate detection of medium and large-size objects in images has been achieved in many applications, accurate detection of small objects, such as a 20x20 pixel duck in an aerial image, remains challenging. Small objects are difficult to detect due to indistinguishable features, low-resolution, complicated backgrounds, limited context information, etc. This is an active research area and many deep learning techniques have been developed in recent years with promising results. Some work showed the importance of combining different feature layers, while others showed contextual infomation is very useful. Moreover, techniques to improve classification accuracy, such as those addressing imbalanced class examples and insufficient training, have achieved good results.

1.1. Scope of this paper

This paper focuses on deep learning techniques for detecting small objects in images. We provide a comprehensive review of related object detection and instance segmentation methods. We identify and analyze major challenges and summarize strategies for improving detection performance on small objects. First, we discuss the challenges in four aspects: 1) features generated by individual layers in basic CNNs do not contain sufficient information for small object detection; 2) context information is lacking for small object detection; 3) imbalance of foreground and background training examples make classification difficult; and 4) insufficient positive training examples for small objects. Then, we summarize existing techniques for small objects from the perspective of 1) combining multiple feature maps, 2) adding context information, 3) balancing class examples, and 4) creating sufficient number of positive examples. We present related techniques developed in four different research areas, including generic object detection, face detection, object detection in aerial images, and instance segmentation. In the end, we

* Corresponding author.
E-mail addresses: ylt5b@mail.missouri.edu (Y. Liu), ps793@mail.missouri.edu (P. Sun), wergelesn@missouri.edu (N. Wergeles), shangy@missouri.edu (Y. Shang).

https://doi.org/10.1016/j.eswa.2021.114602
Received 9 April 2020; Received in revised form 13 October 2020; Accepted 10 January 2021
Available online 19 January 2021

report our experimental results of comparing the performances of several state-of-the-art deep learning methods on benchmark datasets focusing on small objects.

The main contributions of this paper are as follows:

- Provide a comprehensive review of the state-of-the-art deep learning techniques on small object detection.
- Identify challenges for small object detection in four specific aspects, summarize major components of deep learning methods, and categorize existing methods in four aspects.
- Analyze and connect related techniques from four research areas, including generic object detection, face detection, object detection in aerial imagery, and segmentation.
- Empirical performance evaluation of some state-of-the-art deep learning methods on three benchmark datasets of small objects.

1.2. Comparison with previous survey papers

The survey in (Zou, Shi, Guo, & Ye, 2019) covered object detection methods in the past 20 years, including both traditional detection methods and deep learning methods. This paper focuses on deep learning methods for small object detection developed in the last 5 years. (Zou et al., 2019) surveyed deep learning methods for generic object detection, whereas this paper includes methods developed in four research areas, including generic object detection, face detection, object detection in aerial imagery, and segmentation. (Leevy, Khoshgoftaar, Bauder, & Seliya, 2018; Oksuz, Cam, Kalkan, & Akbas, 2019) focused on methods to overcome the class imbalance problem. (Zhao, Zheng, Xu, & Wu , 2019) reviewed several state-of-the-art deep learning frameworks in several object detection tasks and analyzed different methods with experimental results on general object detection. (Liu et al., 2020; Jiao et al., 2019) reviewed the deep learning methods and techniques for object detection. However, they did not provide the experimental analysis for these deep learning methods. (Wu, Sahoo, & Hoi, 2020) reviewed object detection components, models and learning strategies. Even though these works provide a comprehensive review, their focus is on general size objects, not small objects.

Recently, there are also some reviews on the small object detection. Nguyen, Do, Ngo, and Le (2020)) provided a review of the existing object detection methods for small objects and focused on performance evaluation on four models. In comparison, this paper presents a more depth and comprehensive review and different perspective of challenges. Moreover, we summarize the major components of existing deep learning methods, categorize existing detection approaches in four aspects, connect and analyze current deep learning methods from four separate application areas of object detection, and evaluate three models' performances on three different datasets. Tong, Wu, and Zhou (2020) mainly reviewed existing methods from five aspects to improve small object detection and analyzed experimental results on two datasets. In comparison, this paper not only summarizes existing methods in different aspects, but also identifies and analyzes key challenges in four specific aspects, connects and analyzes solutions from several related research areas, and presents empirical results on different datasets.

In summary, this paper differs from the previous review papers in several aspects. First, our review is focus on small objects. Secondly, our review includes summaries of major detection components and state-of-the-art object detection frameworks. Thirdly, we identify the challenges for small object detection, and summarize major techniques to improve small object detection accuracy. In addition, we analyze and connect techniques from four small object detection application areas, which cover a wide range of small object detection tasks. Finally, we provide empirical comparison of three representative deep learning frameworks on small object benchmark datasets.

The rest of the paper is organized as follows. Section 2 presents an overview of deep learning methods and major components for object detection in images. Section 3 presents major deep learning approaches and frameworks for small object detection. Section 4 identifies the challenges and solutions for small object detection and major techniques developed in four related research areas. Section 5 presents experimental results of several leading deep learning methods for small object detection on three benchmark datasets of small objects. Finally, Section VI discusses some future research directions.

2. Overview of deep learning methods for image-based object detection

2.1. Problem definition

The goal of image-based object detection is to detect instances of objects of predefined classes in images and draw a tight bounding box around each object. More specifically, the object detection consists of two tasks: object localization and classification, i.e., finding where objects are located in an image and determining which predefined class each object belongs to.

2.2. Major components of deep learning methods

In this section, we summarize the major components of deep learning methods for image-based object detection, which include backbone networks, region proposals, anchors, object classification, bounding box regression, loss functions, and non-maximum suppression.

2.2.1. Backbone networks

Backbone networks in deep neural network-based object detectors are used to extract high-level features from input images. Most commonly used backbone networks are derived from deep neural network image classifiers that performed well on large-scale image classification datasets, such as the ImageNet classification dataset (Huang, Liu, Van Der Maaten, & Weinberger, 2017; Szegedy et al., 2015; Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016; Newell, Yang, & Deng, 2016; He, Zhang, Ren, & Sun, 2016; Howard et al., 2017; Simonyan & Zisserman, 2014). Typically, the last classification layers are removed from these image classifiers and the rest of the layers are used as the backbone networks. Based on the backbone networks, detection layers are appended to form complete object detectors.

The main design objectives of backbone networks are high detection accuracy and computational efficiency. Some popular backbone networks are as follows.

- VGGNets (Simonyan & Zisserman, 2014) that use small filters of size 3 by 3 pixels in their convolutional layers, followed by 2 by 2 max pooling. VGG16 has 13 convolutional layers, whereas VGG19 has 16 convolutional layers. VGG won the ImageNet Challenge in 2014 and is still one of the most widely used networks.
- Residual networks, or ResNets (He et al., 2016), in which residual blocks were proposed to make training very deep networks possible by overcoming the gradient vanish problem in back propagation by adding a skip connection directly from input of each module. There are several variations of residual networks. The most used versions are ResNet50 and ResNet101. ResNet is much deeper than VGGNet. ResNet won the ImageNet 2015 classification task.
- Inception networks (Szegedy et al., 2015, 2016) that increased the depth and width of networks without increasing computational complexity. The Inception module consists of 1x1, 3x3, and 5x5 filter size convolution layers and max pooling layers stacked parallel with each other. Multiple scales of feature can be extracted simultaneously in one layer. Inception networks are much faster than VGGNet.
- DenseNet (Huang et al., 2017), in which each layer is densely connected to all other layers in a forward manor, so that lower level features are used by all latter layers. DenseNet can alleviate the vanishing-gradient problem.

Many techniques have been developed to improve the backbone networks. For example, Hourglass networks have been designed for capturing multi-scale feature and has been widely used in pose estimation and object detection (Newell et al., 2016). A simple hourglass module consists of convolutional layers and max pooling layers to extract features to a low resolution. After several pooling layers, the network uses upsampling layers and combined features from all different sizes of scaled images. At the end, two 1x1 convolutional layers are applied to generate the final prediction. Stacked multiple hourglass modules followings bottom-up and top-down approaches across different sizes of scaled images. For deep learning models on embedded devices, (Howard et al., 2017) proposed a lightweight network that can run on a mobile device. It replaced a standard convolution layer by a depth-wise convolution and a 1x1 pointwise convolution, and dramatically reduced the amount of computation compared to other deep neural networks. PvaNet (Kim et al., 2016) designed a light and thin feature extraction using CReLU, Inception module and multi-scale outputs.

2.2.2. Region proposals

Major methods for generating region proposals include constrained parametric min-cuts (CPMC), multi-scale combinatorial grouping, selective search and region proposal network (RPN). CPMC (Carreira and Sminchisescu, 2011) is to learn a model to rank the segments. Specifically, a large set of features are extracted from the segments, such as graph features, region features and Gestalt features. The segments are ranked based on their similarity to ground truth. Therefore, the ranking problem is modeled as a regression problem to predict similarity to ground truth. The model is trained on images with various degree of background bias. Then the overlap scores between segments are maximized. Multiscale combination grouping (Pont-Tuset, Arbeláez, Barron, Marques, & Malik, 2016) proposed a hierarchical segmentation method that considers the multiscale information. The image pyramid is constructed based on subsampling and supersampling. Single-scale segmentation is applied on each image resolution. After rescaling and alignment, the different segmentation maps are combined. The combination of multiscale regions is further processed together. Selective Search (Uijlings, Van De Sande, Gevers, & Smeulders, 2013) is a segmentation-based method widely used in object detection. It groups candidate regions hierarchically and generates informative locations containing category-independent objects. However, it is not neural network based, cannot be trained using a dataset, and is slow. RPN (Ren, He, Girshick, & Sun, 2015) is the first CNN based network and can be trained with other detection network. It predicts object regions and object classification confidence scores simultaneously. The input of RPN is an image, and its output contains regions of interests (RoIs) with object scores. Specifically, it uses small networks as sliding windows over the convolutional layers. Each of the sliding windows corresponds to one of the regions in the input image and can be viewed as region proposals with various scales. The features are fed into two prediction layers: classification layer and box regression layer. The classification layer performs binary classification to predict if the region contains any objects or not.

2.2.3. Anchors

Anchors, also called anchor boxes, were first proposed in (Ren et al., 2015). Anchors are a set of pre-defined bounding boxes with various scales and ratios placed regularly on the feature maps. Anchors at different locations on feature maps are projected back to the input images, to be matched with the ground-truth bounding boxes. The stride for each feature map is calculated by H/h and W/w, where H and W are the height and width of an input image, respectively, and h and w are the height and width of a certain feature map, respectively. The scales and ratios of anchors are usually pre-defined to maximize match ground-truth bounding boxes. Some researchers used unsupervised clustering methods to calculate the scales and ratios directly from the training dataset.

During training, anchors are matched with the ground-truth bounding boxes by the IoU (intersection over union) score. Usually, for each bounding box, the anchors with the highest scores or their IoU with ground-truth bounding boxes are higher than a threshold are labeled as positive examples. Those anchors, whose IoU scores with ground-truth boxes are lower than a threshold, are labeled as negative examples. The positive and negative examples are used for training the classifier for object classification. Only the positive examples are further regressed to locate the position of objects. (Wang, Chen, Yang, Loy, & Lin, 2019) proprosed a new anchor sheme called guided anchoring, by using semantic features to dynamicly guid anchor generation. More specifically, the anchors are generated on the feature maps where the prediciotn probability is higher than a certain threshold.

2.2.4. Object classification

The goal of object classification is to predict the class of an object in an image, i.e. predict the probabilities of different class labels for a given region of interest (RoI). Since the localization problem is handled by bounding box regression, the object classification component is similar to a standard classification problem applied to each RoI, instead of the whole images. The object classification and bounding box regression tasks share the same backbone network for generating good results.

2.2.5. Bounding box regression

Bounding box regression is to learn a transformation to map predicted bounding boxes to the corresponding ground-truth bounding boxes. It is either applied to proposals, which is used in the first stage of two-stage object detectors or is used to refine estimated bounding boxes to make them more accurate. In (Girshick, Donahue, Darrell, & Malik, 2014), a linear regression model with the CNN features is trained to better localize bounding boxes. It has four transformation parameters for the center (x and y coordinates) and the width and height of a region proposal. For a region proposal *(P)* and ground truth *(G)*, the four parameters are computed as follows:

$$t_x = (G_x - P_x)/P_w \tag{1}$$

$$t_y = \left(G_y - P_y\right)/P_h \tag{2}$$

$$t_w = \log(G_w/P_w) \tag{3}$$

$$t_h = \log(G_h/P_h) \tag{4}$$

Where $P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$ specifies the pixel coordinates of the center of proposal P^i. Similar $G = \left(G_x, G_y, G_w, G_h\right)$ specifies the ground-truth bounding box. Various bounding box regression loss functions have been proposed. In (He, Zhu, Wang, Savvides & Zhang, 2019), KL loss (Kullback-Leibler) was proposed based on the Kullback-Leibler Divergence of the predicted bounding box distribution and ground truth distribution. In (Lee, Kwak, & Cho, 2018), a bounding box regression neural network was proposed to be trained separately with convolution layers, fully connected layers and ROI-Align layer to minimize IoU loss. (Yu, Jiang, Wang, Cao, & Huang, 2016) proposed the IoU loss for bounding box regression and regress all the bounding box variables together. The IoU loss is not sensitive to scale invariant (Rezatofighi et al., 2019) found that IoU does not provide a strong relationship with minimizing the l_n-norms loss function. Therefore, GIoU is proposed to solve this problem by not only considering the overlap area, but also focus on no-overlapping area. (Zheng et al., 2020) proposed DIoU loss to solve the slow converge issues in earlier work. Moreover, CIoU is further proposed by considering overlap area, central point distance asd well as aspect ratio.

2.2.6. Loss functions

In object detection, multi-task loss functions have been widely used

to simultaneously minimize errors of object classification and bounding boxes regression. For classification, softmax loss has been applied for calculating foreground background classes. For regression loss, SmoothL1, as defined below, has been widely used. This loss is only computed for the bounding boxes predicted as foreground class (i.e. objects).

$$L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geq 1]L_{loc}(t^u,v) \tag{5}$$

$$L_{loc}(t^u,v) = \sum_{i\in\{x,y,w,h\}} smooth_{L_1}\left(t_i^u - v_i\right) \tag{6}$$

where

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 if|x| < 1 \\ |x| - 0.5 otherwise \end{cases} \tag{7}$$

in which p is the probability distribution (per ROI) over all categories, $p = (p_0, \cdots, p_k)p$ is computed by a softmax layer. $L_{cls}(p,u) = -logp_u$ is the log loss for the true class u. The second loss is defined over a tuple of true bounding-box regression for class $u, v = \left(u_x, u_y, u_w, u_h\right)$, and a predicted tuple $t^u = \left(t_x^u, t_y^u, t_w^u, t_h^u\right)$ The L1 loss is used here since it is less sensitive to outliers compared to the L2 loss.

The sentence has been re-written as follows: However, because of the lack of foreground objects in most input images, it is difficult to capture effective patterns and information of foreground objects in the training phase. To overcome this challenge, another type of multi-task loss function has been proposed for object detection to improve the performance. (Lin, Goyal, Girshick, He, & Dollár, 2017). In (Lin, Goyal, et al., 2017), Focal Loss was proposed to address the imbalance problem between foreground and background examples, by down-weighting the easy examples.

$$\text{FL}(p_t) = -\alpha_t(1-p_t)^r log(p_t) \tag{8}$$

Where y is the label, and p if the model's estimated probability. γ is the focusing prarmeter, which reduce the weight on easy examples and α is to control the foreground and background examples balance.

2.2.7. Non-maximum suppression (NMS)

NMS serves as the post-processing step in the inference phase to remove redundant overlapping detection results for the same object. A greedy method is to first sort all detection boxes according to their scores, and then greedily select the detection boxes with the highest scores and suppress other regions if they overlap significantly with the selected boxes, i.e., their intersection-over-union (IoU) score is higher than a designed threshold. NMS is usually applied on each object class independently. In (Rothe, Guillaumin, & Van Gool, 2014), improved solutions were found by treating the problem as a message-passing clustering problem and learning the threshold parameters from training data, instead of pre-defined thresholds. Recently, (Bodla, Singh, Chellappa, & Davis, 2017) and (Hosang, Benenson, & Schiele, 2017) proposed methods with soft thresholds to improve performance. In standard NMS, lower-score regions overlapping significantly with higher-score regions are discarded. However, in soft-NMS (Bodla et al., 2017), lower-score regions are kept in the results. (Hosang et al., 2017) proposed a CNN network to perform NMS by using neighboring regions' detection results to update one region's detection.

3. Major deep learning approaches for image-based object detection

3.1. Anchor-based deep learning approaches

Anchor-based deep learning approaches for object detection can be grouped into two main categories: two-stage object detectors and one-stage object detectors.

Two-stage object detectors, such as Faster R-CNN (Ren et al., 2015), divide the detection problem into two stages: region proposal stage and detection stage. The goal of the region proposal stage is to generate the regions where objects may exist. It outputs the regions locations as well as the object score, 0 or 1, to indicate if there exist objects. In the detection stage, the candidate region proposals are classified into different classes. This stage outputs class probabilities and, optionally, refined region locations. Two-stage detectors achieved state-of-the-art performance; Yet their running speeds were typically slow.

One-stage object detectors, such as YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016) and SSD (Liu et al., 2016), perform region proposal and detection in one deep neural network. Initial regions are pre-defined anchors with various scales and ratios, tiled densely on the image. From the initial anchors, the detectors find those that likely contain objects. Compared with two-stage detectors, one-stage detectors are usually much faster, but achieve less accurate solutions.

Some previous works empirically evaluated the performances of some one-stage detectors and two-stage detectors (Soviany & Ionescu, 2018a, 2018b). It compared the detection accuracy and time between some two-stage detectors and one-stage detectors and concluded that, on average, one-stage detectors asssre faster than two-stage detsssectors, while two-stage detectors tend to be more accurate than one-stage detectors. (Soviany & Ionescu, 2018b) proposed separating images into easy and hard category and training different models on different categories to achieve faster speed and higher accuracy.

3.2. Representative two-stage detectors

R-CNN (Girshick et al., 2014) was the first work transferring deep CNN classification results from ImageNet to object detection. R-CNN adopted the two-stage approach of separate region proposal and object classification. Each region proposal was warped and fed into a deep CNN, and a 4096-dimension feature vector was extracted. During training, the network was first pre-trained on the ImageNet dataset using image level label only. R-CNN made a breakthrough in object detection and improved detection accuracy on the VOC 2012 dataset by more than 30%. This work successfully demonstrated the superior trained CNN features compared with human designed features. However, the CNN feature extraction was applied to each region proposal independently, which made the network very slow.

Fast R-CNN (Girshick, 2015) improved R-CNN by addressing two major issues. First, the training of R-CNN for classification and bounding box regression was done separately in two different stages. Fast R-CNN combined the training for classification and regression. It had two output layers: one for the classification scores and the other for the bounding boxes location offset represented as four values, i.e. the ×, y coordinates of bounding box's center point, and the width and height of the bounding box. A new multi-task loss function was proposed for simultaneously training. Secondly, the feature extraction in R-CNN was applied to each region proposal, which was time and space consuming. Fast R-CNN improved the efficiency by only computing the convolutional feature map once for an entire image. The Region of Interest (RoI) pooling layer was designed to extract fixed size features for different size region proposals. It divided the feature maps into fixed size sub-windows, and max pooled each sub-window to form the fixed size features. Although Fast R-CNN was much faster than R-CNN, it was still based on the traditional region proposal method, which was time consuming.

Faster R-CNN (Ren et al., 2015) further improved the detection speed of Fast R-CNN by replacing the traditional region proposals stage with a convolutional neural network, called Region Proposal Network (RPN). RPN predicted the object region and object confidence scores simultaneously. The main benefit of this design is that the RPN shares the same convolutional layers with the object detection network, which reduces the detection time.

Various anchor sizes captured multi-scale representation and made the computation light weight, compared to image pyramid methods. In

the second stage, the output of RPN were further classified and localized to generate final detection results. Faster R-CNN can be trained end-to-end and the whole network is efficient.

Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017) was proposed as an extension of Faster R-CNN with extra predictions on pixel wise instance segmentation. Mask R-CNN used the Faster R-CNN two-stage pipelines with the same first stage network. In the second stage, it added one extra output, a binary mask for each region proposal, and kept the original classification and bounding box regression. In the loss function used in training, it added one extra term for mask prediction in the form of binary cross-entropy loss. One of the key contributions of Mask R-CNN was the RoI Align layer, which was introduced to fix the misalignment issuses in RoI pooling layer. The idea was to remove the quantization of the RoI boundary, and instead calculating real values. Bilinear interpolation was used to calculate the real feature value on each sampling point. The results were averaged or max pooled from the sampling points. Mask R-CNN achieved the state-of-the-art on instance-level segmentation.

Feature pyramid network (FPN; Lin, Dollár, et al., 2017) focused on solving the problem that lower level feature maps contain more spatial information but less semantic information, whereas the latter layers of a deep neural network contain more high-level semantic information but less spatial information. FPN utilized the hierarchy of a CNN network and implemented a bottom-up and top-down path with lateral connections. In the bottom-up part, an input image was passed through a CNN and pooling layer was used to shrink feature maps size. In the top-down part, the feature maps were up-sampled back into the same size as in the bottom-up part. Moreover, the lateral connection fused the feature maps in bottom-up path and top-down path of same sizes with element-wise addition. FPN generated integrated feature maps that dramatically improve detection accuracy, especially for small objects.

3.3. Representative one-stage detectors

YOLO (Redmon et al., 2016) focused on improving the speed of object detector. It treated the object detection problem as regression problem and removed the region proposal stage in two-stage detectors. Instead of using pre-defined anchors for object region, it divided input images into 7*7 cells and each cell was used to predict the object's center falling into the cell. Each cell predicted bounding box locations, a score for each bounding box, and class probabilities. The network was implemented as convolutional layers followed by fully connected layers. The sum of squared error loss was used to minimize localization and classification error. YOLO was a real-time object detector with 45 frames per second detection speed, which was extremely fast compared to other detectors. However, class probabilities were only predicted within each cell. It does not work well on objects partially located in one cell, could not handle a wide distribution of ground truth objects, and has difficulty to predict bounding box scales and ratios precisely, which results in a low localization accuracy.

YOLOv2 (Redmon & Farhadi, 2017) proposed several improvements on YOLO. In order to increase recall, it removed the fully connected layers and adopted the anchor boxes concept to predict bounding boxes. Unsupervised learning methods were applied to generate bounding box scales and ratios directly from training data. Instead of only predicting one class probability per cell, it predicts both objectness and class for each bounding box, which improved the performance on detecting partially covered objects. For the bounding box regression, it predicted the location relative to the cell left top location, which made the prediction bounds to be between 0 and 1. Other proposed techniques included batch normalization, high-resolution classification, and multi-scale training. All the techniques dramatically improved detection accuracy while keeping the fast speed.

YOLOv3 (Redmon & Farhadi, 2018) proposed more improvement over YOLOv2. For class prediction, it used binary cross-entropy loss instead of softmax loss, to handle the case of multiple classes in one bounding box. It adopted the multi-scale framework and feature pyramid to predict objects in 3 different scales. A new backbone network with ResNet module was proposed for improved speed and accuracy, especially on small object detection.

SSD (Liu et al., 2016) was a single shot detector without a region proposal stage, as shown in Fig. 1. Different from Faster R-CNN that only used the last layer for detection, SSD performed detection using multiple layers to better capture multi-scale objects. Since anchors were applied to multiple feature maps, SSD designed various anchor scale ranges between layers. The lower layers had smaller scales and higher layers larger scales. This design could handle a wide range of objects, resulting in higher recall rate. Different from YOLO that used the fully connected layers for object detection, SSD used fully convolutional layers to predict confidence score and localization offset. With some additional data augmentation and hard negative mining techniques, SSD achieved the state-of-the-art performance on several benchmark datasets. However, SSD performed poorly on small objects, due to shallow layers without deep semantic information.

DSSD (Fu, Liu, Ranga, Tyagi, & Berg , 2017) improved SSD by using a larger network. Their experimental results showed the deep and powerful backbone network ResNet-101 outperformed the VGG network. A deconvolutional module was introduced to add more context information. More importantly, the deconvolutional layers were trainable during training, making DSSD more flexible and achieving better performance. In order to improve the accuracy of anchor scales and ratios, K-means clustering are applied to group training boxes with squared root boxes areas as the distance measurement. DSSD improved the SSD accuracy, especially for small objects.

RetinaNet (Lin, Goyal, et al., 2017) was proposed as a one-stage object detector to reduce the detection accuracy gap with existing two-stage detectors while maintaining fast detection time. This work found that the accuracy gap between one-stage detectors and two-stage detectors was mainly due to the numbers of the positive examples and negative examples as well as easy examples and hard examples used in training were highly unbalanced. The large number of easy examples dominated the loss function, which resulted in a degenerated model. This problem was solved by introducing a new loss function, called focal loss function to reduce the weights of the easy examples adaptively.

4. Challenges and solutions for small object detection

In this section, we identify four major challenges of applying deep neural networks to small object detection and discuss existing solutions.

4.1. Challenges for small object detection

In this section, we summarize the four major challenges for small object detection.

4.1.1. Challenge 1: Individual feature layers do not contain sufficient information for small object detection.

Deep CNN architectures provide hierarchy feature maps due to pooling and subsampling operations, resulting in different layers of feature maps containing different spatial resolutions. It is well known that in the early-layer feature maps, the feature maps are of higher resolution and represent smaller reception fields. At the same time, they do not contain high-level semantic information that is important for object detection. On the other hand, the latter-layer feature maps contain stronger semantics information, which is essential for identifying and classifying objects, including different object poses or illuminations. Even though higher-level feature maps are useful for identifying large objects, they may not be sufficient for small object detection. After down sampling several times in the deep CNN architectures, the latter feature maps lose spatial information. A small object of size 32x32 pixels is clearly visible in earlier (or shallower) feature maps, but not in the

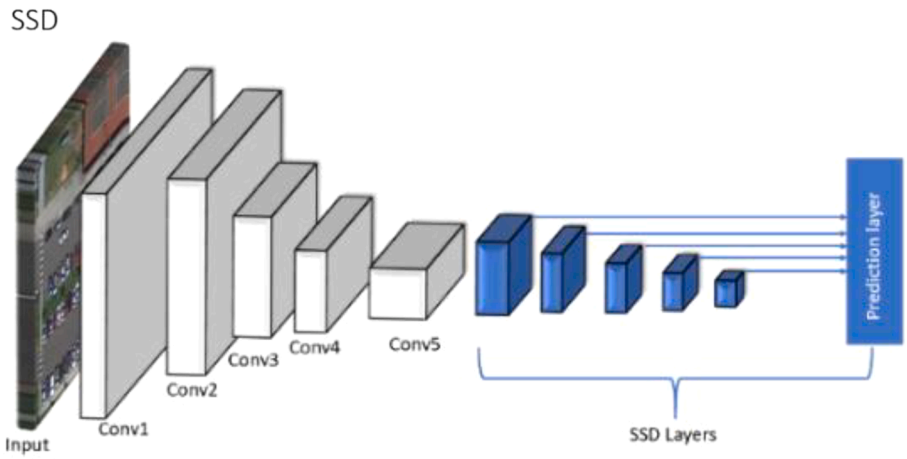

Fig. 1. Architecture of SSD framework.

latter (or deeper) feature maps. Therefore, low-level features alone or high-level features alone are not sufficient for small object detection.

Solution: Combining features from shallow layers and deep layers. To better detect small objects, several deep CNN based methods combine lower-level feature maps and higher-level feature maps together to obtain necessary spatial and semantic information. There are two main approaches to feature map fusion:

1) Bottom-Up Scheme

This scheme is incorporated into the standard feedforward CNN architecture. From early to latter layers, feature maps shrink after pooling operations. The final detection layers directly combine several bottom-up feature maps.

2) Top-Down Scheme

This scheme can be viewed as an attention mechanism that propagates higher-level semantic information back to lower-level feature maps. It usually uses a convolution-deconvolution or encoder-decoder network with an upsampling operation in the decoder to enlarge the feature maps' spatial resolution. Moreover, skip paradigm or lateral connection are often used to connect lower-layer with higher-layer feature maps while bypassing intermediate layers. The fused feature maps are used by detection layers. Typical operations to combine feature maps include summation, production, concatenation, and global pooling.

4.1.2. *Challenge 2: Limited context information of small objects.*

Usually small objects are in low resolutions and it is difficult to recognize low-resolution objects. Since small objects themselves contain limited information, contextual information plays a critical role in small object detection (Torralba, Murphy, Freeman, & Rubin, 2003; Oliva & Torralba, 2007; Divvala, Hoiem, Hays, Efros, & Hebert, 2009; Palmer, 1975). Contextual information has been used in object recognition from a "global" image level to a "local" image level. A global image level considers image statistics from the entire image, whereas a local image level considers contextual information from neighbor areas of the objects. Context features could be categorized into three types (Divvala et al., 2009):

1) Local pixel context: The patches or pixels around an object, such as edges, colors, textures, etc. Local pixel context could be captured by increasing the size of the detection window in object detection networks.
2) Semantic context: The probability of an object to be identify in some surrounding scenes, such as events, activities, or scene categories.
3) Spatial context: The spatial location of other objects in the image, e.g. the likelihood of finding an object in some positions in respect to other objects in the image. For example, in face detection systems, the subject's shoulder and neck are always close to their face.

Solution: Incorporating contextual information in the detection network. The local pixel context is usually added by enlarging filter sizes to capture extra information around the objects. The semantic context is usually added by extracting deeper features from images, such as in the deconvolution layers or recurrent neural networks (RNNs).

4.1.3. *Challenge 3: Class imbalance for small objects.*

Class imbalance refers to the uneven data distribution between classes. There are two types of class imbalances. One is imbalance of foreground and background examples. In object detection, region proposal networks are used to generate the candidate regions containing objects, by densely scanning the entire image. The anchors are pre-defined rectangular boxes densely tiled on the entire input image. The scales and ratios of anchors are pre-defined based on the target objects' sizes in the training dataset. To detect small objects, there is an increase of anchors generated per image compared to detecting large objects. Only the anchors with high Intersection over Union (IoU) with the ground truth bounding boxes are labeled as positive examples. Since most anchors have low or no overlap with the ground truth bounding boxes, they are considered as negative examples. When densely generated anchors are matched with sparsely located real objects in the images, positive examples are a tiny fraction, resulting in a high-class imbalance, e.g. class ratio from 100:1 to 1000:1.

The anchor-based object detection approach has several drawbacks. First, due to the sparseness of ground-truth bounding boxes and the IoU matching strategies between ground-truth and anchors, negative examples highly dominate positive examples, which leads to models favoring the negative class. Second, the dense sliding window strategy has high time complexity, ($O(h^2w^2)$), where h is the height and w is the width of the anchors, which makes training slow.

Solution: Balance positive and negative examples in training. There are two main strategies: 1) data-based and 2) loss function-based. The data-based strategy is to change the foreground and background example numbers to make the examples of the positive and negative class roughly carry the same weights. Hard sampling and soft sampling are two popular methods. Hard sampling selects a subset of samples, whereas soft sampling assigns different weights to examples. For example, random sampling is commonly used to randomly select examples to meet a certain ratio. Another sampling strategy is to sample more of the hard examples with large losses. For example, a machine learning model could be trained first and then the false positives are considered as hard examples, which are weighted heavily in the second round of training. The recently proposed Online Hard Example Mining

(OHEM) method (Shrivastava, Gupta, & Girshick, 2016) performs one forward pass on the calculated region of interest (RoI) and computes losses for all RoIs. Then, examples are ranked based on their loss function values, and the examples with the largest loss are selected to be used in the next round of training since the current trained network model performs the worst on them. (Pang, Chen, et al., 2019) also proposed an IoU-balanced sampling technique in order to sample more training examples from difficult cases. In terms of soft sampling, (Cao, Chen, Loy & Lin, 2020) proposed a technique that selects samples based on their importance where the importance of positive examples is measured by their IoU scores with the ground truth bounding boxes, and the importance of negative examples is calculated by considering both local region and global region properties.

Loss function-based strategies will re-weight examples of imbalanced classes in the loss function in order to balance the foreground and background examples. For example, AP loss (Chen et al., 2019) uses an average-precision loss to re-weight examples. DR Loss (Qian, Chen, Li, & Jin , 2019) re-weights examples based on the distribution of foreground examples over the distribution of background examples.

4.1.4. Challenge 4: Insufficient positive examples for small objects

Most deep neural network models for object detection were trained using objects of various scales. They typically perform well on large objects, but poorly on small objects. Reasons may include an insufficient amount of small-scale anchor boxes generated to match the small objects and an insufficient number of examples to be successfully matched to the ground truth. The anchors are regions in the feature maps of some intermediate layers in a deep neural network, which would be projected back to the original image. It's hard to generate anchors for small objects. Moreover, the anchors need to be matched to the ground truth bounding boxes. A widely used matching method is as follows. If an anchor has a high IoU score with respect to a ground truth bounding box, such as larger than 0.9, it is labeled as a positive example. In addition, the anchor with the highest IoU score with respect to each ground truth box is also labeled as a positive example. Therefore, small objects usually have very few anchors match with the ground truth bonding boxes, i.e. very few positive examples.

Solution: Use methods that can generate more anchors for small objects and match more anchors with small objects. Existing techniques include:

1) Multi-scale mechanism. Multi-scale architectures consisting of separate branches for small, medium, and large-scale objects can generate anchors of different scales.

2) *Matching strategy*. Adaptively setting anchor scales and ratios to help more anchors match to ground truths of small objects.

3) *Increasing positive examples of small objects*. In the region proposal stage, generate more anchors by allowing them to overlap with each other.

4.2. Deep learning techniques for small object detection developed in generic object detection research

In this section, we summarize the deep learning techniques developed in generic object detection research that are effective for small object detection.

4.2.1. Technique 1: Improve feature maps for small objects

Low-level features are important for localization, whereas high-level features are important for classification. Using a combination of low-level and high-level feature maps at the detection layers in several deep neural networks has led to an improvement of results for small object detection.

Using a bottom-up fashion, one group of network architectures merged feature maps from different layers (Fu et al., 2017; Lin, Milan, Shen, & Reid, 2017; Kong et al., 2017). Kong, Sun, Tan, Liu, and Huang (2018) proposed a non-linear feature map transformation by considering both global and local information. The parameters for the non-linear transformation were learnable and shareable with different layers. The transformations were applied to feature maps in different layers and each transformed layer generated detection results. (Yang, Liu, Yan, & Li, 2019) used deconvolutional layers in an "encoder-decoder" architecture in addition to the feature maps from convolutional layers and deconvolutional layers were combined. (Bell, Lawrence, Zitnick, Bala & Girshick, 2016) used skip connections to directly add lower-level feature maps to higher-level feature maps. Features were pooled from convolutional layers with different receptive fields (conv3, conv4, and conv5). These features were normalized and concatenated to be fed into detection modules. (Zagoruyko et al., 2016) also concatenated features from different layers (conv3, conv4, conv5) after normalization. In (Cao et al., 2018), extensive experimental results showed that combining low-level and high-level features improved detection accuracy of small objects. (Jeong, Park, & Kwak, 2017) concatenated features by performing pooling and deconvolution simultaneously, where pooling decreased the size of the low-level feature maps to combine them with high-level feature maps and deconvolution increased the high-level feature maps to combine them with low-level feature maps. (Yu et al., 2018) fused both semantic and spatial information by using Iterative Deep Aggregation (IDA) and Hierarchical Deep Aggregation (HDA). IDA non-linearly two types of features, whereas HDA merged several CNN features in a tree structure. (Zhang, Qiao, et al., 2018) used an extra segmentation module to add more semantic information. (Li & Zhou, 2017) fused features from multiple lower layers, where low-level feature maps were down sampled with max pooling and high-level feature maps were resized with bilinear interpolation. Similarly, (Kong, Yao, Chen, & Sun, 2016) applied max pooling on low-level features and deconvolutional operation on high-level features. Then these feature maps were concatenated together to form the so-called Hyper Feature Map.

Some network architectures used both top-down and bottom-up connections for combing features. ((Shrivastava, Sukthankar, Malik, & Gupta, 2016) added a top-down module as well as lateral connections. The top-down module can generate features with more semantic and contextual information. The lateral connection can help to enrich the top-down features by transmitting lower-level features. (Kong et al., 2017) used reverse connections to add high-level semantic information from latter network layers back to earlier layers. (Ghiasi, Lin, & Le, 2019) proposed a search method to automatic search for good feature pyramid architectures which may make it possible to replace manual design. A recurrent neural network (RNN) served as the controller to merge any two input features with sum or pooling operation. (Pang, Wang, Anwer, Khan, & Shao, 2019) applied the feature pyramid fusion mechanism at two levels. For global information, it constructed an image pyramid and combined the features from four levels of the image pyramid with the original features from the standard SSD framework. For local spatial information, features from both the previous and the current layers were fused together.

4.2.2. Technique 2: Incorporate context information of small objects

Context information of small objects can be divided into local contact and sematic context information. More local context information could be included in deep neural networks through larger bounding boxes and proposal boxes. (Cai, Fan, Feris & Vasconcelos, 2016) added extra local context with bounding boxes 1.5 times of object regions, which was useful to include more of the surroundings of small objects. The extra context information was combined with object features for detection layers. (Zagoruyko et al., 2016) incorporated local context information by increasing the sizes of region proposal boxes with four scales, 1x, 1.5x, 2x, and 4x. The outputs from four regions were pooled with ROI-pooling and concatenated before being fed into the detection and classification layers.

For including more semantic context information, (Fu et al., 2017) used deconvolutional layers with "skip connections", resulting in better

detection results on small objects. Instead of simply stacking deconvolutional layers on top of convolutional layersthe deconvolution layers were designed to be much shallower than convolutional layers, and an element-wise product was used. Similarly, (Cai et al., 2016) also used deconvolutional layers to increase the resolutions of feature maps. (Bell et al., 2016) used four Recurrent Neural Network (RNNs) to capture global information of the input image. The RNNs added semantic context information around the objects and 1x1 convolutions combined all information together. In addition, (Zhang, Wen, Bian, Lei, & Li, 2018) passed higher-level feature maps to lower level features.

4.2.3. Technique 3: Correct foreground and background class imbalance for small objects

Techniques for improving the foreground and background class imbalance can be divided into two part, which are a data-based approach and a loss function-based approach.

In the data-based approach, (Cai et al., 2016) addressed class imbalance using bootstrap sampling that sampled negative examples based on their loss values. (Zhang, Wen, et al., 2018) used a two-step regression method to balance the foreground and background examples. Some easy negatives were dropped to make the ratio between positives and negatives about even. (Kong et al., 2017) implemented an objectness prior to filtering out the bounding boxes without objects. In the loss function-based approach, (Galleguillos & Belongie, 2010) used a loss function that gave hard-negative examples greater weights.

4.2.4. Technique 4: Increase training examples for small objects

Many techniques have been developed to increase the amount of training examples for small objects. They include neural network architectures for multi-scale learning, scale transformation, and adaptive matching of anchor boxes.

Several neural network architectures have been designed for multi-scale learning, i.e. training detector networks for objects of different sizes, to addressing the problem of insufficient examples of small objects in training classifiers. (Singh, Najibi, & Davis, 2018) showed the effectiveness of a training scheme that used objects of various scales and poses to increase the detection performance on small objects due to increased quantity and variety of training examples. Small objects were up sampled and fed into convolutional networks for detection. Only the layers whose feature maps contained target objects within a certain range were activated during training, so that small objects could be trained equally well as medium and large objects. (Najibi, Singh, & Davis, 2019b) proposed a multi-scale network to predict the regions most likely containing small objects and discarded the regions that did not likely contain small objects. The network first predicted a binary segmentation map for small objects, aiming to achieve a large recall rate for small objects. Only the regions that likely contained small objects were used in training. (Yang, Choi, & Lin, 2016) proposed a technique called scale-dependent pooling to assign the appropriate feature maps to objects based on object scales. This approach was based on the idea that small objects mainly have strong activations in lower-level feature maps. This method pools the earlier feature maps for small objects and latter feature maps for large objects. Several other works added prediction and detection layers after different convolutional layers, i.e. using different feature maps for prediction, and enlarged the region sizes to include more local context information (Cai et al., 2016; Li & Zhou, 2017; Chen, Liu, Tuzel, & Xiao, 2016). Scale transformation is another technique used to increase the amount of training examples for small objects (Kim, Kang, & Kim, 2018). Objects of different scales are mapped onto a single scale-invariant space in order to increase the number of examples. Learning the mapping from different scales in the original input images are used to normalized patches. Therefore, images containing objects of all scales can be used in training. (Singh & Davis, 2018a) propsed a novel training scheme called scale normalization for image pyramids (SNIP) that can minimize the scale changes during training.

Instead of pre-defined anchor boxes, machine learning has been applied to find perfect scales and ratios for anchor boxes and to perform adaptive matching of anchor boxes (Redmon & Farhadi, 2017). For example, ground truth bounding boxes can be grouped into clusters based on their scales, and then anchor boxes are matched to ground truth bounding boxes of similar scales.

4.3. Techniques for small object detection developed in face detection research

Face detection has been extensively studied and has achieved great success. Faces have distinct facial features, i.e. the nose, eyes, mouth, and their relative positions with respect to each other, which make face detection different than generic object detection. However, it remains challenging when face sizes are very small, e.g. smaller than 16x16 pixels, and features are not distinguishable. Many techniques have been developed for small face detection. The state-of-art algorithms shown in Table 1.

4.3.1. Technique 1: Improve feature maps for small faces

One of the techniques for combining multiple feature maps is the skip connections used to integrate lower, middle, and higher layer features. (Tian et al., 2018) proposed an iterative feature map generation scheme, which generated features in six different scales, and all feature maps from the backbone network were fed back to the beginning of the network to extract more semantic information for small objects. (Samangouei, Chellappa, Najibi, & Davis, 2018) fed the combined lower and higher layer features to a ROI-based block normalization layers. (Tian et al., 2018) merged four feature maps using skip connections to generate four new feature maps for detection. Specifically, the input features were fused with the next level features by element multiplication. Furthermore, the fused features were added to the original input features to form the final features, which proved to be effective for detecting difficult tiny faces. (Zhu, Zheng, Luu, & Savvides, 2017) combined the lower level features with higher level features by first downsampling the lower level features to the size of higher-level features and then concatenating them with L2 normalization. (Luo, Li, Zhu, & Zhang, 2019) combined the lower level features with neighboring features by using a bilinear upsampling strategy. (Yoo, Dan, & Yun, 2019) designed a feature map generation scheme by recurrently passing the network.

4.3.2. Technique 2: Incorporate context information of small faces

For face detection, (Bai, Zhang, Ding, & Ghanem., 2018) showed that adding context information can improve the performance dramatically for small faces. However, too much context information can also hurt the performance on small faces due to over-fitting. Some methods added context information by enlarging the receptive fields around faces. (Najibi, Samangouei, Chellappa, & Davis, 2017) adopted larger filter sizes, e.g. 5x5 and 7x7 filters in the convolutional network, instead of 3x3 filters. (Wang, Yuan, & Network, 2017) adopted the feature fusion method to combine lower level features with higher level features with an agglomeration connection module. In this module, lower feature maps first pass through an inception like network to increase the semantic information, then it concatenates lower feature maps with higher feature maps. (Samangouei et al., 2018) added context information around each bounding box. In (Tang et al., 2018), extra context information from bodies and shoulders were integrated and a semi-supervised method was used to generate labels for other body parts. (Tian et al., 2018) used a segmentation branch to add extra context and semantic information, without extra annotations, and the segmentation branch shared the same receptive field of detection, which made the segmentation branch an extra source for more discriminative features. (Li, Tang, Han, Liu, & He, 2019) used the structure of the dense block from (Huang et al., 2017) to integrate extracted context features. (Zhu et al., 2017) integrated body information to reduce false positives. The body features were acquired by additional RoI-pooling operations to

Table 1
Face Detection State-of-the-Art Algorithms.

Face Detection						
Network	One-Two stage	Backbone	Strategy	Wider Face validation Set		
				Easy (mAP)	Medium (mAP)	Hard (mAP)
Tiny Face	One-stage	ResNet101	Context reasoning	0.919	0.908	0.823
SSH	One-stage	VGG16	Feature fusion/context reasoning	0.931	0.921	0.845
SRN	Two-stage	ResNet50	Anchor matching /balance classes	0.957	0.946	0.884
S3FD	One-stage	VGG16	Anchor matching /balance classes	0.937	0.924	0.852
EXTD	One-stage	Inverted Residual	Feature fusion	0.912	0.903	0.85
DF2S2	One-stage	ResNet50	Feature fusion/context reasoning	0.969	0.959	0.912
PyramidBox	One-stage	FPN	Feature fusion/context reasoning	0.961	0.95	0.889
PyramidBox++	One-stage	FPN	Context reasoning/balance classes	0.965	0.959	0.912
FA-RPN	One-stage	RPN	Anchor matching /balance classes	0.95	0.942	0.894
Face-MegNet	Two-stage	VGG16	Context reasoning/feature fusion	0.92	0.913	0.85
SFA	One-stage	VGG16	Anchor matching/feature fusion	0.949	0.936	0.866
Face-RCNN	Two-stage	VGG19	Anchor matching /balance classes	0.938	0.922	0.829
RetinaFace	One-stage	–	Context reasoning	0.969	0.961	0.92
RAP	One-stage	Dilated convolutional	Anchor matching	0.949	0.935	0.865

enlarge the receptive fields. The combined face and body features were used in both classification and bounding box regression.

4.3.3. Technique 3: Correct foreground and background class imbalance for small faces

For small face detection, detector networks usually place a lot of small anchors on the images which will usually generate a lot of negative anchors and very few positive anchors, resulting in a high false positive rate. There are two main approaches to handle class imbalance.

a) Filtering anchors. (Zhang et al., 2017) used a max-out background label, which predicted several scores for background labels and selected the largest as the final score. (Chi et al., 2019) used two-step classification on the lower layers to filter out false positives for small faces, which helped to balance the positive and negative examples to improve the classification results.

b) Sampling. (Zhang et al., 2017) applied hard-negative mining to make the ratio between negatives and positives at most 3:1. (Najibi, Singh, & Davis, 2019a) also used hard-negatives mining. Anchors were labeled positive if the overlap with the ground-truth bounding box was larger than 0.5. (Li et al., 2019) proposed a balanced-data-anchor-sampling strategy to select large size and small size anchors with equal probability. (Tang et al., 2018) proposed a Pyramid box that adopted the max-in-out technique on both positive and negative samples to reduce the false positive rate of small objects. (Wang, Li, Ji, & Wang, 2017) applied the online hard exampling mining (OHEM) by sorting the examples based on loss and selecting the top examples with the highest loss as the hard examples. Also, they used a 1:1 ratio for positive hard examples and negative hard examples in each mini batch during training.

4.3.4. Technique 4: Increase training examples for small faces

It is important to make sure that small objects have sufficient anchors to match with, otherwise small objects cannot be trained well, and the trained model's recall will be low. There are three main techniques for increasing training examples for small faces.

a) Matching strategy. (Zhu, Tao, Luu, & Savvides 2018) found the average IoU for small faces and small anchors, which are much lower than large faces. In order to design anchors to match with more small scale objects, a new matching score was proposed to consider face scales and anchor strides so that small face scales can also achieve high IoU scores. Moreover, during training, faces were randomly shifted to match with more anchors. (Chi et al., 2019) identified the mismatch between anchors ratios and receptive fields and proposed inception-styled feature maps to increase the diversity of feature map scales and reduce the mismatches between faces and anchors.

b) Increasing anchors. (Zhang et al., 2017) added extra convolutional layers to generate more anchors for small objects. It also reduced the stride sizes on the lower anchor-associated layer to increase the number of anchors that can potentially match with more small scale objects. Moreover, it proposed a two-stage anchor matching strategy to make sure that each small object has enough anchors to match with. (Zhu et al., 2018) increased the number of anchors by reducing stride and the distance between the face and anchor center, as well as adding extra shifted anchors. Moreover, for the hard faces for which the highest IoU scores were still lower than the matching threshold, the top few anchors with highest scores were selected as positive examples. (Luo et al., 2019) increased the ranges of anchors so that more anchors with small sizes can be matched with small faces.

c) Multi-scale training. (Wang, Chen, Huang, Yao, & Liu, 2017) resized input images to different sizes to generate objects of various sizes and small objects can be resized to larger objects to match more anchor boxes. (Najibi et al., 2017) designed a multi-scale network with three different convolutional branches intended to detect different scales of faces: small, medium and large faces, respectively. Small faces were detected by an element sum feature from conv4 and conv5. Medium faces were directly detected from conv5. Large faces were detected from max pooling after conv5. (Hu & Ramanan, 2017) used an image pyramid, where the input images were scaled to ratio 0.5, 1, and 2 of the original resolution. Then two types of feature maps were applied to capture different scales of faces.

4.4. Techniques for object detection in aerial images

For detecting objects in aerial images, there are mainly four kinds of methods: (i) template matching-based, (ii) knowledge-based, (iii) OBIA-based, and (iv) machine learning-based (Cheng & Han, 2016). In recent years, deep learning based methods achieved the best performance. Commonly, CNNs pretrained on large image datasets, such as the ImageNet and COCO dataset, were fine-tuned on aerial images. In addition, new deep neural networks were proposed for the unique attributes of objects in aerial images, like multi-scale and multi-angle, to achieve better performance. For example, (Dong, Liu, & Xu, 2018) proposed rotation-invariant models to achieve good performance on remote sensing images. Moreover, weakly supervised learning methods (Peng et al., 2018) has been proposed to learn high-level features in an unsupervised manner to capture the structural information of objects in remote sensor images.

4.4.1. Technique 1: Deal with orientation of aerial image objects

Objects in aerial image can have arbitrary orientations or rotations. Deep neural network-based detectors have been designed to address this issue. Rotation-Invariant CNN (RICNN) in (Cheng, Zhou, & Han, 2016) a new rotation-invariant layer in the basic CNN architecture. introduced Rotation-Invariant and Fisher Discriminative CNN (RIFD) in (Cheng,

Han, Zhou, & Xu, 2018) was proposed to contain a rotation-invariant regularizer and fisher discrimination regularizer on multi-scale features from CNN. The rotation-invariant regularizer mapped the CNN feature representations of training samples before and after rotations to be similar, while the fisher discrimination regularizer constrained the CNN features to be similar for within-class examples, but dissimilar for examples in different classes. Recently, anchor rotation methods have been proposed in one-stage object detectors to achieve rotation invariance (Yang et al., 2018). A feature refinement technique was proposed to improve detection performance on aerial images, in which the position information of bounding boxes was encoded to the corresponding feature points through feature interpolation to improve feature reconstruction and alignment. R-Net in (Yang et al., 2018) proposed a network to generate rotatable region proposals.

4.4.2. Technique 2: Incorporate context information of aerial image objects

More context information for small objects have been included in the detection networks through combined feature maps and dilated convolutions. Feature maps from multiple convolutional layers can be concatenated to form a new feature map. Dilated convolutions can be added to CNN models to improve performance on small-scale object detection.

4.4.3. Technique 3: Correcte foreground and background class imbalance for aerial image objects

Based on Faster RCNN, IoU-Adaptive Deformable R-CNN in (Yan et al., 2019) was proposed to address the class imbalance issue in training classifiers in object detectors. By analyzing the different roles that IoU can play in different parts of the network models, an IoU-guided detection framework was proposed to reduce the loss of small object information during training. Besides, an IoU-based weighted loss was designed to learn the IoU information of positive ROIs to improve the detection accuracy. Finally, the class aspect ratio constrained non-maximum suppression (CARC-NMS) was proposed to improve detection precision.

4.4.4. Technique 4: Increase training examples for aerial image objects

Multi-scale network models have been proposed to detect objects of various sizes in aerial images, such as Multi-Scale and Rotation-Insensitive Convolutional Channel Features (MsRi-CCF) in (Wu, Hong, Ghamisi, Li, & Tao, 2018). MsRi-CCF was proposed for geo-spatial object detection by integrating robust low-level feature generation, classifier generation with outlier removal, and detection with a power law.

4.5. Instance segmentation methods for small object detection

Different from the popular bounding-box-based object detectors presented in the previous sections, deep CNNs for instance segmentation have also been applied to object detection. The main drawbacks of segmentation methods are pixel-wise labeling, which is time consuming, and compute and memory intensive. For small object detection, each pixel of an object is important and using pixel information could generate good results.

FCN (Long, Shelhamer, & Darrell, 2015) is one of the first methods that use CNNs for semantic segmentation. FCN employs CNNs without fully connected layers, which allows the input image to have an arbitrary size. It uses pooling layers to reduce computation time and increase the reception field size. Based on FCN, U-Net (Ronneberger, Fischer, & Brox, 2015) was proposed with an encoder-decoder architecture to address the issue of determining appropriate numbers of pooling layers. It has a U-shape architecture to balance the trade-off between good localization accuracy and efficient context information. In the encoder, it uses pooling layers to gradually reduce the layer size, whereas, in the decoder stage, it uses up-convolution to gradually increase the layer size. Moreover, U-Net uses short-cut connections from encoder to decoder to help the decoder recover fine-grain information. Regarding the trade-off between reception field and localization accuracy, large reception fields lead to lower localization accuracy. However, when the reception field is too small, localization accuracy may also decrease due to the lack of context information.

Feature Pyramid Networks (FPNs) combine FCN and Faster R-CNN. On top of the two predictions that Faster R-CNN generates: (i) bounding box localization and (ii) bounding box recognition, FPNs added the third output, (iii) instance mask prediction for segmentation. FPNs also used some new techniques, such as new ROI align layers, multitask training and better backbone networks for further improvement.

For small object detection, some other techniques have also been proposed based on segmentation methods. To include more context information, capsule networks with deconvolutional capsules were proposed to expand the original layers in the network architectures (LaLonde & Bagci, 2018). Segmentations could be refined using bottom-up and top-down network architectures to combine features from different layers (Ronneberger et al., 2015), or using pyramid pooling layers to segment objects in multiple scales as in DeepLab (Chen et al., 2018). To increase training examples for small objects, a more robust embedding could be learned by jointly using unsupervised and supervised learning and combing features from different models to form a multi-scale representa tion (Lin, Milan, et al., 2017). Concept Mask in (Wang Lin, Shen, Zhang, & Cohen, 2018) used a semi-supervised learning method to train a deep neural network with image-level labels. Then, the results were refined and extended to predict attention maps. Finally, an attention-driven class segmentation network was trained.

5. Performance evaluation of deep learning methods for small object detection

In this section, performances of representative state-of-the-art object-detection methods on widely used public benchmarkdatasets are presented. The emphasis is on small object detection.

5.1. Dataset

We used several datasets from three different areas: generic object detection, face detection, and object detection in aerial imagery. Images in the generic object detection dataset were mostly collected under everyday living and indoor settings. The objects are usually of rigid shapes. The difficulty of detection usually comes from illumination and background clutter. In comparison, faces have a common structure containing several regions of fixed parts, such as eyes, noses, mouths, etc., and the relationships between parts are known a prior. In terms of aerial images, they were collected under much diverse conditions, such as camera mounted underneath airplanes, helicopters or UAS (drones), which resulted in straight-down views of the objects, very different viewpoints from images in generic object detection or face detection datasets. We showed the examples of three datasets in Fig. 2.

1) Generic object detection dataset. A combination of images from the Microsoft Common Object in Context (COCO) dataset (Lin et al., 2014) and the SUN dataset (Xiao, Hays, Ehinger, Oliva, & Torralba, 2010) were used in the experiments. COCO consists of 82 K training and 40 K validation images belonging to 80 classes. COCO is a widely used dataset in object detection and a relatively difficult dataset, since the objects sizes are relatively small compared with other datasets. For our experiments, we selected ten small object categories from COCO, where the largest physical dimension is smaller than 30 cm. The selected object categories are mouse, telephone, switch, outlet, clock, toilet paper, tissue box, faucet, plate and jar. Then, we used the ground truth bounding boxes in the COCO and SUN datasets to filter out large objects to create a dataset containing small objects with small bounding boxes. Table 2 shows the statistics of this small object dataset used in our experiments. It contains about 8393 object instances in 4952 images. The mouse category has the largest number of object instances: 2,173 instances in

Fig. 2. Examples of small object subsets of three benchmark datasets used in our experiments for object detection: (a) DOTA, (b) WIDER FACE, and (c) COCO and SUN. All the objects used in our experiments are smaller than 50 × 50 pixels.

Table 2
Three benchmark datasets used to evaluate the performances of representative small object detectors in our experiments.

Dataset	# train	# test	# category	Examples
DOTA	4156	1186	15	Plane, ship, vehicle, harbor, …
WIDER FACE	2730	708	1	Face
COCO + SUN	3655	1560	10	Mouse, telephone, outlet, faucet, …

1,739 images. The tissue box category has the fewest instances: 103 instances in 100 images. The object instances in the dataset are small. The median of relative areas of all the object instances is from 0.08% to 0.58%. As a comparison, the median of relative areas of objects in the PASCAL VOC dataset is from 1.38% to 46.40%. This dataset is challenging in two ways. First, the appearance cue for distinguishing a small object from background clutters is much less due to the small size. Second, the number of bounding box hypotheses for a small object in an image is much larger than that for a big object in VOC.

2) Face detection dataset. We used Wider Face dataset (Yang, Luo, Loy, & Tang, 2016) in our experiments. This dataset contains 32,203 images containing 393,703 annotated faces, 158,989 of which are in the train set, 39, 496 in the validation set and the rest are in the test set. The validation and test set are divided into “easy”, “medium”, and “hard” subsets. This is one of the most challenging public face datasets mainly due to the wide variety of face scales and occlusion. We trained all models on the train set of the WIDER dataset and evaluated their performances on the validation and test sets. Moreover, a subset of small objects was created containing objects smaller than 50x50 pixels.

3) Aerial imagery dataset. The Large-scale dataset for Object Detection in Aerial images (DOTA; Xia et al., 2018) was used in our experiments. It is one of the largest annotated object datasets of aerial images, containing 2,806 high-resolution aerial images with 188,282 object instances of various scales and shapes. It has 15 object categories, including ships, planes, storage tanks, baseball diamonds, tennis courts, basketball courts, ground track fields, harbors, bridges, large vehicles, small vehicles, helicopters, roundabouts, and soccer ball field. DOTA dataset contains challenging and complicated scenes.

In our experiment, we use mean average precision (mAP) as our measurement metric. The definition of mAP was first proposed in the PASCAL VOC challenge (Everingham, Van Gool, Williams, Winn, & Zisserman , 2010). Precision is the percentage of correct predictions and recall is the percentage of true positives over all possible positives. Given a precision-recall curse, AP is defined as:

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,..1\}} P_{interp}(r) \tag{9}$$

$$P_{interp}(r) = \max_{r:r \geq r} p(r) \tag{10}$$

where$p(r)$ is the preicion at recall value r.

5.2. Experimental results

The software we used to generate the results in this paper can be find on GitHub at (Liu, 2020).

Table 3 shows the results of 3 state-of-the-art object detectors, Faster R-CNN, SSD and YOLOv3, on the DOTA dataset. Faster R-CNN, which is a two-stage object detector, performed better than the other two one-stage object detectors. Faster R-CNN achieved 35% on mAP, which is 3% and 11% higher than YOLOv3 and SSD respectively. Among the one stage object detectors, YOLOv3 performed much better than SSD. One reason is that the feature fusion of lower level and higher level and multi-scale training techniques in YOLOv3 are effective for small object detection.

Table 4 shows the results of Faster R-CNN, SSD and YOLOv3 on the small object subset of COCO and SUN datasets.

Their overall mAP is quite low. Again, Faster R-CNN performed the best, with mAP 24.1%. YOLOv3 was slightly worse than Faster R-CNN, while SSD was much woe.

Table 5 shows the results of Faster R-CNN, SSD and YOLOv3, and SSH on the Wider Face dataset. SSH was specifically designed for face detection. Overall, Faster R-CNN performed the best with mAP 33.6% and YOLOv3 the second with 31.5% mAP. Although SSH was designed for face detection, it did not perform well on small faces, worse than Faster R-CNN and YOLOv3. SSH achieved 30.8% mAP. SSD performed poorly with 24.6% mAP.

The execution times of the methods were measured as frames per second (FPS) when predicting the results. All experiments were run on a Dell Alienware computer with GTX 980 M GPU with 8 GB memory running Ubuntu. For images from Wider Face dataset with size 512 × 512 pixels, the running time of Faster R-CNN was 4 FPS, the slowest, due to its two-stage object detection structure. SSH and SSD had similar running times, 14 and 18 FPS, respectively. YOLOv3 ran the fastest, 40 FPS.

6. Future research directions

Different from anchors-based methods, recently several anchors-free algorithms have been proposed and achieved the state-of-the-art results. These methods get rid of the manual design of anchors and can represent bounding boxes in any scales and ratios. (Duan et al., 2019) represented the bounding boxesas pairs of center points and corner points. (Tychsen-Smith & Petersson, 2017) formulated the object detection task as a sparse bounding boxes probability distribution. It first estimated the distribution of four corners’ locations. The features for each corner were constructed with nearest neighbor sampling and bounding box width and height. In its deep neural networks, deconvolutional layers were applied to recover the information lost in pooling layers. (Law & Deng, 2018) defined the object detection problem as detecting and grouping pairs of corners with extra embedding information. The network in was

Table 3
Results of three representative detectors on DOTA dataset.

Method	mAP	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC
FasterR-CNN	0.35	0.574	0.172	0.234	0.175	0.63	0.51	0.721	0.27	0.03	0.434	0.232	0.23	0.506	0.38	0.008
SSD	0.24	0.66	0.0	0.0	0.0	0.60	0.49	0.74	0.09	0.0	0.37	0.0	0.0	0.45	0.22	0.0
YOLOv3	0.32	0.58	0.15	0.20	0.132	0.58	0.41	0.68	0.25	0.01	0.35	0.20	0.21	0.45	0.35	0.0

Table 4
Results of three representative detectors on the small object subsets of COCO and SUN datasets.

Method	mAP	Mouse	Telephone	Outlet	Clock	TP	TB	Faucet	Plate	Jar	Switch
Faster R-CNN	0.241	0.517	0.106	0.368	0.627	0.05	0.0	0.251	0.161	0.06	0.27
SSD	0.17	0.481	0.05	0.155	0.509	0.05	0.0	0.15	0.13	0.02	0.22
YOLOv3	0.23	0.54	0.105	0.397	0.631	0.07	0.0	0.26	0.11	0.06	0.20

Table 5
Results of four representative detectors on a small face subset of Wider Face dataset.

Methods	mAP
Faster R-CNN	0.336
SSD	0.246
YOLO v3	0.315
SSH	0.308

used as the backbone network and heat maps were predicted for both top-left corners and bottom-right corners. The corners belonging to the same object were further grouped with embedding vectors, which were predicted as the similarities between corners. Corner pooling layers were also proposed for combining the prior location knowledge of corners into the feature extraction process. (Wang, Chen, et al., 2017) represented bounding boxes using center points and four corners (top-left, top-right, bottom-left and bottom-right). It divided the feature maps into grid cells and for each cell predicted the probability of center points and corner points in the cell, x-offset and y-offset, as well as the probability of point link. The point link contained two parts: special index (the probability of point linking to cells) and point index (the linking probability between corner points and center point). Different from other methods that estimated the points of bounding boxes, (Zhou, Zhuo, & Krahenbuhl, P, 2019)estimated the four extreme points on an object with fully appearance-based algorithms. It predicted five feature maps: one for the center point and four for corner points. The extreme points were generated with maxpooling. Based on the prediction of any combination of four corners, the center was calculated and verified on the heat map for center. The corner centers with verified center points represented the detected objects. (Duan et al., 2019) improved the network of (Law & Deng, 2018). The work represented bounding boxes using three points (two corners and one center point) and center pooling and cascade corner pooling were proposed to extract strong features.

CRediT authorship contribution statement

Yang Liu: Conceptualization, Writing - original draft, Investigation, Resources, Formal analysis, Software. **Peng Sun:** Writing - original draft. **Nickolas Wergeles:** Resources, Supervision. **Yi Shang:** Conceptualization, Supervision, Project administration.

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.

References

Bai, Y., Zhang, Y., Ding, M., & Ghanem, B. (2018). In *Finding tiny faces in the wild with generative adversarial network* (pp. 21–30). Salt Lake City: IEEE Xplore.

Bell, S., Lawrence Zitnick, C., Bala, K., & Girshick, R. (2016). Inside-Outside Net: Detecting objects in context with skip pooling and recurrent neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 2874-2883). Las Vegas: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2016/html/Bell_Inside-Outside_Net_Detecting_CVPR_2016_paper.html.

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). In *Soft-NMS – improving object detection with one line of code* (pp. 5561–5569). Venice, Italy: IEEE Xplore.

Cai, Z., Fan, Q., Feris, R. S., & Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. European Conference on Computer Vision. 9908, pp. 354–370. Cham: Springer. tps://doi.org/10.1007/978-3-319-46493-0_22.

Cao, G., Xie, X., Yang, W., Liao, Q., Shi, G., & Wu, J. (2018). Feature-fused SSD: fast detection for small objects. Ninth International Conference on Graphic and Image Processing (ICGIP 2017). 10615, p. 106151E. Qingdao, China: Proc. SPIE. https://doi.org/10.1117/12.2304811.

Cao, Y., Chen, K., Loy, C. C., & Lin, D. (2020). Prime Sample Attention in Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR; pp. 11583–11591). IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_CVPR_2020/html/Cao_Prime_Sample_Attention_in_Object_Detection_CVPR_2020_paper.html.

Chen, C., Liu, M. Y., Tuzel, O., & Xiao, J. (2016). R-CNN for Small Object Detection. Asian Conference on Computer Vision (pp. 214–230). Cham: Springer. https://doi.org/10.1007/978-3-319-54193-8_14.

Chen, K., Li, J., Lin, W., See, J., Wang, J., Duan, L., . . . Zou, J. (2019). Towards accurate one-stage object detection with AP-loss. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR; pp. 5119-5127). Long Beach, California: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_CVPR_2019/html/Chen_Towards_Accurate_One-Stage_Object_Detection_With_AP-Loss_CVPR_2019_paper.html.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184

Cheng, G., & Han, J. (2016). A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing, 117*, 11–28. https://doi.org/10.1016/j.isprsjprs.2016.03.014

Cheng, G., Han, J., Zhou, P., & Xu, D. (2018). Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection. *IEEE Trans. on Image Process., 28*(1), 265–278. https://doi.org/10.1109/TIP.2018.2867198

Cheng, G., Zhou, P., & Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sensing, 54*(12), 7405–7415. https://doi.org/10.1109/TGRS.2016.2601622

Divvala, S. K., Hoiem, D., Hays, J. H., Efros, A. A., & Hebert, M. (2009). An empirical study of context in object detection. 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1271–1278). Miami, FL, USA: IEEE. https://dx.doi.org/10.1109/CVPR.2009.5206532.

Dong, C., Liu, J., & Xu, F. (2018). Ship detection in optical remote sensing images based on saliency and a rotation-invariant descriptor. *Remote Sensing, 10*(3), 400. https://doi.org/10.3390/rs10030400

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). CenterNet: Object detection with keypoint triplets. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1904.08189v1.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Fu, C., Liu, W., Ranga, A., Tyagi, A., & Berg, A. (2017). DSSD: Deconvolutional Single Shot Detector. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1701.06659.

Galleguillos, C., & Belongie, S. (2010). Context based object categorization: A critical survey. *Computer Vision and Image Understanding, 114*(6), 712–722. https://doi.org/10.1016/j.cviu.2010.02.004

Ghiasi, G., Lin, T. Y., & Le, Q. V. (2019). NAS-FPN: Learning scalable feature pyramid architecture for object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR; pp. 7036-7045). California: Long Beach. IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_CVPR_2019/html/Ghiasi_NAS-FPN_Learning_Scalable_Feature_Pyramid_Architecture_for_Object_Detection_CVPR_2019_paper.html.

Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV; pp. 1440–1448). Santiago, Chile: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 580–587). Columbus, Ohio: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV; pp. 2961-2969). Venice, Italy: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 770-778). Las Vegas: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding Box Regression With Uncertainty for Accurate Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR; pp. 2888–2897). Long Beach, California: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_CVPR_2019/html/He_Bounding_Box_Regression_With_Uncertainty_for_Accurate_Object_Detection_CVPR_2019_paper.html.

Hosang, J., Benenson, R., & Schiele, B. (2017). Learning Non-Maximum Suppression. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 4507-4515). Honolulu, Hawaii: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2017/html/Hosang_Learning_Non-Maximum_Suppression_CVPR_2017_paper.html.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1704.04861.

Hu, P., & Ramanan, P. (2017). Finding Tiny Faces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, Hawaii: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2017/html/Hu_Finding_Tiny_Faces_CVPR_2017_paper.html.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 4700–4708). Honolulu, Hawaii. Retrieved from https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html.

Jeong, J., Park, H., & Kwak, N. (2017). Enhancement of SSD by concatenating feature maps for object detection. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1705.09587.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., et al. (2019). A Survey of Deep learning-based object detection. *IEEE Access, 7*, 128837–128868. https://doi.org/10.1109/ACCESS.2019.2939201

Kim, K., Hong, S., Roh, B., Kim, K. H., Cheon, Y., & Park, M. (2016). PVANet: Lightweight Deep Neural Networks for Real-time Object Detection. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1611.08588.

Kim, Y., Kang, B. N., & Kim, D. (2018). SAN: Learning relationship between convolutional features for multi-scale object detection. Proceedings of the European Conference on Computer Vision (ECCV), (pp. 316-331). Munich, Germany. Retrieved from https://openaccess.thecvf.com/content_ECCV_2018/html/Kim_SAN_Learning_Relationship_ECCV_2018_paper.html.

Kong, T., Sun, F., Tan, C., Liu, H., & Huang, W. (2018). Deep feature pyramid reconfiguration for object detection. Proceedings of the European Conference on Computer Vision (ECCV), (pp. 169–185). Munich, Germany. Retrieved from https://openaccess.thecvf.com/content_ECCV_2018/html/Tao_Kong_Deep_Feature_Pyramid_ECCV_2018_paper.html.

Kong, T., Yao, A., Chen, Y., & Sun, F. (2016). Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 845–853).

LaLonde, R., & Bagci, U. (2018). Capsules for Object Segmentation. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1804.04241.

Law, H., & Deng, J. (2018). CornerNet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 734–750).

Lee, S., Kwak, S., & Cho, M. (2018). Universal bounding box regression and its applications. Asian Conference on Computer Vision (pp. 373–387). Cham: Springer.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data, 5*(1), 42. https://doi.org/10.1186/s40537-018-0151-6

Li, Z., & Zhou, F. (2017). FSSD: Feature fusion single shot multibox detector. arXiv. Retrieved from https://arxiv.org/abs/1712.00960.

Li, Z., Tang, X., Han, J., Liu, J., & He, R. (2019). PyramidBox++: High Performance Detector for Finding Tiny Face. arXiv. Retrieved from https://arxiv.org/abs/1904.00386.

Lin, G., Milan, A., Shen, C., & Reid, I. (2017). RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 1925–1934). Honolulu, Hawaii: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2017/html/Lin_RefineNet_Multi-Path_Refinement_CVPR_2017_paper.html.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2117–2125).

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

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., & Zitnick, C. (2014). Microsoft COCO: Common Objects in Context. European Conference on Computer Vision (pp. 740–755). Cham: Springer. https://doi.org/10.1007/978-3-319-10602-1_48.

Liu, L.i., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision, 128*(2), 261–318. https://doi.org/10.1007/s11263-019-01247-4

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. (2016). Ssd: Single shot multibox detector. European conference on computer vision, pp. 21-37, (pp. 21–37). Springer, Cham.

Liu, Y. (2020). GitHub. Retrieved from https://github.com/ylt5b/A-Survey-and-Performance-Evaluation-of-Deep-Learning-Methods-for-Small-Object-Detection.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 3431–3440). Boston: IEEE Xplore. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html.

Luo, S., Li, X., Zhu, R., & Zhang, X. (2019). SFA: Small faces attention face detector. *IEEE Access, 7*, 171609–171620.

Najibi, M., Samangouei, P., Chellappa, R., & Davis, L. S. (2017). SSH: Single Stage Headless Face Detector. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 4875–4884).

Najibi, M., Singh, B., & Davis, L. S. (2019a). FA-RPN: Floating region proposals for face detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR; pp. 7723-7732). Long Beach, California: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_CVPR_2019/html/Najibi_FA-RPN_Floating_Region_Proposals_for_Face_Detection_CVPR_2019_paper.html.

Najibi, M., Singh, B., & Davis, L. S. (2019b). AutoFocus: Efficient multi-scale inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9745–9755).

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. European conference on computer vision, pp. 483-499. Cham: Springer.

Nguyen, N., Do, T., Ngo, T., & Le, D. (2020). An Evaluation of Deep Learning Methods for Small Object Detection. 2020, 18. https://doi.org/10.1155/2020/3189691.

Oksuz, K., Cam, B., Kalkan, S., & Akbas, E. (2019). Imbalance problems in object detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence., 1–1*. https://doi.org/10.1109/TPAMI.2020.2981890

Oliva, A., & Torralba, A. (2007). The role of context in object recognition. *Trends in Cognitive Sciences, 11*(12), 520–527. https://doi.org/10.1016/j.tics.2007.09.009

Palmer, T. (1975). The effects of contextual scenes on the identification of objects. *Memory & Cognition, 3*, 519–526. https://doi.org/10.3758/BF03197524

Pang, Jiangmiao, Chen, Kai, Shi, Jianping, Feng, Huajun, Ouyang, Wanli, & Lin, Dahua (2019). Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 821–830).

Pang, Y., Wang, T., Anwer, R., Khan, F., & Shao, L. (2019). Efficient featurized image pyramid network for single shot detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7336–7344).

Peng, Tang, Wang, Xinggang, Wang, Angtian, Yan, Yongluan, Liu, Wenyu, Huang, Junzhou, & Yuille, Alan (2018). Weakly supervised region proposal network and object detection. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 352–368).

Pont-Tuset, J., Arbeláez, P., Barron, J. T., Marques, F., & Malik, J. (2016). Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(1), 128–140. https://doi.org/10.1109/TPAMI.2016.2537320

Qian, Q., Chen, L., Li, H., & Jin, R. (2019). DR loss: Improving object detection by distributional ranking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12164–12172).

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 7263–7271). Retrieved from https://openaccess.thecvf.com/content_cvpr_2017/html/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.html.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv. Retrieved from https://arxiv.org/abs/1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, (pp. 91–99).

Rezatofighi, Hamid, Tsoi, Nathan, Gwak, JunYoung, Sadeghian, Amir, Reid, Ian, & Savarese, Silvio (2019). Generalized intersection over union: A metric and a loss for

bounding box regression.. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 658–666).
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention. 9351, pp. 234–241. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28.
Rothe, R., Guillaumin, M., & Van Gool, L. (2014). Non-maximum suppression for object detection by passing messages between windows. Asian conference on computer vision (pp. 290–306). Springer, Cham.
Samangouei, P., Chellappa, R., Najibi, M., & Davis, L. S. (2018). Face-MagNet: Magnifying feature maps to detect small faces. IEEE Winter Conference on Applications of Computer Vision (WACV). Lake Tahoe, NV, USA: IEEE. https://dx.doi.org/10.1109/WACV.2018.00020.
Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 761–769).
Shrivastava, A., Sukthankar, R., Malik, J., & Gupta, A. (2016). Beyond skip connections: Top-down modulation for object detection. arXiv. Retrieved from https://arxiv.org/abs/1612.06851.
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv. Retrieved from preprint arXiv:1409.1556.
Singh, B., Najibi, M., & Davis, L. S. (2018). SNIPER: Efficient multi-scale training. *Advances in Neural Information Processing Systems*, 9310–9320.
Soviany, P., & Ionescu, R. T. (2018a). In *Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction*. Timisoara, Romania, Romania: IEEE Xplore. https://doi.org/10.1109/SYNASC.2018.00041.
Soviany, P., & Ionescu, R. T. (2018b). Frustratingly easy trade-off optimization between single-stage and two-stage deep object detectors. *Proceedings of the European Conference on Computer Vision (ECCV)*.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–9).
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2818–2826).
Tang, X., Du, D., He, Z., & Liu , J. (2018). PyramidBox: A context-assisted single shot face detector. Proceedings of the European Conference on Computer Vision (ECCV), (pp. 797–813). Retrieved from https://openaccess.thecvf.com/content_ECCV_2018/html/Xu_Tang_PyramidBox_A_Context-assisted_ECCV_2018_paper.html.
Tian, W., Wang, Z., Shen, H., Deng, W., Meng, Y., Chen, B., . . . Huang, X. (2018). Learning better features for face detection with feature fusion and segmentation supervision. arXiv. Retrieved from https://arxiv.org/abs/1811.08557.
Tong, K., Wu, Y., & Zhou, F. (2020). Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing, 97*. https://doi.org/10.1016/j.imavis.2020.103910
Torralba, A., Murphy, K. P., Freeman, W. T., & Rubin, M. A. (2003). Context-based vision system for place and object recognition. Proceedings Ninth IEEE International Conference on Computer Vision (pp. 273–280). Nice, France, France: IEEE Xplore. https://dx.doi.org/10.1109/ICCV.2003.1238354.
Tychsen-Smith, L., & Petersson, L. (2017). DeNet: Scalable real-time object detection with directed sparse sampling. In Proceedings of the IEEE international conference on computer vision (pp. 428-436). Venice, Italy: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_iccv_2017/html/Tychsen-Smith_DeNet_Scalable_Real-Time_ICCV_2017_paper.html.
Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International Journal of Computer Vision, 104*(2), 154–171. https://doi.org/10.1007/s11263-013-0620-5
Wang, H., Li, Z., Ji, X., & Wang, Y. (2017). Face R-CNN. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1706.01061.
Wang, J., Chen, K., Yang, S., Loy, C. C., & Lin, D. (2019). Region proposal by guided anchoring. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 2965–2974). Retrieved from https://openaccess.thecvf.com/content_CVPR_2019/html/Wang_Region_Proposal_by_Guided_Anchoring_CVPR_2019_paper.html.
Wang, J., Yuan, Y., & Network, G. Y. (2017). An Effective Face Detector for the Occluded Faces. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1711.07246.
Wang, X., Chen, K., Huang, Z., Yao, C., & Liu, W. (2017). Point linking network for object detection. arXiv. Retrieved from https://arxiv.org/abs/1706.03646.
Wang, Y., Lin, Z., Shen, X., Zhang, J., & Cohen, S. (2018). Concept Mask: Large-scale segmentation from semantic concepts. Proceedings of the European Conference on Computer Vision (ECCV), (pp. 530-546). Munich, Germany. Retrieved from https://openaccess.thecvf.com/content_ECCV_2018/html/Yufei_Wang_ConceptMask_Large-Scale_Segmentation_ECCV_2018_paper.html.
Wu, X., Hong, D., Ghamisi, P., Li, W., & Tao, R. (2018). MsRi-CCF: Multi-Scale and Rotation-Insensitive Convolutional Channel Features for Geospatial Object Detection. *Remote Sensing*. https://doi.org/10.3390/rs10121990
Wu, X., Sahoo, D., & Hoi, C. S. (2020). Recent advances in deep learning for object detection. *Neurocomputing, 396*(5), 39–64. https://doi.org/10.1016/j.neucom.2020.01.085
Xia, G., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., . . . Pelillo, M. (2018). DOTA: A large-scale dataset for object detection in aerial images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 3974–3983). Salt Lake City: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2018/html/Xia_DOTA_A_Large-Scale_CVPR_2018_paper.html.
Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). SUN database: Large-scale scene recognition from abbey to zoo. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 3485–3492). San Francisco, CA, USA: IEEE. https://dx.doi.org/10.1109/CVPR.2010.5539970.
Yan, J., Wang, H., Yan, M., Diao, W., Sun, X., & Li, H. (2019). IoU-adaptive deformable R-CNN: make full use of IoU for multi-class object detection in remote sensing imagery. *Remote Sensing, 11*(3), 286. https://doi.org/10.3390/rs11030286
Yang, F., Choi, W., & Lin, Y. (2016). Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 2129–2137). Las Vegas: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2016/html/Yang_Exploit_All_the_CVPR_2016_paper.html.
Yang, S., Luo, P., Loy, C., & Tang, X. (2016). WIDER FACE: A face detection benchmark. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 5525-5533). Las Vegas: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2016/html/Yang_WIDER_FACE_A_CVPR_2016_paper.html.
Yang, X., Liu, Q., Yan, J., & Li, A. (2019). R3Det: Refined single-stage detector with feature refinement for rotating object. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1908.05612.
Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., et al. (2018). Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing, 10*(1), 132. https://doi.org/10.3390/rs10010132
Yoo, Y., Dan, D., & Yun, S. (2019). EXTD: Extremely tiny face detector via iterative filter reuse. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1906.06579.
Yu, F., Wang, D., Shelhamer, E., & Darrell, T. (2018). Deep layer aggregation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 2403–2412). Salt Lake City: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Deep_Layer_Aggregation_CVPR_2018_paper.html.
Yu, J., Jiang, Y., Wang, Z., Cao, Z., & Huang, T. (2016). UnitBox: An advanced object detection network. Proceedings of the 24th ACM international conference on Multimedia., (pp. 516–520). Retrieved from https://dl.acm.org/doi/abs/10.1145/2964284.2967274?casa_token=qzC_K8AjZLAAAAAA:8HLv_FdNVhdvhFq5UEX3n_5LZXGAVr4gL3Uw7JnnBTngyaYEnJkhWZxrrGfBlpfFWCSJ4Dw2EVTw.
Zagoruyko, S., Lerer, A., Lin, T.-Y., Pinheiro, P., Gross, S., Chintala, S., & Dollár, P. (2016). A multipath network for object detection. arXiv. Cornell University. Retrieved from https://arxiv.org/abs/1604.02135.
Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-Shot Refinement Neural Network for Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 4203-4212). Salt Lake City: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Single-Shot_Refinement_Neural_CVPR_2018_paper.html.
Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). S3FD: Single shot scale-invariant face detector. Proceedings of the IEEE International Conference on Computer Vision (ICCV; pp. 192–201). Venice, Italy: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_iccv_2017/html/Zhang_S3FD_Single_Shot_ICCV_2017_paper.html.
Zhang, Z., Qiao, S., Xie, C., Shen, W., Wang, B., & Yuille, A. (2018). Single-shot object detection with enriched semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5813–5821).
Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems, 30*(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865
Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In AAAI, (pp. 12993–13000).
Zhou, X., Zhuo, J., & Krahenbuhl, P. (2019). Bottom-up object detection by grouping extreme and center points. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 850-859). Long Beach, California. Retrieved from https://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_Bottom-Up_Object_Detection_by_Grouping_Extreme_and_Center_Points_CVPR_2019_paper.html.
Zhu, C., Tao, R., Luu, K., & Savvides, M. (2018). Seeing small faces from robust anchor's perspective. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR; pp. 5127–5136). Salt Lake City: IEEE Xplore. Retrieved from https://openaccess.thecvf.com/content_cvpr_2018/html/Zhu_Seeing_Small_Faces_CVPR_2018_paper.html.
Zhu, C., Zheng, Y., Luu, Y., & Savvides, M. (2017). CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection. Deep learning for biometrics. Advances in computer vision and pattern recognition (pp. 57–79). https://doi.org/10.1007/978-3-319-61657-5_3.
Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object Detection in 20 Years: A Survey. arXiv. Retrieved from https://arxiv.org/abs/1905.05055.