Neural Networks 139 (2021) 246–254

Contents lists available at ScienceDirect

Neural Networks

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

Detecting slender objects with uncertainty based on keypoint-displacement representation ☆

Zelong Kong [a], Nian Zhang [b], Xinping Guan [c], Xinyi Le [c,d,*]

[a] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Department of Electrical and Computer Engineering, University of the District of Columbia, NW Washington, DC, 20008, USA
[c] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
[d] Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen, China

ARTICLE INFO

Article history:
Received 29 October 2020
Received in revised form 9 February 2021
Accepted 15 March 2021
Available online 26 March 2021

Keywords:
Deep learning
Object detection
Uncertainty prediction
Quality evaluation

ABSTRACT

Slender objects are long and thin objects. Existing object detection networks are not specially designed for detecting slender objects. We propose a method to detect slender objects. We represent slender objects with a keypoint-displacement pattern instead of using axis-aligned bounding boxes, avoiding problems like orientation confusion and wrong elimination. In our network, three parallel branches predict keypoint heatmaps, displacement vector field, and displacement uncertainty heatmap respectively. We add the uncertainty branch to enable our network to give uncertainty together with detection results. The predicted uncertainty provides a continuous criterion to evaluate whether detection results are reliable. In addition, the uncertainty branch can lower the weight of ambiguous training samples, leading to more accurate detection results. We employ our proposed method in two typical practical applications. Edges of electrode sheets and pins of electronic chips are correctly detected as slender objects. Manufacturing quality is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value.

1. Introduction

The rapid development of deep learning has powered many tasks of computer vision. These tasks include image classification, object detection, semantic segmentation and so on. Among them, object detection is a critical task and receives much attention. Many fantastic networks and methods have been proposed to detect objects quickly and accurately. These works perform excellently on general object detection benchmarks like MS COCO (Lin et al., 2014), KITTI (Geiger et al., 2012), and PASCAL VOC (Everingham et al., 2010). Many real-world applications are based on object detection, such as pedestrian detection (Zhang et al., 2018b), defect detection (Zeng et al., 2019), and security protection (González et al., 2020).

Slender objects are long and thin objects. They need to be detected in many cases but existing methods cannot meet the demand properly. In many concrete applications, the objects to be detected are quite different from objects in benchmarks like MS COCO (Lin et al., 2014). The detection network may need to detect objects with a specific size and shape instead of many different kinds of objects. In this paper, we focus on detecting slender objects. Slender objects are quite common. For example, in industrial manufacturing, slender scratches or cracks (Zhang et al., 2019) need to be detected for quality evaluation.

Traditional handcrafted feature descriptors used to be applied widely in object detection, such as HOG (Dalal & Triggs, 2005), LBP (Ojala et al., 2002), and SIFT (Lowe, 2004). However, traditional methods are sensitive to background and illumination. When the environment changes, the parameters need to be carefully adjusted. Traditional methods are not suitable for detecting slender objects because they are less adaptive.

CNN (Bouwmans et al., 2019) is an excellent feature extractor and it is applied in many applications (Chen et al., 2021; Feng et al., 2021; Huang et al., 2020; Le et al., 2020; Xu & Guo, 2021; Zhao et al., 2019; Zhou et al., 2018). As for object detection, many great works based on CNN are proposed. These works represent objects with axis-aligned bounding boxes. Since different objects have different shapes and sizes, using bounding boxes is efficient and intuitive. However, using bounding boxes to represent slender objects is inappropriate and inaccurate. The reasons are as follows. First, the slender object would only occupy little area of the bounding box. Second, using bounding boxes is

☆ The work described in the paper is supported part by Open Fund of Shenzhen Institue of Artificial Intelligence and Robotics for Society (AIRS) (AC01202005016), Aeronautical Science Foundation of China (2019ZE057001), and Shanghai Rising-Star Program (No.20QC1401100)

* Corresponding author at: Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail addresses: bigbigdinosaur@sjtu.edu.cn (Z. Kong), nzhang@udc.edu (N. Zhang), xpguan@sjtu.edu.cn (X. Guan), lexinyi@sjtu.edu.cn (X. Le).

https://doi.org/10.1016/j.neunet.2021.03.024
0893-6080/

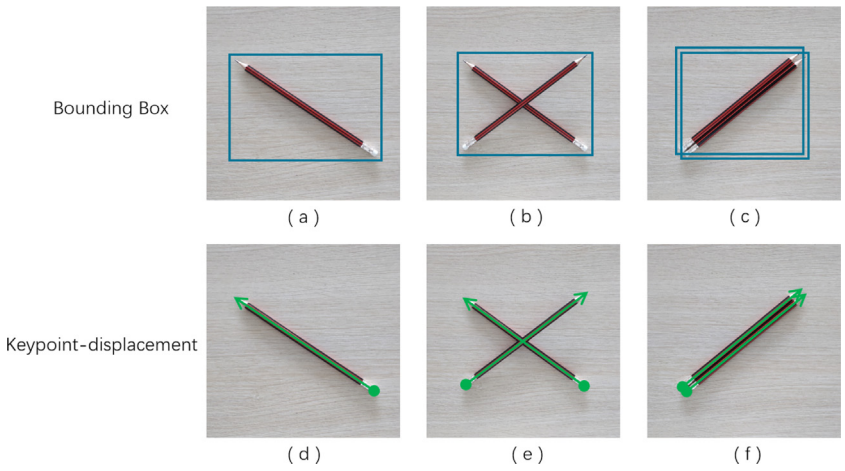

Fig. 1. We take detecting pencils as examples to compare the two ways of representing slender objects. (a) The pencil only occupies little area of the bounding box. (b) Using the bounding box cannot distinguish the orientation of the pencil. (c) The bounding boxes of the two pencils are tightly close to each other. The bounding box with lower classification confidence will be eliminated because of the NMS post-processing step. (d)–(f) Circles and arrows are used to visualize keypoints and displacement vectors. Using the keypoint-displacement pattern we propose is a more appropriate way to represent slender objects.

unable to indicate the orientation of a slender object. Third, when several slender objects are close to each other, their bounding boxes would be almost the same. More seriously, during the Non-Maximum Suppression (NMS) post-processing step, most detection results would be eliminated because the Intersection Over Union (IOU) of these boxes are over the threshold. Therefore, existing object detection networks are not suitable for detecting slender objects. In this paper, we propose a method to detect slender objects. To overcome the problems above, we propose to represent slender objects in a keypoint-displacement pattern instead of using bounding boxes. The comparison between these two ways is shown in Fig. 1. We use circles and arrows to visualize keypoints and displacement vectors. For slender objects, using the keypoint-displacement representation is a properer way than using bounding boxes.

In most cases, deep learning methods will output relatively precise results as we expect. However, what cannot be neglected is that the results may be wrong or even inexplicable in a few cases. Even one incorrect result may cause fatal accidents in applications like self-driving (Kim et al., 2017). Therefore, it is significant for deep learning methods to give uncertainty together with the results. In difficult situations, higher uncertainty should be given to attract attention and get people involved in. As for object detection, generally, a detector will output a confidence score together with the classification and location information. However, the confidence score is a classification score. The location uncertainty is important but neglected. Even worse, detection results with higher location accuracy may be eliminated because of their relatively lower classification confidence during the NMS post-processing step. In this paper, we add an uncertainty branch to our network architecture. Our method will output both classification confidence and location uncertainty when detecting slender objects. Formally, our network will predict a probability distribution instead of a single number to generate the uncertainty output. We reconstruct the loss function and train the uncertainty branch in an unsupervised manner. Furthermore, we utilize the predicted uncertainty to evaluate manufacturing quality in two practical applications.

Quality evaluation problem is generally dealt with as a classification problem. The evaluation results are divided into several discrete classes, such as excellent, good, fair, and poor. Since quality ranges from good to bad in a continuous manner, dividing quality into discrete classes is not appropriate. The relationship between different quality levels should not be opposite. Our method uses the predicted uncertainty value as one criterion of quality evaluation. Therefore, our method evaluates quality in a continuous manner instead of dividing quality into discrete classes. In addition, we do not need quality label information. Higher uncertainty value will be predicted adaptively when the sample is ambiguous owing to low quality.

Our contributions are summarized as follows:

(1) We propose a method to detect slender (long and thin) objects. We propose to represent slender objects with a keypoint-displacement pattern, avoiding problems like orientation confusion and wrong elimination introduced by using axis-aligned bounding boxes.

(2) We add an uncertainty branch to our network and it brings two benefits. First, the predicted uncertainty provides a continuous criterion to evaluate whether the detection results are reliable. Second, the uncertainty branch can lower the weight of ambiguous training samples, leading to more accurate detection results.

(3) We employ our proposed method in two typical practical applications. Edges of electrode sheets and pins of electronic chips are correctly detected as slender objects. Manufacturing quality is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value.

Organization: Some related works about object detection and uncertainty prediction will be introduced in Section 2. The network architecture and training methods are explained in detail in Section 3. Then, some comparison experiments and two practical applications are described in Section 4. Finally, the conclusion is given in Section 5.

2. Related works

2.1. Object detection

Anchor-based methods

Anchor-based methods detect objects through scattering duplicate anchors over images. The network is trained to do classification and regress to offsets of anchors. Two-stage detectors, such as Faster RCNN (Ren et al., 2015), R-FCN (Dai et al., 2016), and Cascade RCNN (Cai & Vasconcelos, 2018), will generate proposals before classification and regression. One-stage detectors, such as SSD (Liu et al., 2016), RefineDet (Zhang et al., 2018a), and RetinaNet (Lin et al., 2017), will predict classification and regression

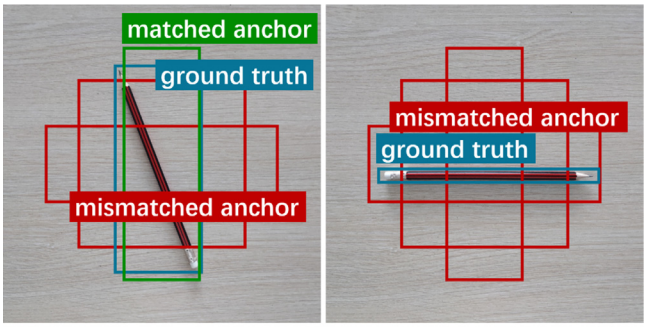

Fig. 2. We take pencils as examples to illustrate the mismatching problem of anchor-based methods. (a) The green anchor can match the pencil. (b) The pencil has no corresponding anchors, leading to the failure of training. Therefore, anchor-based methods are not suitable to detect slender objects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

results directly. Generally, two-stage methods are more accurate and one-stage methods are more time-efficient.

These anchor-based networks have been applied in many practical cases owing to their excellent performance. However, anchor-based methods are not suitable to detect slender objects because of the mismatching problem. In anchor-based methods, anchors are rectangles with different sizes and aspect ratios. The training of the network is feasible when ground truth boxes have corresponding anchors. However, as shown in Fig. 2, slender objects in some cases may have no corresponding anchors. The mismatching problem will lead to the failure of training. Though generating excessive anchors may solve this problem to some extent, new problems like convergence difficulty and imbalance between positive and negative samples would appear.

Anchor-free methods

Anchor-free methods provide innovative ideas to detect objects with competitive performance. These methods do not need to concern about adjusting anchor parameters like size, aspect ratio, and number. CornerNet (Law & Deng, 2018) detects top-left and bottom-right keypoints and generates results through a pairing process. ExtremeNet (Zhou et al., 2019b) detects four extreme points and one center point of objects and groups the five keypoints to form detection results. These methods creatively propose to detect objects in a bottom-up manner. However, after detecting several keypoints, an embedding post-processing step is needed. The extra embedding step has to be carefully designed and adjusted, which is less flexible. CenterNet (Zhou et al., 2019a) detects the center point of an object and outputs the corresponding height and width of the object. For every point in the bounding box of an object, FCOS (Tian et al., 2019) regresses to four distances from the point to the edges of the bounding box. These anchor-free methods do not need an embedding post-processing step, detecting objects in a simpler and more efficient manner. However, they are not suitable to detect slender objects because they use bounding boxes to represent objects. Using the keypoint-displacement pattern to represent slender objects is a more reasonable manner.

2.2. Predicting uncertainty in deep learning

Some research has noticed the importance of predicting uncertainty in deep learning. Kendall and Gal (2017) proposed a novel Bayesian deep learning framework and two kinds of uncertainty. Aleatoric uncertainty captures noise from observations and epistemic uncertainty represents uncertainty in the model. Feng et al. (2018) have applied the same framework in Kendall and Gal (2017) to capture uncertainty when detecting vehicles from Lidia point clouds. In multi-task deep learning, Kendall et al. (2018) proposed to weigh multiple loss functions through considering the uncertainty of each task. These studies are mainly conducted on tasks like depth regression, semantic segmentation, and instance segmentation.

As for predicting uncertainty in the object detection task, He et al. (2019) modified the bounding box regression loss of Faster RCNN (Ren et al., 2015) to generate uncertainty. They improved the location accuracy by merging neighboring bounding boxes based on the predicted uncertainty. Choi et al. (2019) applied the modified YOLOv3 (Redmon & Farhadi, 2018) to autonomous driving. During the inference process, they employed the predicted uncertainty of bounding boxes to reduce the false positive. However, these works are based on anchor-based frameworks and use bounding boxes to represent objects, which is not suitable to detect slender objects. Methods with the keypoint-displacement representation are more reasonable. The predicted uncertainty can be utilized to evaluate quality in practical applications.

3. Approach

3.1. Overview

Detecting slender objects aims to estimate classification information, classification confidence, and location information. Instead of predicting bounding boxes, our network will predict keypoints and corresponding displacement vectors as location information. Furthermore, our network will predict the uncertainty of every displacement vector, providing a criterion to evaluate whether the results are reliable.

The slender object detection framework we propose is shown in Fig. 3. The encoder–decoder feature extractor is a fully convolutional network, receiving an image and generating meta features of the image. Many encoder–decoder networks can serve as the feature extractor, such as Hourglass (Newell et al., 2016) and modified version of DLA (Zhou et al., 2019a). Three parallel branches predict keypoint heatmaps, displacement vector field, and displacement uncertainty heatmap respectively. In every branch, the predictor is several convolutional blocks upon the meta features. Details about the three parallel branches are illustrated in the following discussion. During the inference stage, peaks in keypoint heatmaps will be located. According to these peak locations, the corresponding displacement vectors and displacement uncertainty are extracted to form detection results. In the end, visualization of the detection results is implemented.

3.2. Keypoint heatmap branch (K branch)

An input image $I \in R^{\overline{W}\times\overline{H}\times 3}$ with width $\overline{W}$ and height $\overline{H}$ has several slender objects to be detected. The encoder–decoder feature extractor $F(\cdot)$ will produce meta features $M = F(I)$ of the image I. Let r be the downsampling factor. In the keypoint branch, the predictor $P_K(\cdot)$ will predict the keypoint heatmaps

$$\hat{K} = P_K(M) \in [0,1]^{W\times H\times C} \tag{1}$$

where $W = \overline{W}/r, H = \overline{H}/r$, C is the number of keypoint categories. Generally, the value of C is decided according to the number of slender object categories. The predicted keypoint heatmaps $\hat{K}$ gives the probability of detecting keypoints of certain categories. We train the keypoint branch in the same way proposed in Law and Deng (2018). For each slender object in image I with ground truth keypoint $\overline{p} \in R^2$ and ground truth

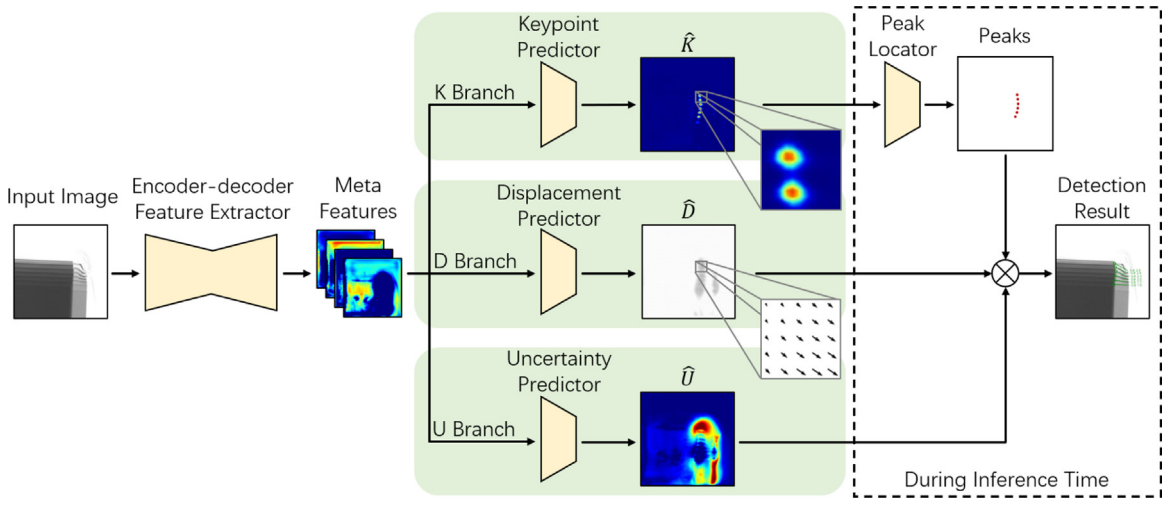

Fig. 3. Overview of the proposed slender object detection framework. Some images are partially enlarged for clarity. We firstly apply an encoder–decoder fully-convolutional network to extract features of an image. Then three parallel branches predict keypoint heatmaps, displacement vector field, and displacement uncertainty heatmap respectively. Every branch has a predictor, which is composed of convolutional modules. The uncertainty branch will predict higher uncertainty adaptively when the sample is ambiguous or difficult. During the inference, peaks in keypoint heatmaps will be located and the corresponding displacement vectors and displacement uncertainty will be extracted to form detection results. We apply the slender object detection framework to two practical applications. Manufacturing quality is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value.

displacement $\overline{d} \in R^2$, low-resolution target keypoint p and target displacement d are calculated.

$$p = \lfloor \overline{p}/r \rfloor = (\tilde{x}, \tilde{y}), d = \lfloor \overline{d}/r \rfloor = (\tilde{\delta}_x, \tilde{\delta}_y) \tag{2}$$

Then every low-resolution target keypoint is mapped onto the target heatmaps $\tilde{K} \in [0, 1]^{W \times H \times C}$ utilizing the Gaussian kernel

$$\tilde{K}_{xyc} = \exp\left(-\frac{(x-\tilde{x})^2 + (y-\tilde{y})^2}{2\sigma_a^2}\right) \tag{3}$$

where σ_a is a scale-adaptive standard deviation. An element-wise focal loss is applied to compute the loss between the predicted keypoint heatmaps $\hat{K}$ and the target keypoint heatmaps $\tilde{K}$. Loss function of the keypoint branch L_K is as follows.

$$L_K = -\frac{1}{N}\sum_{xyc}\begin{cases}\left(1-\hat{K}_{xyc}\right)^{\alpha}\log\left(\hat{K}_{xyc}\right) & if\ \tilde{K}_{xyc}=1\\ \left(1-\tilde{K}_{xyc}\right)^{\beta}\left(\hat{K}_{xyc}\right)^{\alpha}\log\left(1-\hat{K}_{xyc}\right) & otherwise\end{cases} \tag{4}$$

where N is the number of slender object keypoints in image I, α and β are hyper-parameters of the focal loss. The $\log\left(\hat{K}_{xyc}\right)$ and $\log\left(1-\hat{K}_{xyc}\right)$ part in L_K are the commonly used cross entropy loss. The coefficients $\left(1-\hat{K}_{xyc}\right)^{\alpha}$ and $\left(\hat{K}_{xyc}\right)^{\alpha}$ in L_K are used to relieve the imbalance between hard and easy samples. For easy samples, the predicted value is close to the target value. Then the coefficient gets small, reducing the weight of easy samples. The coefficient $\left(1-\tilde{K}_{xyc}\right)^{\beta}$ in L_K is used to relieve the imbalance between positive and negative samples. The coefficient gets small for those negative samples close to positive samples, reducing the weight of those negative samples.

3.3. Displacement vector field branch (D branch)

In the displacement branch, the predictor $P_D(\cdot)$ will predict the displacement vector field $\hat{D} = P_D(M) \in R^{W \times H \times 2}$. The two channels of $\hat{D}$ predict the horizontal and vertical displacement respectively. To relieve the computing burden, all C categories of slender objects share the same predicted displacement vector field. The prediction $\hat{D}_{xy} = \left(\hat{\delta}_x, \hat{\delta}_y\right)$ indicates the displacement of the corresponding keypoint (x, y) in the predicted keypoint heatmaps $\hat{K}$. Loss function of the displacement branch L_D is the L1 loss.

$$L_D = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{D}_{p_i} - d_i\right| \tag{5}$$

where $\hat{D}_{p_i}$ is the predicted displacement of the target keypoint p_i and d_i is the target displacement of p_i. When calculating the loss, only displacements at target keypoint locations will be considered and the others will be ignored.

In order to predict the uncertainty of displacement vectors, we reconstruct the loss function L_D with the uncertainty predicted in the uncertainty branch. We will explain this in detail in the next part.

3.4. Uncertainty heatmap branch (U branch)

In ambiguous or difficult situations, higher uncertainty should be given to attract attention and get people involved in. Therefore, we add an uncertainty heatmap branch to give the uncertainty of the predicted displacement. The predictor $P_U(\cdot)$ will predict the displacement uncertainty heatmap $\hat{U} = P_U(M) \in R^{W \times H \times 2}$. The two channels of $\hat{U}$ are corresponding to the horizontal and vertical displacement uncertainty respectively. The prediction $\hat{U}_{xy}$ indicates the uncertainty of the corresponding displacement $\hat{D}_{xy}$.

Formally, the uncertainty is generated through predicting a probability distribution instead of a deterministic value. In the following discussions, we will take the horizontal displacement $\hat{\delta}_x$ of $\hat{D}_{xy} = \left(\hat{\delta}_x, \hat{\delta}_y\right)$ as an example to illustrate the details. The vertical displacement $\hat{\delta}_y$ is managed in the same way. We use single variable Gaussian distribution for simplicity and predict the distribution of δ_x.

$$\hat{P}_\theta(\delta_x) = \frac{1}{\sqrt{2\pi\hat{\sigma}_x^2}}\exp\left(-\frac{\left(\delta_x - \hat{\delta}_x\right)^2}{2\hat{\sigma}_x^2}\right) \tag{6}$$

θ is the collective name of millions of trainable parameters in the neural network. $\hat{P}_\theta(\delta_x)$ is the predicted distribution of δ_x when

the model parameters is θ. $\hat{\sigma}_x$ is the predicted standard deviation of the Gaussian distribution of δ_x. $\hat{\sigma}_x$ measures the uncertainty of the predicted $\hat{\delta}_x$. When $\hat{\sigma}_x$ is big, the network is uncertain about its prediction. While $\hat{\sigma}_x \to 0$, the network is quite confident of its prediction. The target distribution of δ_x is a Dirac delta function $Dir(\cdot)$.

$$\tilde{P}(\delta_x) = Dir\left(\delta_x - \tilde{\delta}_x\right) \tag{7}$$

where $\tilde{\delta}_x$ is the target horizontal displacement. The reconstructed loss function L_{DU} is used to train the displacement branch and the uncertainty branch together. L_{DU} is derived through making some modifications in KL-Divergence ($D_{KL}(\cdot)$) of the target distribution $\tilde{P}(\delta_x)$ and the predicted distribution $\hat{P}_\theta(\delta_x)$.

$$\begin{aligned} &D_{KL}\left(\tilde{P}(\delta_x) \parallel \hat{P}_\theta(\delta_x)\right) \\ &= \int \tilde{P}(\delta_x) \log \tilde{P}(\delta_x)\, d\delta_x - \int \tilde{P}(\delta_x) \log \hat{P}_\theta(\delta_x)\, d\delta_x \\ &= \frac{\left(\tilde{\delta}_x - \hat{\delta}_x\right)^2}{2\hat{\sigma}_x^2} + \frac{\log\left(\hat{\sigma}_x^2\right)}{2} + \frac{\log(2\pi)}{2} - H\left(\tilde{P}(\delta_x)\right) \end{aligned} \tag{8}$$

where $H\left(\tilde{P}(\delta_x)\right)$ is the entropy of the target distribution. We discard $\log(2\pi)/2$ and $H\left(\tilde{P}(\delta_x)\right)$ in L_{DU} because they do not change while training.

$$L_{DU} \propto \frac{\left(\tilde{\delta}_x - \hat{\delta}_x\right)^2}{2\hat{\sigma}_x^2} + \frac{\log\left(\hat{\sigma}_x^2\right)}{2} \tag{9}$$

In practice, we predict $\hat{\mu}_x = \log\left(\hat{\sigma}_x^2\right)$ in the uncertainty branch to avoid the gradient exploding problem (He et al., 2019). During the inference time, $\hat{\mu}_x$ will be converted back. Therefore, we replace $\log\left(\hat{\sigma}_x^2\right)$ with $\hat{\mu}_x$ in L_{DU}.

$$L_{DU} \propto \frac{1}{2} e^{-\hat{\mu}_x}\left(\tilde{\delta}_x - \hat{\delta}_x\right)^2 + \frac{1}{2}\hat{\mu}_x \tag{10}$$

In order to be consistent with the basic loss function L_D in Formula (5), we make some modifications in L_{DU}. After adding the vertical displacement part and considering all slender objects of image I, the final loss function is derived.

$$L_{DU} = \frac{1}{2N}\sum_{i=1}^{N}\left(e^{-\hat{\mu}_{xi}}\left|\tilde{\delta}_{xi} - \hat{\delta}_{xi}\right| + \hat{\mu}_{xi} + e^{-\hat{\mu}_{yi}}\left|\tilde{\delta}_{yi} - \hat{\delta}_{yi}\right| + \hat{\mu}_{yi}\right) \tag{11}$$

where N is the number of slender object keypoints in image I.

During the training, $\left|\tilde{\delta}_{xi} - \hat{\delta}_{xi}\right|$ and $\left|\tilde{\delta}_{yi} - \hat{\delta}_{yi}\right|$ are relatively larger for those ambiguous samples. Then the network will produce higher uncertainty $\hat{\sigma}_x$ and $\hat{\sigma}_y$ so that the loss L_{DU} will be lower. The network will give more accurate predictions because the training is not dominated by those ambiguous samples owing to their higher uncertainty.

3.5. Loss and inference

The overall loss L can be obtained by weighted summing L_K and L_{DU}.

$$L = L_K + \lambda L_{DU} \tag{12}$$

At inference time, meta features are extracted from the input test image. Then the trained predictors $P_K(\cdot)$, $P_D(\cdot)$, and $P_U(\cdot)$ in three parallel branches will predict keypoint heatmaps $\hat{K}$, displacement vector field $\hat{D}$, and displacement uncertainty heatmap $\hat{U}$ respectively. We use the peak locator to locate peaks on the predicted keypoint heatmaps. (x, y) is a peak of category c if the value $\hat{K}_{xyc}$ is bigger than or equal to its eight adjacent neighbors. The value of peaks indicates the classification confidence score and a threshold τ is set to remove peaks with low confidence. For each remaining peak (x, y) of category c, the corresponding displacement $\hat{D}_{xy}$ and displacement uncertainty $\hat{U}_{xy}$ are extracted from $\hat{D}$ and $\hat{U}$ to form a detected slender object. During the inference, the detection results are produced in a direct way without any post-processing steps like NMS.

4. Experiments

We have deployed our proposed slender object detection method in two typical applications. Edges of electrode sheets and pins of electronic chips are detected as slender objects in these two applications. Manufacturing quality is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value. Owing to the limited space, we will mainly illustrate the first application.

4.1. Background and representation

Positive and negative electrode sheets are the main components of Lithium batteries. To prevent potential safety hazards, X-ray images of electrode sheets are captured for quality evaluation. On these images, the shape and arrangement of electrode sheets indicate whether the product is OK or NG. The edges of electrode sheets can be viewed as slender objects. We set the endpoint of a positive electrode sheet edge as a keypoint. The vector from the keypoint to the endpoint of the negative electrode sheet edge is set as the corresponding displacement. Owing to material property, some electrode sheets with manufacturing defects will appear to be non-rigid. When detecting these non-rigid electrode sheets, the trained model will predict higher uncertainty. According to the predicted higher uncertainty, we evaluate these electrode sheets as NG. In a word, it is because of the non-rigid appearance that we evaluate these electrode sheets as NG.

4.2. Training details

The keypoint heatmap branch and the displacement vector field branch are trained in a supervised manner. The label information includes the coordinates of keypoints and the corresponding displacements. Random scaling, random translation, random flip, and color jittering are applied as data augmentation. We apply a modified version of DLA-34 (Zhou et al., 2019a) network as the encoder–decoder feature extractor. The predictors of three branches share the same structure, which is a separate 3×3 convolution, ReLU, and another 1×1 convolution. We set the downsampling factor $r = 4$ following Newell et al. (2016). We set the hyper-parameters of focal loss $\alpha = 2$ and $\beta = 4$ following (Law & Deng, 2018). $\lambda = 8$ is set as the weight of L_{DU}. Adam Kingma and Ba (2014) is used to optimize the overall target. We train with learning rate 1.25e−4 for 140 epochs, with the learning rate dropped 10 times at 90 and 120 epochs respectively.

4.3. Detection results

After training, test images are fed into the trained network for evaluation. Four test images, their meta features, and their output predictions of three branches are shown in Fig. 4. The meta features have many channels, bringing sufficient information for the following predictions. The predicted keypoint heatmap $\hat{K}$ captures the keypoints on test images accurately. When the image

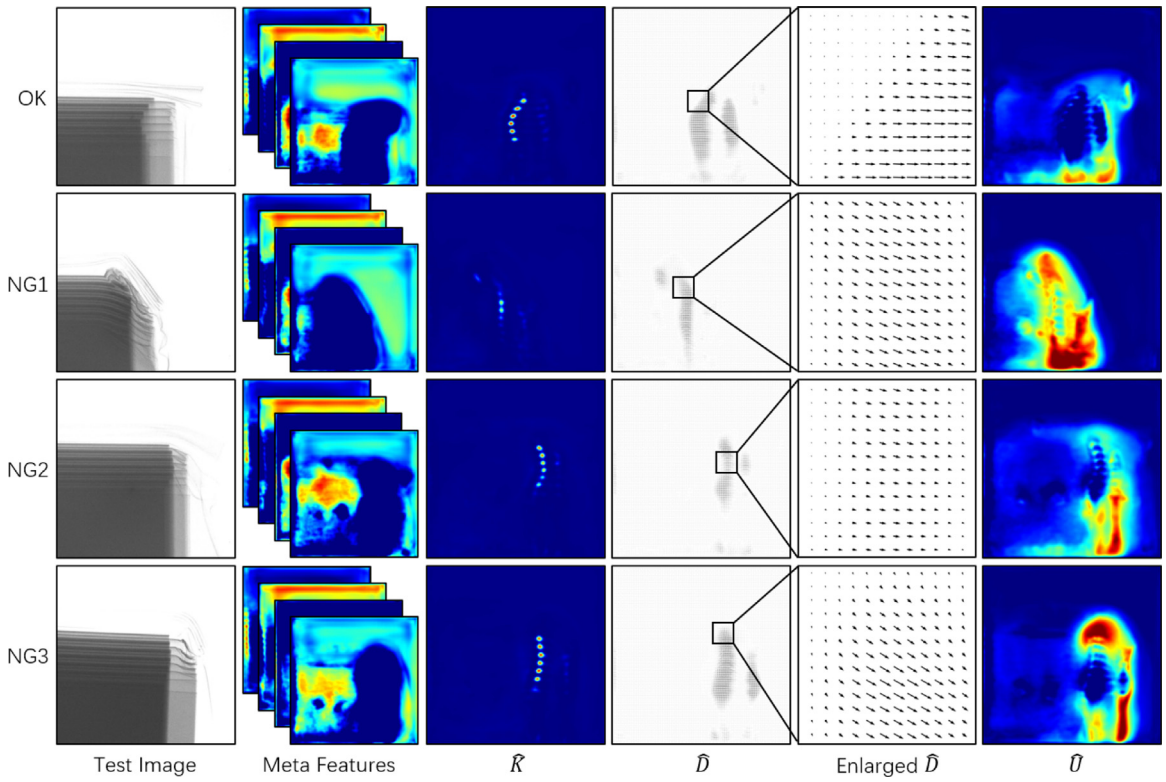

Fig. 4. Four test images (1 OK and 3 NG), their meta features, and their output predictions of three branches. The meta features have many channels, bringing sufficient information for the following predictions. The predicted keypoint heatmap $\hat{K}$ locates the keypoints accurately. The predicted displacement vector field $\hat{D}$ has well continuity and consistency though the supervision acts only at target keypoint positions. $\hat{D}$ is partially enlarged for clarity. The predicted displacement uncertainty heatmap $\hat{U}$ indicates whether the predictions are reliable. From red to blue represents high uncertainty to low uncertainty. The NG1 image has unclear keypoints because of manufacturing defects, leading to fewer detected keypoints in $\hat{K}$. Owing to the fewer detected keypoints, the NG1 image is evaluated as NG. The NG2 image is recognized as NG because the length and angle of predicted displacements go beyond the permissible range. The NG3 image is ambiguous owing to some manufacturing defects, resulting in higher values in $\hat{U}$. Due to the higher uncertainty, the NG3 image is recognized as NG. Therefore, the quality of electrode sheets is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

has unclear keypoints because of some manufacturing defects, the value in $\hat{K}$ is lower. Therefore, we can evaluate product quality according to the number of detected keypoints through setting a threshold. Though the supervision of the displacement acts only at target keypoint positions, we can see the predicted displacement vector fields $\hat{D}$ have well continuity and consistency. If the length or angle of a predicted displacement goes beyond the permissible range, we recognize it as NG. In Fig. 4, the two channels of the predicted displacement uncertainty $\hat{U}$ are weighted summed for brevity. When the image appears to be non-rigid because of manufacturing defects, the value in $\hat{U}$ is higher. Therefore, the detection result with higher uncertainty indicates lower product quality.

After getting the output predictions of three branches, we use the peak locator to locate peaks on the predicted keypoint heatmap. The value of peaks indicates the classification confidence score and we set the threshold $\tau = 0.5$ to remove peaks with low confidence. For each peak, the corresponding displacement and displacement uncertainty are extracted from $\hat{D}$ and $\hat{U}$ to form a detected result. Visualization of some detection results is shown in Fig. 5. Circles and lines are used to represent the detected keypoints and displacements. The two numbers are the predicted keypoint confidence and displacement uncertainty.

To sum up, we evaluate the quality of electrode sheets through the detection results of the X-ray images. Fewer detected keypoints caused by low keypoint confidence, exceeding permissible range of predicted displacement length or angle, and higher predicted displacement uncertainty, will all lead to poor quality decisions. The whole test images are used to judge our proposed quality evaluation method, including 983 OK and 1093 NG images. After analyzing the detection results of an NG test image, if this image is evaluated as NG, a True Positive (TP) is recorded. True Negative (TN), False Positive (FP), and False Negative (FN) can be counted in the same manner. We record the results and get the confusion matrix as shown in Table 1. Through calculating, test results with different evaluation metrics are summarized in Table 2. Among all NG test images, 98.72% are evaluated as NG through our method (Recall=0.9872). Only 1.93% of the OK test images are evaluated as NG by mistake (FPR=0.0193).

Table 1
Confusion matrix on test images.

	Predicted NG	Predicted OK
Actual NG	TP = 1079	FN = 14
Actual OK	FP = 19	TN = 964

Table 2
Test results with different evaluation metrics.

Evaluation metric	Test result
Recall	0.9872
FPR	0.0193
Accuracy	0.9841
Precision	0.9827
F1-Score	0.9849

Table 3
Performance of different backbone networks.

	Training time (h)	Inference time (ms)	FPS	L_K	L_{DU}	L
ResNet-18 (Xiao et al., 2018)	**0.0833**	**16**	**62.5**	0.497	1.683	13.946
ResNet-101 (Xiao et al., 2018)	0.283	30	33.3	0.471	0.811	6.961
DLA-34 (Zhou et al., 2019a)	0.334	25	40.0	**0.086**	**0.354**	**2.921**
Hourglass-104 (Newell et al., 2016)	0.783	61	16.4	0.123	0.405	3.365

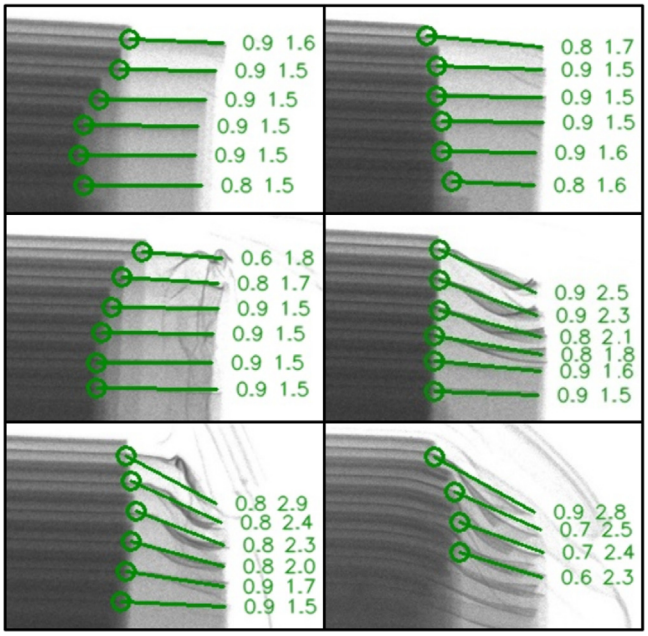

Fig. 5. Detection results of six electrode sheets test images. Images are cropped and enlarged for clarity. The detected keypoints and corresponding displacements are shown as circles and lines. The first number is the predicted keypoint confidence and the second number is the predicted displacement uncertainty. Fewer detected keypoints (such as the last image), exceeding permissible range of predicted displacement length or angle, and higher displacement uncertainty, will all be evaluated as NG. Owing to material property, some electrode sheets with manufacturing defects will appear to be non-rigid (such as the bottom two images). Higher uncertainty is predicted for these non-rigid electrode sheets and we evaluate them as NG according to their higher uncertainty.

4.4. Comparison with general object detection method and segmentation method

To illustrate the advantages of our proposed slender object detection method, we compare our method with general object detection methods and segmentation methods on the electrode sheets application. The famous RetinaNet (Lin et al., 2017) and Mask RCNN (He et al., 2017) are chosen for comparison. We train RetinaNet and our proposed network with the same dataset. Mask RCNN is trained with the segmentation dataset. The three networks are tested on the same test images.

The detection results of these three methods are shown in Fig. 6. RetinaNet cannot detect those horizontal targets because they are too slender. For those inclined targets, RetinaNet is able to detect them with bounding boxes. Using bounding boxes is not clear enough since it cannot indicate orientation. Mask RCNN misses some targets, and problems like wrong connection and insufficient coverage appear. As a segmentation method, Mask RCNN requires a more demanding dataset for training. It uses masks to indicate detection results so some post steps are needed to process the predicted masks. In addition, RetinaNet and Mask RCNN are not able to provide location uncertainty information for quality evaluation. Therefore, general object detection methods and segmentation methods are improper for slender object detection.

4.5. Comparison of different backbone networks

In this part, we conduct experiments with different encoder–decoder feature extractors to evaluate the performance of different backbone networks. The criteria we consider include training time, inference time, FPS, and losses at the convergence. All the experiments are conducted on the same machine. The results are summarized in Table 3. ResNet-18 (Xiao et al., 2018) performs best in speed but the accuracy is relatively low. DLA-34 (Zhou et al., 2019a) achieves the best accuracy at a moderate speed. Therefore, DLA-34 is utilized as the encoder–decoder feature extractor in the electrode sheet application.

4.6. Ablation experiment on uncertainty branch

In order to illustrate the effects of the uncertainty branch we proposed, we conduct an ablation experiment. The uncertainty branch is removed and the loss function of the displacement branch is replaced by the basic L1 loss. The other settings remain unchanged. The comparison of detection results is shown in Fig. 7. With the uncertainty branch, product quality can be evaluated through the predicted uncertainty. In addition, the uncertainty branch can lower the weight of ambiguous training samples, leading to more accurate detection results. Therefore, the uncertainty branch is necessary.

4.7. Application on chip pins

In this application, our proposed method is deployed to detect pins of electronic chips as slender objects. Some electronic chips have slender pins, which may have breakage, bend, or other defects. In order to detect these defects, images of electronic chips are captured. Pins are slender, so using the keypoint-displacement representation we propose is more appropriate. We set the connection point between the chip and the pin as a keypoint. The vector from the connection point to the end of the pin is set as the corresponding displacement. After training, test images are fed into the trained network for evaluation. Visualization of the detection results is shown in Fig. 8. Pins of electronic chips are correctly detected as slender objects. Through analyzing the detection results, manufacturing quality of the electronic chips is evaluated. For example, the pins with breakage, bend, or other defects have higher predicted displacement uncertainty, indicating lower product quality.

5. Conclusion

Existing object detection networks are not suitable for detecting slender (long and thin) objects. We propose a novel framework to detect slender objects. To avoid problems like orientation confusion and wrong elimination introduced by using axis-aligned bounding boxes, we propose to represent slender objects with a keypoint-displacement pattern. Our network has three parallel branches, which predict keypoint heatmaps, displacement vector field, and displacement uncertainty heatmap respectively. We add the uncertainty branch to give uncertainty together with detection results. Formally, the uncertainty is generated through predicting a probability distribution instead of a

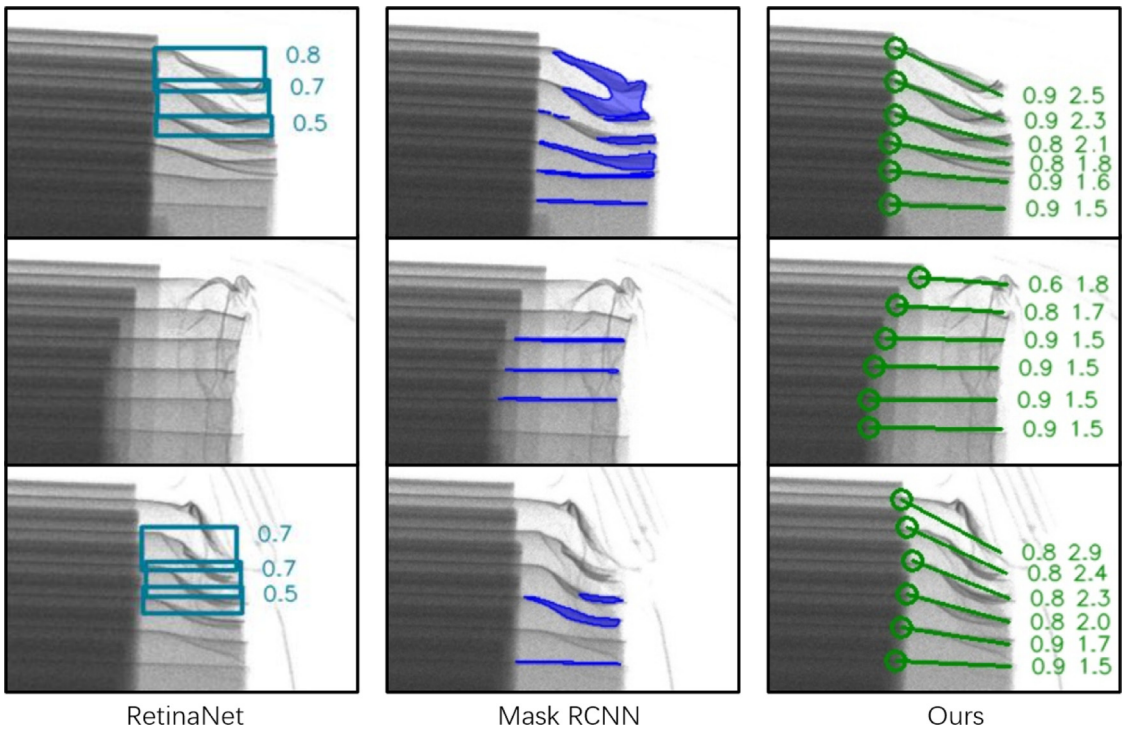

Fig. 6. Detection results of RetinaNet (Lin et al., 2017), Mask RCNN (He et al., 2017), and our proposed network. Images are cropped and enlarged for clarity. Some slender targets are missed by RetinaNet. RetinaNet uses bounding boxes to represent detected targets, indicating no orientation information. Mask RCNN also misses some targets, and problems like wrong connection and insufficient coverage occur. Mask RCNN requires a more demanding dataset for training and it needs some post steps to process the predicted masks. Our method detects targets with the keypoint-displacement representation correctly. In addition, our method provides location uncertainty information for quality evaluation. Therefore, general object detection methods and segmentation methods are improper for slender object detection.

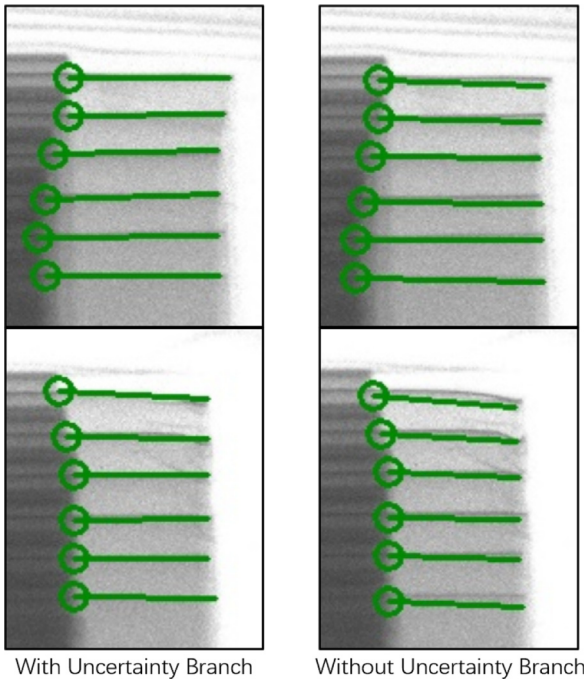

Fig. 7. Comparison between with and without uncertainty branch. Images are cropped and enlarged for clarity. The results are more accurate with uncertainty branch because ambiguous training samples have lower weight. Furthermore, the uncertainty branch provides a criterion to evaluate quality.

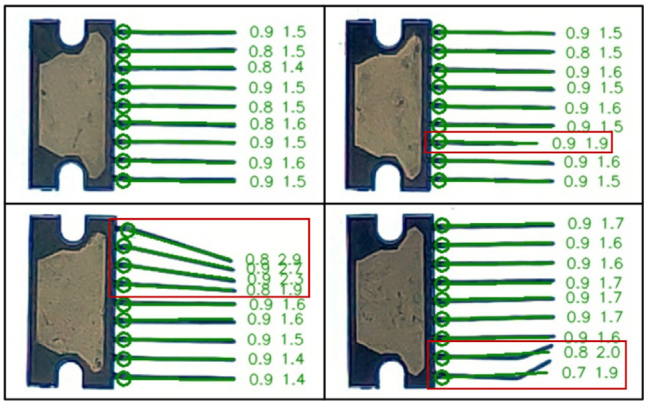

Fig. 8. Detection results of four chip test images. The detected keypoints and corresponding displacements are shown as circles and lines. The first number is the predicted keypoint confidence and the second number is the predicted displacement uncertainty. Those pins with breakage, bend, or other defects have higher predicted displacement uncertainty (emphasized with red rectangles). Therefore, the corresponding test images are evaluated as NG. The product quality of chips is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value.

deterministic value. The predicted uncertainty provides a continuous criterion to evaluate whether the detection results are reliable. Furthermore, the detection results become more accurate because the uncertainty branch can lower the weight of ambiguous training samples. In two typical concrete applications, our proposed method is employed. Edges of electrode sheets and pins of electronic chips are detected as slender objects correctly. Manufacturing quality is evaluated through analyzing the detection results, including keypoint number, displacement property, and uncertainty value.

For future work, we plan to detect slender objects with multiple sources of data like point cloud instead of with only image data. Additionally, we plan to explore more ways to fill the gap between general object detection and practical applications.

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

Bouwmans, T., Javed, S., Sultana, M., & Jung, S. K. (2019). Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Networks*, *117*, 8–66.

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6154–6162). IEEE.

Chen, L., Chen, Y., Xi, J., & Le, X. (2021). Knowledge from the original network: restore a better pruned network with knowledge distillation. *Complex & Intelligent Systems*, 1–10.

Choi, J., Chun, D., Kim, H., & Lee, H.-J. (2019). Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *Proceedings of the IEEE international conference on computer vision* (pp. 502–511).

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems* (pp. 379–387).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 886–893). IEEE.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Feng, D., Rosenbaum, L., & Dietmayer, K. (2018). Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *International conference on intelligent transportation systems* (pp. 3266–3273). IEEE.

Feng, J., Wang, X., & Liu, W. (2021). Deep graph cut network for weakly-supervised semantic segmentation. *Science China. Information Sciences*, *64*(3), Article 130105.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern Recognition* (pp. 3354–3361). IEEE.

González, J. L. S., Zaccaro, C., Álvarez-García, J. A., Morillo, L. M. S., & Caparrini, F. S. (2020). Real-time gun detection in CCTV: an open problem. *Neural Networks*, *132*, 297–308.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2888–2897). IEEE.

Huang, Z., Yu, Y., Xu, J., Ni, F., & Le, X. (2020). PF-Net: Point fractal network for 3D point cloud completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7662–7670). IEEE.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems* (pp. 5574–5584).

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7482–7491). IEEE.

Kim, J., Kim, J., Jang, G.-J., & Lee, M. (2017). Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. *Neural Networks*, *87*, 109–121.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. ArXiv preprint arXiv:1412.6980.

Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision* (pp. 734–750). Springer.

Le, X., Mei, J., Zhang, H., Zhou, B., & Xi, J. (2020). A learning-based approach for surface defect detection using small image datasets. *Neurocomputing*, *408*, 112–120.

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* (pp. 2980–2988).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., & Ramanan, D. (2014). Microsoft coco: Common objects in context. In *Proceedings of the European conference on computer vision* (pp. 740–755). Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., & Fu, C.-Y. (2016). SSD: Single shot multibox detector. In *Proceedings of the European conference on computer vision* (pp. 21–37). Springer.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *Proceedings of the European conference on computer vision* (pp. 483–499). Springer.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. ArXiv preprint arXiv:1804.02767.

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

Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 9627–9636).

Xiao, B., Wu, H., & Wei, Y. (2018). Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision* (pp. 466–481). Springer.

Xu, M., & Guo, L. (2021). Learning from group supervision: the impact of supervision deficiency on multi-label learning. *Science China. Information Sciences*, *64*(3), Article 130101.

Zeng, W., You, Z., Huang, M., Kong, Z., Yu, Y., & Le, X. (2019). Steel sheet defect detection based on deep learning method. In *International conference on intelligent control and information processing* (pp. 152–157). IEEE.

Zhang, H., Chen, Z., Zhang, C., Xi, J., & Le, X. (2019). Weld defect detection based on deep learning method. In *International conference on automation science and engineering* (pp. 1574–1579). IEEE.

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4203–4212). IEEE.

Zhang, S., Yang, J., & Schiele, B. (2018). Occluded pedestrian detection through guided attention in CNNs. In *Proceedings of the IEEE conference on computer vision and pattern Recognition* (pp. 6995–7003). IEEE.

Zhao, B., Le, X., & Xi, J. (2019). A novel SDASS descriptor for fully encoding the information of a 3D local surface. *Information Sciences*, *483*, 363–382.

Zhou, B., He, X., Zhou, Z., & Le, X. (2018). An image-based approach for defect detection on decorative sheets. In *International conference on neural information processing* (pp. 659–670). Springer.

Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. ArXiv preprint arXiv:1904.07850.

Zhou, X., Zhuo, J., & Krahenbuhl, P. (2019). Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 850–859). IEEE.