*Intelligent Automation & Soft Computing*
DOI: 10.32604/iasc.2020.010122
***Article***

# Multi-Scale Boxes Loss for Object Detection in Smart Energy

**Zhiyong Dai[1,*], Jianjun Yi[1], Yajun Zhang[1] and Liang He[2]**

[1]School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai, 200237, China
[2]Shanghai Aerospace Control Technology Institute, Shanghai, 201109, China
[*]Corresponding Author: Zhiyong Dai. Email: tdai05@mail.ecust.edu.cn

**Abstract:** The rapid development of Internet of Things (IoT) technologies has boosted smart energy networks in recent years. However, power line surveillance systems still suffer from the low accuracy and efficiency of the power line area recognition and risk objects detection. This paper proposes a new customized loss function to tackle the disequilibrium of the size of objects on multi-scale feature maps in the deep learning-based detectors. To validate the new concept and improve the efficiency, we also presented a new object detection model. Experimental results are provided to exhibit the advantage of our proposed method in both accuracy and efficiency.

**Keywords:** Object detection; deep learning; smart energy; power line surveillance

## 1 Introduction

The Internet of Things (IoT) allows networked physical objects to be able to identify each other and transfer interoperable information. IoT technologies have changed power line surveillance systems dramatically and video capturing devices and other sensors have been connected to the cloud. Images and other useful data are collected and transferred to the server via the IoT network. The system becomes smart and efficient to manage mass volumes of information from different regions. The IoT technologies based smart energy system can not only perceive dangerous activities and take actions before damages happen but can also predict harmful events by analyzing the collected big data.

Edge computing proposes a distributed computing system to bridge the individual devices with the cloud data centers, and allows device nodes in smart energy IoT networks to work as smart devices and perform tasks such as computation, data storage and transfer, customer interaction interface, alarm trigger, network services and so on. Therefore, most of edge devices in nodes of the network are utilized and smart energy systems are thus made more efficient. Recently, the computation power of embedded devices has been improved greatly, and edge computing is becoming the mainstream in smart energy systems.

The power line surveillance system is a fundamental part of the smart energy network. The most important job of the power line surveillance system is that the power line area automatically recognizes and risk an objects detection, which requires the employed object detection algorithm to detect a variety of risk objects with a fast processing speed. In many real applications, object detectors should read frames from surveillance videos and recognize both the power line area and the risk objects with the predicted location bounding boxes immediately. With the information of the objects and corresponding bounding boxes, the surveillance system is able to predict the safety level of the power line system in a local area. For example, if risk objects such as cranes, construction machineries and so on are too close to the power line area, the surveillance system will raise the alarm since there would be potential risks of collision. Besides that, scenarios such as birds standing on the cable, and animals climbing power poles should also be recognized as dangerous activities to the power line system.

Recently, the performance of object detection algorithms has been significantly improved by the rapid progress of the deep neural networks, which has achieved compelling results in several public detection tasks. Deep learning based object detection technologies have been widely studied in a lot of research fields. Some researchers focus on structures of two-stage detectors such as Faster R-CNN [1], R-FCN [2], and FPN [3] since those structures have an independent region proposal stage to ensure the accuracy of the object region prediction before bounding the boxes regression stage. However, two stage detector structures have the disadvantage in processing speed due to the proposed additional region network, which consumes more computation power. Other researchers are exploring new technologies with one-stage detector structures such as YOLO [4], SSD [5] and RetinaNet [6]. Those methods use a straight forward backbone network directly for object instance prediction and shows having advantages of high efficiency in data processing.

Although there are ample researches on object detection algorithms, some problems in power line surveillance systems are remaining to be settled. For example, some risk objects are too small to be detected accurately and many node devices have limited computational power for the deep learning based models.

This study focuses on the one-stage object detection solutions since many power line surveillance systems have limited computation resources on edge devices. The main contribution of this paper is described as follows:

1. First, we propose a new loss function for object detection models to solve the difficulties of small object detection in the smart power surveillance system by introducing a self-adaptive weight and a global weight as an object size factor, which gives rewards in the loss value when the model finds small objects correctly in each training iteration. By this approach, the performance of the accuracy on the small objects is improved eventually.
2. Second, we use a compact deep neural network model and arrive at a competitive object detection performance with applicable processing speed for the power line zone recognition and related risk object detection. This approach reveals a new way to deploy deep learning models in the IoT edge computing.
3. Third, we conducted a series of experiments to explore the combination of different backbone structures, feature map pyramid structures and their contribution to the performance of the object detection in the power line surveillance system. Those tests could be a general guidance to get an optimized deep learning model for the edge computing application in the IoT networks such as smart energy and so on.

The rest of this paper is organized as follows: Section 2 introduces related works. The background concepts are discussed in Section 3. In Section 4 the detailed customized deep learning method is presented. Experiments and numerical results are shown in Section 5 and Section 6 is the conclusion.

## 2 Related Work

In this section, we go through the related research and give a brief summary of representative works of the object detection and the application in the IoT systems of the power line surveillance.

### *2.1 The Traditional Object Detection Methods*

Before the deep neural network, many research works were engaged on object detection. We highlight some of the most related work here.

To find the location and bounding boxes of objects in the scene, sliding window approaches have been developed and demonstrated impressively in the public data set like PASCAL VOC [7]. But this kind of method should go through all the possible bounding boxes with many redundant computations, which can lead to low efficiency and speed in image processing. To improve the efficiency, Uijlings et al. [8] introduced the selective search method, which uses image segmentation to optimize the object region proposal process. Although the efficiency was improved immensely, the object window sampling speed was still limited for real time applications.

To extract features of the objects for classification, some feature extractors have been proposed, e.g., Dalal et al. [9] introduced the histogram of the oriented gradient (HOG) descriptor to represent features of the object in the scene. Based on the HOG, Felzenszwalb et al. [10] did testing to represent the objects with a discriminately-trained part-based model. Lowe [11] invented a scale-invariant feature transform (SIFT) algorithm, which demonstrated remarkable stability in the feature description for objects with a different size and shape transformation. Bay et al. [12] proposed using a speeded up robust feature (SURF), which the feature descriptor arrives at a quicker process speed than the SIFT. Rublee et al. [13] introduced the Oriented FAST and rotated the BRIEF (ORB) feature detector, which is a fusion of the FAST key point detector and the visual descriptor BRIEF (Binary Robust Independent Elementary Features) with some optimization.

To identify the class of objects from the extracted features, many classifiers have been designed as well. Viola et al. [14] proposed a cascade of AdaBoost classifiers working with haar-like features for face detection. The algorithm was improved by Rainer et al. [15]. The support vector machine (SVM) classifier was introduced by Cortes et al. [16]. The SVM has many advantages in solving the none-linear, high dimensional classification problems and is applicable for object detection tasks. Yan et al. [17] introduced a feature-based model for visual saliency detection. Lee et al. [18] proposed an object detection and tracking method with SURF features. Yan et al. [19] also designed a method of training a dictionary model for the oil pipeline leakage detection. Zhang et al. [20] presented a stacking random forest learning framework for contour detection.

Although there was much progress in the object detection with traditional technologies, the fast and accurate power line area and risk objects detection in complicated outdoor environments still have very challenging problems with those methods.

### *2.2 The Deep Learning Based Object Detection Methods*

The object detection has achieved significant advances due to the rapid progress of the deep neural networks (DNN) in recent years. The convolutional neural network (CNN) based object detectors become a new trend in the detection literature with remarkable test results. The CNN based detectors can be categorized into one and two-stage methods.

The two-stage approach consists of the candidate object region proposal generating process (e.g., Selective Search [8], EdgeBoxes [21], RPN [1]) and the accurate object regions and the corresponding class labels determination process. The two-stage approach shows great success in the object detection and many two-stage detectors were invented (e.g., R-CNN [22], SPPnet [23], Fast RCNN [24], Faster R-CNN [1] and Mask-RCNN [25]) recently. The two-stage approach has achieved remarkable performance on several public data sets (e.g., PASCAL VOC 2007/2012 [7] and MS COCO [26]). Although the two-stage approach promotes the performance of the object detection greatly, it is two-stage structure becomes a disadvantage to the image processing efficiency.

The one-stage approach has attracted attention in recent years due to the compact structure and the high efficiency. Redmon et al. [27] introduced a fast detector called the YOLO to predict object classes and locations within the predefined $N \times N$ girds in the feature map by a single feedforward convolutional network. In YOLOv2, improvements such as adding batch normalization, replacing full connection layers with convolutional layers for bounding-boxes prediction is implemented. After that, the YOLOv3 [28] was proposed with the improved backbone network structure (Darknet53), upsampling layers and concatenating the structure between the feature layer to improve the performance for object detection, especially for method, which use convolutional layers to regress the bounding boxes and classes of objects in multi-layers with different scales. Fu et al. [29] introduced the DSSD with a deconvolution layer to improve the accuracy of the object prediction. RFBnet [30] applies the RFB blocks for high precision object detection and the FSSD [31] introduces a feature fusion layer to improve the performance as well. Recently, RetinaNet [6] was proposed with a focal loss to balance the extreme foreground-background class imbalance during the training of a detector. Although the challenge still exists, the one-stage approach reveals a promising direction to improve both the accuracy and the efficiency of the object detection.

Besides the general object detection models, Xu et al. [32] proposed a CNN based fractal dimension invariant filtering method, which could help in tasks like curve detection. Yang et al. [33] introduced SCRDet, which is an effective multi category rotation detector for small cluttered and rotated objects with feature fusion, attention mechanism and an IoU constant factor for regression loss.

### *2.3 The Object Detection in the Smart Energy System*

Object detection technologies in the power line surveillance system have been studied widely. Many traditional models have been proposed to improve the efficiency of the system. Tan et al. [34] used the edge detection and Ransac matching to detect a power line, Fu et al. [35] proposed an object detection method to identify the power line area with feature descriptors like the Harris operator and the SIFT operator. Nguyen et al. [36] presented an overview of the possibilities and challenges of deep learning approaches (e.g., SSD, YOLO, etc.) for power line inspection by UAV. Xiang et al. [37] introduced a faster R-CNN based approach for engineering vehicles intrusion detection in a smart power gird surveillance system. Tao et al. [38] introduced a method to detect insulator defects by applying two Faster R-CNN models in sequence. The first model detects and crops insulators from the aerial image, and the second model detect defects from the cropped insulators. With the application of latest deep learning-based methods, the power liner surveillance system in the IoT network becomes more intelligent and capable than before.

As Liu et al. [39] mentioned, the improvement of accuracy and efficiency for a smart grid are important. However, the challenges of the object detection in the power line system are different too difficult in public detection tasks. First, there are some very small objects like birds in the area that need to be detected. Second, the detector should figure out both big shapes like the region of the power lines and very small objects like humans in an unbalanced distribution rapidly and stably. Current studies mainly focus on general models like the SSD, YOLO, R-FCN, and Faster R-CNN and so on, which cannot offer promising efficiency and accuracy for the tasks in smart energy systems, especially for very small objects detection in a power line surveillance.

## 3 Background

A typical internet of smart energy system is shown in Fig. 1. The node devices with a video capturing function are widely mounted and play an import role to inspect the distributed power lines.

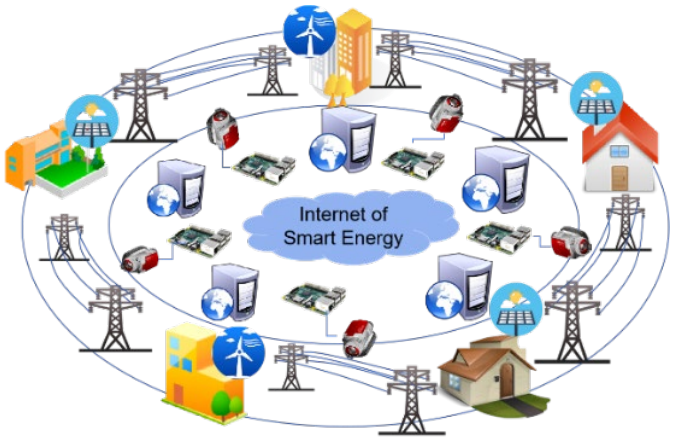



**Figure 1:** A typical network of smart energy

The general work-flow of a monitoring device in the internet of smart energy is shown in Fig. 2. The smart vision system keeps the detection work in cycles and raises an alarm of anomaly when the power line area is not in the image or when risk objects come into the power line area. Therefore, the daily monitoring work of the system is running automatically with the deep learning-based detector until any alert is triggered. Then, the operator will intervene, check the alarm, take the necessary steps and reset the monitoring system.

They do not need to keep eyes on the screen for the tedious monitoring work and all related cost are also saved as well.

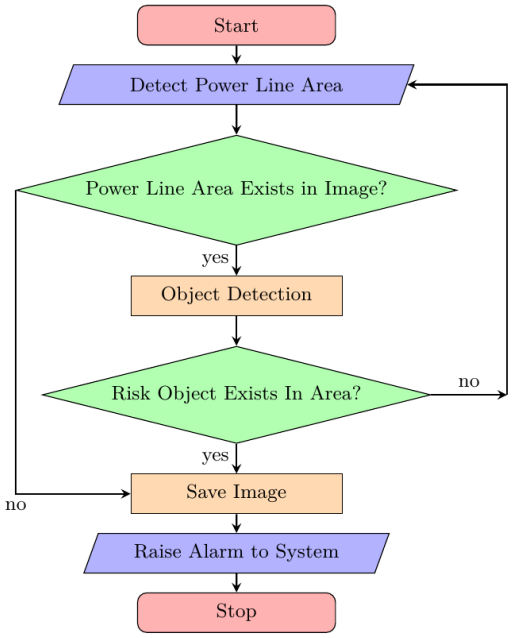



**Figure 2:** The workflow of the node devices in the internet of smart energy

In the work-flow, the risk object detection is regarded as one of the most critical important processes, which requires high efficiency and accuracy since it scans each frame and forwards the detection results from time to time. The risk object should be detected immediately when it comes close to the power line area, otherwise, the power line in the area could get damaged. Some typical risk objects are shown in Fig. 3.

**Figure 3:** Typical risk objects

In recent years, the deep learning-based method becomes the leading technology in many object detection tasks. A typical deep learning-based object detection pipeline is shown in Fig. 4. It comprises a backbone network which, extracts feature from the input image. A feature pyramid network, which provides feature maps in different scales, bounding the boxes regression blocks to predict the locations, classes and confidences of the object bounding boxes, and a Non-Maximum Suppression process to select the best bounding boxes from the predicted candidate results.

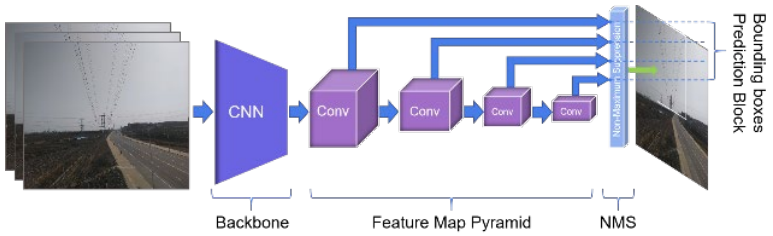



**Figure 4:** Typical object detection network framework

### *3.1 Backbone Network*

With the remarkable performance on the ILSVRC tests, the CNN becomes the mainstream of the backbone framework. Many novel CNN models are proposed and the performance on the general image classification is improved greatly in the last several years, e.g., VGGNet [40] builds a deeper network by stacking 3 × 3 convolution kernels. GoogleNet [41] introduces inception blocks with diverse combinations of the convolution kernel in parallel to enhance the feature extraction capacity. ResNet [42] proposes shortcut connections in the 'basic conv block' and 'bottleneck block' to reduce the training error in deeper networks. Xception [43] introduces the separable convolution layer to improve the feature extracting performance with reduced parameters. DenseNet [44] concatenates features maps from multi-layers densely and reduces the parameters with competitive accuracy. Typical CNN networks structures like the VGG16, Resnet34, Densenet121 and the state of art of light CNN models such as MobileNet V2 [45] and ShuffleNet V2 [46] are shown in Fig. 5.

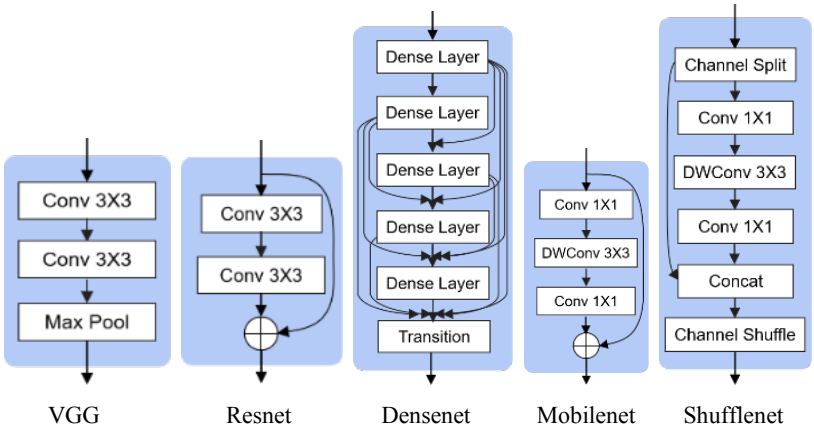
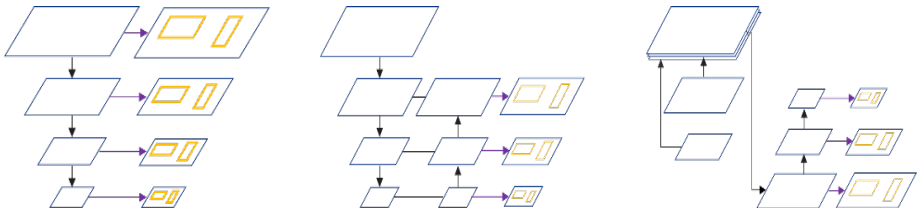



**Figure 5:** The key modules of the typical backbone networks

### *3.2 The Feature Pyramid Network*

The feature pyramid network becomes an essential part of the object detection models since the SSD firstly tries using straight forward multi-layers The FPN structure gets considerable performance on public data sets. The up-sampling structure of the FPN is utilized by using the DSSD and RetinaNet to improve the performance. Besides the two structures mentioned above, the FPN with concatenated multi feature layers is reported by the FSSD and RSSD. Fig. 6 shows the detail of the three FPN structures. The FPN in the SSD model is a straightforward top-down structure. The layers in each downsize step works as a feature layer. In the RetinaNet, the FPN structure is built on a down-top hierarchy with an up sampling operation. To enhance the connection between the feather layers, the FPN in the FSSD concatenates the first three feature layers from the straightforward FPN structure and adds the top-down FPN structure after that.

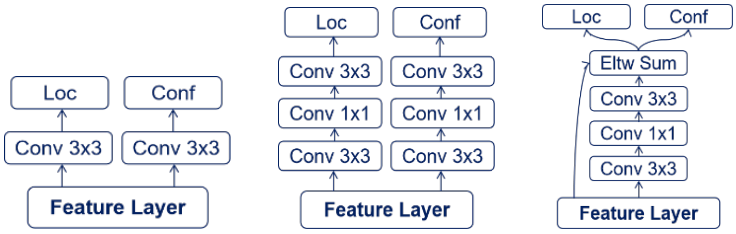


(a) SSD FPN (b) RetinaNet FPN (c) FSSD FPN

**Figure 6:** Typical FPN structures

### *3.3 The Object Detection Blocks*

The object detection blocks predict the location, class and probability of the objects from the feature maps, which are extracted by the FPN. In recent years, most deep learning-based object detection models use convolutional blocks to regress the location, classes and confidence scores of the objects directly. Some typical structures are shown in Fig. 7. In module a, the two convolutional blocks for the location and confidence score regression are introduced to the feature layer in parallel. More convolution units are applied to the convolution blocks in module b. Module c shows a compact solution, which combines the convolution blocks for both locations and the confidence score regression into one block.




(a) Module A (b) Module B (c) Module C

**Figure 7:** Typical object detection blocks

### *3.4 The Non-Maximum Suppression*

The key role of the Non-Maximum Suppression (NMS) is to eliminate the redundant candidate bounding boxes, which are predicted from the different feature layers. Since the bounding boxes prediction blocks provide a list of detection boxes B with related scores S. The NMS operation selects the detection with the maximum scores M, puts it into the set of final detection D and removes it from set B as well. Meanwhile, any box that has an intersection-over-union (IoU) value greater than a threshold with previously selected boxes will be abandoned. The IoU is calculated by Eq. (1).

$$\text{IoU} = \frac{\text{Detection} \cap \text{GroundTruth}}{\text{Detection} \cup \text{GroundTruth}} \tag{1}$$

## 4 The Proposed Deep Learning Method

### *4.1 Loss*

To measure the distance between the ground truth and the prediction of the object detector, the loss function should include both the location loss and classification loss. We introduce a multi scale loss from the multi boxes loss for the object detection:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{2}$$

$$L_{conf} = -\sum_{i \in Pos}^{N} x_{ij}^{p}(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0)$$

$$\text{where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{3}$$

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in x,y,w,h} x_{ij}^{k} \text{smooth}_{L1}(l_i^m - \hat{g}_j^m) \tag{4}$$

In the above $L_{loc}$ it specifies the location loss, which is the sum of the distance between each two corners of the ground truth and the predicted bounding box. $L_{conf}$ shows the classification loss, which is based on a softmax loss. Those losses show good criteria to train the deep learning model for the object detection. However, there still exists the following two problems when trained with them.

***Large variances between the sizes of objects.*** As shown in Fig. 8, (dx, dy) indicates the location distance of the predicted power line area and the ground truth, (dx2, dy2) shows the gap between the predicted bounding box of the bird and the ground truth. From the picture, we find that the detection of the power line area is good since most of the area of prediction is overlapped with the ground truth. However, the predicted bound box of the bird does not fit the ground truth well. We could arrive at a wrong judgement from the loss value since both dx2 and dy2 are less than dx1 and dy1, which indicates that the detection of the bird is better than the detection of the power line area.


**Figure 8:** The difference of the size of objects

***Uneven detection capacity between different feature map:*** The view of the distribution of the object size in the dataset. The mean size of each object category shown in Fig. 9 is calculated by multiplying the mean width and height of objects in each category.

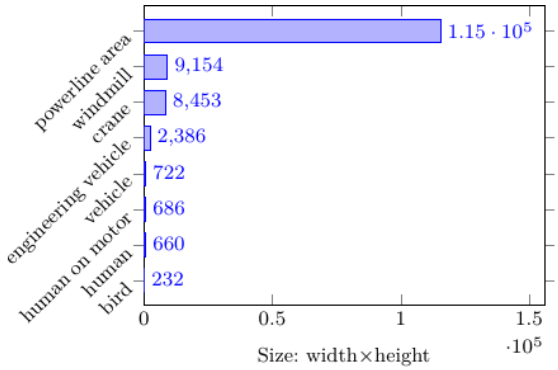



**Figure 9:** Mean size of objects in each category

When we track the objects in each layer of the FPN, we find that larger feature maps are more suitable for the small object detection, while the small feature maps are better for the large object recognition. Fig. 10 shows an example of the comparison of a power line area and a vehicle in different sizes of the feature maps. In the feature map with the size of 64 × 64, the scale of the vehicle is close to the default anchor boxes, which are cantered from the girds between the pixels, while the shape of the power line area is much bigger than any default anchor box in the same feature map. On the contrary, the features of the vehicle in the final layer becomes too tiny for any default box to approximate and the area of power line is downsized to fit the shape of the default anchor boxes well in this layer. Therefore, we expect to detect the small object from the big feature maps and recognize the big objects from the small feature maps as well. However, the loss function above will bring difficulties to train a deep neural network with high performance since all the candidate predictions of the objects from the feature layers of the FPN are treated equally.

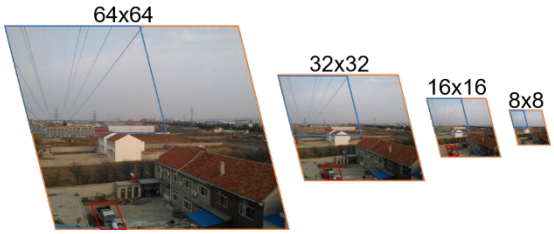



**Figure 10:** Examples of objects on different layers of the FPN

To address these issues, we introduce a new loss function, which has the following characteristics.

(i) To balance the proportion of the contribution by the size of the boxes in the location loss, we propose a vector of size scale factors, **S** to the location loss. The hypothesis is that the object with a large size is easier to be detected and located than the object with the small size. The new location loss $L'_{loc}(x, l, g)$ is shown in Eq. (5) where **S** is multiplied with the location distance of the prediction and the ground truth on each iteration of the training epoch. It gives a discount for the big boxes and a reward for the small boxes in the location regression. $S_i$ is the i[th] component of **S** which, is the logarithm of the maximum area of the bounding boxes in the batch of the samples divided by the area of the i[th] box. **S** and $S_i$ are shown in Eqs. (6) and (7). **S** is a dynamic factor, which is adaptive to the maximum object size of each batch of samples in the training epoch. $l^m$ and $l^g$ are the prediction and ground truth of the object locations. a is the maximum area of the objects in each training batch. $(x_{il}, x_{it})$, $(x_{ir}, x_{ib})$ are the coordinate of the top left and bottom right corners of the i[th] object.

$$L'_{loc}(x, l, g) = \sum_{i\in Pos}^{N} \sum_{m\in x,y,w,h} S_i x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m) \quad (5)$$

$$\mathbf{S} = [S_0, S_1, S_2, S_3, \ldots, S_i, \ldots], \; i \in Pos \quad (6)$$

$$S_i = \log\left(\frac{a}{(x_{ir} - x_{il}) \times (y_{ib} - y_{it})}\right) \quad (7)$$

$$a = \max\left((x_{ir} - x_{il}) \times (y_{ib} - y_{it})\right) \; i \in Pos \quad (8)$$

With the area scale factor, our loss function reflects the distance between the inference and the ground truth with regards to the difference between the object sizes and it works robustly since the value is adaptive on each batch of samples.

(ii) To optimize the object classification from the different layers of the FPN, we introduce a vector of global weight **W**, which is based on the distribution of the object size in our dataset. The classification loss $L'_{conf}$ is the elementwise product of the above classification loss and **W**. The detail is shown in Eq. (9) and $\boldsymbol{w_i}$ is the i[th] element of **W**. $\hat{c}_p$, $\hat{c}_t$ are the predictions and the ground truth of the object classifications.

$$L'_{conf} = -\sum_{i\in Pos}^{N} w_i x_{ij}^p \log(\hat{c}_i^0) - \sum_{i\in Neg} \log(\hat{c}_i^0)$$

$$\text{where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{9}$$

The pipeline of **W** is shown in Algorithm 1. The weight on each object class by feature layers is represented by a matrix ***H***. Since we have 8 classes of objects and 5 feature layers, the shape of ***H*** is 5 × 8. ***T*** is a vector to reflect a reverse ratio of the mean object size in each class. The large object class has a small value in the corresponding element in ***T*** and vice versa. So, the shape of ***T*** is 1 × 8. Hyper parameter $\alpha$ is applied to balance the difference of the object size between the layers. ***T*** with exponents $[(1+2\alpha),(1+\alpha),(1-\alpha),(1-2\alpha)]$ are weights for the 5 feature layers in the sequence. In our new proposed model, we set to 0.1. The exponents list is [1.2,1.1,1,0.9,0.8]. Finally, we sum the weight of each layer by the object classes and get **W**, which is a vector with shape of 1 × 8. Each element in **W** is the weight value of the related class of object.

**Algorithm 1** Global weight for object classification in multi scales of layers

$\mathcal{S}$ is the mean size of bounding boxes in m object classes. $b$ is the mean size of the total bounding boxes in the trainset. $m$ is the index of the classes of objects. The subscript from 0 to 4 is the index of feature layer in the FPN. There are 5 feature layers in our object detection model.

$\mathcal{S} = \{S_0, .., ..., S_m\}$
$b \leftarrow \{\sum_{i \in m} Si/m\}$
$\mathcal{B} = \{S_1/b, .., ..., S_m/b\}$
$\mathcal{T} = \{1/B_0, .., ..., 1/B_m\}$
$\mathcal{H} = \begin{Bmatrix} \mathcal{T}_0^{(1+2\alpha)} & \dots\dots & \mathcal{T}_m^{(1+2\alpha)} \\ \mathcal{T}_0^{(1+\alpha)} & \dots\dots & \mathcal{T}_m^{(1+\alpha)} \\ \mathcal{T}_0 & \dots\dots & \mathcal{T}_m \\ \mathcal{T}_0^{(1-\alpha)} & \dots\dots & \mathcal{T}_m^{(1-\alpha)} \\ \mathcal{T}_0^{(1-2\alpha)} & \dots\dots & \mathcal{T}_m^{(1-2\alpha)} \end{Bmatrix}$
$\mathbf{W} = \{0, ..., ...0\}$
**for** $i = 0$ to 4 **do**
  $\mathbf{W} += \mathcal{H}_{i,}$;
**end for**
**return** **W**

Eq. (10) shows the overall loss function, which is the sum of weighted location loss and confidence loss.

$$L'(x, c, l, g) = \frac{1}{N}(L'_{conf}(x, c) + \alpha L'_{loc}(x, l, g)) \tag{10}$$

*c,l,g,N* are the class confidences, predicted boxes, ground truth boxes and the number of matched default boxes.

### *4.2 The Backbone Network*

To optimize the performance of the power line surveillance in the smart energy IoT system, we propose a hybrid neural network structure as the backbone network. To improve the object detection accuracy, especially for the small objects, we employ resnet34 blocks as the feature extractor for the first feature layer, which has a large perceptive field size than the other feature layers. In each resnet34 block, there is a shortcut connection between the input layer and the output layer. The relationship between the input vector and the output vector is shown in Eq. (11).

$$y = F(x, W_i) + x \tag{11}$$

*x* and *y* represent the input and out vector of the block, $W_i$ is the weights of the neural network in the residual block. $F(x, W_i)$ shows the mapping relationship of the input vector in the residual block. The first feature layer of the FPN is introduced from the resnet34 block. As discussed above, we expect to detect the small objects from this layer since it has a lager feature map size.

From the distribution of the object size in the dataset, we find that the proportion of big objects is quite low. To improve the efficiency of the model, we employ the structures of the shuffle net v2 blocks as feature extractors for the second and third feature layers, which will focus on the objects in the middle and the big size. In this network block structure, the input tensor is split into two parts with the number of channels in half. One of the branches works as the main branch, which is followed with a 1 × 1 convolution, a 3 × 3

depth wise convolution and a 1 × 1 convolution operation in sequence. The other branch is concatenated with the first branch along the axis of the channels. Finally, a channel shuffle operation is applied to explore more information from each group of channels. We expect the resnet34 block to find more small objects with high accuracy and the shuffle net blocks can detect the middle and large objects with high efficiency. Both of them will contribute to the high accuracy and efficiency of the proposed object detector.

### *4.3 The Feature Pyramid Network*

We introduce a hybrid feature pyramid network from the SSD and RetinaNet structure. To enhance the object detection capacity, especially for the small object, we use a down-top structure, which is inherited from the RetinaNet for the first three feature layers. The rest of the two feature layers are based on the top-down structure with the down sampling operation as the FPN structure of the SSD, which is shown in Fig. 6a.

### *4.4 The Overall Network Structure*

The overall structure we proposed in the deep learning model is shown in Fig. 11. To improve the efficiency, we keep the channel size of the feature map as 256 from the conv3 block. Tab. 1 lists the configuration of the key layers of the model. The location and confidence prediction blocks are separate branches, which will comprise 5 convolution units in a sequence with 3 × 3 kernels.

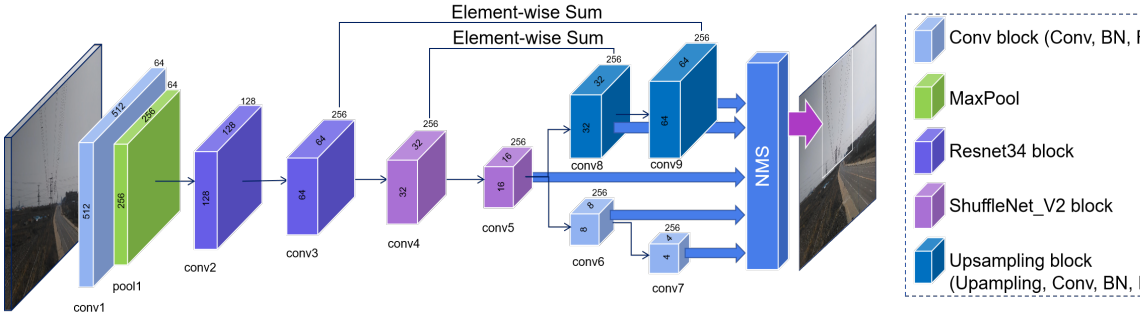



**Figure 11:** Our proposed object detection network structure

**Table 1:** Configuration of the key layers

| Layer Name | Input size (H X W) | Input depth (D) | Output size (H' X W') | Output Depth (D') | Kernel Size (H'' X W'') | Stride ($S_h$ X $S_w$) | Padding ($P_h$ X $P_w$) |
|---|---|---|---|---|---|---|---|
| Conv1 | 512 × 512 | 3 | 256 × 256 | 64 | 7 × 7 | 2 × 2 | 3 × 3 |
| Pool1 | 256 × 256 | 64 | 128 × 128 | 64 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv2 | 128 × 128 | 128 | 64 × 64 | 128 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv3 | 64 × 64 | 128 | 32 × 32 | 256 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv4 | 32 × 32 | 256 | 16 × 16 | 256 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv5 | 16 × 16 | 256 | 8 × 8 | 256 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv6 | 8 × 8 | 256 | 4 × 4 | 256 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv7 | 4 × 4 | 256 | 2 × 2 | 256 | 3 × 3 | 2 × 2 | 1 × 1 |
| Conv8 | 32 × 32 | 256 | 32 × 32 | 256 | 3 × 3 | 1 × 1 | 1 × 1 |
| Conv9 | 64 × 64 | 256 | 64 × 64 | 256 | 3 × 3 | 1 × 1 | 1 × 1 |

## 5 Experimental Results and Discussion

In this section, we evaluate our approach on our manual labelled data set in the VOC2007 format. The data set includes 5418, 1045 and 609 power line surveillance images as a train, validation and test set. In those images, the power line area, engineering vehicles, vehicles, human on a motorcycle, human, bird, wind miller and crane are specified as target objects with information of classes and bounding boxes. We use the mean average precision (mAP) and frame per second (Fps) as metrics to evaluate our approach.

### *5.1 The Experimental Setup*

We designed three experiments to optimize our model and evaluated the overall performance. In the first experiment, we test different backbone networks and compare them with our new proposed Resnet-Shuffle net backbone structure to optimize the backbone for the object detection network. The second experiment is set to test the performance of the different FPN structures with our proposed backbone. The goal of the third experiment is to test our new proposed loss function with the scale factor. Finally, we test our model inference on both the PC platform (CPU: Intel 8700K, GPU: Nvidia GTX1080Ti, RAM:16G).

The initial learning rate is set to 4e-3 and decays exponentially by the steps of iterations. The initial weights of the network are initialized with the kaiming initialization method. The optimizer is a mini-batch Stochastic Gradient Descent (SGD). Settings of those hyper-parameters are shown in Tab. 2.

### *5.2 The Backbone Network*

We test the mainstream of the backbone network with the FSSD network framework. The test result is shown in Tab. 3 and the detail scores of the map on each category of objects are illustrated in Fig. 12.

**Table 2:** The setting of the hyper-parameters

| Items | Value |
|---|---|
| Input Size | 512 × 512 |
| Backbone Layers | 26 |
| Feature Layers | 5 |
| Numbers of Classes | 9 |
| Training Epochs | 500 |
| Batch Size | 8 |
| Optimizer | SGD |
| Initial Learning Rate | 0.004 |
| Momentum | 0.9 |
| Weight Decay | 1e-4 |
| IoU Threshold | 0.5 |
| Negative Positive Ratio | 3:1 |
| Max Detection | 100 |

**Table 3:** The performance of each backbone network

| Backbone network | FPN network | mAP | Fps |
|---|---|---|---|
| Vgg16 | FSSD | 0.502 | 26 |
| Resnet34 | FSSD | 0.511 | 35 |
| Densenet121 | FSSD | 0.506 | 30 |
| MobilenetV2 | FSSD | 0.457 | 38 |
| ShuffleNetv2 | FSSD | 0.440 | 49 |

In this test, Densenet121 and Resnet34 get higher scores of overall accuracy of the power line area and related object detection, while Shuffle net V2 shows better performance in image processing speed. We design a hybrid backbone network with both structures. The test results in Fig.13 shows that the hybrid backbone with the FSSD framework not only performs very well in object detection accuracy like Densenet121 and Resnet34 but also supports faster image processing speed. So, we choose Resnet34+ and Shuffle net V2 structures as the backbone network for our object detector.

### *5.3 The FPN Network*

With our designed hybrid backbone network structure, we test some typical FPN structures on the power line surveillance data set.

The overall performance of each FPN is listed in Tab. 4 and the detail mAP score on each object categories is shown in Fig. 14.

The RetinaNet structure shows better overall performance of the object detection in this test. Since there are only three feature layers in the original RetinaNet, we added more feature layers to the bottom layer of the original RetinaNet FPN structure to improve the accuracy of the power line area and risk the objects detection. Conv6 and Conv7 in Fig. 11 illustrate the added extra feature layers in the RetinaNet FPN framework.

The test results show that the structure of the RetinaNet network with two added features yield better objection detection accuracy with very little loss on the processing speed. This structure is selected as our FPN network.

### *5.4 Loss*

We trained the new proposed object detection model with our new designed multi scale boxes loss function, which introduces an adaptive object area factor to the location loss and a global weight to the classification loss. The loss value per iteration is plotted in Fig. 15. The comparison of the final mAP performance between our objection model trained by the original multi-boxes' loss and our new proposed multi scale boxes loss is shown in Fig. 16. The test result shows that our new proposed loss boosted a 4.41% increase of accuracy in the mAP (from 0.537 to 0.5811) and the average accuracy on the small objects is improved as well.

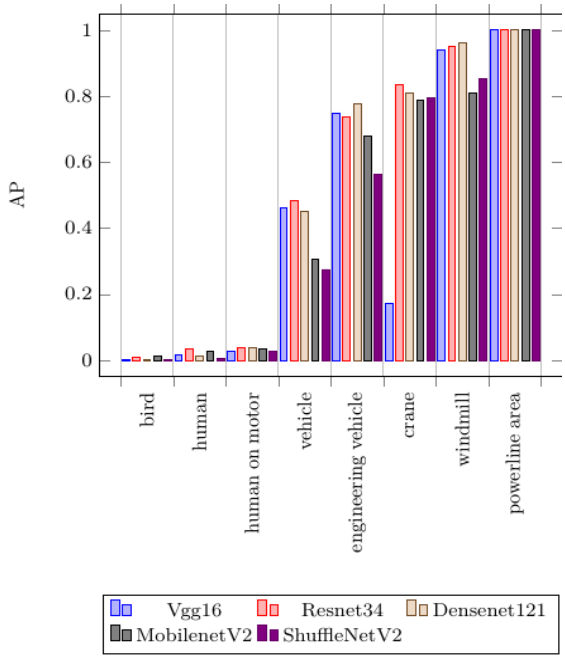



**Figure 12:** The mAP of the backbones in the FSSD on each object of categories

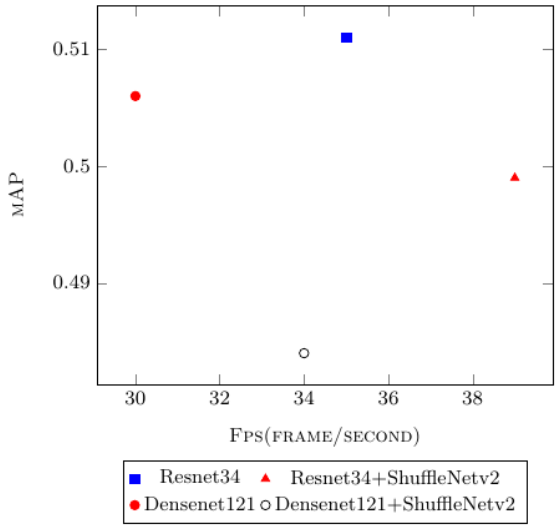



**Figure 13:** The performance of the FPN structures based on the RetinaNet

**Table 4:** Performance of each FPN network

| Backbone network | FPN network | mAP | Fps |
|---|---|---|---|
| Resnet34 + ShuffleNetv2 | SSD | 0.461 | 40 |
| Resnet34 + ShuffleNetv2 | FSSD | 0.499 | 39 |
| Resnet34 + ShuffleNetv2 | RFBNet | 0.466 | 25 |
| Resnet34 + ShuffleNetv2 | YoloV3 | 0.464 | 42 |
| Resnet34 + ShuffleNetv2 | RetinaNet | 0.521 | 38 |
| Resnet34 + ShuffleNetv2 | RetinaNet + 2Convs | 0.537 | 6 |

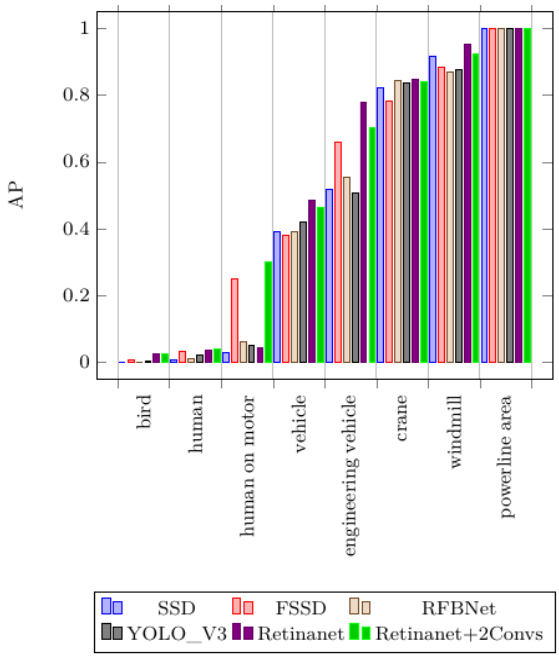



**Figure 14:** mAP of FPNs with our backbone on each object category

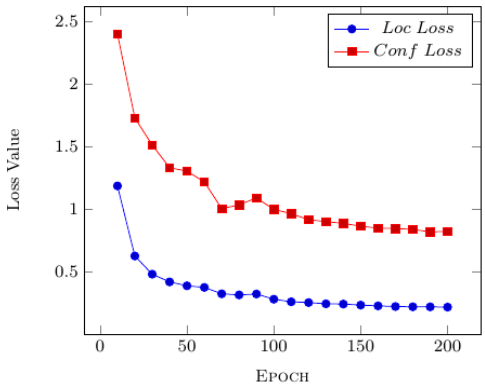



**Figure 15:** Loss in epochs

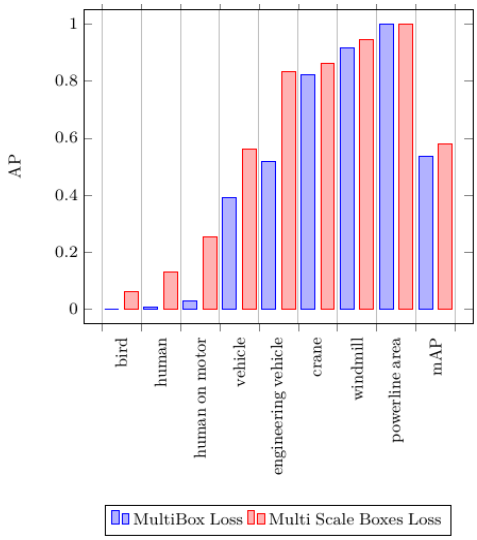



**Figure 16:** The overall performance of the model with our proposed new loss

## 6 Conclusion

In this paper, we present a novel deep neural network-based object detector for power line surveillance tasks in the internet of smart energy, which consists of a customized backbone block, feature map pyramid network and a bounding boxes prediction block. Traditionally, the backbone network is designed generally as an image classification and it is not optimized for object detection in the IoT system. To ensure both accuracy and efficiency, a hybrid backbone network is proposed based on the study of the performance of each typical backbone network on power line surveillance tasks. To further improve the accuracy of the detector, an improved FPN structure, which is based on the structure of the RetinaNet is introduced. To improve the performance of the detector on small objects, an improved multi scale boxes loss, which includes an object size factor as an incentive to the model when small objects are recognized. Test results have been reported on the power line area and the risk object detection. Our proposed object detector, which is trained by our new designed loss out performs other typical object detectors and receives 58.11% in the mAP. The test results also show that our proposed multi scale boxes loss function performs better than the traditional multi-box loss, especially on small objects. In the future, we will continue to study the optimization methods on more object categories for the internet of smart energy.

**Funding Statement:** This work was supported by the Natural Science Fund of China (NSFC) under Grant Nos. 51575186, 51275173, and 50975088, Shanghai Science and Technology Action Plan under Grant Nos.

18DZ1204000, 18510745500, 18510750100, and 19510730600, Shanghai Aerospace Science and Technology Innovation Fund (SAST) under Grant No. 2018-085 and Shanghai Municipal Economic and Informatization Commission Special Founds under Grant No. 201801054.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[2] Y. Li, K. He and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," *Advances in Neural Information Processing Systems*, pp. 379–387, 2016.

[3] T. Y. Lin, P. Doll´ar, R. Girshick, K. He, B. Hariharan *et al.,* "Feature pyramid networks for object detection," in *Proc. CVPR,* pp. 2117–2125, 2017.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR,* pp. 779–788, 2016.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed et al., "SSD: Single shot multibox detector," in *Proc. European Conf. on Computer Vision*, Cham, pp. 21–37, 2016.

[6] T. Y. Lin, P. Goyal, R. Girshick, K. He and R. Dollár, "Focal loss for dense object detection," in *Proc. the IEEE International Conference on Computer Vision,* pp. 2980–2988, 2017.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn and A. Zisserman "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[8] J. R. Uijlings, K. E. Van De Sande, T. Gevers and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision,* vol. 104, no. 2, pp. 154–171, 2013.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR,* pp. 886–893, 2005.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] H. Bay, T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," in *Proc. European Conf. on Computer Vision*. pp. 404–417, 2006.

[13] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. the IEEE Int. Conf. on Computer Vision*, pp. 2564–2571, 2011.

[14] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," *in Proc. CVPR.* pp. 511–518, 2001.

[15] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *in Proc*. *IEEE ICIP*, pp. 900–903, 2002.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] J. Yan, M. Zhu, H. Liu and Y. Liu, "Visual saliency detection via sparsity pursuit," *IEEE Signal Processing Letters*, vol. 17, no. 8, pp. 739–742, 2010.

[18] Y. H. Lee, H. Ahn, H. B. Ahn and S. Y. Lee, "Visual object detection and tracking using analytical learning approach of validity level," *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 205–215, 2019.

[19] J. C. Yan, C. H. Tian, J. Huang and F. Albertao, "Incremental dictionary learning for fault detection with applications to oil pipeline leakage detection," *IET Electronics Letters*, vol. 47, no. 21, pp. 1198–1199, 2011.

[20] C. Zhang, J. Yan, C. Li and R. Bie, "Contour detection via stacking random forest learning," *Neurocomputing*, pp. 2702–2715, 2018.

[21] C. L. Zitnick and P. Dollar, "Edge boxes: Locating object proposals from edges," in *Proc. ECCV*, pp. 391–405, 2014.

[22] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, pp 580–587, 2014.

[23] K. He, X. Zhang, S. Ren and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. ECCV*, pp. 346–361, 2014.

[24] R. B. Girshick, "Fast R-CNN," in *Proc. in ICCV,* pp. 1440–1448, 2015.

[25] K. He, G. Gkioxari, P. Dollar and R. Girshick, "Mask R-CNN," in *Proc. the IEEE Int. Conf. on Computer Vision*, pp. 2961–2969, 2017.

[26] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* "Microsoft coco: Common objects in context," in *Proc. European Conf. on Computer Vision*, pp. 740–755, 2014.

[27] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, pp. 7263–7271, 2017.

[28] R. Joseph and F. Ali, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.

[29] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi and A. C. Berg, "DSSD: Deconvolutional single shot detector," arXiv:1701.06659v1, 2017.

[30] S. Liu, D. Huang and Y. Wang, "Receptive field block net for accurate and fast object detection," arXiv:1711.07767, 2017.

[31] Z. Li and F. Zhou, "FSSD: feature fusion single shot multibox detector," arXiv:1712.00960v1, 2018.

[32] H. Xu, J. Yan, N. Persson, W. Lin and H. Zha, "Fractal dimension invariant filtering and its CNN-based implementation," in *Proc. CVPR*, pp. 3491–3499, 2017.

[33] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang *et al.,* "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. Int. Conf. on Computer Vision*, 2019.

[34] T. Tan, S. Yin, P. Ouyang, L. Liu, and S. Wei, "Efficient lane detection system based on monocular camera," in *Proc. IEEE Int. Conf. on Consumer Electronics*, 2015.

[35] S. Fu, Q. Zuo, Z. G. Hou, Z. Liang, M. Tan *et al.,* "Unsupervised learning of categories from sets of partially matching image features for power line inspection robot," in *Proc. IEEE Int. Joint Con. on Neural Networks*, pp. 2596–2603, 2008.

[36] R. Jenssen and D. Roverso, "Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning," *Electrical Power and Energy Systems*, pp. 107–120, 2008.

[37] X. Xiang, L. Ning, X. Guo, W. Shuai and A. E. Saddik, "Engineering vehicles detection based on modified faster R-CNN for power grid surveillance," *Sensors*, vol. 18, no. 7, 2018.

[38] X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang *et al.,* "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 4, pp. 1486–1498, 2020.

[39] W. Liu, J. He, M. Li, R. Jin and Z. Zhang, "An efficient supervised energy disaggregation scheme for power service in smart grid," *Intelligent Automation & Soft Computing*, 2018.

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

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.,* "Going deeper with convolutions," in *Proc. CVPR*, pp. 1–9, 2015.

[42] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. CVRP*, pp. 770–778, 2016.

[43] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. CVPR*, pp. 1251–1258, 2016.

[44] G. Huang and Z. Liu, "Densely connected convolutional networks," in *Prov. CVPR*, pp. 4700–4708, 2017.

[45] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, pp. 4510–4520, 2018.

[46] N. Ma, X. Zhang, H. T. Zheng and J. Sun, "Shufflenet v2: Practical guidelines for efficient CNN architecture design," in *Proc. ECCV*, pp. 116–131, 2018.