Light-Net: Lightweight Object Detector

YUNBO RAO, (Member, IEEE), GUO YI, JUNMIN XUE, JIANSU PU, JIANGPING GOU, (Member, IEEE), QIUJIE WANG, AND QIFEI WANG, (Member, IEEE)

1School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
2School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China
3School of Computer Science, Guangdong University of Technology, Guangzhou 510006, China
4Department of Electrical Engineering and Computer Sciences (EECS), University of California at Berkeley, Berkeley, CA 94720, USA

Corresponding author: Yunbo Rao (raoyb@uestc.edu.cn)

This work was supported in part by the Science and Technology Service Industry Demonstration Project of Sichuan under Grant 2019GFW126, in part by the Science and Technology Project of Sichuan under Grant 2019YFG0504 and Grant 2020YFG0459, in part by the National Natural Science Foundation of China under Grant U19A2078 and Grant 61806142, and in part by the Sichuan Science and Technology Innovation Project under Grant 2019006.

ABSTRACT Currently, object detectors based on CNN, such as RetinaNet, Faster-RCNN, CornerNet series, can achieve good performance, but have some common drawbacks, like large calculation cost, high model complexity and slow detection speed. In this paper, a new lightweight object detector is proposed, which adopted a density-based approach to merge the real boxes. To reduce calculation cost and improve detection speed, the tactic of multi-scale output is adopted to predict objects of different sizes with features of different scales. Furthermore, a new lightweight network model is proposed, which can show better performance in computation, FPS, and model complexity. Meanwhile, the separation of convolution is used to improve the basic convolution layer, which can achieve better results under the same number of filters. In the experiments, we verified the capability of our methods based on ablation experiment and model evaluation, which demonstrates the superiority of our method. Moreover, we have also conducted deep network and multichannel experiments on MS-COCO2014 datasets and achieved 20.9% mAP performance.

INDEX TERMS Object detection, light-net, prior box, anchor, object detector.

I. INTRODUCTION

Object detection is one of the three fundamental problems of computer vision, which has important applications in automatic driving [1]–[3], image/video [4], [5] retrieval, video monitoring [6], [7] and other fields. At the same time, as being one of the fundamental problems, the research ideas and algorithms in the field of object detection can bring new ideas to the research in other fields of computer vision. As so, The progress of object detection is of huge importance. However, object detection has some limitations. (1) Recent studies had shown that components added or improved in the field of object detection cannot bring about substantial changes. (2) In practical application, the actual effect of object detection will change with different scenes. (3) The improvement of basic components [8], [51] can improve the effectiveness of the whole scene, but its impact on the retrained model is still complex. At present, many researchers focus on the improvement of object detectors. It will be widely recognized and has a good application prospect.

Prior boxes play a major role in object detectors. In object detection stage, some prior boxes are usually set first, and then regressed through the network layer by layer. Prior boxes should be representative and able to represent the size of the boxes in the actual scene sufficiently. Otherwise, it will be difficult for the box to return and the final prediction box won’t fit the real box. In the two-stage detector Faster-RCNN [10], [11], manual settings of aspect ratio and image size will eventually generate prior boxes of 9 different sizes. In the one-stage detector YOLO [12]–[14], the improved k-means [15], [16] algorithms are adopted to merge the true box in the dataset through the calculation of IOU and Distance-IoU (DIoU) [52], and finally generate several groups of boxes, which are of different scales. The artificial method has no explanatory property and strict mathematical proof. Since Faster-RCNN is calculated based on many anchor points, a prior box of the corresponding scale is generated with each anchor point at the center, and unrepresentative aspect
ratio will bring about additional computation overhead in subsequent calculations, such as Non-Maximum Suppression (NMS) [17]–[19].

The object detection model always needs to make a trade-off between detection accuracy and detection speed. Qin et al. [20] proposed that a lightweight network needs to reduce the complexity of the model to prepare for the deployment on mobile devices. However, as a lightweight network also needs to output objects of different sizes, this method is too simple and lacks the process of deep feature extraction and fusion. Aiming to efficiently search for the feature pyramid network as well as the prediction head of a simple anchor-free object detector, Wu et al. [53] used a tailored reinforcement learning paradigm, along with carefully designed search space, search algorithms and strategies for evaluating network quality. However, object detection model always struggles with accuracy.

Based on the above analysis, a new anchor generation algorithm is proposed, which can generate prior boxes more close to its own application scene according to the features of its own dataset, as a substitute for the current scene based on anchor detector with manual settings or k-means. At the same time, a new lightweight network model is proposed, which shows better performance in computation, FPS, and model complexity. Figure 1 shows results of the proposed method and YOLOV3 based on MS-COCO2014 dataset [9], demonstrating the superiority of our method. Meanwhile, the proposed method also achieves a better accuracy, as is shown in Figure 2. In our experiments, we compared it with traditional methods, such as: CornerNetlite, TOLOv3-416, TOLOv3-Tiny416, RFB-Net300-VGG, RFB-Net512-VGG, RFB-Net300-MobileNet, and RFB-Net500-MobileNet. The closer to the origin, the smaller the value, and vice versa. We can see that the proposed algorithm has the
highest average accuracy in several networks, which is less than 10MB.

Our main contributions are summarized as follows.

1) A new generation algorithm of prior boxes is proposed, which adopted a density-based approach to merge [21] the real boxes in the dataset, with the purpose of obtaining the optimal length and width of the boxes and reducing the complexity of subsequent calculation.

2) Multi-scale output is adopted to predict objects of different sizes with features of different scales. Two layers is used to output detection result. Moreover, our proposed method achieves state-of-the-art performance based on MS-COCO2014 dataset.

3) The separation of convolution is used to improve the basic convolution layer and reduce the complexity of the model. A new convolution module is designed and achieved better results under the same number of filters.

The related work is shown in section II. Our proposed Light-Net method will be fully introduced in section III, with datasets and evaluation shown in section IV. Meanwhile, we also compare other methods with ours in many situations. As what Figure.1 and Figure.2 shows, our method performs better than the commonly used methods. The conclusion of this paper is shown in section V.

II. RELATED WORK
In this section, we briefly survey relevant works including unbalanced positive and negative samples, design network, and prior box.

A. UNBALANCED POSITIVE AND NEGATIVE SAMPLES
Sample imbalance [22] mainly refers to the unbalanced sample size of each category during training. Taking object detection [23], [24] based on deep learning as an example, the imbalance of sample categories is mainly reflected in two aspects: the imbalance of positive and negative samples, and the imbalance of hard samples. Generally, the ratio of positive and negative samples is 1:3 (experience value) in the object detection task framework. Samples that can be divided easily take up a large part of the whole samples. And even if the loss function of a single sample is small, the cumulative loss function will dominate the loss function. And as these samples can be well classified by the model anyhow, parameters updated at this stage will not improve the judgment ability of the model. In the training process of indivisible samples, however, the loss function of individual samples is relatively high and has diversity, but the proportion of such samples in the total samples is relatively small. This will lead to low training efficiency, and even failure in convergence.

The size of the object is much smaller than that of the background. For a region-based detector, positive samples are much fewer than negative ones. If such extremely unbalanced data is directly used to train the classifier, the classifier may be inclined to divide all samples into negative samples. One solution is online hard negative mining. Shrivastava et al. [25] proposed an online hard example mining method, which can automatically select those hard examples to join the training during the training process. This method makes the training more effective and fast, and can improve the performance of the network.

Focal loss is proposed to solve low precision problem of one-stage detection algorithm [12], [13]. For the class imbalance problem, the focal loss function focuses on the class imbalance of one-stage detector. This function is modified based on the standard cross entropy loss. This function can make the model focus more on the hard samples during training by reducing the weight of the easily classified samples. However, this method has low detection accuracy.

Two-stage detectors represented by Faster-RCNN [10], [11] generally have relatively high detection accuracy, but slow detection speed. One-stage detectors, represented by YOLO, have fast detection speed, but relatively low detection accuracy. The accuracy of one-stage detectors is not as good as that of two-stage detectors. One important reason is that samples are extremely unbalanced during training. For example, there are only a few hundred candidate boxes after Region Proposal Network (RPN) screening, while one-stage algorithms, such as YOLOv3, usually have tens of thousands of candidate boxes, most of which are negative samples.

B. DESIGN NETWORK
The backbone network for object detection is usually borrowed from the ImageNet [26] classification. In recent years, ImageNet has been regarded as the most authoritative dataset to evaluate the functions of deep convolutional neural networks. Many novel networks aim to achieve higher performance for ImageNet. AlexNet [27] was the first network to try to increase the depth of CNN. To reduce network computation and enlarge the receptive field, AlexNet used 32 step sizes to sample down the feature graph. VGGNet [28] stacked 3 × 3 convolution operations to build a deeper network, while still involving 32 spans in the feature graph. Most of the following studies used a structure similar to VGG and designed a better component at each stage.

GoogleNet [29] proposed a novel starting block to contain more diverse functions. ResNet [30] adopted a “bottleneck” design with residual summation at each stage. ResNet and Xception used a group convolution layer to replace the traditional one. It reduces parameters and improves accuracy. DenseNet [31], [32] connected several layers densely. There are still many studies on effective backbone network, such as [33]–[35].

Since backbone network is typically designed for classification, many backbone networks [36] have recently been redesigned for object detection. Although these methods cannot use pre-trained weights from classification networks, the redesigned object detection networks pay more attention to location detection, not just classification detection.

C. PRIOR BOX
There are two ways to generate pre-selected boxes for object detection: anchor-free and anchor-based. The anchor-free
FIGURE 3. Overview of Light-Net. We propose a new network for object detection and design two output layer for images what have different sizes.

[37] includes the original DenseBox, which uses a single Fully Convolutional Networks (FCN) [38] to simultaneously produce multiple outputs of predictive bboxes and confidence scores. During the test, the entire system takes the picture as the input and outputs the feature map of 5 channels. The output feature map of each pixel is a 5-dimensional vector, including a confidence score and four values from the bbox boundary to the pixel. Finally, each pixel of the feature map is converted into a bbox with mixed scores and then processed after NMS. Later, YOLOv3, CornerNet [39], ExtremeNet [40], and CenterNet [41] skip anchors by predicting some key points. Feature Selective Anchor-Free(FSAF) [42] automatically searches through the network architecture to allow different object to select the most appropriate feature map for prediction, and no longer receive the constraints of the anchor. In the one-stage detector, these candidate areas are the anchors generated by sliding Windows. In the two-stage detector, the candidate region is the proposal for RPN generation, but RPN itself still classifies and regresses the anchor generated by sliding window. In the anchor [43] method, the detection problem is solved by another method. This time, the problem is also divided into two subproblems, namely, determining the center of the object and predicting the four borders. When predicting the center of an object, the specific implementation can either define a hard center area and integrate the center prediction into the object of the category prediction, or predict a soft centerless score. For the prediction of the four borders, it is relatively consistent, which is to predict the distance between the pixel and the ground truth box. In the anchor-based method, although there may be only one anchor per location, the predicted object is matched based on that anchor, while in the anchor-based method, it is usually matched based on the point.

Most object detection algorithms rely on the setting of the initial anchor when generating the prior box. Each set of coordinates represents the coordinates of the upper left corner and the lower right corner of anchor boxes (x1,y1,x2, and y2). According to Spatial Pyramid Pooling network (SPP-net) [44] and a multi-scale image pyramid thought, sufficient anchor boxes can be obtained reversely.

III. OUR METHOD: LIGHT-NET

In our work, we proposed a lightweight object detection model: Light-Net, which takes into full consideration the characteristics and existing shortcomings of the lightweight network. Our proposed overview of light-net is shown in Figure 3. The detailed processing will be described as follows.

A. PRIOR BOX GENERATION ALGORITHM

Faster-RCNN contains a large number of anchor points, with each anchor point producing 9 anchor boxes, which can cover object boxes of various scales and shapes. However, this approach produces too many anchor boxes, resulting in redundant boxes, as is shown in Figure 4.
YOLO detector uses the K-means algorithm to generate the anchor, as is shown in Figure 5. YOLO uses true box size as the input of clustering algorithms. From Fig.5, we can conclude that YOLO’s prior box algorithm is better than previously related algorithm.

Both prior algorithms mentioned above have insufficiency:

1) The method of generating anchor comes from the image feature pyramid model [45]. The logics of anchor can be explained through the reverse calculation of the feature graph. However, the prior box generated by manually setting anchor value cannot represent the true box in the dataset well, for there is no mathematical logic behind the value.

2) K-clustering centers are artificially set, and when the value of the clustering center is updated iteratively, the overlap of all remaining points needs to be calculated, which is too much.

3) K-means randomly initialize the cluster center of the first round of iteration. If the initial cluster center is selected incorrectly, it will be quite time-consuming and even affect the final result.

Meanwhile, we also describe the process of feature extraction of YOLO and Faster-RCNN algorithm based on anchor. This is shown in Figure 6. From Figure 6, this processing includes three stages: multi-layer feature extraction, feature map, and anchor generation.

In our work, to resolve the insufficiency of prior anchor generation algorithms, a novel distance measurement function is proposed to make the cluster center always shift towards the denser place. The proposed algorithm does not need to set the anchor aspect ratio values in advance and only use two artificial parameters $N_a$ and $N_b$ to control the performance of classification and measurement. The input parameter is a dataset containing the length and width of all true boxes. We traverse all points, classify points with high similarity into one class, and finally merge adjacent classes with high similarity. Eventually, the appropriate number of categories and corresponding center point coordinates will be automatically generated. This center point coordinate is the more representative anchor coordinate. The detailed processing will be described in Algorithm 1 and Algorithm 2. We traverse all the data through Algorithm 1 and get preliminary classification data through calculation. Then, all categories are merged with high similarity through Algorithm 2.

We also compared the speed difference between our algorithm and YOLOv3’s algorithm for generating prior boxes. Please see Table 1. Unfortunately, our algorithm is not faster than the algorithm that generates prior boxes in YOLOv3, as the proposed algorithm has a higher time complexity than the other. But the running time of this part is not important in the whole neural network model. Training a neural network requires dozens of hours, or even hundreds of hours, but for a prior box, whether using our algorithm or the YOLOv3 algorithm, takes only a few hundred seconds. We can see that the time of the YOLOv3 method increases with the increase in the number of anchors, but there is a certain fluctuation in the middle. We also found in the experiment that for the same number, the results of multiple experiments are different, sometimes even by more than 50%, and this is because of the instability of the k-means algorithm used by YOLOv3 due to its initialization method.

The proposed algorithm is different from the k-means generation method. The anchor value generated at the beginning is larger, and then the result of this time is entered into the algorithm again to generate a smaller number. Since the proposed method only consume a lot of time for the first result, and the subsequent time consumed can be ignored
FIGURE 6. The picture on the left is the schematic diagram of anchor box of YOLO and Faster-RCNN, respectively. By multi-layer feature extraction, the anchor box on the final feature map maps to the location schematic diagram in the original figure. The picture on the right is the anchors generated by feature extraction.

TABLE 1. We compared the speed of two algorithms for generating a priori boxes. The operating environment is Intel i5 processor, Mac OS X system. The time unit is seconds.

| Anchor Number | k-means(YOLOv3) speed | Ours algorithm speed |
|---------------|------------------------|---------------------|
| 5             | 12                     | 471                 |
| 10            | 16                     | 463                 |
| 15            | 38                     | 461                 |
| 20            | 46                     | 450                 |
| 25            | 37                     | 465                 |
| 30            | 51                     | 445                 |
| 35            | 95                     | 421                 |
| 40            | 226                    | 431                 |

(because the magnitude of the input becomes small), the time consumption is relatively stable.

B. SEPARATION CONVOLUTION AND MULTI-SCALE

For the calculation amount of convolutional layer, the idea of separation is adopted to design step block and replace $N \times N$ convolution with $N \times 1 + 1 \times N$ in the upper level. Our convolution block improves the traditional separable convolution module. While the number of filters being the same, different convolution methods are used for fusion. At the same time, we adopted the idea of fewer channels and multiple layers to construct the network and extract deep information as much as possible. The proposed convolution block is shown in Figure 7.

FIGURE 7. The left is the standard convolution block. The center is the conventional separation convolution block. The right is our proposed convolution block. The proposed convolution module improves the traditional separable convolution module.

The convolution kernels of $1 \times 1$ and $3 \times 3$ is used to give the convolution kernels of $5 \times 5$ or larger sizes and design two output layers for the prediction of images of different...
scales. Compared with the commonly used lightweight network model, the number of channels and network depth is reduced.

A new similarity function is proposed. The overall idea is that the center of the cluster, which always moves towards the direction of the maximum density of the dataset, is updated by the average position of the points inside the circle during every iteration. And the central coordinates of the circle are updated by calculating the most representative size through clustering the true box of the dataset, covering all boxes as much as possible, and discarding the previous anchor generation method.

1) DISTANCE FUNCTION

As we know, if Euclidean distance is directly used as the measurement function, big bounding boxes will generate more errors than small ones. However, we hope to obtain good IOU scores through anchor boxes, and IOU scores are independent of the box size. So we get the new distance function by calculating the IOU. The value of IOU is large, which is what we expect. In our work, \( 1 - \text{IOU} \) is used to make \( D(x) \) as small as possible for the ease of calculation.

Since the prediction phase generates multiple prior boxes based on anchor at multiple points in the image, we only care about the aspect ratio of the box, and not the location of the coordinate of center point. Usually, the label of the dataset is of XML or TXT format, where the coordinate information of the true box can be easily read. The detailed calculation of the IOU function is shown in Algorithm 3. The proposed algorithm assumes that the center of all boxes are at the origin of the coordinates, and is calculated by reading the upper-left and lower-right coordinates from the dataset label. The \( a \) and \( b \) are boxes from the dataset, the value of the first coordinate position is the width, and the value of the second coordinate position is the length. The proposed equation is shown as follows:

\[
D(a, b) = 1 - \text{IOU}(a, b)
\]

In our work, there is \( N \) pieces of data in a given dataset, so the basic form of \( M \) of \( x \) for any point \( x \) in space can be expressed as \( V(x) \), and \( S_h \) refers to a high-dimensional

---

Algorithm 1 The Proposed Prior Box Generation Algorithm for Variable Initialization. It Receives the Original Set of Coordinates and Outputs the Corresponding Category and Center Point

Input: Var: class, classes center, Centers point, Points
Output: classes
Variable initialization for point in Points do
  center = point class, add(center) for point in Points do
    if point != center then
      maxval = 0 temp = V(center, point)
      center+ = temp if D(center, point) > N\beta then
        continues
      end
    else
      if center \notin \text{classes} then
        class, add(point)
      else
        continues
      end
    end
    tval = D(center, point) maxval = max(maxval, tval) if maxval < N\alpha then
      break
    else
      continue
    end
  end
else
  continue
end
return newcenter, newclass

Algorithm 2 The Proposed Prior Box Generation Algorithm for Merging Newclass. It Receives the Output Algorithm to Sort and Merge the Redundant Output

Input: Var: class, classes Function: Merge() => void Merge
Function: Merge two classes
Output: newclass, newcenter
for class in classes do
  center, \in \text{class} for c in classes do
    center \in c if class \notin c then
      if D(center, center) < N\beta then
        Merge(c, class)
      else
        continue
      end
    end
end
return newcenter, newclass

Algorithm 3 IOU Algorithm, \( a \) and \( b \) Are Two Different Boxes, Both of Which Are Binary Groups

Input: Variable: \( a = w_1, h_1 \), \( b = w_2, h_2 \)
Output: \( s_{\text{intersection}} / s_{\text{union}} \)

\[
w = \min(w_1/2, w_2/2) - \max(-w_1, -w_2) \quad h = \min(w_1/2, w_2/2) - \max(-h_1, -h_2) \quad \text{if} \quad w < 0||h < 0
\]

\[
s_{\text{intersection}} = w \times h \quad \text{if} \quad s_{\text{union}}! = 0 \quad \text{then}
\]

\[
s_{\text{union}} = x_1 \times l_1 + x_2 \times l_2 - s_{\text{intersection}} \quad \text{else}
\]

\[
\text{continue}
\]

end
return 0
spherical region with a radius of \( h \). The proposed equation is shown as follows.

\[
V(x) = \frac{1}{k} \sum_{x_i \in S_k} D(x, x_i) \tag{2}
\]

\[
S_k(x) = D(a, b) < N_a \tag{3}
\]

2) RADIAL BASIS FUNCTION KERNEL

Kernel function is widely used and plays an important role in many statistical problems. The basic idea of kernel function is to transform problems that cannot be solved by low dimension into high dimension. This is done by raising dimension. For example, when we classify the points in a plane, cases often occur where it is difficult to divide the points and find the partition function. In this case, we map the dimensions to three-dimensional space and thus transform the two-dimensional non-linear problem into the three-dimensional linear problem, so as to solve it. When the dimension is large, it can hardly find the mapping function, and that is just what the kernel does. The kernel avoids solving the mapping function, which is called the kernel trick.

In our work, a Gaussian kernel function [46] is introduced as the weight parameter, which can make the point with larger IOU to have a larger weight, meaning that the larger the distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel function works as the weight parameter, which can make the point with larger IOU to have a larger weight, meaning that the larger the distance from IOU, the larger the weight. The kernel avoids solving the mapping function, which is called the kernel trick.

In our work, a Gaussian kernel function [46] is introduced as the weight parameter, which can make the point with larger IOU to have a larger weight, meaning that the larger the distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel avoids solving the mapping function, which is called the kernel trick.

3) RADIAL BASIS FUNCTION KERN

Kernel function is widely used and plays an important role in many statistical problems. The basic idea of kernel function is to transform problems that cannot be solved by low dimension into high dimension. This is done by raising dimension. For example, when we classify the points in a plane, cases often occur where it is difficult to divide the points and find the partition function. In this case, we map the dimensions to three-dimensional space and thus transform the two-dimensional non-linear problem into the three-dimensional linear problem, so as to solve it. When the dimension is large, it can hardly find the mapping function, and that is just what the kernel does. The kernel avoids solving the mapping function, which is called the kernel trick.

In our work, a Gaussian kernel function [46] is introduced as the weight parameter, which can make the point with larger IOU to have a larger weight, meaning that the larger the distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel avoids solving the mapping function, which is called the kernel trick.

In our work, a Gaussian kernel function [46] is introduced as the weight parameter, which can make the point with larger IOU to have a larger weight, meaning that the larger the distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel function is just a simple method to calculate the inner product after distance from IOU, the larger the weight. The kernel avoids solving the mapping function, which is called the kernel trick.

D. FOCAL LOSS FUNCTION

An appropriate function is used to measure the contribution of difficult-to-classify and easily-to-classify samples to the total loss. In our work, the focal loss function is used as our loss function. \( \alpha \) is called equilibrium factor, which is used to balance the uneven proportion of positive and negative samples, and the weight of positive samples increases when the value decreases. \( \beta \) is used to balance the uneven proportion of difficult samples. \( \beta \) reduces the loss of easy samples and makes the model focus more on difficult and misclassified samples. In our work, we use values similar with [22], with \( \alpha \) equals to 0.25, and \( \beta \) equals to 2. Both \( \alpha \) and \( \beta \) are empirical values, and we have also verified the influence of other values on the algorithm in the subsequent experiments. The proposed loss function is described as follows:

\[
F_{\text{loss}}(p_t) = -\alpha \times (1 - p_t)\beta \times \log(p_t) \tag{6}
\]

IV. DATASETS AND EVALUATION

A. K-MEANS AND OURS, WHICH IS BETTER?

K-means is the latest algorithm used to generate anchor. In this section, a set of camparsion experiments of K-means with the proposed algorithm will be conducted. Meanwhile, along with the clustering algorithm, the advantages and disadvantages of the two algorithms will also be evaluated.

We investigated the true box distribution in the MS-COCO2014 dataset, as is shown in Figure.9. It can be seen that the size distribution of the box is not uniform, as the amount of larger boxes is less than that of smaller ones, which is more apparent in the verification set. This is especially true in our practical applications.

Our method does not need to pre-set the number of anchors. The algorithm will automatically generate the appropriate number of anchors according to the threshold \( N_a \) and \( N_B \) value and the dataset. It is worth noting the comparison of our algorithm with related ones, such as YOLOv3 and Faster-RCNN. In order to control the variables, the same number of anchors for comparison is used in our work. Through the experimental comparison under MS-COCO2014 dataset, we drew a line chart of the number of anchors and the avg-IOU of YOLO and the prior box algorithm. The experimental results will be shown in Figure.10. From Figure.10, we can find that:
FIGURE 9. The distribution of MS-COCO2014 dataset, including training set and verification set. The x-coordinate is the width, and the y-coordinate is the height. From the frame distribution pictures of the MS-COCO14 dataset, it can be seen that the distribution of the data set is not uniform, and there are fewer pictures with larger length and width than smaller pictures. The true box distribution of the dataset is not uniform, so it is necessary to fit the suitable candidate box value according to the distribution of the true box.

TABLE 2. Compared with our algorithm and YOLO9000, Faster-RCNN in different anchor numbers.

| Anchor Generation | Anchor Numbers | Avg-IOU |
|-------------------|----------------|---------|
| Faster-RCNN       | 9              | 59%     |
| YOLOv3            | 9              | 60.9%   |
| YOLOv3            | 6              | 53.23%  |
| YOLOv3            | 11             | 62.94%  |
| Our algorithm     | 9              | 63.36%  |
| Our algorithm     | 6              | 56.75%  |
| Our algorithm     | 11             | 67.29%  |

(1) The proposed algorithm is always higher as to the avg-IOU than the YOLO series algorithm.
(2) When the number of anchors exceeds 35, the increase in average IOU is not obvious.
(3) Avg-IOU continues to rise with the increase of anchor number, whether it is YOLO or our proposed algorithm.

The Avg-IOU of our algorithm is always higher than YOLOv3 with the same anchor number, as is shown in Table 2. From table 2, the proposed algorithm is always better than the other two methods on Avg-IOU.

TABLE 3. Experimental results of faster-RCNN, YOLO algorithms, and the proposed method based on comparison of backbone and mAP.

| Methods            | backbone | mAP   |
|--------------------|----------|-------|
| Faster-RCNN        | ResNet101| 34.9% |
| Faster-RCNN + Ours | ResNet101| 36.1% |
| YOLOv3 – 416       | darkNet  | 31%   |
| YOLOv3 – 416 + Ours| darkNet  | 31.8% |

TABLE 4. We take some ablation experiment include loss function, prior box algorithm. Focal Loss=FL, Cross-entropy=CE.

| Method               | mAP   |
|----------------------|-------|
| Light – Net(With FL)| 20.9% |
| Light – Net(With CE)| 20.1% |
| FL + K – means       | 19.3% |

TABLE 5. We here verify the influence of $\alpha$ and $\beta$ on our algorithm.

| $\beta$ | $\alpha$ | mAP |
|---------|----------|-----|
| 0.5     | 0.25     | 18.7% |
| 0.5     | 0.5      | 19.7% |
| 0.5     | 0.75     | 19.4% |
| 2       | 0.25     | 20.9% |
| 2       | 0.5      | 19.5% |
| 2       | 0.75     | 20.1% |

We also compared our prior box generation algorithm on different backbone in Table 3, along with Faster-RCNN and YOLO algorithms, to show the improvements to the candidate box algorithm in our work. In this experiment, we used our prior box algorithm to replace the original method in Faster-RCNN and YOLO algorithms, respectively, in order to intuitively show the degree to which the improvement of our prior box algorithm of this network. Experiments show that our algorithm owns portability and effectiveness, and has an effect on other different backbones.

B. ABLATION EXPERIMENT

Based on the light-net network and backbone as LightNet32, we verified the contribution of the prior box algorithm and loss function to this model in Table 4. The first line is the prior box algorithm of our work with the focal loss function. The second line is the prior box algorithm of our work with the cross-entropy loss function. The third line is the focal loss function with the K-means algorithm. Meanwhile, to verify the influence of different values of $\alpha$ and $\beta$ in our work, we compare different values of $\alpha$ and $\beta$ in Table 5. From Table 5, we can find that the better for our work being $\alpha$ set to 0.25 and $\beta$ set to 2.

C. DEEP NETWORK OR MULTICHANNEL

Research shows that the convolution kernel stores the characterization parameters of the entire network, and the stronger the parameter characterization ability, the better the detection performance. In general, the deeper the network is, the more the number of parameters is, and the stronger the representation ability of the model is. Moreover, the higher the
convolution kernel dimension of each layer is, the stronger
the representational ability of the model is. Because each
convolution process will affect the feature map size, the final
feature map size of the same block is kept by controlling the
step size. Finally, when we want to control the number of
parameters for the sake of model complexity, which one is
better, multi-channel or deeper network? Through our exper-
imental comparison, we find that deeper network works better.
We speculate that the deeper layer of the network has the
ability to represent the upper layer of the network, while the
multi-dimensional convolution of a single layer stores more
overall feature information.

We verify the performance of network depth and
multi-channel on the network model through experiments to
find the best balance for shallow networks. In our experi-
ment, light-net and other networks are tested and evaluated under
the MS-COCO2014 dataset to verify the influence of network
depth and channel number on the performance of the network
model. Experimental results are shown in Table 6. In our
experiment, the light-net performance is evaluated with the
same design, but different layers (Light-Net22, Light-Net32,
and Light-Net42) and different channel in convolution kernel.
By combining 2 filters of $N \times S \times S$, we form a filter of
$2N \times S \times S$. Table 6 shows that deep network with tiny chan-
nels works better than shallow network with big channels,
even if they have filters of the same size.

D. MODEL EVALUATION IN MS-COCO2014

In our experiment, multi-scale training, data augmenta-
tion, batch normalization, and all standard content will
be used. The light-net network is used for training and
testing. In this experiment, we compare the traditional
methods (CornerNetlite, RefineDet, CFENet, YOLOv3-416,
YOLOv3-Tiny-416, RFB-Net300, RFB-Net512, RFB-Net300,
and Light-Net) with our method (Light-Net). Experimen-
tal results in Table 7 show that although being not as accurate
as some of the current state-of-the-art algorithms, our method
has reached good results in the MS-COCO2014 dataset. The
main reason is of two points. (1) Training from scratch, and
not using pre-trained weights. (2) The number of iterations is
limited due to hardware constraints. In this experiment, most
algorithms use the pre-trained weights of the classification
network model, which will give them an advantage in the

FIGURE 11. We randomly selected some classes and drew a bar chart, as we can see there are some classes mAP is very low. Experimental results are shown, which is based on MS-COCO2014 datasets.

FIGURE 12. We compared the detection time and model parameter amount of CornerNetlite, YOLOv3-416, RFB-Net300-VGG, RFB-Net512-VGG, and our algorithms. The proposed algorithm has obvious advantages.

| Method          | Parameters | FLOPs | mAP  |
|-----------------|------------|-------|------|
| Light-Net22     | 6.1MB      | 7.26GMac | 19.8% |
| Light-Net32     | 7.3MB      | 7.66GMac | 20.9% |
| Light-Net42     | 9.76MB     | 9.3GMac  | 21.5% |
Table 7. We verify our detector and another detector in MS-COCO2014 dataset. The metric include mAP, detector parameters, detection time, and FLOPs in NVIDIA GTX 1070.

| Method         | Language | Backbone | mAP  | time | parameters | FLOPs      |
|----------------|----------|----------|------|------|------------|------------|
| CornerNetlite  | Python   | Hourglass| 34.4%| 30ms | 12.5MB     | 13.5GMac   |
| RefineNet      | Python   | CENet    | 17.4%| 44.8ms|            |            |
| CFENet         | Python   | CFENet512| 19.1%| 47.6ms|            |            |
| YOLOv3-416     | Python   | DarkNet  | 31.3%| 39ms | 61.5MB     | 32GMac     |
| YOLOv3-Tiny-416| Python   | DarkNet  | 18.9%| —    | 8.85MB     | 2.81GMac   |
| RFB-Net300     | Python   | VGG      | 30.3%| 15ms | 36.53MB    | 68.43GMac  |
| RFB-Net512     | Python   | VGG      | 33.8%| 30ms | 37.42MB    | 68.43GMac  |
| Light-Net      | Python   | Light-Net32| 20.9%| 25ms | 7.3MB      | 7.66GMac   |

Figure 13. Experimental results of the proposed network detection. Dataset from the MS-COCO2014 dataset. The IOU value is set to 0.5.

detection model. At the same time, the accuracy of RFB-net is the same as our proposed method without pre-trained weights and enough iterations. In the future, we believe that if we increase the number of iterations and apply further training techniques such as data augmentation, our detectors can achieve better performance.
Further, to get a fair mAP%, we randomly selected 30 classes from MS-COCO2014 datasets. As is shown in Figure 11, our detector has achieved better performance in most classes.

We evaluate the related works in parameters and times, such as: CornerNetlite, YOLOv3-416, RFB-Net300-VGG, RFB-Net512-VGG, and our model. Experimental results are shown in Figure 12. Our model has great advantages in model parameter and detection time.

At the end, to show that our proposed network has good detection results, we give experimental results in Figure 13.

V. CONCLUSION
In this paper, a novel light-net and a new detector for object detection is proposed, which includes a prior-boxes generation algorithm and a new convolution block. The proposed prior box method takes into account the distribution density of the true boxes of the dataset, and the new convolution block reduces the number of parameters in the traditional convolution module, while keeping a stable accuracy. The proposed light-net method has achieved competitive results on MS-COCO2014 dataset, which is no worse than the latest detectors in training from scratch.

REFERENCES
[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” Int. J. Robot. Res., vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[2] W. Liu, S. Liao, and W. Hu, “Learning efficient one-stage pedestrian detectors by asymptotic localization fitting,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 643–659.
[3] X. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 8445–8453.
[4] J. Siciv and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in Proc. 9th IEEE Int. Conf. Comput. Vis., Oct. 2003, p. 1470. doi: 10.1109/IEEE2003.1238663.
[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2007, pp. 1–8.
[6] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, and P. Burt, “A system for video surveillance and monitoring,” Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., 2000, pp. 1–68.
[7] B. Jiang, R. Luo, J. Mao, T. Xiao, and A. Jiang, “Acquisition of localization confidence for accurate object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 784–799.
[8] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: Real-time surveillance of people and their activities,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 809–830, Aug. 2000.
[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis. Berlin, Germany: Springer, 2014, pp. 740–755.
[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.
