ABSTRACT The ability to identify pets via smart home technology is crucial for managing pet health. However, the present popular object detection method model is too large, support for edge devices is insufficient, and performance is subpar for small object identification in the traditional algorithm, such as SSD. In this study, we propose an improved SSD object detection network algorithm based on a GhostNet network with a dual attention mechanism. A ghost module is introduced to create a lightweight ghost network based on the SSD network. The ghost module is combined with the ECA attention mechanism to dynamically allocate parameters to the target and alter the detection region weights to enhance the model performance. In order to reinforce the model and increase the precision of the small target, the CBAM module was also introduced. It assigns major weights to the output target regions of the large-scale feature layer. According to the experimental findings, the CBAM-GhostNet-SSD network significantly reduces the number of backbone parameters and the calculation amount is reduced by 98.23% when compared to the classic SSD. This lays the groundwork for the self-developed algorithm’s successful deployment of edge devices. The predicted rate synchronization increased by 3.1 times, also mAP was 14.5% higher than SSD. The lightweight model quantitative transformation utilized in edge equipment can accurately evaluate the target region and realize dynamic detection. This has a certain guiding significance for the subsequent detection of household pet targets.

INDEX TERMS Deep learning, object detection, ghostnet, attention modules, edge devices.

I. INTRODUCTION
With the progress of technology, the pace of people’s lives is accelerating, and young people are mostly working away from their parents and friends. Therefore, the problem of companionship bothers most young people [1]. To relieve loneliness and anxiety when living alone, people choose to keep pets to accompany them [2], and among the breeds of pets, most of them are dogs and cats [3]. The number of pet dogs and cats in China is now up to 200 million, and still growing at a rate of 15% per year [4]. However, with owners working outside most of the time, pet dogs and cats cannot have the same environmental adaptability as humans and cannot take care of themselves. Furthermore, pets with a wild nature are more likely to destroy furniture or other property items in the home [5].

Whereas many smart devices entered regular life with the introduction of edge devices. To address the demand of young people for pet dogs and cats, smart pet feeds, pet exercise equipment, pet cat toilets, and other developments have made it easier to care for young people’s pets outside [6]. However, there are few intelligent devices in the pet home market that monitor pets as well as intelligent patterns to identify pet categories, record and judge home pet activity areas and behaviors, analyze pet health status intelligently, control pet activity habits accurately, realize owners’ remote understanding of intervention pet behavior outside, and exclude safety hazards when pets are left alone.

Object detection is an important topic in computer vision and machine learning research, in 2012, after the breakthrough of the convolutional neural network (CNN) in the...
ImageNet image classification task, Girshick et al. successfully introduced CNN into the object detection task and proposed the regional convolutional neural network (R-CNN) model [7]. R-CNN generates regions of interest by selective search and extracts features for all regions, so the detection speed of R-CNN is slow. To solve this problem, Girshick et al. proposed the Fast-RCNN model [8].

Based on multi-task learning, Fast-RCNN training on target classification and bounding box regression synchronous. Compared with R-CNN, the detection speed is nearly 20 times, in the performance on the same VOC dataset, the mAP achieves more than 70 % [9]. However, the common advantage of the above proposal object detection algorithms is that the mAP is relatively high, but the FPS is low and cannot satisfy the requirements of real-time detection. Therefore, people propose one-order object detection algorithms that directly calculate the category and location information of the object, but the mAP of the one-order algorithm is low compared to the second-order algorithm due to the missing step of calculating the proposal regions.

At present, the representative one-order target detection methods are YOLO [10], RETINA-NET [11], and SSD [12]. Although YOLO and SSD and other object detection algorithms have obvious advantages in speed, they are ineffective for small-scale object detection, and the backbone network model is too large, with too many parameters and huge computation, which is not suitable for directly applied in embedded edge devices where computational resources are scarce.

The advantages of the one-order algorithm on real-time detection are considered comprehensively to improve its mAP. Based on the structure of the one-order algorithm model, a lightweight ghost network is introduced to reduce the size of the object detection model and improve its attention mechanism. Finally, the convolutional block attention module (CBAM) [13] is combined to improve the mAP of the algorithm to realize the application of the object detection algorithm on edge devices. Our contributions are mainly concluded as follows.

1) Lightweight modification of the SSD object detection, using a ghost module to generate the SSD’s backbone, replacing its original VGG network, the modified object detection outperforms the SSD network in terms of detection accuracy results.

2) An adaptive attention mechanism module is introduced into the ghost module so that the assign major detection weights to the target and the network automatically care to process key target areas, improving the network detection accuracy furthermore. The modified SSD network adds CBAM modules after the output feature map to improve the results of the poor detection accuracy in small target areas by combining spatial attention and channel attention, which suppress invalid noise in the background region.

3) By removing operators that are not supported by model inference in the RKNN framework, while quantizing the lightweight network model, converting it into an RKNN model under embedded devices, which can be used for object recognition and detection in static images and dynamic video streams.

The remainder of this article is organized as follows. Section II presents an overview of related works. In Section III, we introduce the infrastructure of the GhostNet and the detailed network architecture of CM-GhostNet-SSD. We demonstrate the experiment and evaluation results using real-world data sets, meanwhile applied in the edge equipment in Section IV. In Section V, we conclude this study and give some suggestions for improvement in future research.

II. RELATED WORKS

A. CLASSICAL SSD OBJECT DETECTION ALGORITHM

The SSD object detection algorithm is mainly divided into a backbone layer, an extra feature layer, and a prior frame prediction layer, and the algorithm block diagram is shown in Fig. 1.

In the backbone layer, based on the VGG-16 network, in order to reduce the scale of the original feature map, the SSD algorithm transforms the fully connected layers FC6 and FC7 in the VGG-16 into the convolutional layers CONV6 and CONV7, as well as the pooling layer POOL5 with strides of 2 and convolutional kernel sizes of $3 \times 3$ into pooling layer with strides of 1 and convolutional kernel sizes of $1 \times 1$.

In the VGG-16 convolutional network, CONV4_3 size is $38 \times 38 \times 512$, which is the largest size feature map in the SSD network and is responsible for detecting small targets using shallow semantic, as well as calculating regression parameters as the first feature layer on the detected targets to obtain the location and class information of the targets.

SSD object detection algorithm extra feature layers includes CONV7, CONV8_2, CONV9_2, CONV10_2, CONV11_2, the dimensions of extra feature layers are $19 \times 19, 10 \times 10, 5 \times 5, 3 \times 3, 1 \times 1$.

By the anchor boxes idea from Faster-RCNN [14], different numbers of prior boxes with different levels are preset on each feature map for feature extraction, and the target’s true position is calculated in the steps of classification and bounding-box regression.

In the prior boxes prediction layer, the feature map uses $3 \times 3$ convolution to output two types of detection results: category confidence and bounding box position, and each prior box match a prediction corresponding to a bounding box, according to the feature layer scale and the preset number of prior boxes per layer, it can be calculated that the SSD object detection algorithm matches a total of $38 \times 38 \times 4 + 19 \times 19 \times 6 + 10 \times 10 \times 6 + 5 \times 5 \times 6 + 3 \times 3 \times 4 + 1 \times 1 \times 4 = 8732$ bounding boxes each detection.

The SSD object detection is based on a multi-scale feature map and multi-level prior boxes jointly trained to match regression on real labeled boxes in image sets, but prior boxes
FIGURE 1. Block diagram of SSD object detection algorithm.

FIGURE 2. Ghost module structure diagram. By continuously linearly transforming the feature redundancy, the redundant feature maps of the input feature map in the initial convolutional layer are produced. To create the new ghost features, the redundant features are in contact with the identity features.

FIGURE 3. Schematic diagram of Ghost bottleneck module structure. (a) Stride = 1, Ghost BottleNeck module. (b) Stride = 2, Ghost BottleNeck module. The higher-order features of the network are abstracted by the ghost bottleneck module, which also crops the input feature map. Mimicking the ResNet, the inputs and outputs are connected to facilitate the propagation of parameters during training.

B. MODIFIED GhostNet
GhostNet is a new lightweight neural network model proposed by Huawei Noah Lab in 2020 [15], which is different from previous lightweight neural network models, such as MobileNet [16] and ShuffleNet [17], that focus on reducing the computation of convolutional layers in neural networks. The core of the GhostNet is to batch generate a large number of redundant features in the feature map generated by convolution in the convolutional layer by the linear transformation method and to stitch the redundant feature with the identity feature to form a new ghost feature, and this operation is combined as the ghost module, the related operations and concepts of the ghost module are shown in Fig. 2.

When creating the ghost bottleneck module, the residual structure in the ResNet network [18] is added to address the gradient vanishing issue by connecting the input and output, which makes it easier for parameters to propagate during training. However, too deep a neural network structure will cause the gradient to vanish or explode during the network’s training process, leaving the network untrained.

Based on the ghost bottleneck module, different ghost networks can be constructed to adapt to different deep-learning models. But the SENet attention mechanism [19] is used in the original ghost network to compress the input feature maps dimensions while obtaining the correlation paths between all input channels.

On the one hand, the SENet expression rate is not high, and obtaining the correlation path through the fully connected layer will reduce the network transmission efficiency, on the other hand, the dimensionality reduction is not conducive to the channel attention mechanism to predict the contextual semantics.

Therefore, this paper introduces the ECA attention mechanism [20] instead of the traditional SENet, removes the fully
connected layer in the SENet, and replaces it with a 1D convolution that adaptively adjusts the kernel size dynamically, shares the weights between each group of convolutions, and establishes the relationship between the feature channel dimension and the convolutional kernel size with an approximate linear mapping thus achieving a focused channel convolution information exchange capability. The specific mapping relationship of the convolution kernel size is shown in Equation 1.

$$k = \psi(C) = \left\lfloor \frac{\log_2(C)}{r} \right\rfloor + \frac{b}{r} \mod 2$$ (1)

where $C$ is the channel dimension, $b$ and $r$ are set to 1 and 2, respectively, and the odd operation means taking the nearest odd number to the operand.

The ECA attention mechanism greatly reduces the number of parameters in the convolution and improves the local multi-channel information transfer area, which effectively improves the network learning performance compared to the SENet. The structure of the ECA attention mechanism is shown in Fig. 4.

Finally, the ghost bottleneck module is used to dynamically adjust the VGG16 network structure in combination with the ECA attention mechanism. To reduce the number of parameters, replace the ordinary convolutional module in VGG16 with the ghost bottleneck module; the residual module in the ghost bottleneck network module deepens the neural network and improves the model’s detection accuracy.

### III. DUAL ATTENTION GhostNet-SSD ALGORITHM

Due to the large size of the VGG-16 network, which requires a large amount of parameter space, during the network training period, the model converges slowly, it is difficult to adjust the hyperparameters, and the deeper layers of the network tend to trigger the vanishing gradient problem, which leads to the disappearance of the gradient in the backpropagation, thus reducing the network performance and is not conducive to network’s generalization.

Meanwhile, too large weight parameters make the model unsuitable to be deployed in edge devices directly, which not only wastes device storage resources, but also consumes a large number of computational resources during the operation inference stage, causing the original function of the device to be affected, and even causing frequent crashes and reboots of embedded devices.

Therefore, in this paper, we will improve the VGG-16 backbone network layer of classical SSD object detection, build an improved ghostnet with an improved ghost module to replace the ordinary convolutional module in the VGG network, reduce the parameters of large networks and achieve sufficient accuracy with fewer operations to build an improved GhostNet-SSD network.

At the same time, the attention mechanism module is introduced to strengthen the focus of network attention and realize extracting the network features shallow semantics adaptively with obtaining more effective information.

#### A. CONVOLUTIONAL BLOCK ATTENTION MODULE

The Convolutional Block Attention Module (CBAM) integrates the channel attention mechanism with the spatial attention mechanism, and the single channel attention mechanism networks such as SENet, ECANet, etc. affine feature mapping of the input channels.

Different channel data are scaled according to the task focus, and the weight coefficients of the input feature maps among each channel are redistributed to dynamically focus on key features to achieve network performance improvement.

For the target to be detected in the classification, a spatial attention mechanism network (SANet) is combined with a single channel attention mechanism in the dimension of scale space. This mechanism concentrates on the crucial target information, mutes the irrelevant background noise in the image, gives the target a high weight, and enlarges the important data in the picture space.

In order to create the spatial attention feature map, the input feature map is processed using max-pooling and avg-pooling, the number of feature map channels is adjusted using single channel convolution after contacting, and finally, the feature map weights are obtained by normalizing with the Sigmoid function and multiplying with the original feature map. The specific spatial attention mechanism network and convolutional attention mechanism module are shown in Fig. 5 and Fig. 6.

The CBAM is simple in structure and lightweight in use, suitable for application in light-weight object detection algorithms. In this paper, the CBAM is applied in the modified-GhostNet by adding it after the large-scale feature map in the modified-GhostNet-SSD backbone layers, with almost no additional network parameters.
TABLE 1. Modified GhostNet backbone network architecture.

| Layers | Operations          | Input Size         | Output Size       | Stride | ECANet |
|--------|---------------------|--------------------|-------------------|--------|--------|
| Layer1 | Conv2D              | 300x300x3          | 150x150x32       | 2      | NO     |
| Layer2 | Ghost_Bottleneck    | 150x150x32         | 150x150x16       | 1      | NO     |
| Layer3 | Ghost_Bottleneck    | 150x150x16         | 75x75x24         | 2      | NO     |
| Layer4 | Ghost_Bottleneck    | 75x75x24           | 75x75x24         | 1      | NO     |
| Layer5 | Ghost_Bottleneck    | 75x75x24           | 38x38x40         | 2      | YES    |
| Layer6 | Ghost_Bottleneck    | 38x38x40           | 38x38x40         | 1      | YES    |
| Layer7 | Ghost_Bottleneck    | 38x38x40           | 19x19x80         | 2      | NO     |
| Layer8 | Ghost_Bottleneck    | 19x19x80           | 19x19x80         | 1      | NO     |
| Layer9 | Ghost_Bottleneck    | 19x19x80           | 19x19x80         | 1      | NO     |
| Layer10| Ghost_Bottleneck    | 19x19x80           | 19x19x80         | 1      | NO     |
| Layer11| Ghost_Bottleneck    | 19x19x80           | 19x19x112        | 1      | YES    |
| Layer12| Ghost_Bottleneck    | 19x19x112          | 19x19x112        | 1      | YES    |

Compared to the SSD algorithm, this algorithm gives additional weights to the small target information within the shallow semantics to increase the detection accuracy of the algorithm and improve the shortcomings of the SSD algorithm for small target.

B. CM-GhostNet-SSD ALGORITHM

Firstly, the ordinary convolutional layer of VGG16 is replaced by the ghost bottleneck modules layer in the SSD algorithm. Next, the ECA attention mechanism is used to replace SENet to enhance the network feature detection accuracy in the GhostNet. Finally, the CBAM attention module is introduced to the output feature map to assign the main parameter weights to the large-scale feature map on the spatial target to strengthen the shallow semantic information. The modified-GhostNet-SSD backbone network layer is shown in Table 1.

In the ghost bottleneck module, the base layer consists of ghost modules, compared to a traditional convolutional layer with \( N \) sets of convolutional kernels of size \( K \times K \) transforming a feature map of input size \( W \times H \times C \) to a feature map of \( W' \times H' \times C \) to a feature map of \( W' \times H' \times N \) dimensions, and the number of parameters \( PARA_1 \) as \( N \times C \times K \times K \).

The ghost module down scales the input feature map to \( M \), and then uses a cheap operation on the \( W' \times H' \times M \) feature map, with \( D \times D \) small kernel convolution operation, which quickly generates a large number of ghost features at the scale of \( W' \times H' \times N/M \), for a total of \( S - 1 \) groups, where \( S = N/M \).

Finally, it is contacted with the identity features and recombined into \( W' \times H' \times N \) dimensional features, so the number of parameters \( PARA_2 \) of the ghost module is the sum of the identity feature layer’s parameters and the ghost feature layer’s parameters as \( M \times C \times K \times K + (S - 1) \times M \times D \times D \), then \( PARA_1 \) compared to \( PARA_2 \) is approximated \( S \).

So the ghost module can reduce the number of ordinary convolutional parameters by a factor of \( S \). The specific multiple is related to both the number of identity feature layer channels and the number of original convolution layer channels.

Similarly, it can be deduced that the traditional convolutional layer computation \( FLOPs_1 \) is \( N \times C \times K \times K \times W' \times H' \), while the ghost module computation \( FLOPs_2 \) is \( M \times C \times K \times K \times W' \times H' + (S - 1) \times M \times D \times D \times W' \times H' \), and the ratio of them is also approximated \( S \).

Therefore, the ghost module not only effectively reduce the network weight parameters, but also accelerate the object detection network, which achieves network light weighting and improve the video detection frame rate. The parameters of each model in this paper are shown in Table 2.

The complete Modified-GhostNet-SSD combined with CBAM (CM-GhostNet-SSD) network structure is shown in Figure 7. Based on the Modified-GhostNet-SSD algorithm structure, after the input layer, building 12 layers of GhostNet-VGG backbone layers, the ghost convolutional layer Conv4_3, and the FC7 output feature maps size are 38 \times 38 \times 256 and 19 \times 19 \times 1024 respectively.

The CBAM attention model is applied to the feature maps to strengthen the target feature extraction ability, output with the same scale. In the extra feature layer in classical SSD, the CBAM mechanism is the same applied to the first feature layer Conv4_3, and the FC7 output feature maps size are 38 \times 38 \times 256 and 19 \times 19 \times 1024 respectively.

The CBAM attention model is applied to the feature maps to strengthen the target feature extraction ability, output with the same scale. In the extra feature layer in classical SSD, the CBAM mechanism is the same applied to the first feature map from the extra feature layer, with the five feature maps of 38 \times 38, 10 \times 10, 5 \times 5, 3 \times 3, and 1 \times 1 output from the other network layers, an inverted pyramid SSD was constructed to achieve multi-scale fusion target detection.

C. CM-GhostNet-SSD LOSS FUNCTION

During the training of the CM-GhostNet-SSD network model algorithm, the loss function is calculated as the minimum of the error for the object classification and border regression. Specifically, this is shown in Equation 2.

\[
L(x, c, l, g) = \frac{1}{N} \left( L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right)
\]  

In the output of both weighted sums, \( c \) is the confidence prediction for each class, \( l \) is the predicted value of the position corresponding to the prior boxes, and \( g \) is the position
coordinate of the label boxes. On the right of the equation, \( N \) represents the number of matching prior boxes, \( \alpha \) shows the weight of both the position loss function and the confidence loss function, which is usually set to 1 in the CM-GhostNet-SSD, and the confidence loss function is shown in Equations 3, 4.

\[
\begin{align*}
L_{\text{conf}}(x, c) &= -\sum_{i \in \text{Pos}} x_i^p \log \left( \hat{c}_i^p \right) - \sum_{i \in \text{Neg}} \log \left( \hat{c}_i^p \right) \\
\hat{c}_i^p &= \sum_p \exp \left( c_i^p \right)
\end{align*}
\]

The confidence loss function is composed of cross-entropy which contains the matching prediction box \( i \) and the labeled box \( j \) concerning matching correlation about the prediction probability of category \( p \). \( \{ \hat{c}_i^p \} \) means the confidence of the prediction boxes \( i \) corresponding to class \( p \). The final output value of the neural network is converted into a probability between the interval \([0,1]\) by the softmax function.

As shown in Equation 5, the location loss function reflects the correlation between predicted boxes and labeled boxes concerning the match with one category, adopting the same loss function as Faster R-CNN [21].

\[
\begin{align*}
L_{\text{loc}}(x, l, g) &= \sum_{i \in \text{Pos}} \sum_{m \in \{cx,cy,wh\}} x_i^m \text{smooth}_{L_1} \left( l_i^m - \hat{g}_j^m \right) \\
\text{smooth}_{L_1}(x) &= \begin{cases} 
0.5x^2 & \text{if } |x| < 1 \\
|x| - 0.5 & \text{otherwise}
\end{cases} \\
\hat{g}_j^m &= \frac{\left( g_j^m - d_i^m \right)}{d_i^m} \\
\hat{g}_j^w &= \log \left( \frac{g_j^w}{d_i^w} \right) \tag{8}
\end{align*}
\]

IV. EXPERIMENTAL PROCEDURE AND DATA ANALYSIS

A. PLATFORM

The experiment hardware CPU is Intel I9 10700K, GPU is NVIDIA GTX2080TI, onboard memory 16G; software is Win10 64-bit operating system, Anaconda3, deep learning frame is Python 3.6.13, Cuda10.0, Cudnn7.6.5, Tensorflow1.13.1, Keras2.1.5, Pytorch1.1.0 and NumPy, Pandas, OpenCV-Python, and other third-party packages.

B. DATASET

The data in this experiment are mainly from the web, some of them are images of VOC and COCO datasets. The data set has eight categories of common household pets, including Ragdoll cats, Shiba Inus, Sled dogs, Poodles, Labrador, Orange cats, Chinese LiHua, and American Shorthair.

About 800 images are collected for each type of sample, some images are randomly selected in the dataset for data enhancement. Random flip, random crop and random zoom operations on the randomly selected images expand the data and prevent overfitting caused by insufficient samples, on the other hand, data enhancement enlarges the target details in the image to improve the detection rate. After data enhancement, the total number of sets in this experiment is 8,032, with about 1,000 images per class. The datasets are manually labeled with the LabelImg tool and created in PASCAL VOC2007 format. Finally, the datasets are segmented with an 8:1:1 ratio script for the training set, validation set, and test set.

C. TRAINING

In model training, a stochastic gradient descent optimizer (SGD) [22] was used to train the CM-GhostNet-SSD model. The SGD optimizer randomly selects a set of samples for computing the loss function gradient values so that the learning rate updates the parameters with backpropagation.
The loss function is used to evaluate the effect of the model and represent the difference between the predicted value and the actual value. In this paper, improving the machine learning model by using the SGD optimizer. The specific value of the loss function is meaningless, we use the loss value to judge whether it converges. What is more important is the convergence trend. If the loss of the validation set does not change essentially, the model converges [23].

The SGD algorithm updates the model in real-time because of selecting a set of samples to update the model parameters each time, it also learns very fast.

In the input layer with the SSD model as the base algorithm, making the input samples resize to $300 \times 300$. When the CM-GhostNet-SSD model is training, the model prior boxes size is set to $[30, 60, 111, 162, 213, 264, 315]$, the maximum learning rate of the algorithm is 0.002, the minimum learning rate is 0.00002, the optimizer momentum size is 0.94, the weight decay factor is 0.0005, and the learning rate decay method is CosineAnnealing algorithm [24], model training Epoch is 200(The epoch refers to a complete data set has passed through the neural network once and returned once, that is, the complete training set has undergone a forward and back propagation in the neural network), training batch size is 16, the model weight is stored every 10 generations and calculated mAP. The loss function curves during training are shown in Fig. 8.

D. EVALUATION INDICATORS

Regarding this task which requires object detection to be embedded in edge devices, it is necessary to calculate the mean average precision(mAP) and Frames per second(FPS) of each model, as well as evaluate floating-point operations per second (FLOPs). The mAP is obtained by calculating the AP of M categories in the model and computing a weighted average, as shown in Equation 9.

$$mAP = \frac{1}{M} \sum_{i=1}^{M} \text{AP}_i$$  \hspace{1cm} (9)

The specific AP of a single category needs to be got by calculating the precision and recall with different confidence levels, drawing a ROC curve, and calculating the area under the curve. The precision is calculated as the ratio of true positive(TP) to the total number of true positive (TP) and false positive (FP); the recall is measured as the ratio of true positive(TP) to the total number of true positive(TP) and false negative(FN), as shown in Equation 10, 11.

$$\text{Precision} = \frac{TP}{TP + FP}$$ \hspace{1cm} (10)
$$\text{Recall} = \frac{TP}{TP + FN}$$ \hspace{1cm} (11)

FPS calculates how many images the model processes per second and is the main metric for evaluating the speed of a deep learning algorithm, higher FPS means that the model runs in real-time video stream processing with better results. FLOPs reflects the floating point running speed of the model in the processor at that time, which is positively correlated with the parameter size and generally used to measure the complexity of the algorithm. In edge devices, a model with high computational complexity will take up a huge amount of computational resources in the processor, causing the device to slow down and get stuck.

E. RESULTS ANALYSIS

To objectively evaluate the algorithm merits, in addition to the CM-GhostNet-SSD algorithm, five other control groups are set up: the Faster R-CNN algorithm [25], the classical SSD algorithm, the DSSD algorithm [26], the traditional GhostNet-SSD algorithm based on SENet and the modified-GhostNet-SSD algorithm based on ECANet.

Since the base algorithm SSD is not SOTA, to compare and analyze the CM-GhostNet-SSD algorithm in this paper, some SOTA object detection algorithms proposed in the last three years are also added, such as YOLOX(M) [27], CENTERNET [28], EFFICIENTDET(D0) [29].

All Algorithms are trained under the same deep learning frame Tensorflow and Pytorch, and the mAP of different models in test set at every 10 generations during training is plotted as shown in Fig. 9. After training is finished, the mAP, FPS, and FLOPs are calculated as listed in Table 3.

In Fig. 8, the loss function calculates the loss value decreasing gradually to keep oscillating smoothly. In Fig. 9, similar to the trend of the loss function, eventually, most models tend to have constant mAP values starting at EPOCH 100, but a small number of models start at EPOCH 120(Faster R-CNN, EFFICIENTDET). The mAP curve shows that the GhostNet-SSD model which is based on GhostNet converges faster than the classical SSD in any epoch, and achieves better convergence efficiency and higher mAP values than the recent three-year SOTA algorithms, such as EFFICIENTDET and YOLOX.

As can be seen from Table 3, the classical SSD algorithm with its mAP is 76.73%, FLOPs is high and detection performance is ordinary in motion scenes, FPS is only 27.86. The DSSD algorithm [30] with its mAP is 89.04%, the FLOPs is...
TABLE 3. The test set result of each model.

| Model               | Backbone       | Input Size | Pre-Train | GPU       | FLOPs/G | FPS/(frame/s) | mAP/% |
|---------------------|----------------|------------|-----------|-----------|---------|---------------|-------|
| Faster R-CNN        | VGGENT         | ~600 × 1000| ✓         | RTX2080TI | 226.408 | 17.81         | 79.33 |
| SSD                 | VGGNET         | 300 × 300  | ✓         | RTX2080TI | 62.798  | 27.86         | 76.73 |
| DSSD                | RESNET101      | 320 × 300  | ✓         | RTX2080TI | 32.104  | 10.75         | 89.04 |
| CENTERNET           | RESNET50       | 512 × 512  | ✓         | RTX2080TI | 69.976  | 47.59         | 84.93 |
| YOLOX               | MODIFIED CSP v5| 640 × 640  | ✓         | RTX2080TI | 73.522  | 40.14         | 66.60 |
| EFFICIENTDETON      | EFFICIENTDETON D0 | 512 × 512  | ✓         | RTX2080TI | 5.125   | 28.16         | 83.75 |
| GhostNet-SSD        | GHOSTNET       | 300 × 300  | ✓         | RTX2080TI | 1.072   | 83.37         | 90.25 |
| Modified-GhostNet-SSD| MODIFIED GHOSTNET | 300 × 300  | ✓         | RTX2080TI | 1.071   | 89.46         | 90.86 |
| CM-GhostNet-SSD     | CM-GHOSTNET    | 300 × 300  | ✓         | RTX2080TI | 1.072   | 86.17         | 91.24 |

FIGURE 9. The mAP of each different model during training. Utilizing the mAP in the vertical axis as an indicator of item detection accuracy.

32.104 GFLOPs, and FPS is only 10.75 frames per second, which is the worst performer. The mAP of the Faster R-CNN model based on the VGG network is as high as 79.33%, which is better than the classical SSD algorithm and slightly worse than the GhostNet-SSD algorithm. However, its FPS is only 17.81, which is only better than the DSSD algorithm among the nine algorithms.

The GhostNet-SSD, which is based on the Ghost modules, has an improved mAP value of 13.25%, the computational complexity is only 1.71% of the SSD, 3.33% of the DSSD, 0.46% of the Faster R-CNN, the computation speed is 3 times faster than the SSD, 7.8 times faster compared to the DSSD, and 4.7 times faster than the Faster R-CNN. Therefore, the GhostNet backbone network is comprehensively superior to the VGG backbone network in Faster R-CNN and the classical SSD algorithm. It is reasonable and efficient to replace the backbone of the SSD algorithm with GhostNet, which can make a balance between model prediction accuracy and model consumption of computation resources.

The baseline method we proposed in this paper is not SOTA. SOTA Object detection algorithms proposed in the recent three years, such as YOLOX(M), CENTERNET, and EFFICIENTDETON(D0), as shown in Figure 9 and Table 3, are all better than SSD networks, the best one is the CENTERNET algorithm [31].

YOLOX(M) has the highest FLOPs [32], even surpassing the SSD, while mAP does not perform well enough, scoring only 66.60%, and EFFICIENTDETON(D0) [33] has the worst FLOPs among the three, with only 1/3 of the FLOPs of the GhostNet-SSD. The CENTERNET algorithm, which has the best overall performance, also performs poorly compared to the GhostNet-SSD, with an mAP value 5.32% lower than GhostNet-SSD, and a running speed is only 57.08%.

Although all three algorithms are better than the classical SSD algorithm, as the SOTA object detection algorithm proposed in the recent three years, the model structure uses some newly defined operators in the TensorFlow framework, such as the Lambda operator and logic operator where used in the CENTERNET network, which has not yet supported in some neural network frameworks at the embedded end or have limitations in their use [34], which will lead to the failure of the quantization conversion of the model at the edge device and will be difficult in the practical application at the embedded side.

Taking into account the loss of quantization accuracy [35], the performance of the SOTA algorithm will be lower than the data in Table 3. In contrast, the classical SSD was proposed earlier, the model structure is representative, the related embedded application studies are based on SSD [36], and the technical support is more mature at the embedded device [37]. From the theoretical significance of the research, as shown in Table 3, the accuracy and speed performance of relevant SOTA object detection algorithm in the field of pet detection still lags behind that of the improved GhostNet-SSD algorithm proposed in this paper.

As for the improvement of GhostNet, compared with the traditional GhostNet-SSD algorithm based on the SENet, it can be reflected from the modified-GhostNet-SSD algorithm that the calculated amount of the Modified-GhostNet-SSD is slightly reduced, while the model prediction speed and accuracy are higher by 0.36% and 0.61%.

This demonstrates that the ECANNet module has a certain improvement effect on the tuning accuracy and computation
TABLE 4. The AP test results for specific single-category pets.

| Method              | mAP/% | Ragdoll cats | Shiba Inus | Orange cats | Chinese LiHua | Sled dogs | Labrador | Poodles | American Shorthair |
|---------------------|-------|--------------|------------|-------------|----------------|-----------|----------|---------|-------------------|
| SSD                 | 76.73 | 87.98        | 80.01      | 62.52       | 77.46          | 78.28     | 72.53    | 81.53   | 73.53             |
| DSSD                | 89.04 | 89.99        | 88.78      | 89.65       | 87.33          | 90.75     | 88.32    | 88.50   | 88.98             |
| CENTERNET           | 84.93 | 78.05        | 88.10      | 83.88       | 78.88          | 90.67     | 86.43    | 91.50   | 81.91             |
| Faster R-CNN        | 79.33 | 75.77        | 82.73      | 79.18       | 74.57          | 86.84     | 81.31    | 85.65   | 68.57             |
| YOLOX               | 66.60 | 56.66        | 70.76      | 65.53       | 65.76          | 72.10     | 65.47    | 76.89   | 59.64             |
| EFFICIENTDET        | 83.75 | 79.30        | 87.84      | 79.73       | 75.06          | 87.70     | 84.33    | 95.58   | 80.43             |
| GhostNet-SSD        | 90.25 | 95.19        | 91.19      | 89.44       | 88.33          | 91.11     | 88.53    | 92.08   | 86.11             |
| Modified GhostNet-SSD | 90.86 | 95.31        | 93.84      | 91.89       | 88.18          | 91.91     | 89.02    | 92.14   | 84.57             |
| CM-GhostNet-SSD     | 91.24 | 96.15        | 89.31      | 92.71       | 89.26          | 92.13     | 88.08    | 92.90   | 89.39             |

The CM-GhostNet-SSD algorithm, based on the Modified-GhostNet-SSD, which is introduced to the CBAM attention module in the low feature layer, has no significant improvement in FLOPs compared with the Modified-GhostNet-SSD, while the FPS in video detection only decreases by 0.18%, but the most important mAP value improves by 0.38%. In contrast to the traditional GhostNet-SSD, the mAP of CM-GhostNet-SSD is improved by nearly 1%.

The CM-GhostNet-SSD algorithm achieves a good balance between model inference resource consumption and prediction accuracy and is suitable for deployment in low-cost, weak-performance embedded edge devices to discriminative environmental targets. The mAP of specific single-category pet dogs and cats are shown in Table 4 and Fig. 10.

From Table 4 and Figure 10, it can be seen that the classical detection methods DSSD and Faster R-CNN both outperform the SSD algorithm in single-target, multi-target, and small-target test set image detection accuracy, but DSSD and Faster R-CNN are more complex in network structure [38], as shown in Table 3, the FPS is not as fast as the SSD algorithm, so it is not suitable for video stream detection tasks.

The object detection algorithms proposed in recent years, such as YOLOX, EFFICIENTDET, and CENTERNET, are deficient in detection accuracy compared with traditional detection methods. For example, CENTERNET has missed detection on the test set and has a 20% to 40% lower confidence rate on the small targets and single targets. Compared with the GhostNet-SSD algorithm, the classical method and the algorithm in the last three years are both low in accuracy. In terms of accuracy on a single category, GhostNet-SSD leads all detection algorithms on most categories, so GhostNet-SSD works better on streaming tasks compared to classical detection methods and is slightly stronger than the new detection algorithms in accuracy.

Comprehensively, the SSD algorithm based on the GhostNet backbone and its improved derived algorithms outperform better than traditional SSD in single-target, multi-target, and small-target, there is a 10% to 20% gap in the detection effectiveness of the SSD networks in contrast to the GhostNet network in all different categories. Also in GhostNet, the location of the target to be detected is better and the confidence is greater than SSD, for a single object, GhostNet accurately identifies the object and plots the exact location of the object, for multi-target, it can detect the location and category of each object without false and missing.

The CM-GhostNet-SSD algorithm is the most accurate in determining the location information of the target under multi-category detection, and the bounding boxes are more complete to include the object, the confidence about the object is also higher. In the detection of small targets, the SSD barely recognizes the smaller targets in the test set, while the CM-GhostNet has a perfect and clear description. Therefore, it is feasible and reliable to replace the traditional SSD network with the CM-GhostNet-SSD network, which is more accurate and efficient.

F. MODEL DEPLOYMENT

In research, the CM-GhostNet-SSD algorithm model will be deployed on Rockchip’s RV1126 platform. The RV1126 platform is built with a 2.0 Tera Operations Per Second(Tops), Neural-Network Processing Units(NPU) that supports INT8/INT16 quantization types, which is dedicated to convolutional operations compared to conventional CPUs. As a result, the deep learning model runs efficiently on the RV1126 platform, easily handling multimedia data under high flow and providing inference detection.

The RV1126 platform NPU is designed on the RKNN framework. When used, it is necessary to convert the self-developed algorithms on Tensorflow, Pytorch, Keras and other frameworks into a usable RKNN model. RKNN framework does not yet support all operators in the major frameworks, for Keras framework, RKNN provides support for Dense, Flatten, Reshape, BatchNormalization, Conv2D, RNN and other major operators, and in the design of the network, the size of the convolutional kernel size is preferably $3 \times 3$ size, to reduce the additional computational consumption of NPU, and achieve the maximum utilization of the...
chip platform. Therefore, for the CM-GhostNet-SSD model, fine-tuning the network structure and convolutional kernels is required to meet the quantitative deployment of the model the best.

In this paper, we design a CM-GhostNet-SSD network with the Keras framework. RKNN platform supports the PB model of Tensorflow, which is the most widely used in the industry [39], with a higher conversion success rate, so it is first transformed into PB model using H5 model and then into the RKNN model. Converting the PB model into the RKNN model, asymmetric quantization mode is used to set the input model size as $300 \times 300 \times 3$. The conv1/convolution node is selected as the input and the mbox_conf/concat and mbox_loc/concat nodes are selected as the output in the PB model node to further crop the model and reduce the model size for device deployment. The inference results after the deployment of the RKNN model are shown in Fig. 11.

In Fig. 11, the quantitative model obtained almost the same results as the original CM-GhostNet-SSD network on the single-target and small-target test set, but in the multi-target test, it failed to identify each of the detection targets in the set, but still detected the targets correctly. This is primarily when the model in the quantitative conversion, the original parameters are converted from floating into fixed, and the parameters’ accuracy will inevitably be lost, resulting in quantization error.

In multi-layer neural networks, the quantization error in the lower layers will be further amplified gradually with the quantization to degrade the model. To address this phenomenon, the detection effect can be improved by adding multiple images in the network quantization processing to
continuously adjust the quantization parameters during the model transformation, or by using the model itself parameters to transform the quantization parameters and adapt the original network model to the RKNN framework.

Comprehensively, this paper proposes an attention-based CM-GhostNet SSD algorithm, which is better than the classical SSD object detection algorithm in all aspects, successfully deployed in the embedded edge device, and completes the task of pet cat and dog. Further, it can combine a large number of peripheral sensors with self-researched algorithms to achieve automatic feeding functions, dangerous behavior analysis, and regular uploading of home pet activity videos, etc., to create a wise big data platform that is responsible for home pet life.

V. CONCLUSION

Due to the disadvantage that classical SSD object detection cannot be equipped in embedded edge devices, this paper innovatively proposes an attention-based CM-GhostNet backbone model, which is combined with the SSD network framework to realize the lightweight SSD model and successfully deployed in embedded devices. In contrast to the classical SSD network, the CM-GhostNet-SSD network attains a much better mAP and FPS than the classical SSD on small-sized objects, demonstrating the superiority of the GhostNet backbone and the effectiveness of the attention mechanism. Finally, the successful deployment of the quantitative model on the RV1126 platform evaluates the test set images with a certain degree of accuracy. As a result, the algorithm discussed in this paper has significant application value for the task of object identification in pet dogs and cats, and when combined with embedded and other technological elements, it can be used broadly in the field of home pets.

REFERENCES

[1] S. P. Huang and Q. Q. Li, “A study on life and mentality of the youth living solo in metropolises,” Youth Explor., vol. 4, pp. 75–87, Apr. 2018.
[2] F. X. Yang and Q. A. Li, “Young urban white-collar pet consumption,” Youth Stud., vol. 1, pp. 73–83&96, Jan. 2022.
[3] N. Han, “A first look at the current state of pet economy development,” China Manag. Informationization, vol. 11, pp. 180–181, Jun. 2021.
[4] Y. X. Yang, Y. N. Chen, Y. E. Zhu, S. Q. Kang, and D. Zhao, “Analysis of the causes, trends and pros and cons of the rapid development of the pet economy,” Rural Economy Sci.-Technol., vol. 5, pp. 80–81, May 2020.
[5] F. Y. Wang and Y. Zhu, “Market assessment analysis of pet feeding system,” Ind. Sci. Tribune, vol. 14, pp. 110–111, Jul. 2020.
[6] J. Y. Zhou, Z. Y. Zhang, and Z. Q. Yuan, “Design and application of intelligent feed management control system for household pet,” J. Domestic Animal Econom., vol. 4, pp. 77–80, Apr. 2019.
[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.
[8] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.
[9] K. Wang and W. Zhou, “Pedestrian and cyclist detection based on deep neural network fast R-CNN,” Int. J. Adv. Robotic Syst., vol. 16, no. 2, Mar. 2019, Art. no. 172988141982965.
[10] 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.
[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.
[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis., Oct. 2016, pp. 21–37.
[13] C. Wang, B. Li, and B. Jiao, “Fault diagnosis of rolling bearing based on convolutional neural network of convolutional block attention module,” J. Phys., Conf., vol. 1732, no. 1, Jan. 2021, Art. no. 012045.
[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[15] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More features from cheap operations,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1577–1586.
[16] K. D. Kadam, S. Ahrirao, K. Kotecha, and S. Sahu, “Detection and localization of multiple image splicing using MobileNet V1,” IEEE Access, vol. 9, pp. 162499–162519, 2021.
[17] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6848–6856.
[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[19] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141.
[20] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “Supplementary material for ECA-Net: Efficient channel attention for deep convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 13–19.
[21] G. Xuan, C. Gao, Y. Shao, M. Zhang, Y. Wang, J. Zhong, Q. Li, and H. Peng, “Apple detection in natural environment using deep learning algorithms,” IEEE Access, vol. 8, pp. 216772–216783, 2020.
[22] K. L. Chung, “On a stochastic approximation method,” Ann. Math. Statist., vol. 25, pp. 463–483, Sep. 1954.
[23] K. Janocha and W. M. Czarnecki, “On loss functions for deep neural networks in classification,” 2017, arXiv:1702.05659.
[24] P. Mishra and K. Sarawadekar, “Polynomial learning rate policy with warm restart for deep neural network,” in Proc. IEEE Region Conf. (TENCON), Oct. 2019, pp. 2087–2092.
[25] X. Gao, F. Yang, T. Chen, and J. Si, “Chinese character components segmentation method based on faster RCNN,” IEEE Access, vol. 10, pp. 98095–98103, 2022.
[26] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “SSD: Deconvolutional single shot detector,” 2017, arXiv:1701.06659.
[27] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.
[28] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” 2019, arXiv:1904.07850.
[29] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. ICML, Jun. 2019, pp. 6105–6114.
[30] Y. Yang, J. Yu, and T. Kurihara, “Immature Yuzu citrus detection based on DSSD network with image tiling approach,” in Proc. 66th Annu. Conf. IEEE Ind. Electron. Soc. (IEECON), Oct. 2021, pp. 1128–1133.
[31] Y. Zhou, Y. Bai, and Y. Chen, “Multiframe CenterNet heatmap ROI aggregation for real-time video object detection,” IEEE Access, vol. 10, pp. 54870–54877, 2022.
[32] Y. Zhao, M. Xiao, H. Lv, J. Luo, X. Wang, and D. Luo, “Research on scanning acoustic image defects detection of integrated circuits based on YOLOX,” in Proc. 23rd Int. Conf. Electron. Packag. Technol. (ICEPT), Aug. 2022, pp. 1–4.
[33] F. Sajid, A. R. Javed, A. Basharat, N. Kryvinska, A. Afzal, and M. Rizwan, “An efficient deep learning framework for distracted driver detection,” *IEEE Access*, vol. 9, pp. 169270–169280, 2021.

[34] S. P. Baller, A. Jindal, M. Chadha, and M. Gerndt, “DeepEdgeBench: Benchmarking deep neural networks on edge devices,” in *Proc. IEEE Int. Conf. Cloud Eng. (IC3E)*, Oct. 2021, pp. 20–30.

[35] S. Yang, Z. Niu, J. Cheng, S. Feng, and P. Li, “Face recognition speed optimization method for embedded environment,” in *Proc. Int. Workshop Electron. Commun. Artif. Intell. (IWECIAI)*, Jun. 2020, pp. 147–153.

[36] A. Wong, M. J. Shafooe, F. Li, and B. Chwyl, “Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection,” in *Proc. 15th Conf. Comput. Robot Vis. (CRV)*, May 2018, pp. 95–101.

[37] A. Heredia and G. Barros-Gavilanes, “Video processing inside embedded devices using SSD-mobilenet to count mobility actors,” in *Proc. IEEE Colombian Conf. Appl. Comput. Intel. (ColCACI)*, Jun. 2019, pp. 1–6.

[38] L. Aziz, M. S. H. Salam, U. U. Sheikh, and S. Ayub, “Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review,” *IEEE Access*, vol. 8, pp. 170461–170495, 2020.

[39] A. Jain, A. A. Awan, H. Subramoni, and D. K. Panda, “Scaling TensorFlow, PyTorch, and MXNet using MVAPICH2 for high-performance deep learning on frontera,” in *Proc. IEEE/ACM 3rd Workshop Deep Learn. Supercomput. (DLS)*, Nov. 2019, pp. 76–83.

**XUQING KE** was born in Huangshan, Anhui, China. He received the B.S. degree from the Zhejiang University of Technology, Hangzhou, China, in 2020, where he is currently pursuing the M.S. degree. His research interests include the IoT and multimedia, and embedded technology.

**WEI HOU** received the B.S. degree from the Communication University of Zhejiang, Hangzhou, China, in 2020. He is currently pursuing the M.S. degree with the Zhejiang University of Technology, Hangzhou. His research interests include the Internet of Things and multimedia communication.

**LIMING MENG** received the Ph.D. degree in communication and electronic systems from Zhejiang University, Zhejiang, China.

She is currently a Professor with the School of Information Engineering, Zhejiang University of Technology. Her research interests include wireless communication and networks, streaming media transmission, and the IoT communications. She is a member of the China Computer Federation (CCF).

---

**XUQING KE** was born in Huangshan, Anhui, China. He received the B.S. degree from the Zhejiang University of Technology, Hangzhou, China, in 2020, where he is currently pursuing the M.S. degree. His research interests include the IoT and multimedia, and embedded technology.

**WEI HOU** received the B.S. degree from the Communication University of Zhejiang, Hangzhou, China, in 2020. He is currently pursuing the M.S. degree with the Zhejiang University of Technology, Hangzhou. His research interests include the Internet of Things and multimedia communication.

**LIMING MENG** received the Ph.D. degree in communication and electronic systems from Zhejiang University, Zhejiang, China.

She is currently a Professor with the School of Information Engineering, Zhejiang University of Technology. Her research interests include wireless communication and networks, streaming media transmission, and the IoT communications. She is a member of the China Computer Federation (CCF).