Fast object detection based on binary deep convolution neural networks

Siyang Sun\textsuperscript{1,2}, Yingjie Yin\textsuperscript{1,2}, Xingang Wang\textsuperscript{1,2} \textsuperscript{✉}, De Xu\textsuperscript{1,2}, Wenqi Wu\textsuperscript{1,2}, Qingyi Gu\textsuperscript{1,2}

\textsuperscript{1}Research Centre of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Haidian District, Beijing, People’s Republic of China
\textsuperscript{2}University of Chinese Academy of Science, 19, Yuquan Road, Shijingshan District, Beijing, People’s Republic of China
E-mail: xingang.wang@ia.ac.cn

Abstract: In this study, a fast object detection algorithm based on binary deep convolution neural networks (CNNs) is proposed. Convolution kernels of different sizes are used to predict classes and bounding boxes of multi-scale objects directly in the last feature map of a deep CNN. In this way, rapid object detection with acceptable precision loss is achieved. In addition, binary quantisation for weight values and input data of each layer is used to squeeze the networks for faster object detection. Compared to full-precision convolution, the proposed binary deep CNNs for object detection results in 62 times faster convolutional operations and 32 times memory saving in theory, which’s more, the proposed method is easy to be implemented in embedded computing systems because of the binary operation for convolution and low memory requirement. Experimental results on Pascal VOC2007 validate the effectiveness of the authors’ proposed method.

1 Introduction

Visual object detection is one of the important problems in computer vision, and it is widely used in vision-based tasks such as video surveillance \cite{1, 2}, visual navigation \cite{3, 4}, human–computer interaction \cite{5–8}. Shallow learning based on handcrafts features \cite{9–15} and deep learning \cite{16–22} are two main ways for visual object detection. Many vision-based tasks implemented on embedded computing systems adopt handcrafts features and shallow learning to achieve object detection and obtain real-time performance and low energy consumption.

Kyrkou et al. \cite{9} proposed a dedicated hardware architecture that provides an efficient platform for generic real-time object detection. The depth and edge processing mechanisms and support vector machine classification are implemented on a field-programmable gate array (FPGA) platform, which results in significant speed-ups and improved detection accuracy. Jiang et al. \cite{11} introduced a high-speed all-hardware scale-invariant feature transform (SIFT) architecture with parallel and pipeline technology for real-time extraction of image features for object detection. The overall time of the proposed system to extract SIFT features for an image having 512×512 pixels is only 6.55 ms, which was sufficient for real-time applications. Farrugia et al. \cite{12} presented a parallel architecture for fast and robust face detection implemented on FPGA hardware. The system used the convolutional face finder (CFF) algorithm, which achieved both real-time requirements in an embedded context and face detection robustness within complex backgrounds. Ma et al. \cite{13} proposed a full-image evaluation methodology to explore the FPGA implementation of HOG using reduced bit width. The improved HOG algorithm of image detection was implemented on FPGA, which can reach real-time processing with no loss of accuracy. Hiromoto et al. \cite{15} introduced a kind of hardware architecture for object detection based on an AdaBoost learning algorithm with Haar-like features as weak classifiers. The system was implemented on a Virtex-5 FPGA, which achieves real-time object detection at 30 fps on VGA video and improves the total processing speed. Most vision algorithms implemented on embedded computing systems adopted handcrafts features and shallow learning methods which have low accuracy in complex scenes.

Recently, the method based on deep convolution neural networks (CNNs) is widely and successfully used in object detection \cite{16–22} and achieves high accuracy, compared to the methods based on handcrafts features and shallow learning.

Most of current state-of-the-art object detection methods are based on region CNNs, which usually adopt separate steps such as proposals prediction \cite{23, 24}, resampling pixels or features for each box, applying a high-quality classifier \cite{25–28} and so on to achieve object detection. Girshick et al. \cite{16, 17} proposed a region-based CNN (R-CNN) for object detection. R-CNN method extracts around 2000 region proposals and computes features for each proposal using a deep CNN and then classifies each proposal using class-specific linear support vector machines. He et al. \cite{19} proposed spatial pyramid pooling (SPP) on the basis of R-CNN, and SPP can use a deep CNN to extract the fixed length feature of an image whose size does not need to be constant. The SPP network (SPPNet) is 24× to 102× faster than the R-CNN for the target detection task. Girshick \cite{18} proposed a fast R-CNN network (Fast R-CNN) based on the SPPNet and R-CNN, which improves the speed of object detection. Ren et al. \cite{20} proposed a faster R-CNN detection method which introduces a region proposal network (RPN) that shares full-image convolutional features with the detection network and further improves the accuracy and speed of object detection.

The separation of proposals prediction and classification limits the speed of R-CNN-based detection methods. Recently, some unified networks are proposed for object detection. Redmon et al. \cite{21} proposed a unified network named YOLO, which uses one deep CNN to predict bounding boxes and class probabilities directly from input images. Liu et al. \cite{22} proposed a single shot detector (SSD) for multiple categories which uses small convolutional filters applied to feature map of different scales to predict category scores and box offsets for a fixed set of default bounding boxes. Though unified-network-based detection methods improve the detection speed, large amounts of memory and computational power are still needed since deep full-precision CNNs are used.

Several famous full-precision CNNs are usually used in CNN-based object detection methods. Krizhevsky et al. \cite{29} proposed a deep CNN called AlexNet for classification and
achieved better results compared to the previous state-of-the-art methods.

AlexNet has <60 million parameters (249 MB of memory) and performs <1.5 billion high precision operations to classify one image. Simonyan and Zisserman [30] proposed VGGNet which increases the depth of CNN to 16–19 weight layers using the architecture with very small \((3 \times 3)\) convolution filters. VGGNet further improves classification precision, but it has more 138 million parameters (573 MB of memory). He et al. [31] proposed a residual learning framework called ResNet, which explicitly reformulates the layers as learning residual functions with reference to the layer inputs. ResNet ensures that deeper CNNs including more than one hundred layers can be trained effectively and achieve higher precision.

However, deeper CNNs need more memory and computational power. These full-precision CNNs quickly overtax the limited storage \([\text{memory}]\). These full-precision CNNs can be quantised as binary numbers \([\text{binary number}]\) and achieve higher precision. In order to reduce the time cost and space complexity of the network effectively, the parameters and the input data of each layer of full-precision CNNs can be quantised as binary numbers accordingly.

### 2 Principle of binary deep CNNs

In order to reduce the time cost and space complexity of the network effectively, the parameters and the input data of each layer of full-precision CNNs can be quantised as binary numbers accordingly.

#### 2.1 Binary weight network

A binary filter \(B \in \{+1, -1\}^{c \times w \times h}\) and a scaling factor \(\alpha \in R^+\) are used to approximate real-value weights \(W \simeq aB\). Thus, the convolution operation of the network can be approximated as

\[
I * W \simeq (I \oplus B)\alpha
\]

where \(\oplus\) indicates a convolution without any multiplication. Only the relevant addition and subtraction operations are needed to complete the convolution operation using (1).

In order to solve the approximation problem of \(W \simeq \alpha B\), we use the minimisation of the mean square error to solve the optimal solutions \(a^\alpha, B^\alpha\) as shown in the following equation:

\[
\begin{align*}
a^\alpha, B^\alpha &= \arg\min_{a, B} J(B, a) \\
&= \alpha^\alpha B^\alpha B - 2aW^T B + W^T W
\end{align*}
\]

where

\[
J(B, a) = \| W - \alpha B \|^2
\]

By expanding (3), we have

\[
J(B, a) = \alpha^2 B^2 B - 2aW^T B + W^T W
\]

Since \(B \in \{+1, -1\}^{c \times w \times h}\), \(B^2 B\) is a constant. Because \(W\) represents the network weights, \(W^T W\) is also a constant. The optimal problem can be solved by maximising the following constrained optimisation:

\[
B^* = \arg\max (W^T B)
\]

The analytical method as shown in [33] is used to solve the optimal problem (3).

Through the processing of optimisation, the estimation of a binary weight filter can be express as

\[
B^* = \text{sign}(W)
\]

The optimal factor \(\alpha^*\) can be expressed as

\[
\alpha^* = \frac{1}{n} \| W \|_1
\]

where \(\| \cdot \|_1\) is 1-norm.

Thus, the optimal estimation of a binary weight filter can be simply achieved by taking the sign of weight value and the optimal scaling factor is the average of absolute weight values.

#### 2.2 Binary weight and binary input data network (BWBBDN)

According to a principle of binary weight network (BWN), we introduce BWBBDN [33], whose weights and input data of each layer are both quantised as binary numbers.

![Processing of the unified network for object detection](image)
We can use (8) to approximate dot product between the input data \( X \in \mathbb{R}^p \) and the weights \( W \in \mathbb{R}^q \).

\[
X^T W \approx \beta H^T a B
\]  
(8)

where \( H, B \in \{+1, -1\}^n \) and \( \alpha, \beta \in \mathbb{R}^p \). The minimisation of the mean square error is also used to solve the optimal solutions \( \alpha^*, \beta^*, H^* \) as shown in the following equation:

\[
\alpha^*, \beta^*, H^* = \arg\min_{\alpha, \beta, H} \|X^T W - \beta a H^T B\|
\]  
(9)

Let \( Y = XW, C = HB \) and \( y = \alpha \beta \), where \( Y \in \mathbb{R}^n, C \in \{+1, -1\}^n \) and \( y \in \mathbb{R}^p \). From (9) can be written as

\[
y^*, C^* = \arg\min_{\gamma C} \| I^T Y - \gamma I^T C \|
\]  
(10)

where \( I \) is an \( n \)-dimensional vector where all of its entries are 1.

The analytical method as shown in [33] is used to solve the optimal problem (10), and the analytic solution \( \gamma^*, C^* \) can be expressed as

\[
\gamma^* = \sum_{n} \frac{|\gamma_i|}{n} = \frac{\sum_{n} |\gamma_i||W_i|}{n} \approx \left( \frac{1}{n} \|X\|_1 \right) \left( \frac{1}{n} \|W\|_1 \right)
\]

\[
C^* = \text{sign}(Y) = \text{sign}(X^T) \text{sign}(W) = H^T B^*
\]

(11)

3 Object detection based on binary CNNs

We unify the processes of proposals prediction and object classification and propose a fast unified network for object detection. Unlike Faster R-CNN [20], our fast unified detection network does not need to extract proposals.

3.1 Architecture of unified detection networks

VGG-16 [30] is used as the base CNN in our proposed unified detection network, as shown in Fig. 2. In the last convolution layer Conv5_3 of VGG-16, another seven convolution layers (Conv6_1, Conv6_2, Conv6_3, Conv7_1, Conv7_2, Conv7_3, Conv8) are added. After the last layer, convolution kernels of different sizes are used to get the layer Conv9 which includes predicted bounding boxes and classes. Conv9 includes a set feature map of different sizes and each feature map consists of 25 channels which include 21 class channels (20 object classes and 1 background class) and four position channels. The different sizes of convolution kernels are set from \( 1 \times 1 \) to \( kw \times kh \), so there are \( kw \times kh \) convolution kernels of different sizes in Conv9.

3.2 Training of our proposed unified detection network

When one image is input into the network, we will get feature maps, which are used to predict multiple bounding boxes and corresponding classes due to the different sizes of convolution kernels following Conv8 layer. Every position in feature maps of Conv9 is corresponding to a default box in the original input image. The corresponding process is as follows.

Let the width and height of an input image be \( w_a \) and \( h_a \), respectively, and let the width and height of the feature maps in Conv8 layer be \( w_i \) and \( h_i \) respectively. The scale factors \( w_i \) and \( h_i \) between conv8 and the input image can be calculated using the following equation:

\[
W_i = \frac{w_a}{w_i}, \quad H_i = \frac{h_a}{h_i}
\]

(12)

Since the size of convolution kernels in Conv9 is \( k_i \times k_i \), \( i = 1, 2, 3, \ldots, w \) and \( j = 1, 2, 3, \ldots, h \), the size of each feature map in Conv9 can be calculated as

\[
\begin{cases}
H_i = H_i - (k_i - 1), & j = 1, 2, 3, \ldots, h \\
W_i = W_i - (k_i - 1), & i = 1, 2, 3, \ldots, w
\end{cases}
\]

(13)

There are \( \sum_{i=1}^{w} \sum_{j=1}^{h} \) \( W_i \times H_i \) default boxes whose feature maps of Conv9 are corresponding to the original input image and coordinates of each default box are calculated using the following equation:

\[
A_{x_{\min}} = (m - 1) \times W_i, \quad m = 1, 2, 3, \ldots, W_i, \quad i = 1, 2, 3, \ldots, w \\
A_{x_{\max}} = (m + k_i - 1) \times W_i, \quad m = 1, 2, 3, \ldots, W_i, \quad i = 1, 2, 3, \ldots, w \\
A_{y_{\min}} = (n - 1) \times H_i, \quad n = 1, 2, 3, \ldots, H_i, \quad j = 1, 2, 3, \ldots, h \\
A_{y_{\max}} = (n + k_i - 1) \times H_i, \quad n = 1, 2, 3, \ldots, H_i, \quad j = 1, 2, 3, \ldots, h
\]

(14)

The key difference between training our fast unified detection network and training a typical detector with region proposals is that ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Let \( x_{ij} \in \{0, 1\} \) be an indicator for matching the \( ij \)th default box to the \( ij \)th ground truth box of category \( p \), and we can have \( \sum_{ij} x_{ij} \geq 1 \).

Because different sizes of convolution kernels following the last convolution layers are used to predict class probabilities and positional regression simultaneously, the overall objective loss function is defined as follows:

\[
L(x, c, l, g) = \sum_{ij} \left( L_{\text{conf}}(x_{ij}, c) + \alpha L_{\text{loc}}(x_{ij}, l, g) \right)
\]

(15)

where \( k_w \times k_h \) is the number of convolution kernels.
In the testing phase, each default bounding box is classified as positive or negative based on the proportion of positive and negative samples selected at 1:3. The negative samples are sorted using the highest confidence scores, and the top ones are picked. The hard negative mining method is adopted to select negative samples if their values of intersection over union with all ground truth are greater than or equal to 0.5, and they are dropped. The localisation loss is the mean squared error between the predicted box \( \hat{x}_c \) and the ground truth box \( x_c \):

\[
L_{\text{loc}}(x, y, w, h) = \sum_{i=0}^{N_c} \sum_{j=1}^{N_x} \alpha_{ij} \left( \frac{1}{w_i} \left( x_{ij} - \hat{x}_{ij} \right)^2 + \frac{1}{h_i} \left( y_{ij} - \hat{y}_{ij} \right)^2 + \frac{1}{w_i} \left( w_{ij} - \hat{w}_{ij} \right)^2 + \frac{1}{h_i} \left( h_{ij} - \hat{h}_{ij} \right)^2 \right)
\]

where \( \alpha_{ij} \) is a variable for controlling the number of negative samples.

In the training phase, these default bounding boxes are divided into positive samples or negative samples by calculating the intersection between the candidate boxes and the ground truth. Positive samples are those whose intersection with any ground truth box is greater than or equal to 0.5, and they are kept. The others are discarded.

In practice, most of the bounding boxes are negative samples, so the hard negative mining method is adopted to select negative samples. Negative samples are sorted using the highest confidence loss in descending order and the top ones are picked. The proportion of positive and negative samples selected is kept at 1:3. The others are discarded.

In the testing phase, each default bounding box is classified as different categories, including objects and non-object, and the position of every object default bounding box is regressed to a rectified position by our proposed network. The processing of bounding boxes is shown in Fig. 3. The default bounding boxes of an input image are shown in Fig. 3a, bounding boxes output by our network are shown in Fig. 3b. Fig. 3c shows the final object bounding boxes by using non-maximum suppression step.

### 3.3 Binary unified detection network

As shown in Section 2, binary operations for CNNs can effectively relieve the limitation of large amounts of memory and computational power. The convolution layers in the proposed unified detection network, as shown in Fig. 2, are converted into binary convolution layers, which are represented as BinConv. Similar to [33], the implementation of BinConv is as follows in order: batch normalisation, binary activation layer, binary convolution, and pooling operations. The architecture of binary unified detection network is shown in Fig. 4.

Another seven convolution layers (Conv6_1, Conv6_2, Conv6_3, Conv7_1, Conv7_2, Conv7_3, Conv8) are binarised. Because VGG-16 [30] is used as the base CNNs in our proposed unified detection network, and in order to save training time, it is not necessary to train the base network. In addition, Conv9 cannot be binarised because we predict classes and bounding boxes of multi-scale objects directly in the Conv9 feature map of a deep CNN. The probability of each class and the positions of bounding boxes in the Conv9 are real float parameters.

The similar strategies in [33] are used to train our proposed binary unified detection network. Weights are binarised during the forward pass and back propagation. In the parameter update process, the full-precision weights are still used. The parameters are updated by using the SGD update with momentum or ADAM [35].

### 3.4 Complexity analysis

The proposed network is analysed in terms of both data storage space and speed, which is shown in Sections 3.4.1 and 3.4.2.

(i) **Memory:** In terms of memory, because the weight parameters and the input data are binary, 32x the memory storages are saved compared to the real weight parameters and the input data. (ii) **Speed:** In terms of speed, the total number of operations is \( cN_wN_f \) in the standard convolution, where \( c \) is the number of channels, \( N_w \) is the size of convolution kernels and \( N_f \) is the size of input data. According to XNOR-Net [33], our proposed binary unified detection network obtains equivalent speed improvements. The speedup can be computed by

\[
S = \frac{cN_wN_f}{(1/64)cN_wN_f + N_f} = \frac{64cN_w}{cN_w + 64}
\]

### Fig. 3 Processing of bounding boxes

- **a** Default bounding boxes
- **b** Bounding boxes output by our unified network
- **c** Object bounding boxes

### Fig. 4 Binary network detection
According to (18), the proposed binary deep CNN for object detection results in about $62 \times$ faster convolutional operations compared to the full-precision convolution in theory in the case that $c = 512$, $N_W = 3$.

4 Experimental results

The object detection was performed on the PASCAL VOC2007 dataset [36] and the proposed drogue dataset.

PASCAL VOC2007 dataset has 10,022 train images and 4952 test images and includes 20 categories. On this dataset, we compared our non-binary and BWN and BWBDN using our method. We also compared our method with Fast R-CNN [18] and Faster R-CNN [20].

The proposed drogue dataset including 24,654 images is used to train and evaluate our proposed network. The resolution of images is $1280 \times 960$ pixels. All the images in the data set are collected in the ground simulation environment and the drogue is made according to the real 3D model of drogue for autonomous aerial refuelling, the experiment platform is shown in Fig. 5. The ground-truth bounding boxes for images are obtained by manual. There are pose variety (in the plane or out of plane), illumination changes, scale changes, occlusion, and cluttered background in the drogue dataset.

In every iteration of training, images were resized to 256 pixels at their smaller dimension without changing their aspect ratios. Because our network had not full connection layers, which is a full convolution network, it is not necessary to fix the size of the image, which is different with Fast RCNN [18] and Faster RCNN [20]. There is no deformation of the object. The training algorithm was run for 50 epochs or more. In each epoch, 1000 images were randomly selected for training. ADAM [35] was run for 50 epochs or more. In each epoch, 1000 images were stochastically selected for training. ADAM [35] and based on Alexnet achieve 89.5 mAP, which is the main evaluation standard on object detection.

Table 1 evaluates the detection results of our non-binary network based on VGG and based on Alexnet, respectively. We primarily evaluate detection mean Average Precision (mAP), which is the main evaluation standard on object detection. We compared with Faster R-CNN [20], the accuracy of our non-binary network based on VGG-16 had only 1% loss, but the speed of our non-binary network based on VGG-16 is faster 10 times than Faster R-CNN [20], which is shown in Table 1. Our method has a frame rate of 20 fps (frames per second). Our unified network and Faster R-CNN [20] were tested on Titan X GPU with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40 GHz. For comparison fairly, Faster R-CNN [20] was implemented in Torch.

From Table 3, we can see that our proposed binary deep CNNs for object detection achieves acceptable accuracy compared with the non-binary network. What’s more, compared to non-binary network, the proposed binary deep CNNs for object detection results in $62 \times$ faster convolutional operations and $32 \times$ memory saving in theory with an acceptable loss of accuracy.

The required memory for binary, float and double precision weights in two different architectures (AlexNet, VGG-16) are

| Category | VGG/BWBDN | VGG | Based on Alexnet | Faster RCNN | Faster RCNN |
|----------|-----------|-----|------------------|-------------|-------------|
| person   | 64.9/69.9 | 64.9 | 68.9/70.5    | 62.5/62.2   | 62.5/62.2   |
| car      | 61.2/67.1 | 61.2 | 68.9/70.5    | 62.5/62.2   | 62.5/62.2   |
| bike     | 90.0/96.1 | 90.0 | 96.9/98.5    | 85.9/85.9   | 85.9/85.9   |
| bus      | 67.9/67.2 | 67.9 | 65.9/65.5    | 64.9/65.5   | 64.9/65.5   |

Table 3 Results of VGG-based BWN and BWBDN on PASCAL VOC2007 (mAP, %)

| Method | Based on VGG/BWN VGG/BWBDN VGG | Based on Alexnet/BWN Alexnet/BWBDN Alexnet |
|--------|---------------------------------|------------------------------------------|
| mAP, % | 69.9/70.5    | 68.9/70.5    | 62.5/62.2 |
| fps    | 20               | 20             | 20         |
| input resolution | 300 × 300        | 300 × 300        | 300 × 300 |
A fast object detection algorithm based on binary deep CNN is proposed. Convolution kernels of different sizes are used to predict classes and bounding boxes of multi-scale objects directly in the last feature map of a deep CNN. In this way, rapid object detection with acceptable precision loss is achieved.

In addition, the binary quantisation for weight and input data of each layer is adopted to squeeze the networks for faster object detection. The proposed binary unified detection network effectively relieves the limitation of memory storages and computing power in a CNN-based detection method. Experimental results on Pascal VOC2007 validate the effectiveness of our proposed methods.

In the future, hardware acceleration techniques such as acceleration on FPGA, dedicated hardware and so on will be used to implement our proposed fast object detection method. The proposed method can be implemented in embedded computing systems efficiently because the binary operation for convolution makes a large number of the multiplications can be replaced by bit operations, and the proposed binary detection network has fewer network parameters.

## 5 Conclusion

Table 4 Drogue dataset detection results for different methods (mAP, %)

| Class      | Method       | VGG   | Faster RCNN | BWBDN VGG |
|------------|--------------|-------|-------------|-----------|
| drogue     | Based on VGG | 81.6  | 60.9        | 80.3      |

Table 5 Drogue dataset detection results for binarised layers (mAP, %)

| Class      | Method                     | VGG conv6 | VGG conv7 | VGG conv6 and conv7 | VGG conv6, conv7 and conv8 |
|------------|----------------------------|-----------|-----------|---------------------|---------------------------|
| drogue     | Based on VGG               | 81.6      | 54.4      | 62.8                | 80.3                      |

Table 6 VOC2007 dataset detection results for binarised layers (mAP, %)

| Class      | Method                               | Based on VGG | Binarised conv6 and conv7 | Binarised conv6, conv7 and conv8 |
|------------|--------------------------------------|---------------|---------------------------|----------------------------------|
| VOC2007    |                                      | 88.9          | 45.6                      | 44.3                             |

shown in Fig. 6. Compared to full-precision convolution, our binary-precision convolution can save 32% memory in theory. As shown in Fig. 6, e.g. the double precision convolution base on VGG needs 1106 MB memory, but the binary precision only requires 17.28 MB memory in theory.

In order to verify the generalisation and practical application of the proposed network on the other dataset, a drogue dataset for autonomous aerial refuelling of unmanned aerial vehicles based on the probe-and-drogue refuelling system is proposed in this paper. The detection results for the proposed network on the proposed drogue dataset are shown in Table 4.

The results, as shown in Table 4, show that the accuracy of our non-binary network based on VGG-16 and our BDBWM network achieve 81.6 and 80.3% mAP on the proposed drogue dataset, respectively. The proposed non-binary network is effective on the drogue dataset with higher accuracy, which is much higher than Faster RCNN. We can further see that our proposed binary deep CNNs for object detection also achieves acceptable accuracy compared with the non-binary network on the drogue dataset. The value of the mAP based on BWBDN VGG with 80.3% is even higher than Faster RCNN with only 60.9%. These experimental results show the generalisation of the proposed non-binary network.

In order to verify the effectiveness of different binarised layers in our network, some different layers are binarised. For example, ‘conv6’ is binarised, ‘conv6’ and ‘conv7’ are binarised and ‘conv6’, ‘conv7’ and ‘conv8’ are all binarised same as BWBDN. According to the different binarised layers, the detection results on the proposed drogue dataset are shown in Table 5.

From Table 5, we can see that our proposed binary deep CNNs with binarised ‘conv6’, ‘conv7’ and ‘conv8’ has the best result among these different binary layers methods. The value of mAP based on binarised ‘conv6’, ‘conv7’ and ‘conv8’ (same as BWBDN) is slightly lower than non-binary network and the parameters of BWBDN are minimal. The performance of the proposed network with only binary ‘conv6’ is worst. The reason may be that the layers of the proposed network are all non-binary layers except for ‘conv6’. The gradient propagation for only one binary layer ‘conv6’ is discontinuous for its input and output. The result of the network with binarised ‘conv6’ and ‘conv7’ further confirmed this view. With two continuous binary layers, the gradient propagation did not change quickly. The network with binarised ‘conv6’, ‘conv7’ and ‘conv8’ is adopted in our network named as BWBDN.

We also compared the different binarised layers on VOC2007, and the corresponding results are shown in Table 6. The value of the mAP is worse and worse with the increase in the number of binarisation layers on the VOC2007. The value of mAP based on binarised ‘conv6’, ‘conv7’ and ‘conv8’ (same as BWBDN) decreases 24.6% compared with the non-binary work, and it is also lower than the value of mAP based on binarised ‘conv6’ and ‘conv7’.

## 6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 61573349, 61703398 and 61673376 and in part by the National High Technology Research and Development Program of China (863 Program) under grant no. 2015AA042308. This manuscript is an extension of a conference paper presented at the 13th Chinese Conference on Intelligent Robotics.

## 7 References

[1] Shafie, A.A., Ibrahim, A.B.M., Rashid, M.M.: ‘Smart objects identification system for robotic surveillance’, Int. J. Autom. Comput., 2014, 11, (1), pp. 59–71

[2] Collins, R.T., Robert, T., Lipton, A.J., et al.: ‘Introduction to the special section on video surveillance’, IEEE Trans. Pattern Anal. Mach. Intell., 2000, 22, (8), pp. 745–746
[3] Wang, J., Kong, B., Mei, T., et al.: ‘A lane detection algorithm based on temporal-spatial information , matching and fusion’, CAAI Trans. Intell. Technol., 2017, 2 (4), pp. 275–291
[4] Bonin-Font, F., Ortiz, A., Oliver, G.: ‘Visual navigation for mobile robots: a survey’, J. Intell. Robot. Syst., 2008, 53 (3), pp. 263–264
[5] Pavlovic, V.I., Shamma, R.T., Huang, S.: ‘Visual interpretation of hand gestures for human-computer interaction: a review’, IEEE Trans. Pattern Anal. Mach. Intell., 1997, 19 (7), pp. 677–695
[6] Sacha, D., Zhang, L., Sedlmair, M., et al.: ‘Visual interaction with dimensionality reduction: A structured literature analysis’, IEEE Trans. Vis. Comput. Graphics, 2017, 23 (1), pp. 241–250
[7] Wang, X., Yuan, C.: ‘Recent advances on human-computer dialogue’, CAAI Trans. Intell. Technol., 2016, 1 (4), pp. 303–312
[8] Crouser, R.J., Franklin, L., Endert, A., et al.: ‘Toward theoretical techniques for measuring the use of human effort in visual analytic systems’, IEEE Trans. Vis. Comput. Graphics, 2017, 23 (1), pp. 121–130
[9] Kyrkou, C., Tsiofis, C., Theoharides, T.: ‘A hardware architecture for real-time object detection using depth and edge information’, ACM Trans. Embedded Comput. Syst., 2013, 13 (3), p. 54
[10] Zhuang, X., Kang, W., Wu, Q.: ‘Real-time vehicle detection with foreground-based cascade classifier’, IET Image Process., 2016, 10 (4), pp. 289–296
[11] Jiang, J., Li, X., Zhang, G.: ‘SIFT hardware implementation for real-time image feature extraction’, IEEE Trans. Circuits Syst. Video Technol., 2014, 24 (7), pp. 1209–1220
[12] Farrugia, N., Mamelet, F., Roux, S., et al.: ‘Fast and robust face detection on a parallel optimized architecture implemented on FPGA’, IEEE Trans. Circuits Syst. Video Technol., 2009, 19 (4), pp. 597–602
[13] Ma, X., Najjar, W.A., Roy-Chowdhury, A.K.: ‘Evaluation and acceleration of high-throughput object detection on FPAGs’, IEEE Trans. Circuits Syst. Video Technol., 2015, 25 (6), pp. 1051–1062
[14] Chen, S.Y., Zhang, J., Guan, Q., et al.: ‘Detection and amendment of shape distortion based on invariant moments for active shape models’, IET Image Process., 2011, 5 (3), pp. 273–285
[15] Hiromoto, M., Sugano, H., Miyamoto, R.: ‘Partially parallel architecture for adaboost-based detection with haar-like features’, IEEE Trans. Circuits Syst. Video Technol., 2009, 19 (11), pp. 41–52
[16] Girshick, R., Donahue, J., Darrell, T., et al.: ‘Rich feature hierarchies for accurate object detection and semantic segmentation’, IEEE Conf. on Computer Vision and Pattern Recognition, Columbus, Ohio, 2014, pp. 580–587
[17] Girshick, R., Donahue, J., Darrell, T., et al.: ‘Region-based convolutional networks for accurate object detection and segmentation’, IEEE Trans. Pattern Anal. Mach. Intell., 2014, 38 (1), pp. 142–158
[18] Girshick, R.: ‘Fast R-CNN’, IEEE Int. Conf. on Computer Vision, Santiago, Chile, 2015, pp. 1440–1448
[19] He, K., Zhang, X., Ren, S., et al.: ‘Spatial pyramid pooling in deep convolutional networks for visual recognition’, IEEE Trans. Pattern Anal. Mach. Intell., 2014, 36 (9), pp. 1904–1916
[20] Ren, S., He, K., Girshick, R.: ‘Faster R-CNN: towards real-time object detection with region proposal networks’, IEEE Int. Conf. on Computer Vision, Columbus, Ohio, 2015, pp. 1707–1715
[21] Redmon, J., Divvala, S., Girshick, R.: ‘You only look once: unified, real-time object detection’, IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, Nevada, 2016, pp. 777–786
[22] Liu, W., Anguelov, D., Erhan, D.: ‘SSD: single shot multibox detector’, IEEE Conf. on European Conf. on Computer Vision, Amsterdam, Netherlands, 2016, pp. 21–37
[23] Uijlings, J.R.R., Van De Sande, K.E., Gevers, T., et al.: ‘Selective search for object recognition’, Int. J. Comput. Vis., 2013, 104 (2), pp. 154–171
[24] Zitnick, C.L., Dollár, P.: ‘Edge boxes: locating object proposals from edges’, IEEE Conf. on European Conf. on Computer Vision, Zurich, Switzerland, 2014, pp. 391–405
[25] Wang, Q., Garry, G.M., Tiedje, J.M.: ‘Navie Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy’, Appl. Environ. Microbiol., 2007, 73 (16), pp. 5261–5267
[26] Keerthi, S.S., Shevade, S.K., Bhattacharyya, C.: ‘Improvements to Platt’s SMO algorithm for SVM classifier design’, Neural Comput., 2001, 13 (3), pp. 637–649
[27] Suykens, J.A.K., Vandewalle, J.: ‘Least squares support vector machine classifier’, Neural Process. Lett., 1999, 9 (3), pp. 293–300
[28] Toba, H., Ming, Z.Y., Adriani, M.: ‘Discovering high quality answers in community question answering archives using a hierarchy of classifiers’, Inf. Sci., 2014, 261, pp. 101–115
[29] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ‘Imagenet classification with deep convolutional neural networks’, in Proc. Int. Conf. Neural Inf. Process. Syst., California, USA, December 2012, pp. 1097–1105
[30] Simonyan, K., Zisserman, A.: ‘Very deep convolutional networks for large-scale image recognition’, arXiv preprint arXiv:1409.1449, 2015
[31] He, K., Zhang, X., Ren, S.: ‘Deep residual learning for image recognition’, arXiv preprint arXiv:1512.03385, 2015
[32] Butters, T.D., Güttel, S., Shapiro, J.L.: ‘Detecting and reducing redundancy in alarm networks’, IEEE Int. Conf. on Automation Science and Engineering, Dublin, Ireland, 2015, pp. 1224–1229
[33] Rastegari, M., Ordonez, V., Redmon, J.: ‘XNOR-Net: ImageNet classification using binary convolutional neural networks’, arXiv preprint arXiv:1603.05279, 2016
[34] Courbariaux, M., Bengio, Y., David, J.P.: ‘Binaryconnect: training deep neural networks with binary weights during propagations’, Advances in Neural Information Processing Systems, Montreal, Canada, 2015, pp. 3123–3131
[35] Kingma, D., Ba, J.: ‘Adam: a method for stochastic optimization’, arXiv preprint arXiv:1412.6980, 2014
[36] Everingham, M., Eslami, S.M.A., Van Gool, L., et al.: ‘The PASCAL visual object classes challenge’, Int. J. Comput. Vis., 2015, 111 (1), pp. 98–136