High-Definition Object Detection Technology Based on AI Inference Scheme and its Implementation

Hiroyuki Uzawa1), Shuhei Yoshida1), Yuukou Inuma1), Saki Hatta1), Daisuke Kobayashi1), Yuya Omori1), Ken Nakamura1), Shuichi Takada2), Hassan Toorabally2), and Kimikazu Sano1)

Abstract Video artificial intelligence (AI) applications for edges/terminals, such as non-visual drone flight, require object detection in high-definition video in order to detect a wide range of objects with one camera at once. To enable satisfying this requirement, we propose a new high-definition object detection technology based on an AI inference scheme and its implementation. In the technology, multiple object detectors cooperate to detect small and large objects in high-definition video. The evaluation results show that our technology can achieve 2.1 times higher detection performance in full HD images thanks to the cooperation of three object detectors.

Key words: AI, inference, high definition video, object detection, real time

Classification: Integrated circuits

1. Introduction

Video artificial intelligence (AI) applications for edges/terminals, where an AI inference scheme for object detection is executed in real time, have recently gathered attention [1, 2]. One application is non-visual drone flight. A drone has to detect objects in images input from a mounted camera in real time for safe flight, especially for avoiding flying directly over people’s heads. Another application is surveillance cameras, and object detection has to be done in a terminal for complying with personal information protection requirements, such as the general data protection regulation (GDPR). Future applications include autonomous driving.

In these applications, the number of mounted cameras is limited due to weight and power restrictions, which requires object detection in high-definition images in order to detect a wide range of objects with one camera. In a high-definition image such as full HD and 4K, objects near and far from the mounted camera can coexist in the same image due to a wide angle of view, and thus both large and small objects can be included in the image. In particular, small objects can account for a larger percentage of the objects in the image. Therefore, object detection in high-definition images has to be able to detect not only large objects but also small objects with high accuracy.

Conventionally, various AI inference schemes for object detection have been proposed, such as you-only-look-once (YOLO) [3-5] and single-shot-multibox-detection (SSD) [6], and dedicated engines have also been proposed for executing these schemes in real time [7-20]. The inference scheme has a convolutional neural network (CNN) that predicts object bounding boxes and class probabilities in a single evaluation. The input-layer size of a trainable CNN is limited to suppress the computational complexity and the training process difficulty. For example, the largest input-layer size in the standard model of YOLOv3 is 608 × 608 pixels. Thus, when a high-definition image is input, the input image is shrunk to the limited input-layer size and the inference scheme is executed with the shrunk input image. Large objects in the input image can be detected, but small objects are difficult to detect because the characteristic parts for identifying objects are also shrunk. Dividing the input image into a limited size can also be considered to prevent shrinking the characteristic parts [21-26], but this means large objects that straddle the divided images cannot be detected because the characteristic parts are also divided. In other words, the conventional approaches are unsuitable for object detection in high-definition images.

For this reason, we propose a new high-definition object detection technology based on an inference scheme and its implementation. In the technology, multiple object detectors cooperate to detect small and large objects in high-definition images. For real-time operation, each object detector is executed in parallel, which is achieved by assigning one inference engine at every object detector.

The remainder of the paper is organized as follows. In sections 2 and 3, we give details of the proposed technology and its implementation, respectively. In section 4, we present evaluation results. Section 5

---

1) NTT Device Innovation Center, NTT Corporation, Morinosato Wakamiya, Atsugi, Kanagawa 243-0198, Japan
2) ArchiTek Corporation, Kitahorie, Nishi-ku, Osaka 550-0014, Japan
a) hiroyuki.uzawa.zy@hco.ntt.co.jp

DOI: 10.1587/elex.18.20210323
Received August 02, 2021
Accepted October 04, 2021
Publicized October 19, 2021
concludes this paper.

2. Proposed technology

2.1 Overview of proposed technology

Our proposed technology enables both small and large objects to be detected in high-definition images by multiple object detectors cooperating on the basis of an inference scheme (Fig. 1). In this technology, an input image is divided into images of limited size, and objects are detected on the basis of an inference scheme for every divided image. In parallel, object detection in the whole image shrunk to a limited size is also done to detect large objects that straddle the divided images. By combining these object detection results, a final object detection result is obtained. Here, the combination is a process where the undetected objects in the shrunk whole image are interpolated by the object detection results obtained with the divided images. As a result, both small and large objects can be detected in high-definition images.

As YOLO is used as an example, we explain the mechanism by which the high-definition object detection can be achieved with the above-mentioned technology. YOLO divides the input image into \( S \times S \) grids and predicts bounding boxes and class probabilities for each grid. The resolution for detecting objects depends on only the grid size. Our technology reduces the grid size. Figure 2 shows an example where the input image is divided into four images. In the division process, since each divided image has \( S \times S \) grids, the total of the four divided images can be regarded as \( 2S \times 2S \) grids. This means that the grid size is halved. On the other hand, the division process causes the receptive field to narrow because it does not execute the convolutional operations for the whole image and thus cannot detect large objects. In contrast, in the whole process, although the detection resolution is low, a wide receptive field can be obtained. Therefore, in our technology, the division process with high detection resolution and small receptive field and the whole process with low detection resolution and wide receptive field are combined, which achieves the object detection with high detection resolution and wide receptive field. Concretely, when an image with 1920 \times 1080 \text{ pixels} is input, the minimum grid size without our technology (YOLOv3 only) is 60 \times 34 \text{ pixels} because \( S \) is 32. In contrast, with our technology, the minimum grid size becomes 30 \times 17 \text{ pixels} when there are four divisions, and higher detection resolution can be provided.

This technology is suitable for hardware implementation because all object detection processes can be executed in parallel. The same weight coefficients can also be used between all object detection processes. In addition, any AI inference scheme for object detection can be applied, and re-training for applying our technology is not needed. Moreover, the detection resolution can be improved by increasing the number of divisions. An extension with hierarchical structure, shown in Fig. 3, suppresses the penalty caused by increasing the number of divisions because of a finer combination, which enables our technology to support 4K/8K image inputs.

2.2 Detail of combination process

In the combination process, the undetected objects in the shrunk whole image are selected from the object detection results obtained with the divided images. The detected objects in the shrunk whole image and the selected objects are output as the final object detection results.

This selection requires judging whether an object detected in the divided image is an undetected one in the shrunk whole image or not. For this judgement, the
combination process calculates two indicators: a multiplicity between the detected object in the divided image and that in the shrunk whole image, and a ratio between the area size of the detected object in the divided image and that in the shrunk whole image.

When both these indicators are high, the process judges that the detected object in the divided image is the same as the detected one in the shrunk whole image and excludes it from the objects to be selected. In contrast, when either one of those indicators is low, the process judges that the detected object in the divided image is different from the detected one in the shrunk whole image. This judgement is performed for all detected objects in the divided images in combination with all detected objects in the shrunk whole image, and the non-excluded detected objects in the divided images are selected as the undetected objects in the shrunk whole image.

In detail, the above-mentioned multiplicity $I_{\text{multiplicity}}$ is a dedicated indicator for enabling comparison between the divided object and the whole object, which is expressed as (1).

$$I_{\text{multiplicity}} = \frac{A_{\text{overlap}}}{A_{\text{div}}} \quad (1)$$

Here, $A_{\text{overlap}}$ is the overlapped area size between the divided object detected in the divided image and the whole object detected in the shrunk whole image, and $A_{\text{div}}$ is the area size of the divided object, as shown in Fig. 4(a). Although intersection over union (IoU) is often used as multiplicity, it is unsuitable for comparison between a divided object and a whole object because it assumes comparison between whole objects. Therefore, we newly define the dedicated indicator $I_{\text{multiplicity}}$. When $I_{\text{multiplicity}}$ is larger than or equal to a threshold $\alpha$, the multiplicity is determined to be high. In the high multiplicity condition, the divided object is likely to be the same object as the whole object.

On the other hand, even if the multiplicity is high, the detected object in the divided image may be different from the detected object in the shrunk whole image when the area size ratio between those objects is high, as shown in Fig. 4(b). The above-mentioned size ratio $I_{\text{ratio}}$ is used for judging this condition. $I_{\text{ratio}}$ is given by

$$I_{\text{ratio}} = \frac{A_{\text{div}}}{A_{\text{whole}}} \quad (2)$$

where $A_{\text{whole}}$ is the area size of the detected object in the shrunk whole image. In the high multiplicity condition, when $I_{\text{ratio}}$ is larger than or equal to a threshold $\beta$, the detected object in the divided image is determined to be different from the detected object in the shrunk whole image.

By doing the above-mentioned judgement with those indicators, the undetected objects in the shrunk whole image are selected from the detected objects in the divided images, and both large and small objects can be detected while avoiding duplicate detection of the same object.

3. Implementation

For scalable implementation, the proposed technology is achieved with a multi-chip configuration. Each chip mainly has a central processing unit (CPU), an image processing engine for general purposes, such as image division and shrinking, and an AI inference engine, as shown in Fig. 5. The AI inference engine has a dedicated interface for multi-chip cooperation in addition to an inference accelerator for executing object detection in real time.

Figure 6 shows the multi-chip configuration with the inference engines. In Fig. 6, the proposed technology is implemented under the condition that there are two divided images. The master chip performs the object detection in the shrunk whole image and broadcasts its result to all slave chips. Each slave chip also performs the object detection in the divided image in parallel while the object detection is performed in the master chip. After receiving the detection result in the master chip, each slave chip selects only the objects undetected in the master chip and forwards the selected result to the master chip. The master chip outputs the detected objects in the master chip and the selected objects from the slave chips as the final detection result. These processes are executed with a pipeline, as shown in Fig. 7. Unlike the conventional multi-chip configuration where the interim result of the matrix calculation for the convolutions is shared between chips [20], our configuration for the proposed technology distributes only the object detection result (metadata) between chips. Therefore, the required interface rate for multi-chip configuration is much smaller than that for the conventional configuration. For example, in the configuration shown in Fig. 6, the maximum data rate for forwarding and receiving metadata is 92 Mbit/s when the YOLOv3 model with 608 x 608 pixels is used as the inference scheme. Thus, a scalable implementation can be achieved with a low speed interface of 1 Gbit/s or less, which is easy to scale up.

The inference accelerator in the AI inference engine consists of a feature-map-read (FmapRD) block, a feature-map-write (FmapWR) block, a kernel-read (kernelRD) block, a multiply-accumulate (MAC) block, and a neural network control (NNCTRL) block (Fig. 5). The NNCTRL block controls other blocks in accordance with the neural network configuration based on the
inference scheme. To pursue real-time operation, the maximum kernel size is specialized for image processing and fixed to $3 \times 3$. Although a fixed-point calculation circuit is adopted in the MAC block, the position of the decimal point is flexible in each layer, which enables the dynamic range in each layer to be changed and improves the detection performance. An 8-MB cache memory provides efficient external memory access. In particular, when the feature map size in a layer is small, all pixels included in the feature map are kept in the cache in order to remove the external memory access for the next layer.

To assess feasibility, we incorporated the above-mentioned inference engine inside ArchiTek Intelligence®️ real-time AI and image processing LSI, which was fabricated with 12-nm CMOS technology. The ArchiTek Intelligence®️ [27] can easily integrate various user-defined engines. Table I shows a feature summary for this chip, and Figure 8 shows a photograph of the chip layout under fabrication. In these design specifications, the inference engine can theoretically execute the YOLOv3 model with $608 \times 608$ pixels in real time at 20fps. The power consumption of our inference engine is 0.421 W, which is a time-averaged value estimated with RTL-level power estimation tool. The interface line rate for multi-chip cooperation is 800 Mbit/s. With three chips, the real-time object detection in full HD video can be achieved when there are two divided images in the division process.

4. Object detection performance

To verify the effectiveness of the proposed technology, we applied it to the standard YOLOv3 model with $608 \times 608$ pixels and evaluated the object detection performance. In our technology, all object detectors use the same weight coefficients. We used the weight coefficients published in [28]. The published coefficients are pre-trained with the Microsoft Common Objects in Context (MS COCO) dataset [29], which is a widely used object detection dataset with 80 classes. In the proposed technology, the thresholds $\alpha$ and $\beta$ described in section 2.2 were set to 0.5 and 0.1, respectively. There are two divisions.

First, as a basic evaluation, we used 5000 images in the COCO dataset val2017 and measured the AP value for each class. The AP value is a value obtained by the precision and recall. Although the image included in the MS COCO dataset is a standard-definition image such as $600 \times 400$ pixels, the large objects accounts for a higher percentage of the objects in the image and thus we can confirm the image-division penalty in our technology by comparing the APs with and without our technology. Here, the measured AP is one averaged by 10 IoU thresholds in 0.05 increments from 0.5 to 0.95.

Figure 9 shows the measured results. The AP value was improved in almost all classes and by a maximum of 1.2
times in these results. This means that our technology can suppress the image-division penalty for large objects and can improve the detection performance even in standard-definition images.

Next, we evaluated the object detection performance with high-definition images. We selected 150 high-definition (1920 × 1080) images from VisDrone2019 dataset [30] and evaluated the precision and the number of detected objects. The VisDrone dataset is large-scale drone-captured dataset with 10 classes and includes full HD images. For the weight coefficients, we used the same coefficients pre-trained by the COCO dataset as above. To calculate the precision, we remapped people and pedestrian labels to person label and van label to car label in the VisDrone dataset, and only the common class labels between COCO and VisDrone were evaluated.

Table II summarize the detection results, and Figure 10 shows the example images obtained in this evaluation.

Table II. Summary of the detection results (image size: 1920 × 1080 pixels).

| Class      | W/o our technology (standard YOLOv3 model only) | W/ our technology |
|------------|-----------------------------------------------|-------------------|
|            | Precision [%]                                 | Total number of objects detected in 150 images | Precision [%] | Total number of objects detected in 150 images |
| Person     | 62.5                                          | 176               | 62.8          | 298               |
| Car        | 83.9                                          | 1426              | 81.5          | 2049              |
| Bicycle    | 50.0                                          | 6                 | 54.5          | 11                |
| Truck      | 17.1                                          | 111               | 19.3          | 176               |
| Bus        | 33.3                                          | 6                 | 15.0          | 20                |
| Motorbike  | 61.5                                          | 13                | 59.5          | 37                |

Fig.9 AP value of each class (image size: about 600 × 400 pixels).

Fig.10 Examples of detection results (a) without our technology (standard YOLOv3 model only) and (b) with our technology.
From Table II, our technology enables the number of detected objects to be increased while maintaining the precision. For example, the number of detected objects in the person class is 1.7 times higher with our technology than without it. Across all evaluation classes, it is 2.1 times higher on average.

Figure 11 shows the size distribution of detected objects. From Fig. 11, we can see that our technology can detect much smaller objects. Specifically, the minimum width size is halved from 12 pixels without our technology to 6 pixels with it. This is because $2S$ (width) $\times S$ (height) grids are achieved by our technology when there are two divided images in the division process, as described in section 2.

These results show that our technology can improve the object detection performance in not only standard definition images but also high-definition images with suppressing the image division penalty.

5. Conclusion

To enable a wide range of objects to be detected with one camera at once for edge/terminal AI applications, we proposed a high-definition object detection technology based on an inference scheme and its implementation. The proposed technology detects objects with a high detection resolution and a wide reception field by combining the division process with multiple divided images and the whole process with the shrunk whole image. For real-time operation, these processes are executed in parallel by using multiple inference engines with a multi-chip cooperation interface.

We applied our technology to the YOLOv3 model with 608 $\times$ 608 pixels and evaluated the object detection performance. The evaluation results show that our technology can improve the object detection performance in not only standard definition images but also high-definition images.

References

[1] X. Wu et al., “Recent Advances in Deep Learning for Object Detection,” Neurocomputing, vol. 396, pp.39-64, 2020.
[2] L. Jiao et al., “A Survey of Deep Learning-Based Object Detection,” IEEE Access, vol. 7, pp. 128837-128868, 2019.
[3] J.Redmon et al., “You only look once: Unified real-time object detection”, IEEE CVPR, pp. 779-788, Jun. 2016.
[4] J.Redmon et al.,“YOLO9000: Better faster stronger”, IEEE CVPR, pp. 6517-6525, Jul. 2017.
[5] J.Redmon et al., “YOLOv3: An Incremental Improvement,” [online] Available: https://arxiv.org/abs/1804.02767
[6] W. Liu et al., “SSD: Single shot multibox detector,” [online] Available: https://arxiv.org/abs/1512.02325.
[7] H.Nakahara et al, “A demonstration of FPGA-based you only look once version2 (YOLOv2),” IJPL, pp. 457-4571, Aug. 2018.
[8] Y. J. Wai et al., “Fixed Point Implementation of Tiny-Yolo-v2 using OpenCL on FPGA,” JACSA, vol 9 Issue 10, 2018.
[9] D.T.Nguyen et al., “A High-Throughput and Power-Efficient FPGA Implementation of YOLO CNN for Object Detection,” IEEE Trans. VLSI, vol. 27, no. 8, pp.1861-1873, 2019.
[10] C.Ding et.al., “REQ-YOLO: A resource-aware efficient quantization framework for object detection on FPGAs,” [online] Available: https://arxiv.org/abs/1909.13396v1.
[11] Y. J. Wai et al., “A scalable FPGA based accelerator for Tiny-YOLOv2 using OpenCL,” IJRES, 8, pp. 206–214, 2019.
[12] S. Zhang et al., “An FPGA-Based Reconfigurable CNN Accelerator for YOLO,” IEEE ICET, pp. 74-78, 2020.
[13] Z. Wang et al.,“Sparse-YOLO: Hardware/Software Co-Design of an FPGA Accelerator for YOLOv2,” IEEE Access, vol. 8, pp. 116569-116585, 2020.
[14] K.Xu et al., “A dedicated hardware accelerator for real-time acceleration of YOLOv2,” J Real-Time Image Proc 18, 2021.
[15] H. Fan et al., “A Real-Time Object Detection Accelerator with Compressed SSDLite on FPGA,” FPT, pp. 14-21, 2018.
[16] F. Sun et al., “A High-Performance Accelerator for Large-Scale Convolutional Neural Networks,” IEEE ISPA/UIUC, 2017.
[17] A.Maki et al., “Weight Compression MAC Accelerator for Effective Inference of Deep Learning,” IEICE Trans.Electron, vol.E103-C, No.10, pp.514-523
[18] A. Zhou et al., “Incremental network quantization: Towards lossless CNNs with low-precision weights”, Feb. 2016, [online] Available: https://arxiv.org/abs/1702.03044.
[19] JJo et al., “DSIP: A Scalable Inference Accelerator for Convolutional Neural Networks,” IEEE J. Solid-State Circuits, vol. 53, no. 2, pp. 605-618, Feb. 2018.
[20] B.Zimmer et al., “A 0.11 pJ/Op, 0.32-128 TOPS, Scalable Multi-Chip-Module-based Deep Neural Network Accelerator with Ground-Reference Signaling in 16nm,” VLSI circuits, June, 2019.
[21] V. Ruzicka et al., “Fast and accurate object detection in high resolution 4K and 8K video using GPUs,” IEEE HPEC, 2018.
[22] G. Plastiras et al., “Efficient ConvNet-based Object Detection for Unmanned Aerial Vehicles by Selective Tile Processing,” ICDSC, Sep. 2018.
[23] D.Vorobjov et al., “An Effective Object Detection Algorithm for High Resolution Video by Using Convolutional Neural Network,” LNCS, vol.10878, pp.503-510, 2019
[24] R.Bohush et al., “Object detection algorithm for high resolution images based on convolutional neural network and multiscale processing,” CMIS-2021, April 2021
[25] M.Gao et al., “Dynamic Zoom-in Network for Fast Object Detection in Large Images,” IEEE CVPR, June 2018.
[26] C. Tang et al., “Multi-view object detection based on deep learning,” Applied Sciences 2018, vol. 8, no. 9, p. 1423
[27] ArchiTek Corporation, “aIPE: The New Blueprint for AI” https://architek.ai
[28] https://pjreddie.com/darknet/yolo/
[29] T.Y. Lin et al., “Microsoft coco: Common objects in context,” in European conference on computer vision, 2014, pp. 740–755.
[30] P.Zhu et al., “Vision Meets Drones: A Challenge,” arXive-prints, page arXiv:1804.07437, Apr. 2018.