Object Detection Using Zynq-7000 FPGA For An Embedded Application

Miss Riddhi K. Yerne[1], Dr. Mr. B.G. Patil[2]
[1] Dept of Electronics Engineering
[2]HOD, Dept of Electronics Engineering
[1, 2] Walchand college of Engineering, Sangli, Maharashtra

Abstract- *Object detection basically means detecting an object or any relevant thing from the image. In this research we have used Petalinux i.e. Operating system for the Zynq-7000 FPGA. Many object detection algorithms are there to detect the objects. But YOLO object detection is one of the fastest algorithms. It has best accuracy in detecting the objects. Abundant amount of applications is there where we can use Zynq-7000 FPGA such as autonomous vehicles, military reconnaissance and robotics. Thus Zynq-7000 FPGA is a good platform for all these types of application as it provides good processing speed, time required is less and cost is also reliable. It offers flexibility and scalability, meanwhile it provides high performance and low power. Most of these applications demand low power consumption, compact and lightweight design, and high-speed computation platform for processing image data in real time.*

Keywords- Zynq-7000 FPGA, Petalinux, BSP files, YOLO (You only look once), Vivado, HLS.

I. INTRODUCTION

The field of Computer vision is now widely progressing including deep learning and CNN i.e. Convolutional Neural network. Object detection is most crucial ability required by most computer and robot vision systems.

Classifying and finding unknown number of objects in an image was very difficult task earlier. But now many companies are productizing this task with very ease, companies like Google and IBM are widely known for many of its software and applications that are being used for the object detection. In many computer vision systems, object detection is the first task to be performed as it gives us all the detailed information about the scene. Once the object is detected it is further possible to carry out much more information about that particular thing. Object detection has been used in many applications such as human computer interaction, robotics, consumer electronics, tracking, autonomous and assisted driving. Each of these applications has special requirement such processing time and detection if pose is changed. While many require only one object class others require multiple object classes. We are still far from gaining human-level performance, in terms of open-world learning. There are still many applications left behind where object detection is not used. Although object detection can provide much more help such as service robots and autonomous machines. Finally, we need to consider that we will need object detection further such as to explore depth parts of the sea or other planets and also new object classes will be formed. It will get high value as our object detection system will learn to detect various class objects around us. In such cases a real time learning ability is much more important. As the object detection techniques are widely increasing there are many more complex algorithms that are emerging in this new world. All of these algorithms have maximum speed and identification time is very less. By using FPGA it's easy to use in the field. Object detection has much more potential. When it comes to future the possibilities are endless. Sports broadcasting will be employing this technology in instances such as detecting when a particular football team is about to score and will directly notify the fans via mobile phone or to create a virtual reality setup. In video collaboration leader will be able to count the number of attenders of a meeting. Object detection can be majorly used for security industry. There are various object detection methods. Some of the methods are as follows:

1) Region Proposals
2) R-CNN, Fast R-CNN, Faster R-CNN
3) YOLO (You only look once)

All of these methods vary differently according to their time of detecting the objects. Also, some methods may have drawback of losing important data in the image.

II. LITERATURE SURVEY

Since in embedded applications the used platform and object detection techniques are mandatory part, where major focus needs to be given. Hence comprehensive of review is presented here. Following are the few literatures

explaining various methods of object detection and also giving brief information about the platform that have been used.

Abdelkader BEN AMARA [2018] [1] presented, an FPGA based mobile feature detection and tracking solution is for complex video processing systems. Presented algorithms include feature (corner) detection and robust memory allocation solution to track in real-time corners using the extended Kalman filter. Target implementation environment is a Xilinx Zynq SoC FPGA based. In this paper they have used Harris corner detection and Kalman filter w.r.t non max suppression algorithm. The temporal performances of their approach are roughly 50% better than these of classic implementations. Also, this application is being used for visually impaired persons so typically they only detected corners but not the whole object such as trees. Many obstacles can be there but this approach is only good for corner detection. [6] *Wenchao Liu* presented an FPGA-based moving object detection and tracking system isintroduced for image processing application. The algorithms presented include object detection based on dynamic background difference, Kalman filter for object tracking. The target device for the implementation is a Xilinx Zynq-7000 FPGA.

In the dynamic background, the system realized various objects but while at tracking we have to differentiate between the objects that are important and are necessary for any application this is time consuming. There are various limitations of the Kalman filter - It assumes that both the system and observation models' equations are both linear, which is not realistic in many real-life situations. Also, it assumes that the state belief is Gaussian distributed. [7] In this paper, Hanaa M. Abdelgawad shown how HLS constraints and optimization directives were applied for timing and area optimization. The implementation of stream-based canny edge detector processing using C-based HLS is presented. The results show that hardware accelerators enhance the complex computation of the processing functions. The hardware accelerators on FPGA enhance the computational performance: the CPU utilization drops down and the frame rate increases, in ZYNQ platform it is up to 60 fps for a resolution of 1280 x 1024. There are many computer visions application which can take advantage of hardware accelerators to enhance performance of real-time highly computational applications. When targeting HLS design flow, the implementation of C/C++ code is rapidly developed for hardware accelerator. In the Future work, besides improving the quality of edge texture map result by distributed canny edge detection algorithm. Implementation of augmented reality (AR) pipeline is considered to make use of the cooperation between CPU and FPGA. Highly computational video and image processing operations of augmented reality will be as hardware accelerators. This will enhance the real time performance of AR applications.

III. PETALINUX AND YOCTO-FLOW

Petalinux is an operating system for Zynq-7000 FPGA. It is used to build and deploy embedded Linux solutions on Xilinx processing system. In other words, we can say that it is a tool chain to generate Linux kernel images, root file system and kernel modules. Petalinux is much beneficial we can use this kernel images instead of using any other cross compilation tool. Petalinux requires a number of standard development libraries that should be installed in Linux system. A peta project directory contains configuration files of the project, the linux subsystem and the components of the subsystem. The petalinux build command builds the project with those configuration files. Petalinux-config command is to modify them. The petalinux project has two layers under -/project-spec.

- meta-plnx-generated – this layer contains all bbappends and configuration fragment for all the components. All files are generated by the tool based on the HDF and user configuration. These files are generated from the petalinux-config and petalinux-build commands
- meta-user – this layer is a place where all user-specific changes are done. We can add here bbappends and configuration files.

The Yocto project is an open source collaboration project that helps developers to create tradition linux based system for embedded products irrespective of the products hardware architecture. Yocto project provides a flexible toolset and a development environment that allows embedded device developers to collaborate through shared technologies, software stacks, and configurations used to create linux images. Yocto project has many benefits such as both in systems and applications development and memory utilization. It also has software customizations and can be interchanged with in multiple hardware. Poky is Bitbake, open embedded core and Board support packages. Poky is a reference OS kit. Poky consists of Bitbake and open embedded- core as well as set of metadata so that we can make our own distribution. Poky comprises of meta-poky and meta-yocto bsp. Yocto Project (YP) Umbrella open source project that builds and maintains validated open source tools and components associated with embedded linux Poky Yocto project open source reference embedded distribution. Open Embedded (BB) Meta data is the key element of the yocto project. It is basically used to construct a linux distribution. The meta data consists of which data should built the software and also what

parts are necessary. Also, yocto by itself are used to fix bugs and customizes the software for a particular situation. The build instructions consist of commands that helps the linker and archiver 26 for packaging tools and other programs. Each software components on the system i.e. individual program has their own separate file to express its meta-data.

Now it is the time to articulate the research work with ideas gathered in above steps by adopting any of below suitable approaches:

IV. PROPOSED SYSTEM

The proposed system can be Object detection using Zynq7000 FPGA for various embedded applications. Object detection is basically detecting the instances of relevant objects of certain class in digital image. With the deeper research on object detecting more and more complex algorithms are presented which leads to the larger calculation and throughput. The target device for implementation of this algorithms and over-all procedure is Xilinx Zynq-7000 FPGA. Zynq-7000 includes dual package that is dual core ARM Cortex-A9 based Processing System (PS) and Xilinx Programmable Logic in a single device. Zynq7000 FPGA is a good platform, it has good processing speed, and time required is less and reduces the cost. It offers flexibility and scalability, meanwhile it provides high performance, low power. Vivado Design Suite has been used for analysis of HDL designs, perform timing analysis and examine RTL diagrams and also configure the zynq-7000 FPGA with a programmer. Earlier versions of Xilinx ISE are also their but Vivado consists of tool chain that converts C code into Programmable logic. Also, many processors can be integrated and lots of peripherals can be dumped into it. Considering all these advantages, Xilinx Zynq-7000 FPGA is chosen to be supreme platform for this work. After studying of algorithm, we found that its implementation is little tricky but in accuracy and speed these are better. Various object detection algorithms are introduced now. Comparison chart of object detection algorithms is shown in Table 1, also given that how Yolo is fast and its accuracy is better than other algorithms. Some are as follow-

• R-CNN,
• SPP,
• Fast R-CNN, Faster R-CNN,
• Yolo

YOLO is an extremely fast real time multi object detection algorithm. YOLO stands for “You Only Look Once”. The algorithm applies a neural network to an entire image. The network divides the image into an S x S grid and comes up with bounding boxes, which are boxes drawn around images and predicted probabilities for each of these regions. The method used to come up with these probabilities is logistic regression. The bounding boxes are weighted by the associated probabilities. For class prediction, independent logistic classifiers are used.

V. METHODOLOGY

The methodology for the Object detection using Zynq-7000 FPGA for various embedded applications is as follows:

Firstly, we need to download Board support packages as per our requirement. For the system design we need petalinux version 2020. Further methodology is explained below:

- Download petalinux 2020 and board support package file.
- After installing petalinux OS we need to make sub directory in the petalinux2020 parent directory. To establish peta environment we should execute the .sh file.
- Both BSP file as well as sub directory should be linked to each other.
- After linking both the files we will get one new sub directory i.e. Xilinx qemu run. This is because we have linked both the files. All this work will be done in terminal 1.
- Then we will dump the peta that is one qemu virtual window will be formed. It will ask for password and id. Both password as well as id are root. It will give all the detail about the custom based image which was already included in the bsp and peta 2020 version.
- Now, on the other terminal we need to open again the same project that was being created in petalinux2020 directory.
- After opening that directory, we need to create secure shell terminal with secure copy protocol command to transfer the files between the virtual and the host window. Secure copy protocol in generally used in linux. As it securely copies the files. And this security is maintained by the secure shell. It gives safety and security to our communication. Also, called ssh has two keys in use i.e. private as well as public key.
- Now on virtual window we can see that our files are being copied, and thus we can run our python script. And for the output one folder will be created on the

desktop with name new output. After opening this folder, we can see processed image over there.

- Hence object detected successfully.

The proposed system can be Object detection using Zynq7000 FPGA for various embedded applications. Object detection is basically detecting the instances of relevant objects of certain class in digital image. With the deeper research on object detecting more and more complex algorithms are presented which leads to the larger calculation and throughput.

VI. RESULTS

The output for the Object detection using Zynq-7000 FPGA for various embedded applications is:

Input image

Output image

VII. CONCLUSION

Petalinux as well as Zynq-7000 FPGA both can be used for object detection purpose. QEMU virtual window is very relaxed to use. Also, secure communication can take place.

- Accuracy of the object detection is 80 to 90 percent as we can see from the following output.
- Thus, Petalinux2020 version and Board support package is very useful as it already contains Zynq-7000 FPGA block design with all the dependencies.
- Object detection using FPGA is very efficient. It is good for parking system as well as is very useful in self-driving cars.
- Further future objective can be detecting the objects such as pedestrian, car, Human beings with their correct counting numbers.

VIII. ACKNOWLEDGEMENT

This work is partially supported by Dr. Mr. B.G. Patil, Head of the Electronics department at Walchand College of Engineering, Sangli. I am eternally grateful to him for his support and guidance. He has always guided me with his valuable suggestions and encouraged me a lot.

REFERENCES

[1] Abdelkader BEN AMARA (*IJACSA)* (Zynq FPGA based and Optimized Design of Points of Interest Detection and Tracking in Moving Images for Mobility System), University of Rouen Normandy & CNRS FR 3638 Rouen, France, *Vol. 9, No. 10, 2018*

[2] (An Object Detection System Based on YOLO in Traffic Scene) Jing Tao, Hongbo Wang, XinyuZhang,Beijing, China 978-1-5386-0493-9/17 ©2017 IEEE

[3] (Pedestrian Detection Based on YOLO Network Model) Wenbo Lan, Jianwu Dang, Yangping Wang and Song Wang Proceedings of 2018 IEEE International Conference on Mechatronics and Automation August 5 - 8, Changchun, China

[4] (FPGA Implementation for Image Processing Algorithms Using Xilinx System Generator) Neha. P. Raut, Prof.A.V. Gokhale, *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 2, Issue 4 (May. – Jun. 2013),*

[5] (Optimizing CNN-based Object Detection Algorithms on Embedded FPGA Platforms) Ruizhe Zhao1, Xinyu Niu1, Yajie Wu2, Wayne Luk1, and Qiang Liu3 Imperial College London fruizhe. zhao15, niu. xinyu10, w.lukg@imperial.ac.uk

[6] (MOVING OBJECT DETECTION AND TRACKING BASED ON ZYNQ FPGA AND ARM SOC) *Wenchao Liu, He Chen*, Long Ma (Beijing Institute of Technology, Beijing 100081), China*

[7] (High Level Synthesis of Canny Edge Detection Algorithm on Zynq Platform) Hanaa M. Abdelgawad, Mona Safar, Ayman M. Wahba. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol: 9, No: 1, 2015