

Improving Medical Imaging Diagnostics by Utilizing Compression Techniques and Implementing FPGA Acceleration with Vivado HLS

Karthikeyan. R¹, Dr. Hariharan Illango², Rajakumar. P³, Dr. Sumathi Sokkanarayanan^{4*},
Chithrakkannan. R⁵

Submitted: 04/02/2024 Revised: 12/03/2024 Accepted: 18/03/2024

Abstract this paper aims to provide efficient medical image compression/decompression using vivado HLS (High level synthesis). Data contrast compression method has been proposed to perform image compression using vivado HLS, HLX design blocks for implementation using Arty z7 kit. Data contrast compression method provides efficient image compression by performing 'C/C++' code in vivado HLS environment and effective design blocks are designed using VHDL code in vivado HLX environment. Performances of above two environments are implemented in Arty z7 20 kit. Results for Software simulation, Hardware simulation, design blocks, synthesis, implementation, bit generation, elaborated design or schematic view are generated accordingly. Hardware implementation result can be viewed in Arty z7 kit. Performance parameters like compression ratio, latency are concentrated and tabulated accordingly. Medical Image compression applications are widely used in the field of telehealth for storage and communication purpose.

Keywords: Vivado HLS, Vivado HLX, Data contrast compression, Image compression.

1. Introduction

Medical image processing widely used in the applications like detection of particular region in an image, segmentation mainly used for identifying tumor cells, bones and skeleton segmentation. Major part in segmentation process leads to detect the cancer cells in prior condition and evaluating thickness of cartilage in human knee part. Medical image compression plays vital role in transferring of medical images from one medical center to another which is frequently used in telemedicine to provide better for the people in rural and remote areas. Depending on the image, the decompression method adapts to either lossy or lossless compression techniques. The implementation was carried out in Vivado HLS using the Arty Z7-20T kit. Xilinx Vivado Design Suite offers an effective feature for designing logic by incorporating IP modules into the overall design structure. The majority of designs are created using the IP catalog, which encompasses various modules such as the system generator for DSP designs and the Vivado repository for HLS designs. Additionally, third-party IP can be integrated and packaged using the Vivado IP Packager tool, which

operates in both project and non-project modes.

In project mode, RTL designs are developed in accordance with the project guide available on the Xilinx forum, and the outputs are generated after completing the simulation and synthesis processes. Subsequently, the implementation process proceeds. In non-project mode, outputs are generated based on the completion of the simulation and synthesis processes, with reference to the Xilinx Core Instance (XCI) file. The Vivado implementation process involves a series of steps for placing and routing the netlist onto the device while adhering to logical, physical, and timing constraints. It supports both Synopsys design commands and Xilinx design commands and follows a well-defined procedure for converting C, C++, and system code into RTL design. Initially, it creates an RTL implementation, followed by transformations aimed at achieving low area and high throughput implementations without altering the original 'C' code. The implementation process consists of several sub-processes, including optimized design, power-optimized design, place design, post-place power-optimized design, post-place physical-optimized design, route design, and post-route physical-optimized design. Upon the completion of the implementation process, a bitstream is generated using the 'write bitstream' command, resulting in a '.bit file' that can be programmed into an FPGA device for further processing. The implementation phase encompasses logic optimization, placement of logic cells, and routing of connections between cells.

2. Established Methodologies

High level synthesis: Productivity, performance and

¹Department of ICE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India.

²School of Electronics Engineering, Vellore Institute of Technology, Chennai Campus., Tamil Nadu, India.

³Department of EEE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India

⁴Department of ECE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India
Sumathi.ECE@sairam.edu.in

⁵Department of ICE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India

software constraints (Yun Liang et al., 2019) [22] in this paper, it explains about the efficiency of HLS tool and also it describes about the evaluation of how HLS tool is applicable to the real world applications. Review of stereo matching has been done for the applications of image denoising, image retrieval, feature extraction and face recognition. The proposed method provides best platform to interconnect software and hardware designs using HLS. Stereo matching algorithm achieved 3.5x to 67.9 x speeds with 5x reduction in design effort. The above achievement is less than manual RTL design. **Design and Evaluation of an FPGA based Hardware Accelerator for Deflate Decompression (Morgan et al., 2019) [23]** this paper aims to provide efficient data transfer using deflate lossless data compression/decompression with the proposed hardware accelerators. It explains about the effective implementation of deflate decompression algorithm using HLS designs coded in C++, and this code dumped in xilinx virtex ultrascale + class FPGA. While decompressing, input and output throughputs are 70.7 (246.4) and 130.6 (386.6) MB/s dynamically and statically encoded files. Maximum throughput can be achieved upto 375 MB/s. **HLS – Based Optimization and Design Space Exploration for Applications with Variable Loop Bounds (Young – Kyuchoi et al., 2018) [24]** in this paper, proposed framework design space exploration (DSE) has been introduced to improve the efficiency of FPGA's and to obtain efficient FPGA design parameters. DSE framework has been designed using high level synthesis tool (HLS). HLS based FPGA optimization and DSE framework provides high performance design blocks even in different loops. This application is used for effective baseline implementation throughput of 75x. **Study on digital image inpainting method based on multispectral image decomposition synthesis (Jia Zhaoyang et al., 2018) [4]** this paper aims to provide the analytical report of image reproduction for affected painting arts. The Proposed method multispectral image decomposition synthesis, it removes the structure and texture of the input image used. During inpainting process, the proposed method reduces the accuracy of output image due to affected area. To avoid this, synthesis correlation has been performed between colors components present in the input image. It concentrates on two parameters, MSE and PSNR. Value obtained has been indicated as MSE = 2.7951 and PSNR= 44.1681. **A High capacity reversible watermarking scheme based on shape decomposition for medical images (Xin zhong et al., 2018) [4] [5]** this paper describes about the proposed algorithm, bottom up saliency detection has been applied to medical images to detect the region of interest (ROI). It generates square shape for non-region of interest (NROI). It has been experimented on OASIS medical image dataset which consists of 416 subjects. The proposed algorithm produced effective watermarking capacity and improved image

fidelity. **A New RSA image encryption algorithm based on singular value decomposition (Kai Zhu et al., 2018) [7]** in this paper RSA encryption algorithm based on singular value decomposition has been proposed, it analyse the datasets in the form of statistical order and in secured basis. Better encryption efficiency has been obtained and it is applied in the areas of military, medical and in digital images.

Design of Embedded Architecture for Pedestrian Detection in Image and Video (Taoufik Salem et al., 2017) [15] in this paper, detecting pedestrian in a particular time is a main factor. Here hardware architecture for pedestrian detection system has been proposed. The system consists of effective extractor and classifier which help to detect an object in a particular time. Hardware architecture consists of many design blocks. These design blocks are designed by using xilinx HLS tool, software development kit (SDK) for hardware – software co design. Implementation of a proposed system shows effective classification with less energy and time consumption. Proposed system has a capability to detect a pedestrian for high definition video (HD) at 180 frames per second. **High Throughput 2D spatial image filters on FPGA's (Ab- dullah Al- Dujaili et al., 2017) [14][8]** this paper aims to provide clear explanation about implementation of two dimensional spatial filters on FPGA's. It shows improved performance DSP blocks, which has an eligibility to produce effective pixel arrangement. Nowadays FPGA's are an exploited heterogeneous resource, which allows user functions to be implemented with greater performance, low power and less area. The above DSP blocks are added with number of features, interconnectivity, architectures to provide efficient two dimensional spatial filter implementation. **Real Time System Implementation for Video Processing (Dr.Wagdy.H.Mahmoud et al., 2016) [1] [27] [24]** this paper aims to provide, implementation of canny edge algorithm on real time hardware/software video processing system in zynq FPGA platform. The above proposed work carried out in xilinx environment by using vivado HLS tool and it has achieved detection of efficient edges for the input stream of 1080p full HD in a real time with effective time consumption.

3. Proposed Methodology

The proposed study focuses on enhancing image compression efficiency through the utilization of the Data Contrast Compression method, implemented on the Arty Z7 kit. An input image with dimensions of 345 x 230 pixels was subjected to this method within the Vivado HLS environment. This research primarily caters to applications in remote sensing, the medical field, and telehealth. The compression process is specifically tailored for the rapid transmission of image datasets. In the medical field,

compressed images are swiftly transmitted for real-time consultations. The Data Contrast Compression method yields both a successful compression outcome and an impressive compression ratio for storage and transmission.

Now, the proposed method has been coded in 'C' to enable image compression with an initial pixel size of 256x256. In this process, the 'core data' file is dynamically generated based on the input and function values. Verification of the 'core data' file is a prerequisite in the 'make file rules' before advancing to the 'C' simulation phase. Following the completion of the 'C' simulation with 'csim.exe,' 'csim.mk' files are produced. Subsequently, the process proceeds to 'C synthesis' and the associated build procedures, generating comprehensive reports. Figure 1 represents flow diagram of proposed data contrast compression method.

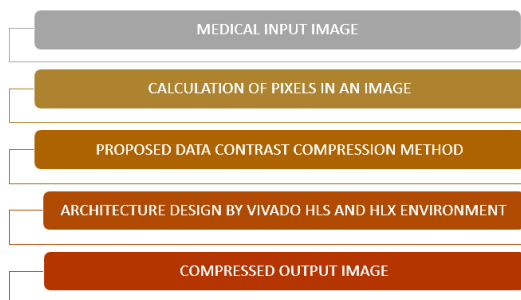


Fig 1 Flow diagram of proposed method

Sequential steps to be executed within the Vivado HLS environment:

In the Vivado HLS IDE environment, the input image 'test.jpg' with dimensions of 345x230 pixels has been taken for image compression. Subsequently, declared a unit integer function with an image size of 1080x1920. To facilitate the compression processes, a memory copy function has been introduced. Within the memory copy function, the image was filtered using AXI-Stream and Inter-Pix, treating the input image as a bit stream with 4x4 rows and columns.

In the HLS dataflow, the 'S-axilite' HLS interface function has been employed to connect Vivado HLS and HLX environments. This created ports for both input and output images with 'm-axi' depth or offset values. The 'conv' function was used to convert the integer unit function into image format. Dataflow coefficients [i, j] with a size of [3][3], i.e., {-1,-2,-1}, {0, 0, 0}, {1, 2, 1} for loop operations has been applied.

The 'mat' function then compared HLS data flow with depth values against the image pixel values, initially set at 1080x1920. The 'AXI to mat' function has been, which involved defining source, destination, and anchor ports for storing input images, output images and filter operations. The result was a compressed output of 256x256 pixels stored in the anchor port. kernel function has been incorporated to eliminate noisy textures from the output of

compressed image. Throughout this process, 'C' simulation and 'C' synthesis took place in accordance with the specified values. During 'C' simulation and synthesis, the design blocks created in the 'HLX' environment has been interfaced. Following the successful execution of Vivado HLS and HLX tools, results of image compression as well as hardware simulations and latency reports has been viewed. The steps for proposed work are outlined below.

Sequential steps to be executed within the Vivado HLX environment:

In the Vivado HLX environment, creating architecture design blocks using VHDL code has been done. A 'Design Wrapper' was established, and VHDL code was written to define clock ports, a DDS compiler, I/O ports, AXI stream interfaces for input and output, PWM ports, HDMI ports for both receiving and transmitting data, LED & RGB ports, and software interfaces.

In this context, various ports such as clock, address, DDR, HDMI receiver, and some fixed ports were designated as 'in/out,' signifying their roles as input and output ports. HDMI transmitter and receiver data were specified as [2:0], LED outputs as [3:0], RGB outputs as [2:0], and software interfacing as [1:0]. The values for 'HDMI' transmitter and receiver were determined in accordance with the Digilent INC reference link, and these values were established prior to the block design.

Initially, the 'RGB to DVI' video encoder served as the source for the image pre-production process. A video timing controller was employed to manage image IN and image OUT timings. Following this, 'AXI video direct memory accesses' were employed to process AXI stream input and output data. The 'AXI stream subset converter' was responsible for converting 'AXI' values to 'Mat' values, specifically for compression with a size of [256x256]. The 'AXI Stream' handled the image IN and image OUT during the compression process, as per the values defined in the 'HLS' environment. A video timing controller was also utilized on the output side to monitor image IN and OUT timings. Finally, the 'DVB2RGB' image decoder (sink) was integrated into the design, taking into account reset and encoding values. Subsequently, external pin connections are established to ensure proper hardware interfacing. Following the design of architecture blocks, the workflow involves software simulation, synthesis, and implementation steps. Upon concluding the implementation process, the final step is the generation of the bitstream file. With these design values in place, the overall architecture can be visualized as follows:"

Procedure for Implementation of Arty z7 kit:

Upon generating the .bitstream file, access the Vivado HLX environment, open the Hardware Manager, and initiate an auto-connect by right-clicking. The kit number,

identified as [XC7Z020], will be displayed, enabling hardware connection to both Vivado IDE environments. LED lights will indicate successful hardware connection with a steady white color. Subsequently, initiate the 'Program device' process to load the .bitstream file onto the device, ensuring hardware interconnectivity across both Vivado environments.

Upon executing the image compression process, the hardware simulation output and image decompression implementation can be observed through LED lights blinking in RGB colors as per the programming instructions. In the Vivado HLS environment, you can view the output images from the compression, along with the hardware simulation results and latency reports. Meanwhile, the Vivado HLX environment provides access to design blocks, software simulation, synthesis, implementation, bitstream generation, and a project summary report. Screenshots of these outputs are presented in the next section.

4. Results and Discussions

Simulation Result

Name	Value	[999,999 ps]	[999,999 ps]
> WIDTH[31:0]	0000030	00000300	00000300
> HEIGHT[31:0]	0000020	00000200	00000200
> INFILE[67:0]	6b6f64696432532e696579	6b6f64696432532e696579	6b6f64696432532e696579
> START...[31:0]	0000006	00000064	00000064
> HSYNC...[1:0]	000000a	000000a0	000000a0
> VALUE[31:0]	0000006	00000064	00000064
> THRES...[1:0]	0000005	0000005a	0000005a
> SIGN[31:0]	0000000	00000000	00000000
> sizeOf...[31:0]	0000000	0000000e	0000000e
> sizeOf...[31:0]	0012000	00120000	00120000
> ST_IDLE[1:0]	0	0	0
> ST_VSY...[1:0]	1	1	1
> ST_HSY...[1:0]	2	2	2
> ST_DATA[1:0]	3	3	3

Fig 2 Simulation result using vivado for Arty z7 20kit

Synthesis Design

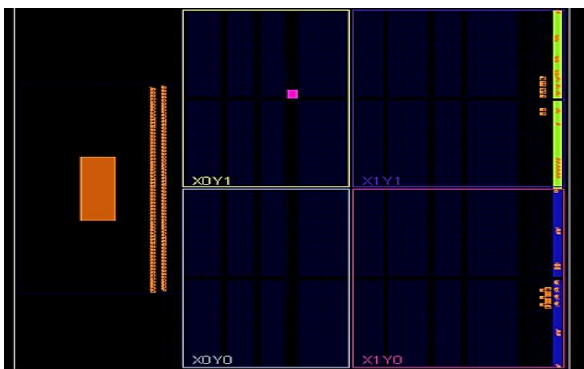


Fig 3 Synthesis result using vivado for Arty z7 20kit

Implementation Design

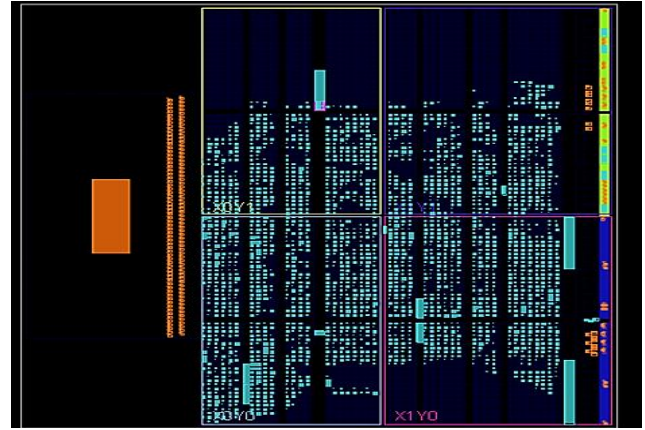


Fig 4 Implementation result using vivado for Arty z7 20 kit

Project summary report

Project Summary

Overview | Dashboard

Settings Edit

Project name: Arty-Z7-20-OOB
 Project location: D:/Arty-Z7-20-OOB-2018.2-1/Mivado_proj
 Product family: Zynq-7000
 Project part: xc7z020clg400-1
 Top module name: design_1_wrapper
 Target language: Verilog
 Simulator language: Mixed

Synthesis

Status: ✔ Complete
 Messages: ! 405 warnings
 Active run: synth_2
 Part: xc7z020clg400-1
 Strategy: Vivado Synthesis Defaults
 Report Strategy: Vivado Synthesis Default Reports

Power

Total On-Chip Power: **1.872 W**
 Junction Temperature: **46.6 °C**
 Thermal Margin: 38.4 °C (3.2 W)
 Effective θ_{JA} : 11.5 °C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low
 Implemented Power Report

Implementation Summary | Rout

Status: ✔ write_bitstream Complete!
 Messages: ! 1 critical warning
! 23 warnings
 Active run: impl_2
 Part: xc7z020clg400-1
 Strategy: Vivado Implementation Defaults
 Report Strategy: Vivado Implementation Default Reports
 Incremental implementation: None

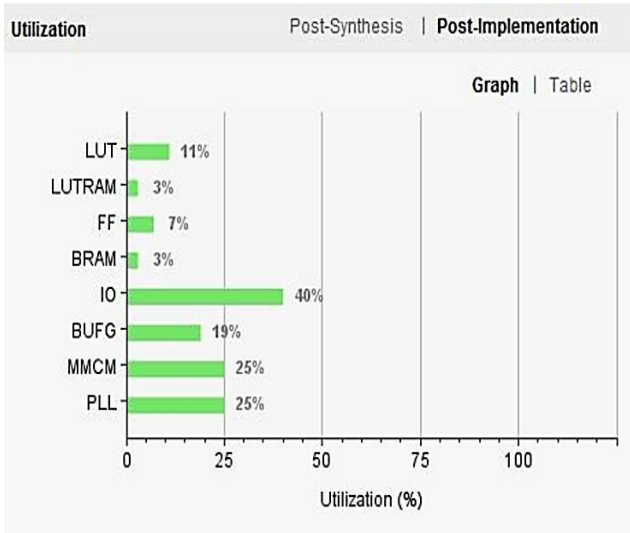


Fig 5 Project summary report and Bitstream generation using vivado for Arty z7 kit

Input image



Fig 6 Input image with the size of 345 x 230 Compressed output image

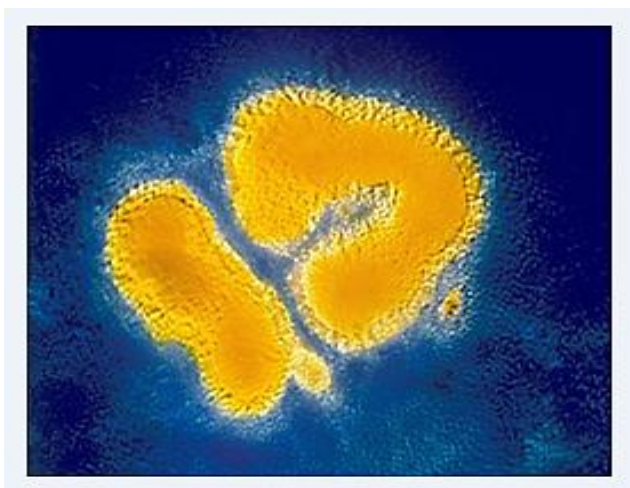


Fig 7 Compressed output image with the size of 256x256

Performance Parameters

Compression ratio = Uncompressed size / compressed size. Value of compression ratio obtained **5.61**. Percentage of compression level obtained is **89.5%**.

5. Conclusion

This paper demonstrates the successful achievement of efficient medical image compression through the Data Contrast Compression method, resulting in a noteworthy compression ratio. The implementation on the Arty Z7 kit was executed seamlessly, leading to the generation of a bitstream. Enhanced compression output was attained through the utilization of Vivado HLS and HLX environments. The paper includes screenshots illustrating software simulation, synthesis, implementation, hardware simulation, and synthesis, alongside a comprehensive project summary report. The successful interfacing of Vivado HLX and HLS with the Arty Z7-20 kit is detailed, and performance parameters have been evaluated.

References:

- [1] Marcin kowalczyk, Dominika przewlocka, Tomasz kryjak, Real time implementation of contextual image processing operations for 4k video stream in zynq ultrascale + MPSOC, 2018, 978-1-5386-8237-1/18, IEEE. DOI:10.1109/DASIP.2018.8597105
- [2] Saniya Zafar, Sobia Jangsher, Ouns Bouachir, Moayad Aleqaily, Jalel Ben Othman, Qos enhancement with deep learning based interface prediction in mobile IOT, computer communications 148(2019)86-97[Elsevier]. <https://zuscholars.zu.ac.ae/works/2857/>
- [3] Amir.Hajirassoulina, Andrew J.Tabenner, Martyn P.Nash, Poul M.F.Nielsen, Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms, Signal processing: image communication, 2018, 68(2018)101-119[Elsevier]. <https://doi.org/10.1016/j.image.2018.07.007>
- [4] Jia ZhaoYang et.al, Study on digital image inpainting method based on multispectral image decomposition synthesis, International journal of pattern recognition and Artificial intelligence, 2018[World scientific]. <https://doi.org/10.1142/S0218001419540041>
- [5] Biomedical Imaging and Sensing Conference", Proc. SPIE 11140, Biomedical Imaging and Sensing Conference, 1114001 (21 April 2019); <https://doi.org/10.1117/12.2535451>
- [6] Xin Zhong et.al., A High capacity reversible watermarking scheme based on shape decomposition for medical images, International journal of pattern recognition and Artificial intelligence, 2018[World scientific]. <https://doi.org/10.1142/S0218001419500010>
- [7] Kai zhu et.al., A New RSA image encryption

- algorithm based on singular value decomposition, *International journal of pattern recognition and Artificial intelligence*, 2018[World scientific]. <https://doi.org/10.1142/S0218001419540028>
- [8] Pranitha.K, Dr.G.Kavya, Data compression with high peak signal to noise ratio using Bisectional Cylindrical wavelet transform for a satellite image, *International journal of Engineering and Technology(UAE)*, 2018, volume 7, No (4.6). <https://doi.org/10.14419/ijet.v7i4.6.28648>
- [9] Ghislain takam Tchendjou, Emmanuel Simeu, Rshdee Alhakim, Fuzzy logic based objective image quality assessment with FPGA implementation, *Journal of system architecture*, 2017, doi: 10.1016/j.sysarc.2017.12.002.
- [10] Bhaumik vaidya, Mustafa surti, Parthvaghya siya, Jay Bordiya, Jenish Jain, Hardware acceleration of image processing algorithms using vivado high level synthesis tool, *International conference on intelligent computing and control systems (ICICCS 2017)*. 10.1109/ICCONS.2017.8250732
- [11] Ahmed Aliyu, et.al, Towards video streaming in IOT environments: Vehicular communication perspective, *computer communications*, S0140-3664(17)30512-1[Elsevier]. <http://doi.org/10.1016/j.comcom.2017.10.003>.
- [12] Ajay Rupani, Pawan Whig, Gajendra sujediya, Piyush vyas, A robust technique for image processing based on interfacing of raspberry-PI and FPGS using IOT, *International conference on computer, communications and Electronics (comptelix 2017)*. 10.1109/COMPTELIX.2017.8003992
- [13] Hemanth, D.J., Vieira Estrela, Deep Learning for Image Processing Applications , *Book of IOS Press*, December 2017. DOI:10.3233/978-1-61499-822-8
- [14] Pranitha.K, Dr.G.Kavya, Literature Survey Of Image Compression/Decompression Techniques For Telehealth applications in *Proceedings IEEE International Conference on Photonics and High Speed Optical Networks ICPHON 2018* pp. 63-68.
- [15] Pranitha.K, Dr.G.Kavya, A Systematic Method for Hardware Software Codesign using Vivado HLS, *International Journal of Recent Technology and Engineering (IJRTE)*, Volume-8 Issue-4, November 2019, PP 467-472. DOI:10.35940/ijrte.d7008.118419
- [16] R. Liu, M. Liu, D. Li, Y. Zhang, and Y. Zheng, "Fpga architecture for real-time ultra-high definition glasses-free 3d system," in *Computer Vision*, DOI:10.1007/978-981-10-7299-4_31.
- [17] A.cortes, I.velez, A.Irizar, High level synthesis using vivado HLS for zynq SOC: image processing case studies, 978-1-5090-4565-5/16, 2016, IEEE. DOI:10.1109/DCIS.2016.7845376
- [18] Leandro Y.Mano, et.al, Exploiting IoT technologies for enhancing health smart homes through patient identification and emotion recognition, *computer communications*, S0140-3664(16)30068-8[Elsevier]. DOI:10.1016/j.comcom.2016.03.010
- [19] Z. Shi, "Rapid Prototyping of an FPGA-Based Video Processing System," PhD thesis, Virginia Tech, 2016.
- [20] Pranitha.K, Dr.G.Kavya, Literature Survey Of Image Compression/Decompression Techniques For space and Telehealth applications, *oxidation communications*, book 2 volume 42(2019), pp 151 – 159. <https://www.researchgate.net/profile/Kavya-G/publication/336686746>
- [21] M. Nieto, A. Cortés, O. Otaegui, J. Arróspide, and L. Salgado, Real-time lane tracking using rao–blackwellized particle filter. *Journal of Real-Time Image Processing*, vol. 1, no. 1, 2016, pp. 179-191, DOI :10.1007/s11554–012–0315–0. DOI:10.1007/s11554-012-0315-0
- [22] H. M. Abdelgawad, M. Safar, and A. M. Wahba, High level synthesis of canny edge detection algorithm on zynq platform, *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2015, vol 9(1) pp 148–152. doi.org/10.5281/zenodo.1337955
- [23] Jinalkumar. D, Joshi et.al, Implementing a Cloud based Xilinx ISE FPGA Design Platform for Integrated Remote Labs, in *Proceedings of IEEE ICACCI 2015*, pp-533-537. DOI:10.1109/ICACCI.2015.7275663
- [24] Pranitha.K, Dr.G.Kavya, M.Arun Kumar, Implementation and Elaborated Block design for Zybo kit Using Vivado High Level Synthesis tool, *Test Engineering, and Management*, 2020, pp 14623-14629. <https://www.testmagazine.biz/index.php/testmagazine/article/view/9755>
- [25] Shivank. D, Zalak. D, Joshi et. al, Reconfigurable Image Processor using an FPGA-Raspberry pi Interface, *International Journal of Computer Applications Proceedings ICCT 2015*, pp 11-15. <https://www.ijcaonline.org/proceedings/icct2015/number5/22664-1565>
- [26] Haxhimusa.Y, The Structurally Optimal Dual Graph Pyramid and its Application in Image Partitioning, *Book of IOS Press*, May 2007. <http://hdl.handle.net/20.500.12708/22348>.

- [27] Pranitha. K, Dr.G.Kavya, M.Arun Kumar, “A Detailed Illustration of VLSI Block Design Implementation Process Using Vivado HLS and Arty kit”, Universal Journal of Electrical and Electronics Engineering, 7(3):201-208, 2020, DOI:10.13189/ujeee.2020.070304.