UNDERGRADUATE FINAL YEAR PROJECT REPORT

Department of Computer & Information System Engineering

NED University of Engineering and Technology

# Software-Defined Camera for Outdoor Surveillance Applications

**Group Number: 08** **Batch: 2018 – 2022**

**Group Member Names:**


| | |
|---|---|
| Soniya Shafi | CS-18133 |
| Iqra Irfan | CS-18123 |
| Misha Akram Baig | CS-18118 |
| Hoor Soomro | CS-18110 |


Approved by

…………………………………………………………………………………………………...

Ms Fakhra Aftab

Lecturer

Project Advisor

## Author's Declaration

We declare that we are the sole authors of this project. It is the actual copy of the project that was accepted by our advisor(s) including any necessary revisions. We also grant NED University of Engineering and Technology permission to reproduce and distribute electronic or paper copies of this project.

| | | | |
|---|---|---|---|
| Signature and Date | Signature and Date | Signature and Date | Signature and Date |
| ................................. | ................................. | ................................. | ................................. |
| Soniya Shafi | Iqra Irfan | Misha Akram Baig | Hoor Soomro |
| CS-18133 | CS-18123 | CS-18118 | CS-18110 |

aa9391438@gmail.com syedaiqrairfan950@gmail.com mishaakrambaig@gmail.com hoorksoomro2@gmail.com

## Statement of Contributions

This report was written collaboratively by all four group members, Soniya Shafi, Misha Akram, Iqra Irfan, and Hoor Sooro. This report is divided into four chapters. Whereas Group Leader Soniya Shafi writes the background information and chapter 2 covers the literature review, Iqra Irfan covers the aims and objectives, methodology, and report outline from chapter 1 as well as the table of contents and list of abbreviations. Misha Akram covers Chapter 3 which covers the methodology used for the proposed work, while Hoor Soomro covers Chapter 4 which covers the conclusion and recommended future work, as well as the pictorial table in Chapter 3 discussing methodology.

## Executive Summary

Traditional cameras are not flexible and also unaware of their environment. They cannot provide constant and standard image quality in every environment because of their fixed (non-programmable) internal architecture. Along with image quality, image pre-processing is also crucial. The images need to be pre-processed before they can be used for useful results. For this purpose, FPGAs are used which are programmable semiconductor devices that are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. As opposed to Application Specific Integrated Circuits (ASICs), where the device is custom built for the particular design, FPGAs can be programmed to the desired application or functionality requirements. Along with the FPGA, Omnivision OV5640 image sensor is used to 'preprocess' the camera input before the capture of images. After that, the images were examined in MATLAB, and specific parameters were determined. Brightness, color, saturation, sharpness, and luminance were among the criteria. The parameters were utilized to identify the most important ones, which could then be manipulated during pre-processing. The camera was put in a fixed position at one particular location and was programmed to generate a live feed and take photographs at regular intervals. Before using the FPGA board and a connected camera, camera equipped with sensor OV5640 was to be used for the purpose of environmental profiling which wasn't available, so after searching various resources, we found out that that the image sensor in Pi camera i.e. OV5647 was similar in many aspects.  Sensor parameters can be modified using command line interface. Parameter values range between 0 to 100 where 0 represented minimum and 100 maximum value. This is proof of concept that sensor state is modifiable and not fixed like traditional and conventional cameras. The implemented solution provides a way to realize a design on the FPGA board to pre-process the live video stream being captured in real-time. The proposed design for this can be extended with further image processing blocks and environment adaptable capturing features for further applications as well.

# Acknowledgments

[Write the acknowledgments here. Use single-spacing between lines. You can acknowledge any faculty member, any national or international organization, or sponsor for helping you out in completion of this project. They may have provided you valuable resources in the form of data, personal experience, disciplinary expertise, instruments, or technical support. Ideally it should be one or two short paragraphs.]

# Dedication [Optional]

[Write dedication here. You may want to dedicate this work to immediate family members or members of a board or society that works for your cause. Ideally, it should be only one or two lines in length. However, if you do not wish to include a dedication, delete this page. After deleting this page and updating the table of contents, this page will automatically disappear from the table of content. In case, you have decided to delete this page, the table of contents should begin from page v.]

# Table of Contents

## List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **FPGA** | Field Programmable Gate Array |
| **HLS** | High Level Synthesis |
| **IP** | Intellectual Property |
| **CLBs** | Configurable Logic Blocks |
| **ASICs** | Application Specific Integrated Circuits |
| **OTP** | One-Time Programmable |

## United Nations Sustainable Development Goals

The Sustainable Development Goals (SDGs) are the blueprint to achieve a better and more sustainable future for all. They address the global challenges we face, including poverty, inequality, climate change, environmental degradation, peace and justice. There is a total of 17 SDGs as mentioned below. Check the appropriate SDGs related to the project.

- ☐ No Poverty
- ☐ Zero Hunger
- ☐ Good Health and Well being
- ☐ Quality Education
- ☐ Gender Equality
- ☐ Clean Water and Sanitation
- ☐ Affordable and Clean Energy
- ☐ Decent Work and Economic Growth
- ☐ Industry, Innovation and Infrastructure
- ☐ Reduced Inequalities
- ☐ Sustainable Cities and Communities
- ☐ Responsible Consumption and Production
- ☐ Climate Action
- ☐ Life Below Water
- ☐ Life on Land
- ☐ Peace and Justice and Strong Institutions
- ☐ Partnerships to Achieve the Goals

# Similarity Index Report

Following students have compiled the final year report on the topic given below for partial fulfillment of the requirement for Bachelor's degree in Computer System Engineering.

**Project Title** **Software Defined Camera for Outdoor Survillance Applications**

| S. No. | Student Name | | Seat Number |
|---|---|---|---|
| | | | |
| 2. | | | |
| 3. | | | |
| 4. | | | |

This is to certify that Plagiarism test was conducted on complete report, and overall similarity index was found to be less than 20%, with maximum 5% from single source, as required.

Signature and Date

..................................

[Name of Advisor]

[Plagiarism Test must be conducted on the project report by the project advisor using Turnitin Software, or any other software approved by the university. Project Advisor is required to sign the Plagiarism Certificate if the similarity is found as per rules defined in the guidelines. Plagiarism report must be included (only the first and last page) generated by Turnitin. A sample of the plagiarism test report is provided with the FYDP template package.]

[Attach here the first and last page of the plagiarism report generated by Turnitin.]

# Chapter 1
# Introduction

## 1.1 Background Information

### 1.1.1 Field Programmable Gate Array -FPGA

FPGAs are programmable semiconductor devices that are based around a matrix of Configurable Logic Blocks (CLBs) connected through programmable interconnects. As opposed to Application Specific Integrated Circuits (ASICs), where the device is custom built for the particular design, FPGAs can be programmed to the desired application or functionality requirements. Although One-Time Programmable (OTP) FPGAs are available, the dominant type is SRAM-based which can be reprogrammed as the design evolves. Due to this programmability, FPGAs are ideal for a large variety of markets such as ASIC prototyping, Aerospace and Defense, Automotive, Communications, High Performance Computing, Industrial, Medical and Video and Image Processing.

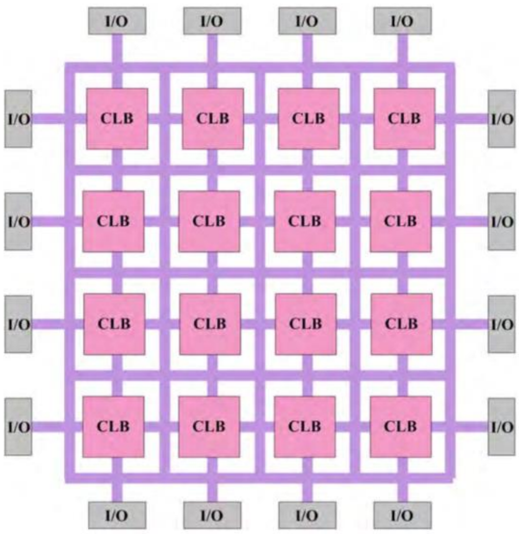



Figure 1.1 Overview of FPGA architecture

### 1.1.2 Image Sensor

Our goal was to identify a programmable image sensor that will provide a way to 'preprocess' the camera input before the capture of images. For this reason, Omnivision OV5647 image sensor is used.

## 1.2 Significance and Motivation

All the computer vision algorithms that perform image analysis and processing require high quality images. The conventional cameras available now a days have fixed internal architecture i.e. they are not programmable; hence not aware of their environment and non-adaptive. These cameras can't provide a constant and standard image quality in different scenarios.

All real time applications need to process data as fast as possible; the images need to be pre-processed as well before they can be used to generate useful results according to the requirements of an application. If the environment diverges from normal conditions or noise is generated, conventional cameras fail to maintain the standard image quality. As a result, more computationally intensive & time-consuming algorithms will be required to generate clean and clear images before using them in computer vision applications.

This leads to the need for a software-defined camera; where imaging sensor inside camera is adaptive to respond environment. Therefore, processing the images directly on the camera, taking advantage of the sensors to become environmentally conscious. Such a camera takes over the load of pre-processing images. The strategy is to make our camera smart enough so that it can sense its surroundings and noise, adjusts its internal hardware and select most suitable parameters for the situation even before the image is captured. In this way, the quality of image will be constant. However, in a practical scenario, some of the images might be affected but there would not be in depth data loss hence data recovery will be possible.

This project provide ease to customers of any environment varying image processing application [9] e.g.

- In surveillance systems, as it is used to recognize the suspected person or vehicle from a still image or video.
- In airports, automated systems could be used to check luggage.
- In biometric recognition, to identify an individual based on some biological feature such as eye, ear, fingerprint, etc.

## 1.3 Aims and Objectives

- To study and determine exact parameters which can be manipulated to produce high quality images.
- To integrate different models of the environment for an efficient solution.
- To exploit the internal architecture of the image sensor.
- To develop image pre-processing IP cores and implement a solution that will integrate all the cores.
- To test the software-defined camera for outdoor surveillance applications under various environmental conditions.

## 1.4 Methodology

Our project's aim is to make camera smart enough, such that it produces high quality images in varying environment. The methodology we will use to achieve our target comprise of:

A. **Environmental Profiling:**
Conventional cameras will be used for capturing images in different environments at regular intervals. These images help in determining the most varying parameters.

B. **Image Sensor Integration:**
OV5640 camera module will be connected to the FPGA board and a set of open-source Vivado IP cores will be used for its configuration in software.

C. **Image Processing Cores using Vivado HLS:**
Vivado High Level Synthesis will be used for generation of different soft IP cores

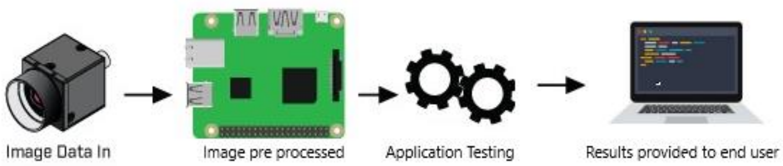



Figure 1.2 Working Methodology smart camera

## 1.5 Report Outline

Now a day's image processing has become very powerful tool in the field of medical imaging, digital photography, video surveillance etc. Image processing usually requires very large number of operations and high-speed data transfer, therefore parallel processing or multiprocessing hardware is essential. Because of this, FPGA is one of the best alternatives for image processing as it performs the operations in parallel fashion.

This report is divided in three main Chapters, each one of those includes smaller sections and possibly subsections.

Chapter 2 provides the literature review useful to understand the scope of this project. At first, it discussed the need of software defined camera. It also offers information about the benefit of edge computing. Finally, it gives brief overview about HLS.

Chapter 3 begins with a brief introduction about the methodology adopted. It further delivers the parameters obtain from environmental profiling.

Chapter 4 summarizes the results generated along with the project summary. Finally, it provides some future improvements and potential work.

# Chapter 2
# Literature Review

## 2.1 Introduction

Traditional cameras are not flexible and also unaware of their environment. They cannot provide constant and standard image quality in every environment because of their fixed (non-programmable) internal architecture. Along with image quality, image pre-processing is also crucial. The images need to be pre-processed before they can be used to generate useful results. Therefore, ample amount of work done only to improve image quality and making pre-processing faster.

The authors of [1] use Field Programmable Gate Array (FPGA) architecture for making pre-processing faster. As the FPGA architecture has the ability to perform parallel processing, it will shorten the processing time and the efficiency will increase.

## 2.2 Edge Computing

The pre-processing generally occurs at server side. Server solution is simple and effective but it is costly and time consuming. Edge computing eliminates the need to send image data to server-side for processing. Provides a way to process the images directly on the camera by taking advantage of sensors and the use of end devices to take over the load of processing [3].

In [4] the detection and recognition tasks for surveillance are executed locally by edge devices. Only when devices are not able to execute the recognition task, a recognition request is sent to the server.

## 2.3 High Level Synthesis

The properties of FPGA (Field Programmable Gate Array) make them a desired technology to implement digital image and video processing. HLS (High Level Synthesis) Vivado is one of the most used tools which can directly transform a C description to a hardware IP, described on Verilog or VHDL. FPGA reconfigurable circuit with its HLS software tools is become a very promising technology to be widely used in many applications encompassing all aspects and requirements of embedded system [5].

In [6] authors state that, escalating system-on-chip design complexity is pushing the design community to raise the level of abstraction beyond register transfer level. Despite the unsuccessful adoptions of early generations of commercial high-level synthesis (HLS) systems, we believe that the tipping point for transitioning to HLS methodology is happening now, especially for field-programmable gate array (FPGA) designs. The latest generation of HLS tools has made significant progress in providing wide language coverage and robust compilation technology, platform-based modeling, advancement in core HLS algorithms, and a domain-specific approach.

## 2.4 Summary

Image processing applications are widely used nowadays which need to analyze and process a large number of images and video streams in real-time. This real-time constraint can be handled if the live camera feed is preprocessed right at the edge device i.e. an FPGA. The FPGA coupled with an image sensor is used to develop a camera that is adaptive and can preprocess the incoming live camera feed using high-level synthesis.

# Chapter 3

# Methodology

## 3.1 Introduction

In this chapter we will focus on the methodology of our work; the environmental profiling is done in order to determine the relevant parameters i.e. brightness, contrast etc. which can be later manipulated in preprocessing of the video stream captured by the image sensor used later.

## 3.2 Environmental Profiling

By using this approach various environments were chosen. With one particular location, the camera was placed in a fixed position and was made to generate live stream and capture the images at regular intervals. These images were then analyzed in MATLAB and their certain parameters were calculated. These image parameters comprised of brightness, hue, saturation, sharpness, and luminance. Studying these parameters and their change that result in images of varying degree of quality helped determine the most relevant ones which can be later manipulated in pre-processing of the video stream captured by the image sensor used later.

| Image # | Time | Mean Brightness | Median Brightness | Mean Saturation | Median Saturation | Mean Hue | Median Hue | Sharpness | Mean Luminance | Median Luminance |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6:32:00 | 0.550520803 | 0.584313725 | 0.142395522 | 0.070588235 | 0.298135255 | 0.273809524 | 5.949145965 | 136.5677421 | 145 |
| 2 | 6:33:00 | 0.567795377 | 0.603921569 | 0.157405229 | 0.077348066 | 0.296065907 | 0.272222222 | 6.092881948 | 140.4469348 | 149 |
| 3 | 6:34:00 | 0.580999818 | 0.619607843 | 0.145564895 | 0.069651741 | 0.299879782 | 0.277777778 | 6.612048545 | 143.9801842 | 153 |
| 4 | 6:35:00 | 0.595612374 | 0.635294118 | 0.145356 | 0.068571429 | 0.299151288 | 0.277777778 | 6.991325781 | 147.555352 | 158 |
| 5 | 6:36:00 | 0.592914617 | 0.631372549 | 0.143014786 | 0.067669173 | 0.300115478 | 0.277777778 | 6.995509533 | 146.9746856 | 157 |
| 6 | 6:37:00 | 0.603293894 | 0.643137255 | 0.135822801 | 0.068627451 | 0.31015093 | 0.291666667 | 7.136078657 | 149.5317785 | 159 |
| 7 | 6:38:00 | 0.59942015 | 0.639215686 | 0.136748751 | 0.065789474 | 0.303042577 | 0.277777778 | 7.633507411 | 148.7260745 | 158 |
| 8 | 6:39:00 | 0.601251116 | 0.639215686 | 0.136862795 | 0.066176471 | 0.299531191 | 0.270833333 | 7.673207469 | 149.1829456 | 159 |
| 9 | 6:40:00 | 0.601856553 | 0.639215686 | 0.136285214 | 0.065989848 | 0.296831473 | 0.270833333 | 7.64822469 | 149.3608073 | 159 |
| 10 | 6:41:00 | 0.60081587 | 0.639215686 | 0.135705359 | 0.066666667 | 0.295252254 | 0.269230769 | 7.655987238 | 149.0937587 | 159 |
| 568 | 15:59:00 | 0.53083274 | 0.564705882 | 0.175689592 | 0.12804878 | 0.348272606 | 0.321428571 | 8.165185715 | 128.7013513 | 135 |
| 569 | 16:00:00 | 0.531632546 | 0.564705882 | 0.170819297 | 0.124324324 | 0.349545399 | 0.333333333 | 7.77066837 | 129.0550019 | 136 |
| 570 | 16:01:00 | 0.537557534 | 0.57254902 | 0.170660854 | 0.123076923 | 0.345973425 | 0.3125 | 8.338961885 | 130.5605199 | 138 |
| 571 | 16:02:00 | 0.538943964 | 0.57254902 | 0.170912778 | 0.123188406 | 0.345447643 | 0.3125 | 8.279745209 | 130.8921528 | 138 |
| 572 | 16:03:00 | 0.537760061 | 0.57254902 | 0.1718341 | 0.125 | 0.346854277 | 0.316666667 | 8.396601902 | 130.515653 | 138 |
| 573 | 16:04:00 | 0.530813684 | 0.560784314 | 0.176561741 | 0.125827815 | 0.343021836 | 0.305555556 | 7.642751882 | 128.8661748 | 134 |
| 574 | 16:05:00 | 0.532722302 | 0.564705882 | 0.174071373 | 0.12568306 | 0.347890765 | 0.321428571 | 8.044829986 | 129.1828665 | 136 |
| 575 | 16:06:00 | 0.529400766 | 0.564705882 | 0.174009919 | 0.126760563 | 0.346495061 | 0.3125 | 8.188513509 | 128.3808111 | 135 |
| 576 | 16:07:00 | 0.527202796 | 0.560784314 | 0.171836395 | 0.126373626 | 0.347144191 | 0.314814815 | 7.968723546 | 127.937907 | 135 |

Figure 3.1 Environment profiling using Matlab

### 3.3 Camera Analysis

Before using the FPGA board and a connected camera, camera equipped with sensor OV5640 was to be used for the purpose of environmental profiling which wasn't available, so after searching various resources, we found out that that the image sensor in Pi camera i.e. OV5647 was similar in many aspects. So, for proof of concept, we started with available resources in order to fulfill our objective, using this camera sensor integrated with raspberry Pi board we tried to access, read and modify the internal registers states by varying the sensor parameters of the sensor, by exploring different possibilities using protocols like UART and I2C, but due to OS and library specific restrictions register level information of the image sensor could not be made accessible. After thorough analysis and research, we found out that camera sensor parameters can be modified using command line interface, and that the parameter values ranged between 0 to 100 where 0 represented minimum and 100 maximum value which gave us proof of concept that sensor state is modifiable and is not fixed like traditional and conventional cameras.

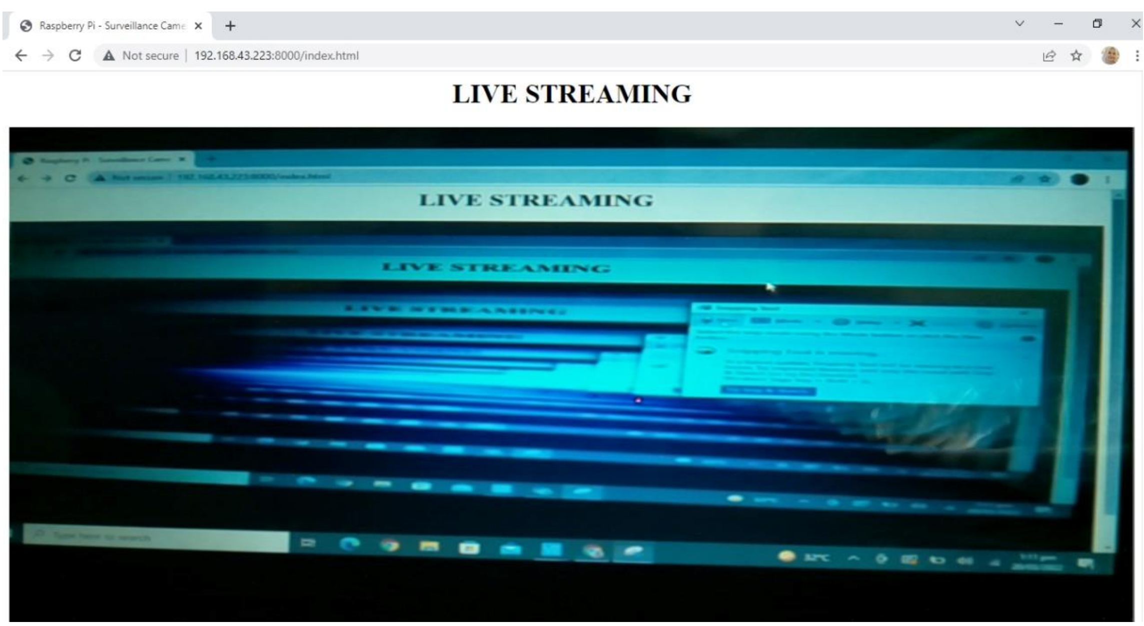
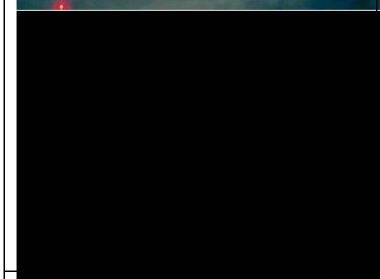
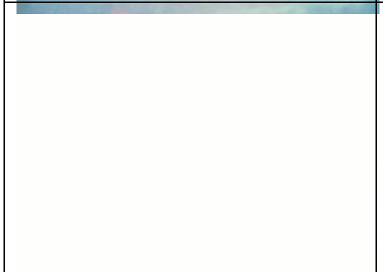
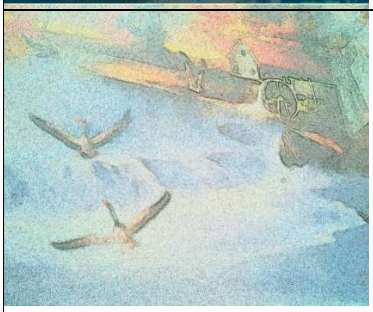



Figure 3.2 Live stream generated through raspberry pi that can be remotely accessed

Table 2.1 Different images obtained by changing sensor modes and observing their effects.

| Image Obtained | Sensor Mode | Effect Observed |
|---|---|---|
|  | original | No change |
|  | Zero brightness | Unclear |
|  | Increased brightness | More clear |
|  | Full brightness | Unclear |
|  | Increased contrast | Clear image |

| | | |
|---|---|---|
|  | Decreased contrast | Unclear |
|  | Color swap | Colored image |
|  | Negative | Unclear |
|  | Exposure | Unclear |
|  | Saturation decreased | Unclear |

# Chapter 4

# Conclusions

## 4.1 Summary

Image processing applications are widely employed nowadays in situations where a high number of images and video streams must be analyzed and processed in real time. This real-time constraint can be overcome by preprocessing the live camera feed right at the edge device, such as an FPGA. The FPGA is used in combination with an image sensor to create an adaptive camera that can use high-level synthesis to preprocess the incoming live video feed.

## 4.2 Recommendations for Future Work

In order to configure registers according to environmental conditions IPs will be designed on FPGA. Testing and validation will be done with respect to each aspect to ensure correctness of the product.

# Appendix A
# [Title of Appendix]

Include appendices, if applicable. Appendices should include the information that is not the primary part of the main body of the project report. This means that if this information is removed from the main body, it would not negatively affect the flow of ideas in the main body of the report. For example, it may include long computer programming codes while keeping a flow chart of the code in the main body of the report. It may also include lengthy numerical data while keeping their graphs in the main body of the report.

If the project report has only one appendix, then the label of this chapter should be "Appendix" (not "Appendix A"). To cite an appendix in the text, write full title of the appendix that is "Appendix A", "Appendix B", etc. Each appendix should start from a new page and should also include a short description of what is included in that appendix.

# References

[1] M. Yildirim and A. Çinar, "Simultaneously Realization of Image Enhancement Techniques on Real-Time Fpga," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2019, pp. 1-6

[2] D. Tsiktsiris, D. Ziouzios and M. Dasygenis, "HLS Accelerated Noise Reduction Approach Using Image Stacking on Xilinx PYNQ," 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 2019, pp. 1-4, doi: 10.1109/MOCAST.2019.8741574

[3] Ahmed, E., Ahmed, A., Yaqoob, I., Shuja, J., Gani, A., Imran, M. and Shoaib, M. (2017). Bringing Computation Closer toward the User Network: Is Edge Computing the Solution?. IEEE Communications Magazine, 55(11), pp.138-144.

[4] H. Kavalionak, C. Gennaro, G. Amato, C. Vairo, C. Perciante, C. Meghini and F. Falchi, "Distributed video surveillance using smart cameras," Journal of Grid Computing, 17(1), 2019

[5] M. S. AZZAZ, A. MAALI, R. KAIBOU, I. KAKOUCHE, M. SAAD and H. HAMIL, "FPGA HW/SW Codesign Approach for Real-time Image Processing Using HLS," 020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP), EL OUED, Algeria, 2020, pp. 169-174, doi: 10.1109/CCSSP49278.2020.9151686.

[6] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers and Z. Zhang, "High-Level Synthesis for FPGAs: From Prototyping to Deployment," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, no. 4, pp. 473-491, April 2011, doi: 10.1109/TCAD.2011.2110592.

[7] S. Lahti, P. Sjövall, J. Vanne and T. D. Hämäläinen, "Are We There Yet? A Study on the State of High-Level Synthesis," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 5, pp. 898-911, May 2019, doi: 10.1109/TCAD.2018.2834439.

[8] O'Loughlin, Declan & Coffey, A. & Callaly, F. & Lyons, D. & Morgan, F.. (2014). Xilinx Vivado High Level Synthesis: Case studies. 352-356

[9] Bansal, M., Kumar, M. & Kumar, M. 2D Object Recognition Techniques: State-ofthe-Art Work. Arch Computat Methods Eng (2020).

https://doi.org/10.1007/s11831- 020-09409-1

## Glossary [Optional]

| **Term** | **Definition** | [Page at which it first appeared] |
|---|---|---|
| **Azimuth** | *angle between North, measured clockwise around the observer's horizon and sun* | Pg. 1 |
| **Zenith** | *an imaginary point directly above the observer/system, complement of altitude angle* | Pg |