Video Pipeline for Machine Computer Vision

DESIGN DOCUMENT

Team Number: sdmay25-01
Team Name: Project ELM
Advisors: Dr. Zambreno
Client: JR Spidell

Team Members:
James Minardi — Hardware Integration
Eli Ripperda — Embedded Systems
Lindsey Wessel — ML Face & Eye Detection
Mason Inman — Semantic Segmentation Optimization

Contact Information:
sdmay25-01@iastate.edu
https://sdmay25-01.sd.ece.iastate.edu/

Executive Summary

Individuals with mobility and cognitive impairments, including those with Cerebral Palsy, often face serious obstacles to independence and safety due to the limitations of traditional wheelchairs. These devices typically lack integrated technologies that support autonomy, communication, and timely medical response. Caregivers and healthcare professionals are further burdened by the absence of real-time alerts for critical events like seizures, resulting in delayed interventions and increased risk.

To address these challenges, our client is leading the development of a next-generation assistive wheelchair equipped with advanced mobility features and real-time seizure detection capabilities. The goal is to enhance user independence, improve safety, and alleviate caregiver workload. In support of this initiative, our team has developed an edge-computing subsystem that enables real-time pupil tracking, laying the foundation for future enhancements to the client's innovative solution. Our solution adheres to our client's requirements of low latency, modularity, and specific technology requirements such as an Ultra96v2 development board and MIPI camera sensor, and transition documents to pass onto future engineering teams.

Please note that our team is operating under a Non-Disclosure Agreement (NDA) with our client. As a result, some technical details may appear limited in this document, but we've taken care in presenting the project clearly while respecting the confidentiality terms set by our client.

Learning Summary

Development Standards & Practices Used

- Version Control
- Hardware Abstraction
- Product verification and validation
- Agile project development – meeting with client on a weekly basis, adjusting our goals according to progression the past week and what serves the client the best.

Summary of Requirements

- Accurate Eye tracking
- Real-Time Performance
- Hardware Constraints
- Development & Runtime Tools
- System Output
- Documentation & Handoff

Applicable Courses from Iowa State University Curriculum

- CprE 288, CprE 381, ComS 474, CprE 487

New Skills/Knowledge Acquired That was not Taught in Courses

- Containerization (especially using Docker)
- Exposure to the open-source framework, PyNQ, for embedded python development
- Conflict resolution between opposing stakeholder interests
- Understanding and applying AI principles including industry standard libraries

Table of Contents:

Figures

Glossary

Term	Definition
ML	Machine Learning
SS	Semantic Segmentation
CNN	Convolutional Neural Network
VART	Vitis AI Runtime
ROI	Region of Interest

1. Introduction

1.1 Problem Statement

People with mobility and cognitive impairments, such as Cerebral Palsy, face significant challenges in maintaining independence and safety. Traditional wheelchairs lack the advanced technologies needed to support these users, leaving gaps in autonomy, communication, and safety. Healthcare professionals and caregivers also struggle with the absence of real-time alerts for medical emergencies like seizures, increasing the risk of delayed responses. These challenges not only affect the quality of life for wheelchair-bound individuals but also limit opportunities for proactive care.

Our client seeks to address these issues by developing assistive wheelchair technologies with features such as advanced mobility assistance and real-time seizure detection. This system is meant to increase wheelchair user autonomy, improve safety, and reduce caregiver stress. Our team is collaborating with the client to develop a subsystem that detects, locates, and presents information on the user's eyes in real time that will be used in future iterations of the client's project.

1.2 Intended Users

The primary users of our client's assistive wheelchair technologies will be wheelchair-bound individuals, such as those with Cerebral Palsy, along with their caregivers and healthcare providers. However, the specific subsystem we are developing is designed to detect and locate the user's eyes in real time. It will primarily serve as a foundational tool for our client and future engineering teams, while providing high modularity and performance needed for our client's long-term goal of developing assistive wheelchair technology.

1.2.1 Client

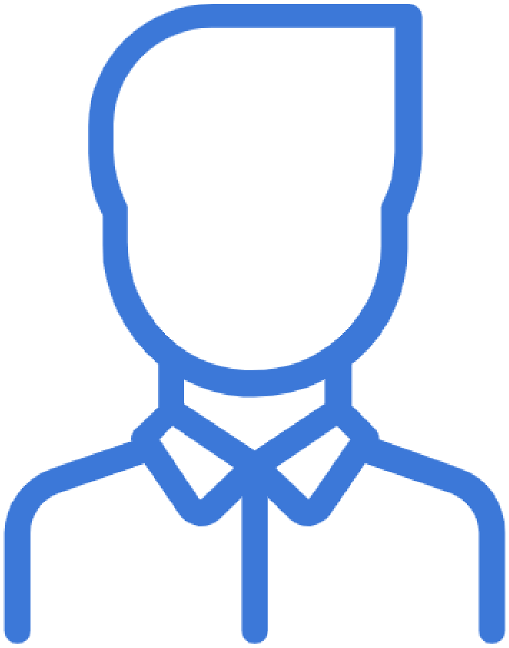


Figure 1.1 Client Picture

User Description

Our client, JR Spidell, is an experienced software engineer who formerly volunteered to help individuals with cerebral palsy. They are the primary source of project requirements and hold the high-level vision of the project.

Empathy Map

Due to his career as a principal software engineer at Collins Aerospace, our client lacks the free time necessary to create and design this embedded system. He is passionate about this project after a first hand experience supporting individuals with Cerebral Palsy while volunteering at a hospital. This opportunity made him more empathetic and knowledgeable about the daily struggles of living with mobility and cognitive impairment. Naturally, being an engineer, he has many technical thoughts and questions he poses to the group in regard to our system. He works with several college student groups across the United States, which also means he is passionate about helping students learn from him.

Key Characteristics

- Balancing their professional commitments as an engineering and student mentor limits the time dedicated to hands-on project work.
- The client has extensive experience in software engineering, giving him a deep understanding of system design.
- The client is passionate about student development opportunities to teach and guide future engineers.
- The client's past volunteer experience helping individuals with cerebral palsy gives them personal motivation to ensure his assistive technologies project is practical and effective.

User Needs

- The client needs a way to help people with cerebral palsy because he sympathizes with their challenges.
- The client needs a way to turn their high-level vision into a functional product because he has limited time to develop the embedded system themself.
- The client needs a way to check if all functional and non-functional requirements are met because system performance and reliability are critical for the success of the assistive technology.

Connection to Project

The client is the sole reason this project is in existence. While there are some requirements from ISU and our professors, the client holds the primary responsibility and ownership of the project requirements. He also holds the vision for the project, a strong technical background, and first-hand experience with the end goal. He possesses all the key components to be the project owner, of which our project is a sub-project.

1.2.2 Future Engineering Teams

Figure 1.2 Image of Iowa State University

User Description

This is a long-term project with many senior design teams contributing in the past decade, and many more teams in the future. Our team's starting point for this project picks up some elements of the last Iowa State University team. Similarly, the progress our team makes will benefit the next team to follow.

Empathy Map

These future engineering teams may be taking multiple classes, working multiple jobs, and being away from their home countries and/or families. They are in the last stretch of their undergraduate career and look to finish well. They will spend a lot of time trying to understand the wide scope of this project. They will not understand their specific goals at the beginning of their project and will have a steep learning curve to understand the technology they will be working on. These engineers will likely be connected to this project because they indicated some desire to work on it. To understand the technology, they might not be excited about investing extra time into the project when our team can easily provide good resources for them to efficiently learn from.

Key Characteristics

- Future team members will work on the project while balancing multiple academic and personal responsibilities as seniors.
- Future teams will be able to understand technical documentation and engineering concepts, which is important for benefiting from our documentation.
- They will appreciate clear documentation to help reduce the learning curve and focus on the project's development.
- They will heavily rely on the handoff process to maintain continuity and pick up when the previous team leaves.
- Future teams will have the same client, meeting with them for feedback and direction.

User Needs

- Future teams need a way to learn how to interface with and develop the technology to accomplish their goal.
- Future teams need a way to easily pick up a comprehensive project and understand project requirements to finish their own senior design project.
- Future teams need clear documentation and well-organized codebases to work from to ensure continuity between teams.

Connection to Project

This group, and possibly multiple groups, will directly work on the project. They will reference this project for their in-class assignments and will work with the same client that our team is currently working with.

1.2.3 Our team

User Description

As seniors in college working towards graduate school and full-time jobs, our team seeks real project development opportunities. This enables them to apply project management, organization, teamwork, and technical skills to a successful project that mimics the work life and expectations of a full-time job. As team members are majoring in Software Engineering or Computer Engineering, the team has a variety of skill sets.

Empathy Map

Communicating with advisors, the client, team members, and all others involved, our team will have to cipher through many messages from many different sources. Additionally, other responsibilities in individuals' lives (part-time jobs, full-time course loads, family, etc.) are an ongoing challenge team members must handle. Being young engineers, team members strive to learn new skills and solve problems, especially in their respective fields. Our team also has to learn from previous teams and participate in a handoff process. Taking on a large project can be overwhelming, and it's important to split it up into workable sections.

Key Characteristics

- Team members balance full-time course loads, part-time jobs, and personal responsibilities while contributing to this project.
- Each member brings specialized knowledge that adds value to the project in fields such as embedded systems, machine learning, and computer graphics.
- The team is motivated to grain hands-on experience in this project to prepare for post-graduation employment.

User Needs

- The team needs a way to gain relevant experience to build a strong foundation for our future careers.
- Our team needs a way to deliver a system that future teams can iterate on, because the system must seamlessly integrate into the client's broader assistive technologies vision.

Connection to Project

Our team plays an important part in creating this subsystem; we are responsible for communicating, facilitating, and developing all aspects of the project. With a focus on education and preparing for future success, the team is motivated to dedicate our time to make the best project possible. Before coming together, each team member expressed interest in this specific project, and now, our combined skills will make the project come to life.

2. Requirements, Constraints & Standards

2.1 Requirements & Constraints

2.1.1 Functional Requirements

Our system must:

- Capture real-time video of the user
- Detect and locate the user’s eyes
- Segment the pupil from the user’s eye
- Output relevant results to a display

2.1.2 Non-Functional Requirements

Modularity

The system must be modular such that each subsystem can be independently tested and interchanged. This allows for easier debugging, testing, and replacement of subsystems without affecting the rest of the system. Additionally, our client and future engineering teams will have a clean and maintainable foundation to build upon.

Accuracy

The system must reliably detect and track the user’s pupils. Inaccurate results would render the system useless. For the client’s goal of real-time seizure detection, false positives or missed signals could affect user safety. Therefore, accuracy directly impacts the reliability and safety of the assistive technology that the client envisions.

Frames Per Second

While extremely high frames-per-second (FPS) is required for tracking the saccades movement of the eye needed for seizure detection, our client’s requirement is to reach the highest frame rate we can within the constraints of our system. This involves optimizing our processing algorithms and image capture from the IMX219 camera sensor.

2.1.2 Constraints

The majority of our constraints are confined to the use of certain hardware and software frameworks requested by the client described below.

Open-Source Semantic Segmentation Model

In order to run the real-time system, the team is utilizing an award-winning open-source semantic segmentation model to optimize and target for our hardware platform. The model utilizes Pytorch, which is indirectly another constraint for our optimized versions of the model.

PYNQ Framework

PYNQ is an open-source project that provides a Python-based framework for using Xilinx platforms, making it easier to communicate between the programmable logic and processing system through Python APIs and Jupyter notebooks. Its use also aligns with prior teams' work and client expectations, smoothing continuity across past and future development.

Vitis AI & DPU-PYNQ

Vitis AI is Xilinx's official development platform for running AI inference on Xilinx hardware such as the Ultra96v2, and DPU-PYNQ is an open-source tool that simplifies integration with the PYNQ operating system. These tools allow us to perform model optimizations, such as pruning, and quantization while targeting the supported deep processing units (DPUs) on our hardware platform.

Ultra96v2 Development Board

The system must run on an Ultra96v2 development board allowing the system to be easily replicated and commercially available. The board is the central processing unit for the system that runs our eye detection algorithm and semantic segmentation model. It features a Xilinx Zynq Ultrascale+ MPSoC, which includes both ARM processing cores and an FPGA ideal for parallel processing and acceleration.

IMX219 Camera Sensor

The Sony IMX219PQH5-C (IMX219) camera sensor is used to capture video of the user's eyes for our real-time system. With its 8MP sensor, it's capable of 47 FPS at 1080p and even higher at lower resolutions.

2.1.3 Transitional Requirements

Documentation

As a non-functional requirement, the client has requested supportive documentation of system design and research completed by our team. This is to support current and future teams working on the project and aid in the handoff process. Documentation includes slide decks over system frameworks to be used within the system, environment and package management, and the system design itself. Along with presentations, traditional documents will be used for note taking and resource gathering.

Aid in Project Handoff

This is an additional non-functional requirement requested by our client. Our team is the 22nd Senior Design team to work with our client, so we are building upon what was done before us. Many teams will follow us, so we will help the following semester teams get up to speed by having meetings, sharing notes and designs, and to answer questions.

2.2 Engineering Standards

Engineering standards are important because they give everyone a common set of rules for designing and maintaining systems. They help with reliability, safety, and quality for all disciplines and solutions in order to avoid risk and comply with regulations.

2.2.1 IEEE 1016-2009

Information Technology – Systems Design – Software Design Descriptions

Description & Purpose

This standard defines the structure and content of Software Design Descriptions (SDDs). These are used to document and communicate software design information to clients and other stakeholders. It is meant to ensure that designs are well documented for leading code development and "reverse engineering" existing software.

Relevance

SDDs are highly relevant to this project, ensuring that our design process is well-documented and transparent. Since our group is focussed on assistive technologies, an SDD is crucial for communicating design choices to our client, advisors, and each other as team members. Since this standard applies to both high level and low level design descriptions, it would be useful for all stages in the project, from high-level designs to implementation details.

Modifications to Design

Implementing this standard will benefit the project with better communication, traceability, and transparency in our design process. With a well-structured SDD that has all aspects of the standard including system architecture and interfaces, we can make sure all our stakeholders have a good understanding of our project's design.

2.2.2 ISO/IEC TS 4213:2022

Information Technology — Artificial Intelligence — Assessment of Machine Learning Classification Performance

Description & Purpose

This standard details several methodologies of machine learning, best practices, and statistical testing strategies to measure performance. Additionally, it outlines common terminology used in the machine learning classification space. It provides a foundation for accurately assessing the effectiveness of machine learning models in classification tasks, which is critical for validating optimizations.

Relevance

This standard directly correlates to one of our client's requirements for needing statistics to validate our optimizations. Since we are focusing on AI-driven assistive technologies, applying the best practices in performance evaluation will ensure our models meet quality standards. It is particularly important for testing the classification accuracy of our machine learning solutions.

Modifications to Design

Incorporating this standard will ensure a more robust statistical foundation when reporting performance metrics. Adopting the terminology and methodologies from this standard will lead to consistency across team discussions and with our client, ensuring that the machine learning model's optimizations are well-documented and supported by empirical data.

2.2.3 ISO/IEC 19776-3:2015

Information Technology — Computer Graphics, Image Processing, and Environmental Data Representation — Extensible 3D (X3D) Encodings Part 3: Compressed Binary Encoding

Description & Purpose

This standard describes how to encode X3D files in a compressed binary format. It covers the compression of large datasets generated by 3D graphics, which are often required for real-time rendering in dynamic environments. The document outlines the tools, techniques, and methodologies used to compress these files while maintaining the integrity of the 3D data.

Relevance

As we will be working with image processing and optimizing for high frame rates (FPS), reducing data throughput is critical. Using this standard allows us to efficiently compress 3D data, meeting both our FPS and optimization requirements. It ensures that we are using best practices for managing large datasets, which is directly aligned with the performance goals of our project. While this can provide vital insight and knowledge to our project, fully implementing this standard in our project would be unnecessary, since there will be no X3D files specifically.

Modifications to Design

Implementing this standard will allow us to handle 3D data more efficiently, particularly when dealing with image processing and rendering. With reduced output channels and optimized binary encoding, we can meet our performance requirements without sacrificing the quality of the visual outputs. The standard will ensure our design is optimized for both speed and data throughput.

3. Project Plan

3.1 Overview

Our team has subdivided the project into Hardware Implementation, Region of Interest, and Semantic Segmentation. Each component has its own unique set of tasks and challenges to overcome. By utilizing team members' specialties and interests, group members have been assigned appropriate tasks. Generally, James and Eli will be working on Hardware Implementation, Lindsey on the region of interest algorithm, and Mason on the semantic segmentation model.

3.2 Project Management

Our team is adopting a combination of waterfall and agile methodologies meant to support the independent development of each subsystem while ensuring alignment for later integration. This hybrid approach allows us to complete initial planning and allows each team member to work autonomously with clear objectives. From the waterfall approach, we ensure each subsystem aligns with our overarching system requirements, while agile's flexibility allows us to adjust as our needs change.

To maintain coordination, we hold weekly meetings where team members provide status updates, discuss cross-system dependencies, and address any roadblocks. These function similarly to Agile standups, promoting ongoing collaboration. Additionally, we track individual contributions through a GitHub repository hosted by our client, allowing for accessible version control and progress tracking.

3.3 Task Decomposition

In the figure below, we have three main areas of focus: Region of Interest, Hardware, Pipeline, and Semantic Segmentation. Each focus area consists of several subtasks. This task decomposition is shown in our Gantt Chart as well.

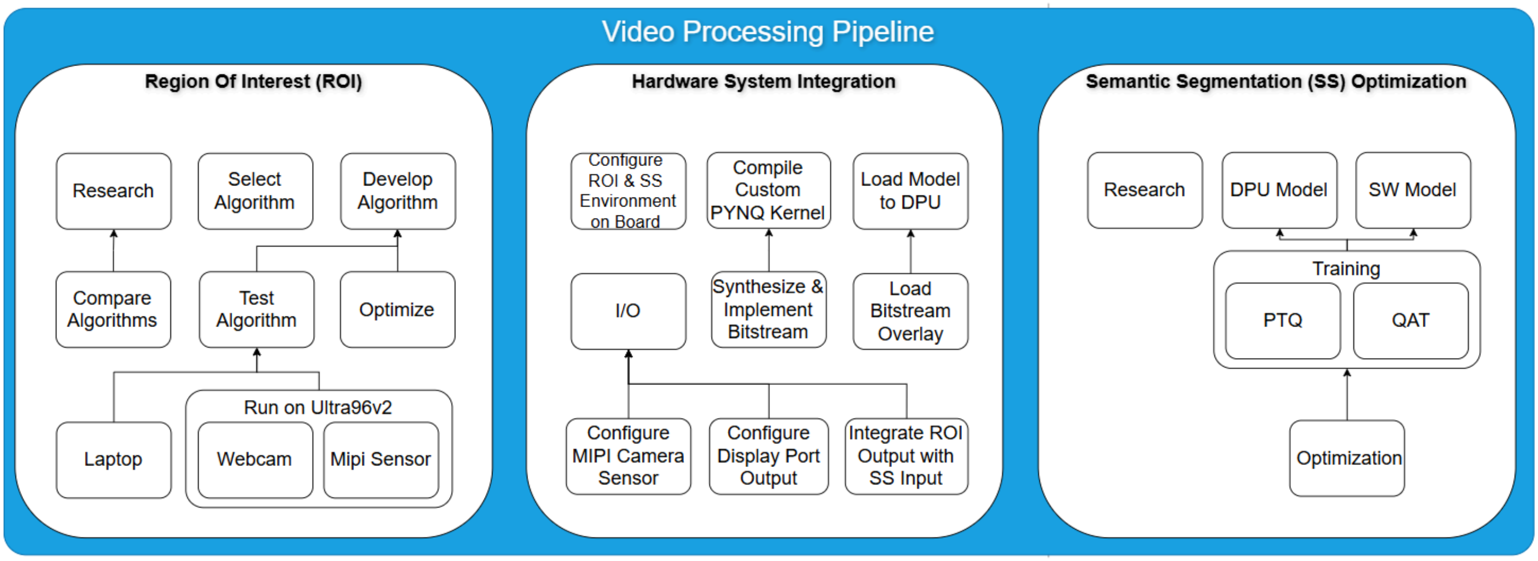

Video Processing Pipeline
Region Of Interest (ROI)
Research
Select Algorithm
Develop Algorithm
Compare Algorithms
Test Algorithm
Optimize
Laptop
Run on Ultra96v2
Webcam
Mipi Sensor
Hardware System Integration
Configure ROI & SS Environment on Board
Compile Custom PYNQ Kernel
Load Model to DPU
I/O
Synthesize & Implement Bitstream
Load Bitstream Overlay
Configure MIPI Camera Sensor
Configure Display Port Output
Integrate ROI Output with SS Input
Semantic Segmentation (SS) Optimization
Research
DPU Model
SW Model
Training
PTQ
QAT
Optimization

Figure 3.1 Task Decomposition Flow Chart

3.4 Proposed Milestones, Metrics, and Evaluation Criteria

Our project involves creating a real-time filtering system. Due to client needs, the system must be made with readily available resources, and small enough to fit on a wheelchair. In order to fulfill requirements, our software must be efficient enough to run on the FPGA board with high accuracy and low latency. We are utilizing and optimizing a computer vision algorithm coupled with a machine learning algorithm to reduce computational expense and increase throughput.

3.4.1 Environment and Hardware Setup

Our goal for the first semester was to get the Hardware running the previous team's progress demo and to set up environments on personal computers to fulfill our goals. To do this, our team set up the hardware and different development environments on personal computers.

The hardware also requires a custom hardware design and PYNQ kernel that we compiled ourselves to fulfill our model's memory and performance needs.

3.4.2 Locate the Region of Interest

The Region of Interest Algorithm should take in an image, with a minimal level of computations, output a cropped image of any eye(s) from the input image. The eyes do not need to be detected when closed, partially obstructed, or facing away from the camera. The algorithm is optimised, a six times FPS increase, to improve throughput of the pipeline.

3.4.3 Optimize the Semantic Segmentation Algorithm

The overarching goal of the SS optimization is to obtain baseline metrics from an open-source model, research and implement neural network pruning strategies, and retrain the optimized model.

Ultimately, by the end of the second semestester, the model had been converted to QAT compatible operations as well as some complex fusion operations. This model is optimized for the deployed environment rather than the training environment, increasing accuracy when running on DPU. To evaluate the model performance, torch summaries, utility testing scripts, loggers, and graphs are used to evaluate model performance.

3.4.4 Deploy ML and ROI Algorithms on the FPGA board

Using Vitis AI we compiled the SS model and deployed it to the board. Additionally, the ROI algorithm will be loaded onto the board so that, at runtime, it can be utilized during demo execution.

3.4.5 Run System & Display Output

In order to understand how well our system performs, it is necessary to see the system's output. This output is what we use to analyze performance and debug. When working on the ROI algorithm and SS model, debugging output was utilized as well as visual inspection through a display. Depending on the test, demo application outputs can be viewed through the Jupyter Notebook server browser or to a display via PYNQ's displayport IP.

3.5 Project Timeline

3.5.1 Gantt Chart

The following Gantt Chart has four subsections of tasks and the plan for completion during our design and prototyping phase of development.

	Task Title	Task Owner	Week										
			1-5	6	7	8	9	10	11	12	13	14	15
1	Hardware System Integration												
1.1	Tensil Research & Debugging	Eli	■	■	■	■							
1.2	Develop Hardware Design (.bit)	James				■	■	■					
1.3	Generate PYNQ - Linux Image (.img)	James				■	■	■					
1.4	Load Bitstream Overlay on FPGA	James					■	■	■	■			
1.5	Load Compiled Model to DPU	James							■	■	■		
1.6	Configure Model I/O	Eli								■	■		
1.6.1	Configure Displayport	James								■	■		
1.6.2	Output Model Results to Displayport	James								■	■		
1.6.3	Connect & Setup Webcam	Eli								■	■		
1.7	Connect ROI Output to SS Input	Team									■	■	■
2	Region of Interest (ROI)												
2.1	Research	Lindsey	■										
2.1.1	Compare Algorithms	Lindsey	■										
2.2	Algorithm Selection	Lindsey	■	■	■	■							
2.3	Develop Algorithm	Lindsey	■	■	■	■	■	■	■	■	■	■	

2.3.1	Test Algorithms on Laptop	Lindsey	■	■	■	■	■						
2.3.2	Run on FPGA Board with Webcam	Lindsey					■	■	■	■	■	■	
2.3.3	Run on FPGA Board with MIPI Sensor	Lindsey					■	■	■	■	■	■	■
2.3.4	Optimize Algorithm	Lindsey						■	■	■	■	■	■
3	SS - Semantic Segmentation Optimization												
3.1	Setup Optimized Model	Mason	■										
3.2	Post-Training-Quantization	Mason	■										
3.3	Setup Quantization Aware Training	Mason	■										
3.4	Perform QAT	Mason		■	■	■	■	■	■	■	■		
3.5	Compile Deployable Model	Mason			■	■	■	■	■	■	■		
3.5.1	Attempting to Compile with Tensil	Mason			■	■	■						
3.5.2	Compile with VAI_C_XIR	Mason						■	■	■	■		
3.6	Stretch Goal: Pruning Model	Mason											
4	Course Deliverables												
4.1	Prepare Handoff Materials	All						■	■	■	■	■	
4.2	Poster & Presentation	All								■	■	■	■

Figure 3.2 Team Task Decomposition Gantt Chart

Our project schedule is organized into a structured timeline, outlined in a Gantt chart format, detailing the weekly progress for each primary task and sub-task. This schedule adopts a hybrid Agile-Waterfall approach, with clear task decomposition and specific milestones for each team member to meet.

3.5.3 Hardware System Integration

For this high-level task, the team focused on configuring I/O, and deploying the ROI algorithm and SS model on the board. This involves researching the tools, frameworks, compilation, connectivity, and display methods.

3.5.4 Implement Region of Interest Algorithm

This task involves developing the Region of Interest algorithm, beginning with preliminary setup and research to determine the most suitable algorithm for our project. Then, we shift to implementing the selected solution and optimizing and testing as necessary.

3.5.5 Optimize Base Semantic Segmentation Model

For this task, we will gather training data, obtain a baseline model, and identify and implement optimizations. More specifically, PTQ and QAT versions to increase the deployed performance accuracy, which is ultimately what we care about. The model will then be retrained with these optimizations to increase model efficiency and accuracy.

3.6 Risks and Risk Management

Below, we identified salient risks for each main task in our project, with estimated probabilities for each risk factor. It is difficult to accurately estimate risks due to requirement volatility, but we've done our best to take that into account. For risks exceeding 0.5, we discuss mitigation plans, including alternative solutions and adjustments.

3.6.1 Environment and Hardware Setup

This part of our project involves minimal risk, mostly consisting of straightforward tasks like configuring our hardware and development environments. However, potential risks arise if the previous team's project setup and code are incomplete or incompatible with how we've set up ours.

This could create delays as we adapt or troubleshoot their configuration. To mitigate this, our team is keeping close contact with the current team and is reviewing their documentation early to ask questions as they arise.

3.6.2 Run ML Model on the FPGA

The model fails to run in real-time on hardware (0.6)

There is some risk that our ML model will reach the hardware limitations of the board. This could lead to bottlenecks, slowdowns, or even failure to run. To address this, we will monitor performance closely during initial trial runs and adjust the model using Tensil's architecture definition file to reduce complexity or optimize for certain requirements. We can also adjust our inputs by downsampling the video feed or reducing the frame rate.

Incompatible I/O to display (0.5)

There is some risk in setting up our demo to display the video feed and results from the model. There are unknowns regarding the frameworks we will use to display information to the display and what our outputs will look like from each subsystem. We plan to mitigate some of this by keeping each team member up to date on each subsystem and what we expect our I/O to look like. This will make it easier for us to plan ahead on what needs to be done to hook up each subsystem.

3.6.3 ROI - Region of Interest

Failure to meet real-time performance (0.6)

Similar to other subsystems, there is some risk in achieving real-time performance from this algorithm. We plan to mitigate this by spending adequate time researching the best solution to meet our needs. Additionally, we are planning to spend time researching and implementing optimizations.

Algorithm Inaccuracy (0.4)

We plan to mitigate algorithm inaccuracy primarily through our algorithm selection done during our research phase. If an algorithm fails to provide sufficient accuracy, we can adopt optimizations or change our algorithm approach. Also, there is some room to increase the desired region size to accommodate for inaccuracies, but that comes at the cost of performance.

Algorithm Selection (0.3)

As mentioned above, there is some risk that the algorithm we select and start developing does not meet our needs. We plan to mitigate this by documenting several algorithms and narrowing them down based on our performance and accuracy requirements.

3.6.4 Semantic Segmentation Optimization

Lost Training Progress (0.3)

To minimize the risk of losing potentially weeks of time during training due to an unforeseen error, checkpoint files will be created to save the progress of training sessions. Additionally, extensive research will be performed before implementation on the model itself. This will also reduce the re-training efforts of the model.

Model Complexity Exceeds FPGA Capacity (0.6)

Similar to a risk from above, the model complexity may exceed the hardware capabilities of the FPGA. If this occurs, we can mitigate it by considering solutions such as simplifying the model architecture or completing more aggressive optimizations that reduce the hardware requirements of the model at the cost of accuracy.

3.7 Personnel Effort Requirements

3.7.1 Task Breakdown with Estimated Man-Hours

WBS #	Task Title	Est. Hours	Reasoning
2.1.1	Research Tensil-AI	8	Requires exploring Tensil documentation and understanding how to compile ML models with it.
2.1.2	Make First Compilation	12	Initial compilation may involve trial and error to address compatibility issues with the FPGA environment.
2.1.3	Compile Mason's Model	10	Following initial compilation, adjustments for Mason's specific model should take slightly less time.
2.2	Download Model onto FPGA	6	Assuming the model has been successfully compiled, downloading and setting it up on FPGA should be straightforward.
2.3.1	Connect & Setup Webcam	5	Setting up hardware should be quick, but may need some adjustments for compatibility with the FPGA system.
2.3.2	Understand & Define ML Model Output	8	Reviewing model output specifics and understanding how it should be processed or displayed.
2.3.3	Determine How to Display Output	6	Deciding the most effective display method may involve testing several approaches.

2.3.4	Display Output	4	Implementing the chosen display method based on prior testing and research.
3.1	Gather Initial Understanding	6	Reviewing documentation and requirements to understand the scope of Region of Interest (RoI) analysis.
3.2	Environment Setup	8	Setting up and testing the environment for RoI research and model development.
3.3.1	Compare Algorithms	10	Requires evaluating different algorithms for RoI analysis to determine the best fit.
3.3.2	Test Model	12	Testing baseline models with different algorithms to assess performance.
3.3.3	Compare Models	10	Comparison analysis to select the model with optimal performance for further development.
3.4	Create Optimized Solution with Research	14	Using prior research to develop a solution tailored to project requirements, including potential fine-tuning.
4.1	Research	20	Reviewing existing literature and resources on semantic segmentation optimization techniques.
4.2	Environment Setup	8	Configuring and testing the development environment for semantic segmentation work.
4.3	Obtain Training Data	1	Acquiring and organizing relevant data for model training, with consideration for data quality.
4.4	Obtain Baseline Model to Put on Board	10	Preparing a functional baseline model to understand current performance and areas for improvement.
4.5	Identify/Research Points of Optimization	12	Identifying specific aspects of the model architecture or processing pipeline that can be optimized.

4.6	Implement Optimizations	15	Modifying model code or architecture based on research findings, potentially requiring iterative adjustments.
4.7	Train Model with Optimizations	18	Training the optimized model, likely with several trials to evaluate performance and finalize adjustments.

Figure 3.3 Task Time Allocation Estimations

3.7.2 Estimation Variance

WBS #	Task Title	Est. Hours	Actual Hours	Variance	Reason for Variance (if significant)
2.1.1	Research Tensil-AI	8	80	74	Tensil.AI was an initial constraint, eventually proven to not be compatible, thus many man-hours were used to debug an unsupported framework.
2.1.2	Make First Compilation	12	80	68	This is linked to issues related to Tensil-AI and also general complexity of quantization.
2.1.3	Compile Mason's Model	10	10	0	
2.2	Download Model onto FPGA	6	6	0	
2.3.1	Connect & Setup Webcam	5	12	7	Irrelevant bugs.
2.3.2	Understand & Define ML Model Output	8	24	16	Additional time needed to understand quantization processes to deploy model.
2.3.3	Determine How to Display Output	6	6	0	

2.3.4	Display Output	4	12	8	Irrelevant bugs
3.1	Gather Initial Understanding	6	20	14	After dropping Tensil.AI, we had to gain new understanding of Vitis-AI and VART.
3.2	Environment Setup	8	40	32	Learning and using Docker, Conda, and frameworks.
3.3.1	Compare Algorithms	10	10	0	
3.3.2	Test Model	12	20	8	Slight learning curve on how to evaluate results.
3.3.3	Compare Models	10	20	10	Many strategies were researched.
3.4	Create Optimized Solution with Research	14	100	86	Many errors during quantization and deployment steps that documentation did not support.
4.1	Research	20	20	0	
4.2	Environment Setup	8	24	16	Learning and using Docker, Conda, and frameworks.
4.3	Obtain Training Data	1	1	0	
4.4	Obtain Baseline Model to Put on Board	10	10	0	
4.5	Identify/Research Points of Optimization	12	12		

4.6	Implement Optimizations	15	15		
4.7	Train Model with Optimizations	18	18		

Figure 3.4 Estimation Variance

4. Design

4.1 Design Context

4.1.1 Broader Context

Context of Public Health, Safety, and Wellness

The long term goal of this project directly relates to the health, safety, and wellness of wheelchair bound individuals. With the vision of real-time seizure detection, the project has potential to improve aid past even health-care professionals (of course utilizing both the clients final system and healthcare professionals simultaneously would be a more realistic approach). The system can signal healthcare professionals to render proactive aid to the wheelchair bound individual if a seizure is possibly detected.

Context of Design Switch (Tensil.AI to Vitis-AI)

Originally, we were tasked with utilizing a limited set of technologies to accomplish our goal of a real time pupil segmentation; such technologies include the Ultra96v2, Tensil.AI, PYNQ, and the Vitis AI quantizer. After researching throughout the first semester, having success deploying other convolutional neural networks, we had reason to believe Tensil.AI was capable of compiling our semantic segmentation model. However, we soon learned that our model has a few unique operations that are unsupported in Tensil.AI's subset of operations. With too little time to dedicate to attempting to integrate these operations ourselves within the Tensil.AI source code, we had to pivot our design to something else. Additionally, in early 2025, the Tensil.AI domain went down. While we would have liked to ask for Tensil.AI's op subset to expand to the ones we need, we soon learned that support for Tensil.AI stopped and was no longer being supported.

So, after presenting this to the client, our team proposed a new plan for what our final requirements will look like. The client was equally shocked and tasked us with completing the real-time ROI algorithm with the MIPI sensor, and do our best to integrate the semantic segmentation model.

4.1.2 Prior Work

There are two main sources of prior work utilized by our team: MIPI camera algorithm and a non-optimized U-Net model. The MIPI camera requires a much more complex algorithm to capture the image, so our team was able to “black box” the algorithm that a previous senior design team created (also under the same client) to focus purely on the ROI algorithm. Finally, instead of starting from scratch, the client provided an open-source model to optimize for real-time use, thus only needing to focus purely on the optimizations. Both sets of prior work allowed our team to focus on the main aspects of our project.

4.1.3 Technical Complexity

Our project demonstrates significant technical complexity through the use of advanced tools and frameworks required to create the system. Across working with the IMX219 camera sensor to optimize and compile a machine learning model to an FPGA, our project covers a large variety of topics in real-time systems like FPGA development, ML acceleration, data flow management, and computer vision algorithms. All subsystems of our project have niche complexities that engineers must learn to progress in the project. There is more than enough scope for all four team members to have rigorous work to complete, which will meet if not excellent career workloads.

Complexity of the ROI Algorithm

Managing an algorithm with a large search space (an image) results in high computational complexity. Time was dedicated to researching and making optimizations to decrease computational stress, and increase framerate. Details of optimization and complexity are held under NDA and can not be shared in depth.

Complexity about the ML model

Understanding the “black box” of ML models is inherently complex. Optimizing an ML model requires an understanding of the underlying mathematical operations and activation functions, as well as the purpose of certain training strategies. For example, Stochastic Gradient Descent is an essential component to many training algorithms in ML, yet it is backed with Linear Algebra and Calculus backgrounds. All of which is just background knowledge needed to begin to understand the model's behavior. With that said, working with the ML model from our client is a sufficiently complex task.

Complexity of Integration

Successfully integrating all parts of this system onto the board requires a base-level understanding of the algorithms,and a good understanding of the following: computer architecture, hardware design, embedded systems, PYNQ operating system, the camera sensor, and their respective relevant I/O. A specific challenge that our group overcame was: get the model running on the FPGA. This inherently comprises many layers of challenges, and given the system knowledge needed to solve them, is certainly on par with that of some industry challenges.

4.2 Design Exploration

4.2.1 Design Decisions

Camera Sensor

Initially, our team considered using a standard webcam for the system. This was because the purpose of our project was more focused on the implementation and optimization of machine learning models and computer vision algorithms. However, with the successful use of the Sony IMX219PQH5-C (IMX219) camera sensor by the previous team, we've chosen to utilize this camera instead.

The IMX219 camera sensor is capable of high-speed image capture, aligning with our client's need for low-latency performance. Additionally, it will be easier to build upon the existing work and documentation of the previous team. This decision is important because the camera sensor directly impacts the accuracy and responsiveness of the system. Rather than being bottlenecked by the low FPS of a digital webcam, we can analyze our real-time system performance more accurately with the higher camera throughput.

Region Of Interest Algorithm

The previous senior design team made a pipeline that displayed camera input on a screen. In order to achieve a high FPS, they reduced their video size to increase throughput. So that we can build off of this solution, the ROI Algorithm will locate the eye, then crop the video frame to only include the eye. By reducing the video size, we can increase the throughput of the system with higher resolution images. Finding a fast and efficient algorithm will enable the system to accurately and quickly locate the ROI and reduce the data used to analyze it. An efficient algorithm also reduces the chance of clogging the pipeline, which would delay the output and remove the system's ability to work as a real time system.

Semantic Segmentation

The U-Net model architecture was ultimately chosen as it is an award winning model that is great at specific tasks like semantic segmentation. Additionally, this was a model previous teams had worked on and the client approved of. This specific model has high accuracy due to the skip connections greatly reducing data loss from encoding layers. The high accuracy of the model was the original design decision as it is imperative the model correctly segments the pupil before making any optimization decisions. Ultimately, this model is known to be precise and accurate, but not known for its speed. This leads to our team's role, play to the strengths of the model architecture while optimizing its poor performance.

Tensil AI

Initially, Tensil.AI was set as a technology requirement forour team. After research, Tensil.AI looked promising but there were concerns about integration. When it came to integration, we ran into many issues that ultimately led to a need to step away from Tensil.AI. We switched to Vitis AI and VART as they are under the same framework, so integration would be easier and we would avoid similar issues we were running into with Tensil.AI, mentioned throughout this document. Ultimately, this switch to Vitis AI and VART was key to integration success.

Vitis AI Compiler & VART

Because Tensil does not support many operations that the U-Net model heavily uses – therefore cannot compile our model – we needed an alternative compiler and respective runtime environment. By utilizing technologies all under the same framework, the team planned to quantize and compile the model using the Vitis AI Compiler to run the model on a DPU within the FPGA . This alternative approach needed to be quickly researched and implemented.

4.2.2 Ideation

Model Optimization Techniques

To reach our final model, we first started one of the best open-source models. Our team looked into Quantization techniques to have the best deployed performance in terms of accuracy, and speed as well.

This was a very large model, so we looked into making a mathematically equivalent model with optimized operations in place of standard convolutions. With the help of parallel work with other senior design teams, this was implemented and trained very quickly.

By utilizing the tools within the Vitis-AI workspace we already were working in, we had access to model inspectors, simplifiers, and quantizers. All of which at the start required research to understand the purpose. Firstly, inspectors are used to gain an understanding of potential bottlenecks within the model. Secondly, simplifiers do as the name suggests and minimize unneeded operations. Thirdly, quantizers are responsible for turning the floating-point-32 models into unsigned-integer-8 bit models, which are able to be deployed to the FPGA itself. These quantizers are the primary focus of SS model optimizations for our team.

Because the quantizers were an essential part of deployment and a key point of accuracy loss, we looked into the best ways to specifically make an integer-based model. We came up with two primary ways: Post-Training-Quantization (PTQ) and Quantization-Aware-Training (QAT).

PTQ is relatively simple; train the model normally, and then run a separate quantization script that generates a quantized model (where the complex work is done behind the scenes via pytorch_nndct).

Quantization Aware Training (QAT) is an excellent option when the model will run in an uint8 environment, or some other low accuracy environment. Since the model will be given a grayscale image, the values will fall within unsigned 8-bit integers. If you quantize a floating-point model without QAT, there often is accuracy loss when the weights essentially are truncated. QAT prevents this by performing quantization and dequantization steps throughout the forward pass of the model during training.

This quantization/dequantization process happens before and after developer specified operations (eg. Convolution, BatchNorm2D, or LeakyReLU). Then, the weights are constantly truncated, but the precision during the operations themselves are maintained. Ultimately, the weights converge onto a more robust set that will have significantly less data loss when deployed to the FPGA.

Model Evaluation Techniques

There are many well known evaluation methods to evaluate machine learning models. In our research, we looked into TensorBoard, Grad-CAM, and Vitis-AI technologies specifically.

First, we looked into TensorBoard. TensorBoard offers easy-to-implement loggers, which can generate graphs of several metrics tracked through the model, like weights. We have implemented a tensorboard onto the open-source model as a proof of concept to show the client.

Next, we looked into Grad-CAM, or Gradient-Class-Activation-Mapping. This is a well researched technique of generating heat maps, to help understand what the model is doing. This technique requires adding Global Average Pooling layer, which quantizes the model and subsequently a ReLU layer, which would remove any undesirable weight (ie. negative values). This seems like a good option, however further research is needed to know if the compilers will optimize the sparse network.

Finally, we were able to find an extensive library of tools with Vitis-AI. With a well documented API, there are many tools to extract data in the model. With Vitis-AI quantizations we were able to test the model output before deployment, which was key for evaluation.

Overall, these technologies help evaluate the model quantitatively, qualitatively, and visually.

ROI Algorithm: Option One

This algorithm requires two cameras. One normal camera to find the pupil as a black circle. The other camera is infrared, IR. This is out of our project scope due to the added expense of adding another camera as well as the additional complexity of pipelining an IR video feed.

ROI Algorithm: Option Two

This algorithm uses contrasts in lighting to locate facial regions. It is extremely fast with few computations. However, the speed results in lower accuracy. Specifics of the algorithms are confidential under the NDA.

4.2.3 Decision-Making and Trade-Off

Criteria	Weight	Camera Sensor (IMX219)	ROI Algorithm		Semantic Segmentation (U-Net)	Quantization (Vitis-AI)
			Option One	Option Two		
Accuracy	0.3	9	8	6	10	9
Speed	0.25	7	8	10	6	6
Ease of Integration	0.15	8	5	7	8	5
Cost	0.2	6	10	10	10	10
Client Approval	0.1	10	5	5	10	7
Averages	1	8	7.2	7.6	8.8	7.65

Figure 4.1 Weighted Decision Matrix

The weighted decision matrix ranks several components and ideations of our project based on key criteria. We have weighted accuracy as the most important criteria, closely followed by speed. After that, cost, integration, and client approval is taken into consideration.

Based on the weighted decision matrix scores, algorithm two was chosen.

4.3 Final Design

4.3.1 Overview

Our design focuses on creating a real-time system that tracks a user's pupil for specific use in assistive wheelchair technology. Providing accurate and fast-tracking data can be used for mobility control and safety features, such as seizure detection. Despite many roadblocks through development, the team was able to achieve a fully-integrated video processing pipeline to achieve these goals.

With the combined use of OpenCV, Pytorch, Vitis-AI, and PYNQ frameworks/libraries, a camera sensor will capture images of the user's eyes at a high frame rate to be processed on our hardware board using computer vision and machine learning to locate the pupils in each frame. The result will then be displayed on a monitor for testing and demonstration purposes.

Lastly, an optimized ML model has been delivered to our client as the final step in the pipeline. This model is already being used by a future team. The ML model has been integrated into the pipeline, officially creating the full end-to-end pipeline for the client.

4.3.2 Detailed Design and Visuals

Our system consists of a few separate components: region of interest (ROI) cropping algorithm, a trained pupil locating machine learning model, and research takeaways on VART. These subsystems come together to form our video processing pipeline, highlighted below with our software flow diagram and architecture design block diagram.

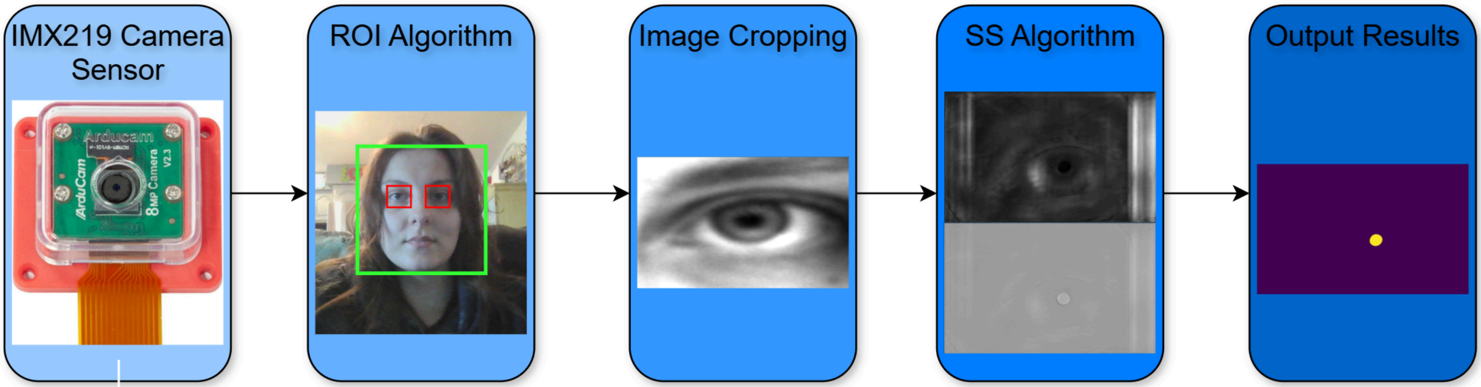


Figure 4.1 System Software Flow Diagram

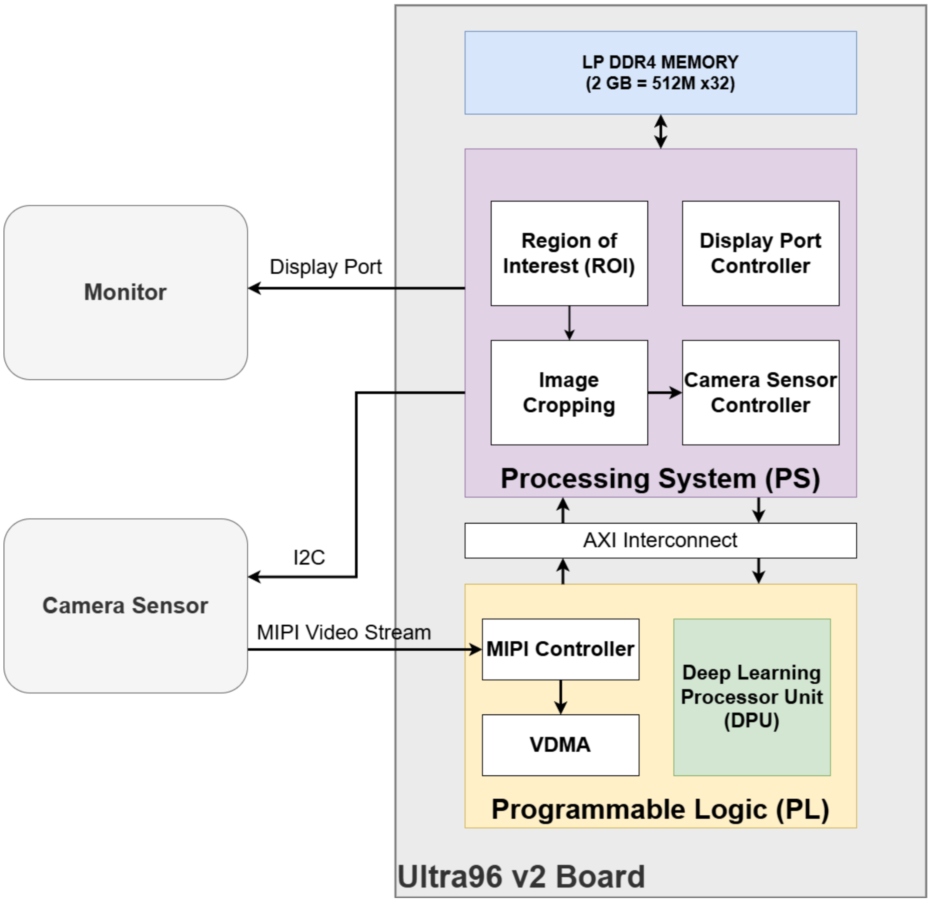


Figure 4.2 High-level Architecture of Ultra96 Board

Above is a high-level block diagram of the software-data-flow, system architecture, and workflow (in that order) showcasing primary components respectively. The software-data-flow diagram allows for a great visual of what the pipeline is doing at every stage. Whereas the system architecture diagram helps understand how the pipeline runs on hardware.

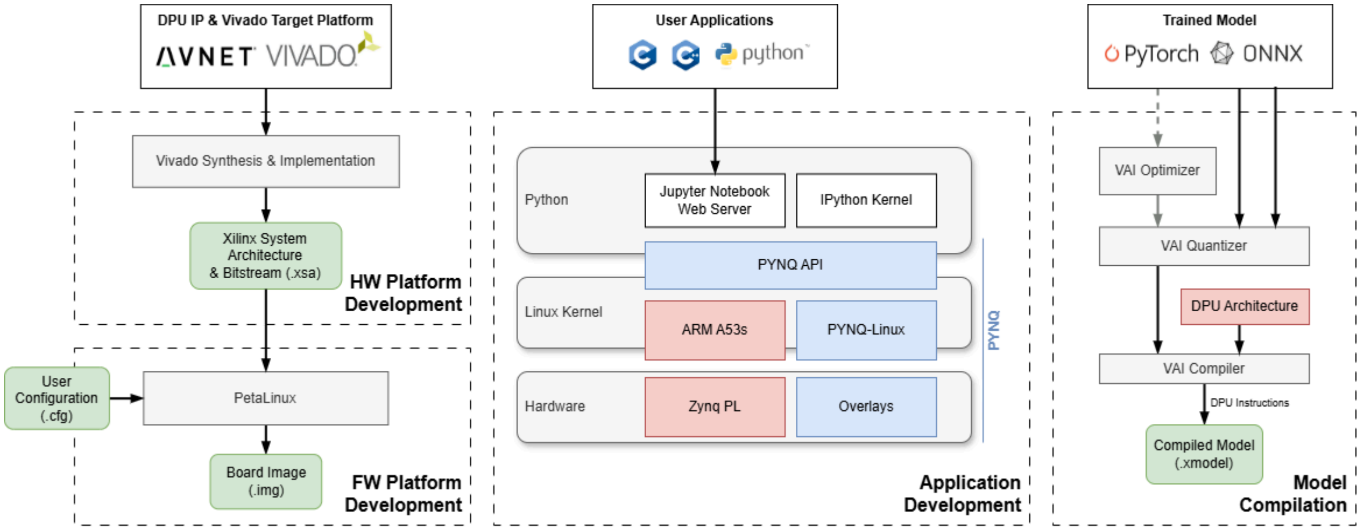


Figure 4.3 Development Environments

The workflow diagram above helps the audience grasp what the development stages look like for the team and possibly future teams for the client. On the left is our hardware and firmware platforms that output our hardware design overlay and PYNQ-Linux kernel to be used on the board. On the right are our steps for optimizing and compiling a machine learning model to the XModel platform for a specific DPU architecture.

In the middle, application development, the team connects to a Jupyter Notebook webserver on the board that we can execute C++ and Python code from a remote machine. These notebooks interface with PYNQ that abstracts the interfaces to the hardware programmable logic.

Camera Sensor

The IMX219 camera will capture continuous, high FPS video of the user's eye. This camera sensor has been used by previous teams, allowing for simple integration with passed-down documentation and code. It can record video at a high FPS due to the ability to configure a cropped video feed to reduce bandwidth. The feed will picked up from the VDMA to the processor to be sent through our various algorithms.

Ultra96 v2 FPGA Board

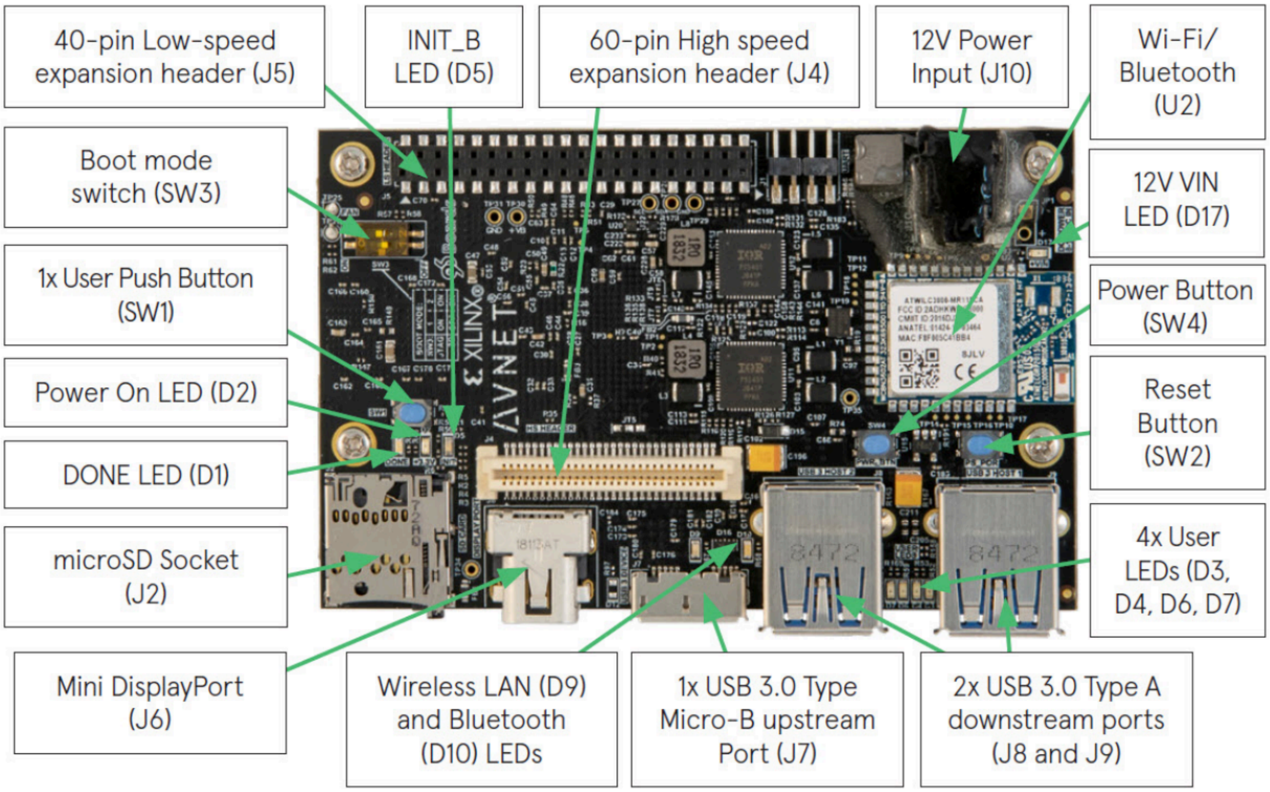

40-pin Low-speed expansion header (J5)
INIT_B LED (D5)
60-pin High speed expansion header (J4)
12V Power Input (J10)
Wi-Fi/ Bluetooth (U2)
Boot mode switch (SW3)
12V VIN LED (D17)
1x User Push Button (SW1)
Power Button (SW4)
Power On LED (D2)
Reset Button (SW2)
DONE LED (D1)
microSD Socket (J2)
4x User LEDs (D3, D4, D6, D7)
Mini DisplayPort (J6)
Wireless LAN (D9) and Bluetooth (D10) LEDs
1x USB 3.0 Type Micro-B upstream Port (J7)
2x USB 3.0 Type A downstream ports (J8 and J9)

Figure 4.4 Ultra 96 v2 Topology

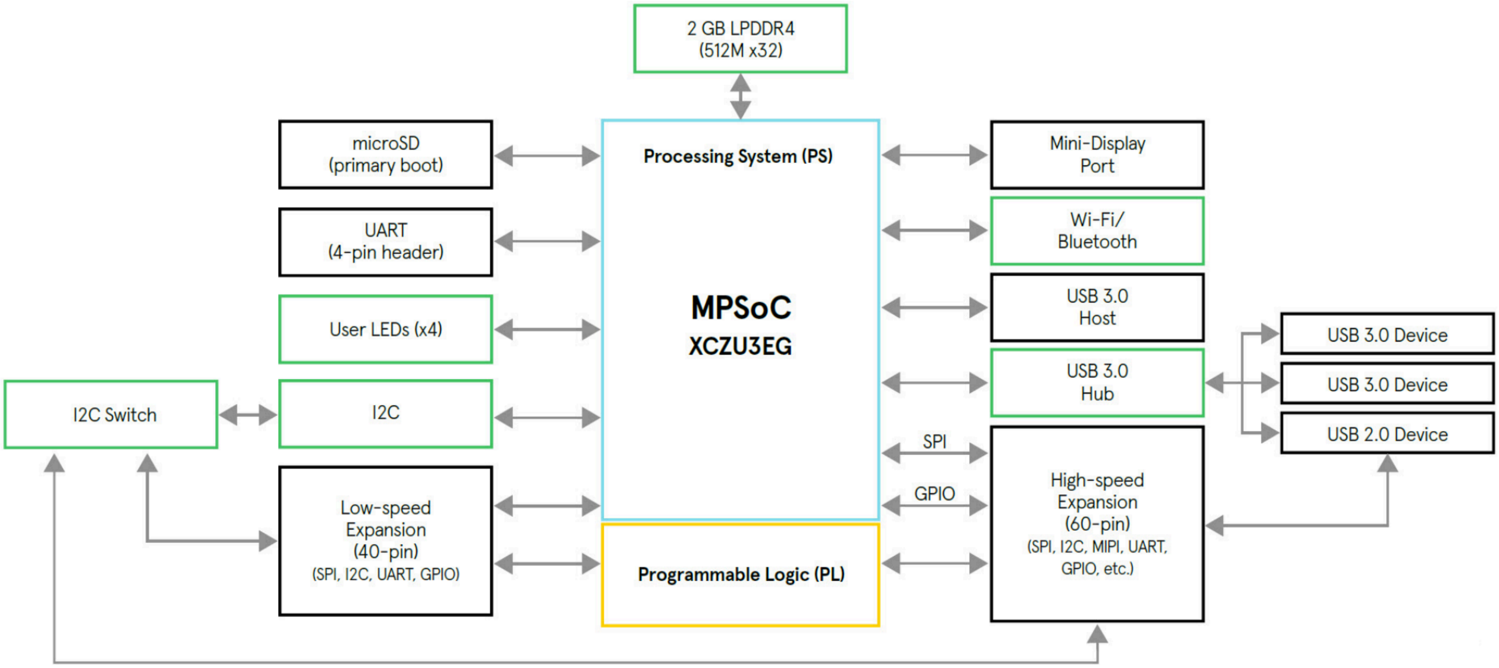


Figure 4.5 Ultra96 v2 Block Diagram

Equipped with a Xilinx Zynq Ultrascale+ MPSoC, the Ultra96 v2 board is capable of parallel processing and memory management suited for this high performance system. Outside of the important computer vision and machine learning algorithms, the board will be primarily used for moving data between each subsystem. Once the cropped video feed is received, it will be moved into the semantic segmentation ML model compiled onto the FPGA using Vitis AI.

Intermittently, the ROI algorithm will re-analyze the video feed to update the cropped image region. If a new region is determined, the camera registers will be adjusted to output that new region.

The board will also be responsible for visualizing our system using a display. To verify system functionality, it will show the processed data, including the cropped video feed, pupil location, and performance metrics.

Region of Interest & Cropping Algorithm

The ROI Algorithm is mostly compared on its computational expense, the number of math operations it must use to accomplish a goal. This expense can be compared by looking at the code and the mathmatic equations and by comparing how long, in milliseconds, it takes to run. Both of these are important aspects to consider within this project.

The algorithm can not use too many computations, because we are working with a Ultra96 board that can not quickly run thousands of computations, for both the ROI and Semantic Segmentation Algorithms. Additionally, the overall runtime can not be too large, because it will cause a clog in the pipeline.

The cropping algorithm resulted in a 6x increase of FPS.
The details of the cropping algorithm can not be shared due to NDA constraints.

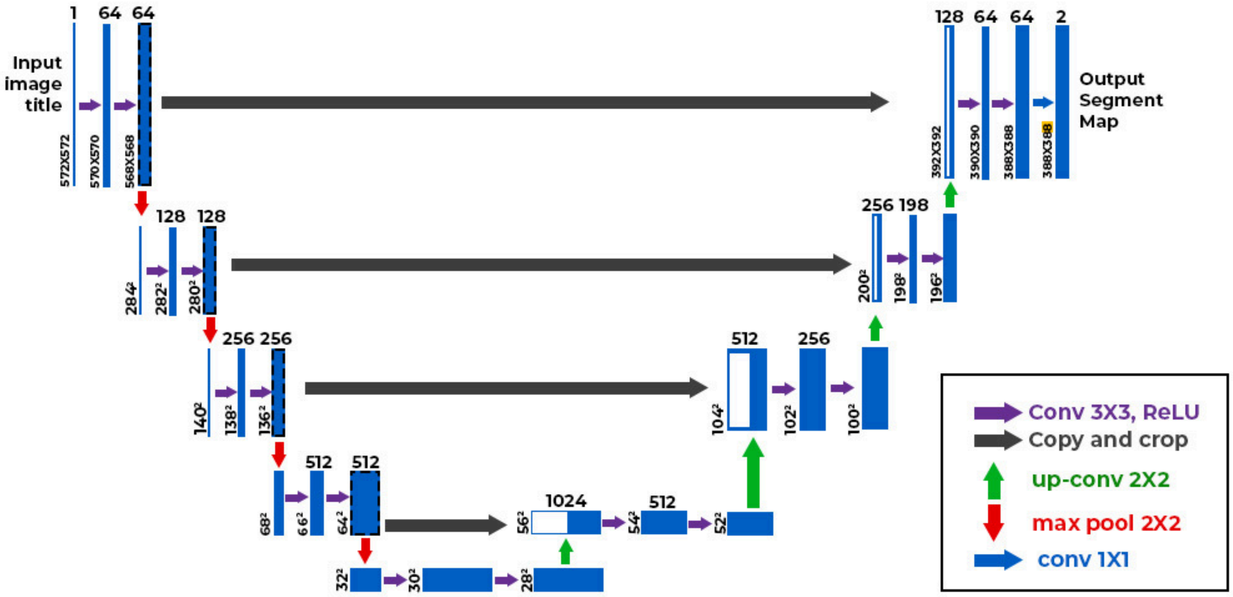


Figure 4.6 U-NET Convolutional Model Diagram

Semantic Segmentation ML Model

The semantic segmentation model is a convolutional neural network based on a U-net architecture. This model is excellent at specialized tasks, such as pupil detection. Taking in an input feed of video frames, the data is encoded through several layers, passed through the bottleneck connecter layer (bottom of the U), decoded back to original resolution, and finally an output map is produced, where pixels color indicates its grouping (ie. white pixels represent pupils and black pixels represent everything else).

4.3.3 Functionality

In the scope of our team's project. The system will have a live video feed through a camera, which is sent to the board for processing. After being processed, the system will display the live semantic segmentation results.

In a longer term perspective, a camera will be attached to a wheelchair and will be pointed at the user. Additional processing may be implemented for additional functionality, which is enabled by our team’s real-time semantic segmentation system.

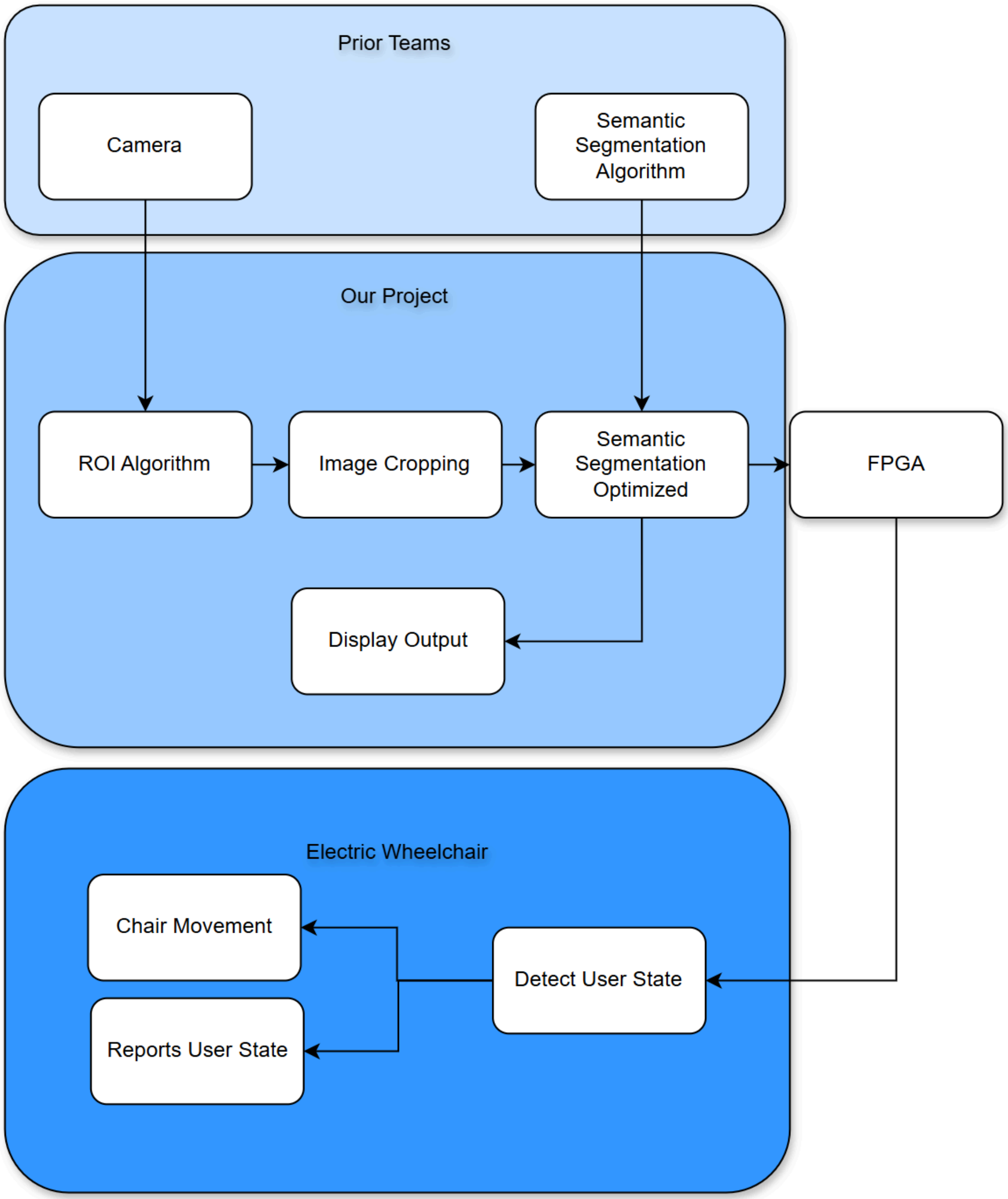


Figure 4.7 Project Functionality and Handoff Flow Diagram

4.3.4 Areas of Challenge

In the first semester, our team noted a few concerns meeting the real-time, accurate performance for assistive technology and the seamless integration of all subsystems. This turned out to be true. With Tensil.AI issues noted throughout, our team quickly pivoted late in the project to Vitis-AI, which had much more success.

Besides our initial concerns, our team has ongoing concerns for the performance of the systems, despite integration being complete. Speed and accuracy on non-dataset images is not up to requirement standards.

To mitigate these issues, we have already begun working on post-processing logic to improve output results. Also, we discussed this with our client and set plans of future work and research to be done in regards to performance of the pipeline. The client found this conversation very valuable. Additionally, we met with Dr. Zambreno to acquire his insight into our problems and successes throughout the project.

4.4 Technology Considerations

4.4.1 Ultra96 v2

Strengths	High processing power, optimized for AI/ML tasks, FPGA-based for efficient parallel processing.
Weaknesses	Limited memory and storage, requires specific FPGA programming knowledge.
Trade-Offs	Balancing processing power and memory for complex tasks may require offloading some computations.

4.4.2 PYNQ

Strengths	Simplifies FPGA programming with Python, supports rapid prototyping and visualization on Ultra96v2.
Weaknesses	May lack the low-level control needed for highly customized hardware tasks.
Trade-Offs	Ease of use vs. low-level FPGA programming for maximum efficiency.

4.4.3 U-NET Model

Strengths	High accuracy due to skip connections; encoding provides spatial, decoding provides contextual information—ideal for specialized tasks.

Weaknesses	High computational complexity (many convolutional layers) and memory intensity (storing data for skip connections).
Trade-Offs	Reducing complexity increases risk of data loss, potentially impacting segmentation accuracy.

4.4.4 ROI Algorithm

Strengths	Increases efficiency by focusing on specific regions of interest, reducing processing time, and improving speed.
Weaknesses	Smaller regions create a risk of losing important data, possibly lowering segmentation accuracy.
Trade-Offs	Faster processing with smaller ROIs may sacrifice detail and accuracy.

4.4.5 Vitis-AI

Strengths	Compatible with the SS model. Large framework with many tools available to the team.
Weaknesses	Requires certain dependency versions.
Trade-Offs	Initial Camera Sensor code dependencies not compatible with Vitis-AI runtime dependencies and will need to be modified.

5 Testing

5.1 Interface Testing

Our integration testing transition points of our pipeline when data is passing from one system to the next. To have consistent interface testing, all of our systems are able to run on a static image, saved as a file. This allows interfaces to be tested individually and asynchronously with the whole system. This strategy of testing with just image files was vital to integration success as it allowed for debugging of separate systems and partial demos.

5.2 Integration Testing

The critical integration path in our design is getting the full pipeline running on the FPGA. This pipeline is the focus of our project – all of our requirements are based on improvements to this foundation. It is tested by running our full pipeline:

Camera Sensor → ROI → Pre-Processing → SS → Post-Processing → Display Port

Verifying that the output is correct, and that the logs represent correct steps to get to the result.

5.3 Regression Testing

As the team inherited the IMX219 Camera Sensor Code from the client, our first testing/task was to replicate and verify the code works on our board. This acted as a milestone for the team and gave the green light to move forward to adding our sub-systems to this existing system. This specific regression test is driven directly by our requirements to use the IMX219 Camera Sensor, thus tests to ensure its functionality were created. Additionally, as we continue to integrate parts to our pipeline, we take regression testing seriously by ensuring that stuff we had before the integration still works.

5.4 Acceptance Testing

As the team started creating video demos, we shared them with our client in our weekly client meeting as a form of acceptance testing. In general, the video demos were received well by the client; we received little to no criticism when we provided demo videos. These videos acted as a way to show the current status of the project, demonstrating functional and non-functional requirements to the client.

5.5 Results

There are two ways to look at the results of our project, both offering unique insights. The first way is to look at the integration task the team completed, in which the team accomplished many challenging engineering goals providing direct value to the client. The second way is to look at the performance of our system, which highlights the future work needed before the client’s long-term goal is met. Our project addresses requirements and user needs. The primary value that our team struggled, debugged, researched, and eventually achieved was the full integration of the pipeline; however, our team left room for improvement in terms of performance metrics of subsystems.

6 Implementation

6.1 Feature Implementation

6.1.1 Software Implementations

Project ELM delivered standard software implementations of the ROI algorithm, SS Model(s), and demo/testing scripts. The SS Model also provided a new QAT model architecture as well as various trained models used for testing and deployment.

6.1.2 Hardware Implementations

Besides for the software testing versions. Project ELM integrated an ROI pipeline for a standard webcam and MIPI camera sensor. The SS Model was quantized into a DPU compatible version such that the SS Model successfully runs on the Ultra96v2. We are able to run the webcam based pipeline through the SS model successfully on the Ultra96v2. Due to dependency issues, the MIPI Sensor requires some package management (as future work) to run the SS Model.

6.1.3 Overall Implementation

All of our sub-systems and the pipeline itself are successful deliverables provided to the client from our team. However, we were not able to reach the performance targets due to a time constraint and the quick turnaround on Vitis-AI implementation and integration.

6.2 Design Analysis

As this project is a continuation of several senior design projects from universities throughout the country, there is a large amount of progress our team is able to build off. During the first semester the main priority was understanding the complex tools, technologies, algorithms, and models being utilized. Each team member has been assigned a role to be the subject matter expert in their respective technology area. This allows for specialization, while still encouraging collaboration between team members.

Our design can be thought of as a working pipeline with several steps. First, we use the IMX219 Camera Sensor and capture an image frame from a video stream. Second, pass the frame into the ROI algorithm, which will output coordinates corresponding to an eye. Third, perform image preprocessing utilizing those coordinates. Fourth, run the preprocessed image through the SS model. Fifth, perform image-post processing. Finally, sixth, output via display port to screen. This creates a video processing pipeline, shown through our recorded demo.

Drilling into the operations: Through iterations of research, design, and prototyping we have our SS model successfully ported to Xmodel, ONNX, and TorchScript format and compiled using Vitis AI Quantizer. Additionally, the Ultra96v2 has a setup PYNQ environment that can run JuPYter Notebooks, where a lot of our testing and development occurred.

We find that despite the model having a near-perfect performance during training and testing prior to performance. However, the model does have a significant accuracy drop during the deployment (quantization) process with PTQ models. Initial QAT models show much more promising results for accuracy on the deployed board, showing better results with 10 times fewer epochs for training.

This is a note of future work and improvements for the client. The ROI algorithm is limited to the quality and speed of the camera capabilities and CPU processing speed. This aligns with our expectations through testing. Overall, the biggest accomplishment of what works well is achieving a fully integrating pipeline, in which there is room to improve the subsystems in the future.

7 Ethics and Professional Responsibility

7.1 Areas of Professional Responsibility

Area	Description	Team Use Case
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Our team and client set goals that require work competence.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Our team is not financially responsible for the project, but we are expected to use

		provided technology appropriately.
Communication Honesty	Report work truthfully, without deception, and are understandable to stakeholders.	Honest communication is a key part of our team's day-to-day tasks.

Figure 7.1 Areas of Professional Responsibility

Our team demonstrates high work competence as each team member specialized in sections of the project in order to deliver high quality work. As the team worked through bugs, we were fortunate enough to have many supplies, boards, and labs to work in. However, if we were not as well-funded from our client and did not have access to campus resources our team may have fallen to financial struggles.

7.2 Four Principles

Broader Context Areas	Four Principles			
	Beneficence	**Nonmaleficence**	**Respect for Autonomy**	**Justice**
Public health, safety, welfare	Enhance user safety	Avoid false positives that harm users	Enable use control with accessible interfaces	Provide equal access to assistive technology
Global, cultural, societal	Increase accessibility for wheelchair-bound individuals	Avoid cultural insensitivity	Respect diverse user needs and preferences	Address unfairness in assistive technologies
Environmental	Use energy-efficient hardware	Avoid waste by upgrading existing wheelchairs	Respect environmental regulations	Ensure fair distribution of resources
Economic	Develop a cost effective solution	Avoid unnecessary expenses for users	Give users cost effective design tailored to their needs	Avoid financial barriers to assistive technology

Figure 7.2 Project Context on the Four Principles

One of the most important broader context-principle pairs is public health, safety, and well being paired with beneficence; because, enhancing user safety is the end goal of our clients long term project. The long term goal of seizure detection will greatly enhance wheelchair bound individuals safety and well being. By integrating the video processing pipeline, our team played a part in improving the safety of our end-users.

While our project benefits the health and well being of our users, there is room for improvement on the development cost of an effective solution; specifically, the cost of the final product. Our research shows a lack of competitive products, this highlights our key problem:a lack of availability of technology due to high costs. It is important to note that our end-users are likely to already have many high medical-expenses, so mitigating cost would be an important part of the final product that Project ELM did not have as a primary focus.

7.3 Virtues

Our team highly values honesty, hard-work, and communication. Throughout the two semesters, our team has had quick responses for team members asking for help at all hours of the day and maintaining both synchronous and asynchronous communication on a weekly, and even daily, basis. Each team member led their status reports with honesty about time commitments and their availability, in which the team were able to be flexible around. The team has continued to communicate effectively and take responsibility for their subsections of the project, resulting in a well-functioning team.

Eli’s Virtues

Respect is a virtue that I carry with me everywhere I go – especially in this 9-month long Senior Design journey. This is important to me because this underlies a safe and positive environment. I have found that when I respect others and my environment, respect is returned to me. This enables each member of our team to understand one another, to understand their perspective, and be okay with failure. This contributed to a more collaborative environment and a more productive team as well. I demonstrated this by *seek[ing] first to understand, then to be understood,* one of Stephen Covey’s 7 Habits of highly effective people. Faith in Jesus is a virtue that I highly value; and it is not one that I have demonstrated very much in my senior design work thus far. It is important to me because God made me, Jesus died for my sins, and they enable me to live a life worth living. I can demonstrate this by casually bringing this up in conversation.

James’s Virtues

A virtue I believe I’ve demonstrated well in my senior design work is confidence. This is import to me because being able to clearly communicate ideas and trust the knowledge I’ve gained over the course of my undergraduate education is essential when working on a highly collaborative and technically complex project. Throughout this project, especially during major turning points discussed earlier, I relied on my confidence to navigate technologies I’ve previously never used. Being able to make informed decisions has helped the team and I stay focused and moving despite uncertainty.

On the other hand, a virtue I wish to demonstrate more is open-mindedness and adaptability. It is easy to stick to an original plan and idea, but as I have learned during these two semester, a shift in direction is often needed. I’d like to be more proactive in the future about seeking alternative approaches and inviting feedback earlier in the process.

Lindsey’s Virtues

Throughout this project, I have worked to increase inclusion for individuals with mobility impairments. The daily struggles that people with disabilities experience are commonly overlooked. After having a knee surgery and being told to expect more in the future. My mobility was impaired for a few months. With this came a recognition and understanding for a lack of infrastructure and the systematic disregard for individuals with mobility impairments. Our team's subsystem will be used to increase wheelchair bound individuals accessibility and independence. I also worked on and tested my algorithm to locate the eyes of individuals of all races, genders, facial details (freckles), and accessories (glasses).

Another virtue that is important to me is leadership. However, due to our team's consistent teamwork and a lack of conflicts there was never a need for a leader in our group. We all had equal but shifting roles, and maintained a true democracy. I typically value leadership because conflicts are typical. However, each individual's self accountability and dedication to the team, outweighed the typical importance of leadership. While this is not realistic for all teams, I hope this team dynamic can be shared going forward in the professional field, otherwise I may step into a leadership role by accepting responsibility for parts of projects.

Mason’s Virtues

Throughout the time spent on this project, I believe I have demonstrated cooperativeness to work with team members and compromise on solutions. This is important to me because I believe it is key to having a high-functioning team. I have demonstrated this throughout any and all communication with my team members. A virtue I could work on is my commitment to objectivity, sometimes I do not communicate my thoughts objectively and I could improve on this.

8 Conclusions

8.1 Summary of Progress

Overall, the team has completed an ROI algorithm, a SS model, research, and documentation that resulted in an integrated video processing pipeline deployed on an FPGA board. Throughout development cycles, the team researched and documented main attempts that either did not work or simply did not make the final design, and of course the final design is well documented as well. Additionally, the team has boosted other teams our client works with in their progress. The fully integrated pipeline will be a big help for the client and directly completes project goals. The client will be able to improve submodules of the pipeline in the future with a well-modularized codebase.

8.2 Value Provided

Our team provided value to the client far surpassing the simple scope of the project. For example, all the time spent helping other teams is time spent providing value to the client, but not necessarily helping our team meet the project's goals. For our clients needs, we were able to push this long-term project forward to its next step as our demo begins to see the project's high-level vision become reality. We also have provided more specific metrics of our system to the client, which are not shared here for NDA purposes.

8.3 Next Steps

The next steps overall now that the system is integrated is to work on performance. This can be done in many ways: concurrency, SS model reduction, HW acceleration, or even algorithm design. In fact, our team has already begun helping future teams with these exact goals in mind to continue providing value to our client. These are the most logical next steps as our team had to go through many iterations of integration to get the pipeline running in the first place, which meant performance took a back-seat compared to having the pipeline functioning. These steps are imperative to our clients end goal, who needs to obtain certain performance measures, for the safety of the end-users (wheelchair-bound individuals).

9 References

List technical references and related work / market survey references.

[1] "Ultra96-V2 Single Board Computer Hardware User's Guide," Ultra96-V2 Single Board Computer Hardware User's Guide, https://www.avnet.com/wps/wcm/connect/onesite/b85b9556-0b2a-42b3-ad6a-8dcf3eac1ff9/Ultra96-V2-HW-User-Guide-v1_3.pdf?MOD=AJPERES&CACHEID=ROOTWORKSPACE.Z18_NA5A1I41L0ICD0ABNDMDDG0000-b85b9556-0b2a-42b3-ad6a-8dcf3eac1ff9-nDNP5R3 (accessed May 4, 2025).

[2] "Vitis AI User Guide (UG1414)," AMD Technical Information Portal, https://docs.amd.com/r/en-US/ug1414-vitis-ai/Vitis-AI-Overview (accessed May 4, 2025).

10 Appendices

A.1 Operation Manual

Due to our team being under an NDA, we are not able to disclose step-by-step instructions on how to run the system.

A.2 Original Design

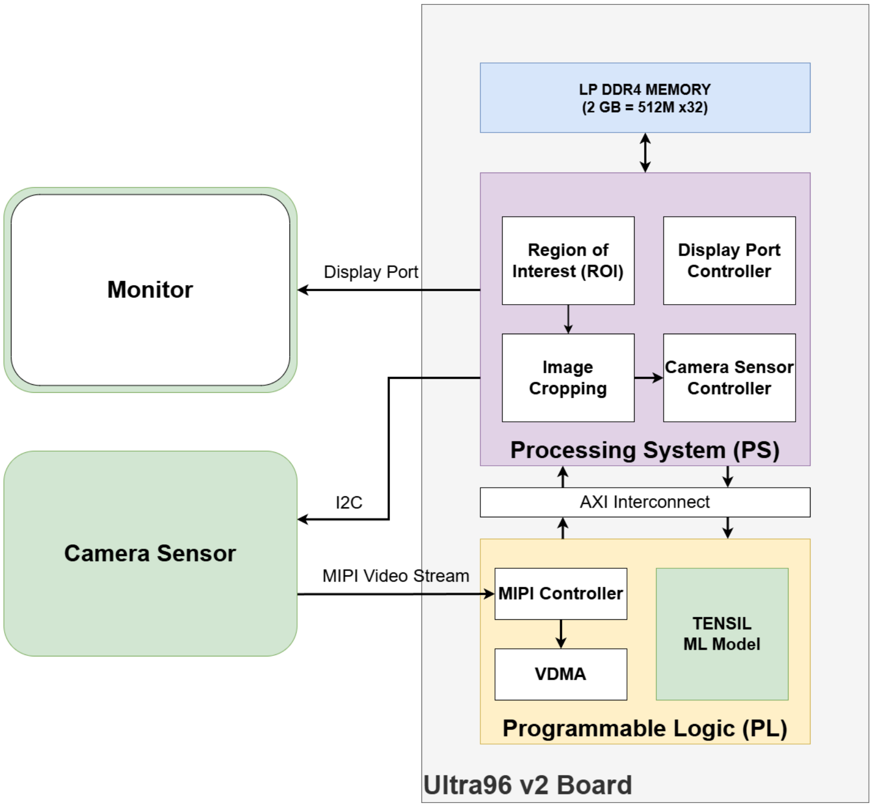
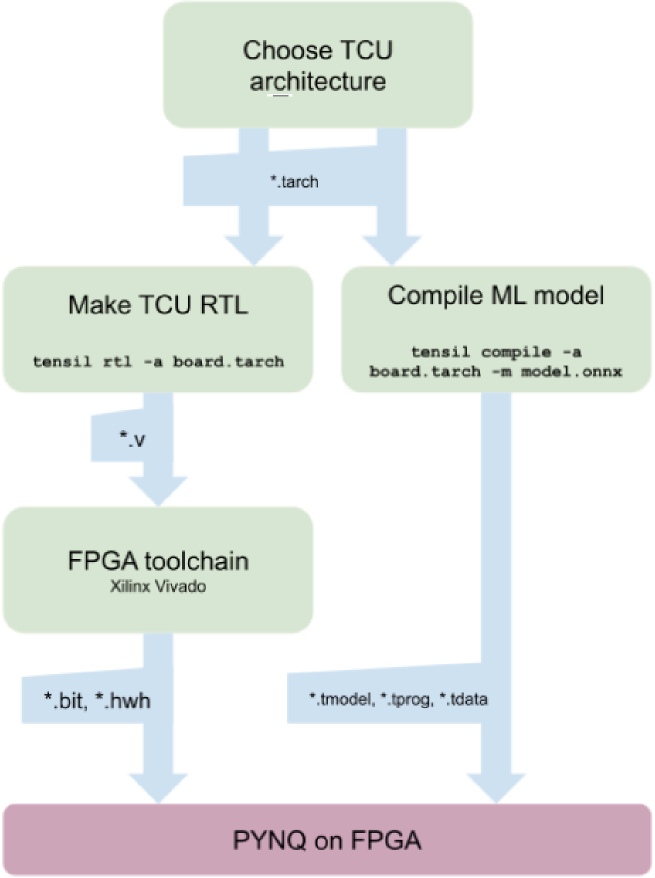


Figure #



A.3 Other Anecdotes

Our team decided the team leader based on a very (not) serious game of Rock, Paper, Scissors each week.

A.4 Code

Due to the team's NDA with the client, we are unable to supply the code repository to our project.

A.5 Team Contract

Advisor Meetings: Thursday at 1 pm, and as needed on Fridays at 2. Unless communicated otherwise, we will meet face-to-face as a team over Microsoft Teams with the advisors.

Client Meetings: Sunday at 4 pm on telegram with the client. The team will meet in person at SIC unless otherwise communicated.

Team Meetings: Thursdays from 1-4 pm or Fridays after 2 as needed. In-person unless communicated otherwise.

A.5.1 Methods of Communication

Discord: Informal team communication related to team meetings and class assignments. Additionally, it is used as a platform for remote team meetings.

Telegram: Private communication between the team and our client focussed on all project related communication. This includes technical questions, client meeting summaries, research, and weekly deliverables.

Email / Teams: Communication with advisors. Where we will send out meeting information and links.

In-Person: Will meet in person in class, during meetings, and as needed outside of class.

A.5.2 Decision-making

The team will follow the consensus view for decision-making. For large project decisions, input from the advisors and client will be considered, while minor decisions will be left up to the team with preference towards any product owners.

A.5.3 Record Keeping

The team will work together to take meeting notes in a shared Google document for all team, advisor, and client meetings. The team leader will lead client meetings and summarize the meeting and associated deliverables to post in Telegram. For our client, we will complete weekly updates within a shared Google slide deck containing each team member's accomplishments for the week, current challenges, and future plans.

A.5.4 Team Expectations

Attendance, Punctuality, and Participation

All team members are expected to attend all meetings. Team members are expected to communicate absences and remote participation as soon as possible via Discord. They are also expected to lead rescheduling efforts as needed.

For client meetings, team members are expected to communicate via Telegram for absences. Since the team meets in person and remotely connects with the client, if a team member needs to meet remotely, that should be communicated via Discord. The same applies to advisor meetings, but communication should include the Advisor(s) via email.

While our communication methods are diverse, this follows the preferences of our client and advisors as best as possible.

Responsibilities

The team determines deliverables at each client meeting with the approval of the assigned individual(s). Team member(s) shall communicate issues and need help as early as possible to stay on schedule.

Communication

Communication should be frequent. This includes when help is needed, questions, meeting attendance, in-class assignments, and when the communication may assist with other team members' tasks. The elected Team Leader of the week will also be responsible for team announcements.

Commitment

Team members are encouraged to give thoughts on all team decisions. Outside of class, team members are expected to invest a minimum of 5 hours per week and more than 15 hours per week are not expected. Resolving blockers of other team members take priority.

A.5.5 Leadership

Leadership Roles

A team leader is designated each week, responsible for leading client meetings and making announcements. A new weekly leader is chosen via three-way rock-paper-scissors games. Each team member gets one self-veto, allowing themselves to be skipped for that week. This allows members with a particularly busy week to not have to put an additional responsibility on top of it. A final rock-paper-scissors game will be played between the last two members to determine the team leader.

Support Strategies

Communicate. By fostering a supportive, open environment, each team member will communicate the reasonable support needed. In weekly standups, the team will see how each team member is performing and will be able to apply support as needed.

Recognizing Contributions

At each meeting, the team leader will walk through the team meeting slides that note the accomplishments of every team member in the past week. Team members will also self-report any accomplishments they make in a week.

A.5.6 Collaboration and Inclusion

Individual skills, Expertise, and Unique Perspectives

Eli Ripperda
Eli has experience in embedded software development and testing. Additionally, Eli has industry experience in data engineering. Eli will develop software that will receive info (eye box, and associated metrics (from semantic segmentation) from Mason's work and will display the info in a "useful format" to other stakeholders. Additionally, Eli will provide support to James, the hardware lead, and debugging knowledge to the rest of the team as integration takes place.

James Minardi
James has experience in several areas including embedded systems, computer graphics, machine learning, computer vision, and hardware architecture. These areas of expertise have been obtained through coursework, personal projects, and profession experience. Most recently, James interned at Garmin developing and maintaining multithreaded Vulkan & OpenGL graphics libraries for aviation flight decks. Through these experiences, James will bring a unique hardware focused perspective to the project.

Lindsey Wessel
Lindsey will use Machine Learning to track the user's face and eyes. The coordinates of the eyes will be sent to Mason for further processing. With no prior knowledge of machine learning, the self-driven, & inquisitive nature will bring success to the role. Additionally, she had years of experience in effective communication, conflict resolution, and teamwork which will help the team run effectively.

Mason Inman

Mason has experience creating custom data processors, as well as optimizing them, with his work with Kingland Systems. With a large variety of technical skills, Mason has developed a portfolio of projects: Social Networking Gym App, Cancer Prediction Models, Facial Recognition software, and many more. Mason plans on pursuing an MS in Artificial Intelligence, and hopes to utilize and expand skills in the Machine Learning space with this project.

Encouraging and Supporting Contributions and Ideas

The team will self report their achievements and contributions every week during our client meetings, by writing what they did that week on their slide and then sharing it at the beginning of the meeting. Additionally, we will all congratulate each other on both small and large accomplishments by noting and verbally or virtually congratulating/tanking them.

Identifying & Resolving Collaboration or Inclusion Issues

It is important that as a team we build an environment where team members can openly discuss any collaboration or inclusion issues they are having. The team should acknowledge and discuss those issues and create a plan to avoid them in the future.

A.5.7 Goal-Setting, Planning, and Execution

Team Goals for the Semester

Collaborate. Plan. Execute. Review. Repeat. Work as a team with agile team practices.

Strategies for Planning & Assigning Individual Work

Each team member will be an “expert” in a certain project area. They will be the primary developer for that portion of the project. However, communication and collaboration will be encouraged/required throughout the year.

Strategies for Staying on Task

Have regularly scheduled meetings to keep status of weekly meetings. If unsure or blocked on something technical, reach out to team members first, and then reach out to client or professor if needed.

A.5.8 Consequences for Not Adhering to Team Contract

Handling Infractions

If a team member fails to meet any group expectations defined in this document, the situation should be discussed as a team before any escalation. The team should prioritize the issue so as to not delay progress, and determine the cause. Solutions may only require an open discussion with the team or better planning. Under extenuating circumstances, the issue may require escalation to the professors or advisors for guidance.

Continued Infractions

Given that continued infractions have occurred following open team discussions about the issue, the issue should be escalated to the professors and advisors to seek further guidance and figure out how to move forward.

---

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*

b) *I understand that I am obligated to abide by these terms and conditions.*

c) *I understand that if I do not abide by these terms and conditions, I will suffer the*

consequences as stated in this contract.

1)	Eli Ripperda	Date: 9/17/2024
2)	James Minardi	Date: 9/17/2024
3)	Lindsey Wessel	Date: 9/17/2024
4)	Mason Inman	Date: 09/17/2024