EE/CprE/SE 491 DESIGN DOCUMENT

Video Pipeline for Machine Computer Vision

Team Number: sdmay25-01
Advisors: Dr. Jones and Dr. Zambreno
Client: JR Spidell

Team Members:

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

Table of Contents:

1. PROBLEM STATEMENT

People with mobility and cognitive impairments, such as Cerebral Palsy, face significant challenges in maintaining independence and safety. Traditional wheelchairs often 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 wants to address these issues by developing assistive wheelchair technologies with features such as advanced mobility assistance and real-time seizure detection. This system aims 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 vision.

2. INTENDED USERS

The primary users of our client's vision for assistive wheelchair technologies will be wheelchair-bound individuals, such as those with Cerebral Palsy, along with their caregivers and healthcare providers. These technologies enhance autonomy, improve safety with real-time seizure detection, and alleviate caregiver stress. However, the specific subsystem we are developing, designed to detect and locate the user's eyes in real time, will primarily serve as a foundational tool for our client and future engineering teams. This subsystem will provide high performance and accuracy to help our client's long-term goal of assisting wheelchair users and their caregivers in everyday life.

User 1 – The Client

User Description

Our client is a highly respected software engineer who formerly volunteered to help individuals with cerebral palsy. He is the primary source of project requirements and holds the high-level vision of the project.

Empathy Map

Due to his successful 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 very 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 a senior 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.
- They are passionate about student development and value 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.

User 2 – Future Engineering Teams

User Description

This is a long-term project with over twenty senior design teams contributing in the past decade. There will be many more teams contributing in the future. Our team's starting point for this project is where the last team left off. Similarly, the progress our team makes is where the next team will begin their work.

The future engineers – senior design teams – are university students who will begin working on this project at the beginning of a semester and will contribute to this project for a total of two semesters.

Empathy Map

These students balance a lot. They will be taking multiple classes, possibly working multiple jobs, and might be 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.

User 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.

3. REQUIREMENTS, CONSTRAINTS, & STANDARDS

3.1 Requirements & Constraints

Functional Requirements

Frames Per Second

Our client provided a functional requirement for the system to have a throughput capable of processing at least [NDA] frames per second (fps). A high frame rate ensures the system can capture and analyze eyes accurately, such as blinking and fast movement.

Low Latency

The system must be capable of processing images faster than it takes them in, preventing backlogs and ensuring smooth operation. If the system becomes overloaded with old frames, it could miss important events such as rapid eye movement or blinks. High latency would hinder our ability to fulfill the goal of creating a real-time system.

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.

Resource Requirements

FPGA - Ultra96 Board (constraint)

The system must be able to run on an Ultra96 board. This constraint ensures the system can be easily replicated and is not abundantly expensive.

USB Camera or Sensor Input (constraint)

The system must receive images of the user from a USB Camera or Sensor as provided by our client.

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 power-point presentations over system frameworks to be used within the system, environment and package management, and the system design itself. Along with power-point presentations, traditional word documents, like this, will be used for note taking and resource gathering.

Aid in Project Handoff

This is a non-functional requirement. Our team is the 22nd Senior Design team to work on this overall project. Our team is building off of the previous team's finished projects. There will be many senior design teams after our team. Therefore, we must help the next team(s) "get up to speed" on their project. This will include meeting with the next team to show them what we documented, to show hardware configurations, to answer questions, etc.

3.2 Engineering Standards

Importance of 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.

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.

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.

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.

4. PROJECT PLAN

4.1 Project Plan

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 speaking, James and Eli will be leading in the Tensil-AI and Hardware Implementation work, Lindsey will be working on the Region of Interest algorithm, and Mason will be in charge of the Semantic Segmentation Machine Learning Model.

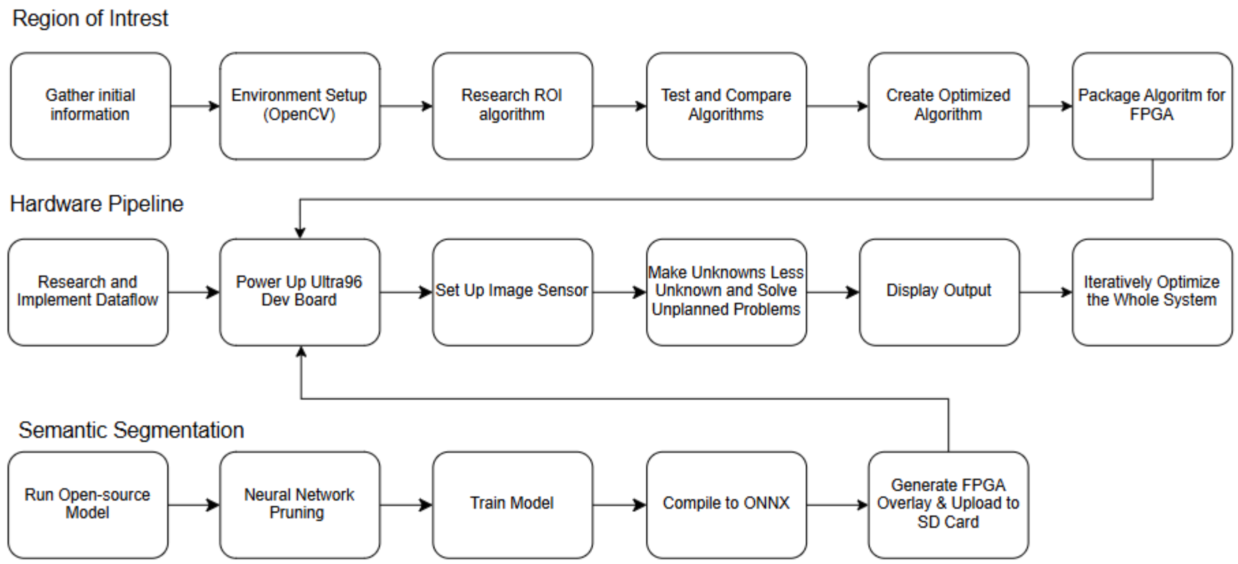
Project Management

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.

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.


Region of Intrest
Gather initial information
Environment Setup (OpenCV)
Research ROI algorithm
Test and Compare Algorithms
Create Optimized Algorithm
Package Algoritm for FPGA
Hardware Pipeline
Research and Implement Dataflow
Power Up Ultra96 Dev Board
Set Up Image Sensor
Make Unknowns Less Unknown and Solve Unplanned Problems
Display Output
Iteratively Optimize the Whole System
Semantic Segmentation
Run Open-source Model
Neural Network Pruning
Train Model
Compile to ONNX
Generate FPGA Overlay & Upload to SD Card

Project 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 with a total price under $2,500, and small enough to fit on a wheelchair. Subsequently, our team chose to work with a ULTRA96 FPGA board. In order to reach real-time functionality our software must be efficient enough to run on the FPGA board with high accuracy and low latency. We are utilizing two machine learning algorithms, and optimizing them, to reduce computational expense and increase the speed of throughput.

Environment and Hardware Setup

Our goal for the first semester is to get the Hardware Pipeline running up to displaying output. To do this, Our team must set up the hardware and different development environments. This includes receiving the hardware, understanding what we are working with, and familiarizing ourselves with the operating system on the Ultra96 Dev Board.

Locate the Region of Interest

The Region of Interest Algorithm should take in an image once a second and, with a minimal level of computations (to be determined after research), output the (x,y) coordinates of any eyes in the image. The eyes do not need to be detected when closed, partially obstructed, or facing away from the camera; the algorithm should return a corresponding value.

Obtain and Optimize the Semantic Segmentation Algorithm

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

By the end of the semester, the model will have at least one optimization implemented, and then a prototype will be trained to five epochs or less. Metrics will be recorded and further research and optimization will be performed subsequently.

By the end of the project, the model will be capable of processing images up to 240 frames per second, while using the four cores of the Ultra-96.

Display Output

In order to understand how well our system performs, it is necessary to see the system's output. This is necessary to analyze performance and debug. The specifics of executing this task are to be determined. Additional research is needed here. It is likely we will port finished data/images to a data storage and/or to a monitor.

Upload ML and ROI Algorithms to the FPGA board

Using Tensil.ai we will compile the ML model onto the FPGA board. This will run on FPGA startup. Additionally, the ROI algorithm will be loaded onto the FPGA dev board storage that, at runtime, it can be utilized.

Run Real-Time System

The whole system should process images at a rate of over 240 frames per second. This threshold is important because it is the starting point for the ability to gather data on human stress and cognitive overload.

Project Timeline

Gantt Chart

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

<table>
<tr><th rowspan="2">WBS #</th><th rowspan="2">Task Title</th><th rowspan="2">Task Owner</th><th colspan="9">Week</th></tr>
<tr><th>1-7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td><b>1</b></td><td colspan="11"><b>Environment and Hardware Setup</b></td></tr>
<tr><td>1.1</td><td>Meet Client</td><td>All</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1.2</td><td>Get Hardware From Client</td><td>James</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1.3</td><td>Setup PYNQ on Ultra96</td><td>James</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>1.4</td><td>Acclimate to Project</td><td>All</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td><b>2</b></td><td colspan="11"><b>Run Compiled ML Model on FPGA</b></td></tr>
<tr><td>2.1</td><td>Create Compiled ML Model</td><td>James</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.1.1</td><td>Research Tensil.ai</td><td>Eli</td><td colspan="3">X</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.1.2</td><td>Make first Compilation</td><td>James</td><td></td><td></td><td></td><td colspan="2">X</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.1.3</td><td>Compile Mason's Model</td><td>James</td><td></td><td></td><td></td><td></td><td colspan="2">X</td><td></td><td></td><td></td></tr>
<tr><td>2.2</td><td>Download Model onto FPGA</td><td>James</td><td></td><td></td><td></td><td></td><td></td><td colspan="2">X</td><td></td><td></td></tr>
<tr><td>2.3</td><td>Connect & Setup I/O for ML Model</td><td>Eli</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.3.1</td><td>Connect & Setup Webcam</td><td>Eli</td><td></td><td></td><td colspan="2">X</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.3.2</td><td>Understand & Define ML Model Output</td><td>James</td><td></td><td></td><td></td><td></td><td colspan="2">X</td><td></td><td></td><td></td></tr>
<tr><td>2.3.3</td><td>Determine How to Display Output</td><td>Eli</td><td></td><td></td><td></td><td></td><td></td><td colspan="2">X</td><td></td><td></td></tr>
<tr><td>2.3.4</td><td>Display Output</td><td>James</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">X</td><td></td></tr>
<tr><td><b>3</b></td><td colspan="11"><b>Region of Interest (ROI)</b></td></tr>
<tr><td>3.1</td><td>Gather Initial Understanding</td><td>Lindsey</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.2</td><td>Environment Setup</td><td>Lindsey</td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3.3</td><td>Research</td><td>Lindsey</td><td></td><td colspan="5">X</td><td></td><td></td><td></td></tr>
<tr><td>3.3.1</td><td>Compare Algorithms</td><td>Lindsey</td><td></td><td></td><td colspan="5">X</td><td></td><td></td></tr>
<tr><td>3.4</td><td>Create Optimized Solution</td><td>Lindsey</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">X</td></tr>
</table>

3.5	Test Solution & Refine	Lindsey									X
4	Semantic Segmentation Optimization										
4.1	Research	Mason	X								
4.2	Environment Setup	Mason				X					
4.3	Obtain Training Data	Mason						X			
4.4	Obtain a baseline model to put on board	Mason							X		
4.5	Identify/Research points of optimization	Mason								X	
4.6	Implement Optimizations	Mason									X
4.7	Train model with Optimizations	Mason									X

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.

We've divided our project into four main parts:

Environment and Hardware Setup

This task includes meeting with the client, setting up the Ultra96 v2 with PYNQ, and refining project requirements.

Run ML Model on the FPGA

For this task, the team will focus on creating, compiling, and implementing an ML model onto the FPGA, as well as setting up necessary I/O. This stage involves researching tools, compilation, connectivity, and display methods.

Implement Region of Interest (ROI) 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.

Optimize Base Semantic Segmentation Model

For this task, we will gather training data, obtain a baseline model, and identify and implement optimizations. The model will then be retrained with these optimizations to increase model efficiency and accuracy.

Risks and Risk Management

Below, we've identified salient risks for each main task in our project, with estimated probabilities for each risk factor. It is important to note that there are still a lot of unknowns, so it is difficult to accurately estimate risks, 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.

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.

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 fail 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.

Region of Interest (ROI)

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.

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.

Personnel Effort Requirements

Task Breakdown with Estimated Man-Hours

WBS Number	Task Title	Estimated Man-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.

5. Design

5.1 Design Exploration

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 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 now 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 video frame rate of [NDA] fps, they reduced their video size to _x_. So that we can build off of this solution while looking at a human eye, the ROI Algorithm will locate the eye and then reduce 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 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 SDMay25-01's role, play to the strengths of the U-net model while optimizing its weaknesses, the speed.

Ideation

Neural Network Pruning Techniques

Pruning a neural network is rather complex, and there are a couple of generalized techniques and technologies that are commonly used. Our research primarily focused on TensorBoard, Grad-CAM, and the Vitis-AI technologies.

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. Our team plans to utilize the knowledge from TensorBoard and Grad-CAM research while leveraging the Vitis-AI tools to prune the model.

ROI Algorithm: Bright Eye - Dark Eye

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: Haar Like Features

This algorithm uses high contrast lighting to locate facial regions, like eyes. It is extremely fast with only x^2 operations due to its bounding boxes and superior use of linear algebra. However, the spread results in lower accuracy.

Decision-Making and Trade-Off

Weighted Decision Matrix:

Criteria	Weight	Camera Sensor (IMX219)	ROI Algorithm (Bright Eye - Dark Eye)	ROI Algorithm (Haar-like Features)	Semantic Segmentation (U-Net)	Neural Network Pruning (Vitis-AI)
Accuracy	0.3	9	8	6	10	7
Speed	0.25	7	8	10	6	9
Ease of Integration	0.15	8	5	7	8	6
Cost	0.2	6	10	10	10	10
Client Approval	0.1	10	5	5	10	8
Averages		8	7.2	7.6	8.8	8

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 ROI Algorithm scores, the Haar algorithm received the highest score out of the algorithms currently being researched. Vitis-AI has a much lower score than the U-Net model as a whole since its integration is much lower. These scores depict the strengths and weaknesses of the processes noted.

5.2 Proposed Design

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. Our system is built around a few key components that work together to achieve these goals.

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 each frame. The results and statistical data will then be displayed on a monitor for testing and demonstration purposes.

Detailed Design and Visuals

Our system consists of a few key components including a camera sensor, FPGA board, region of interest cropping algorithm, and pupil locating machine learning model. These subsystems are integrated to continuously capture and process eye data for later analysis. Below is a figure describing the real-time data flow in our system, followed by a high-level block diagram.

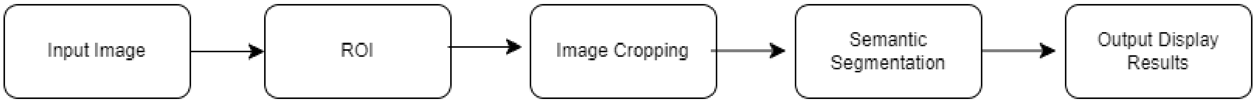


Figure 5.1: System Data Flow

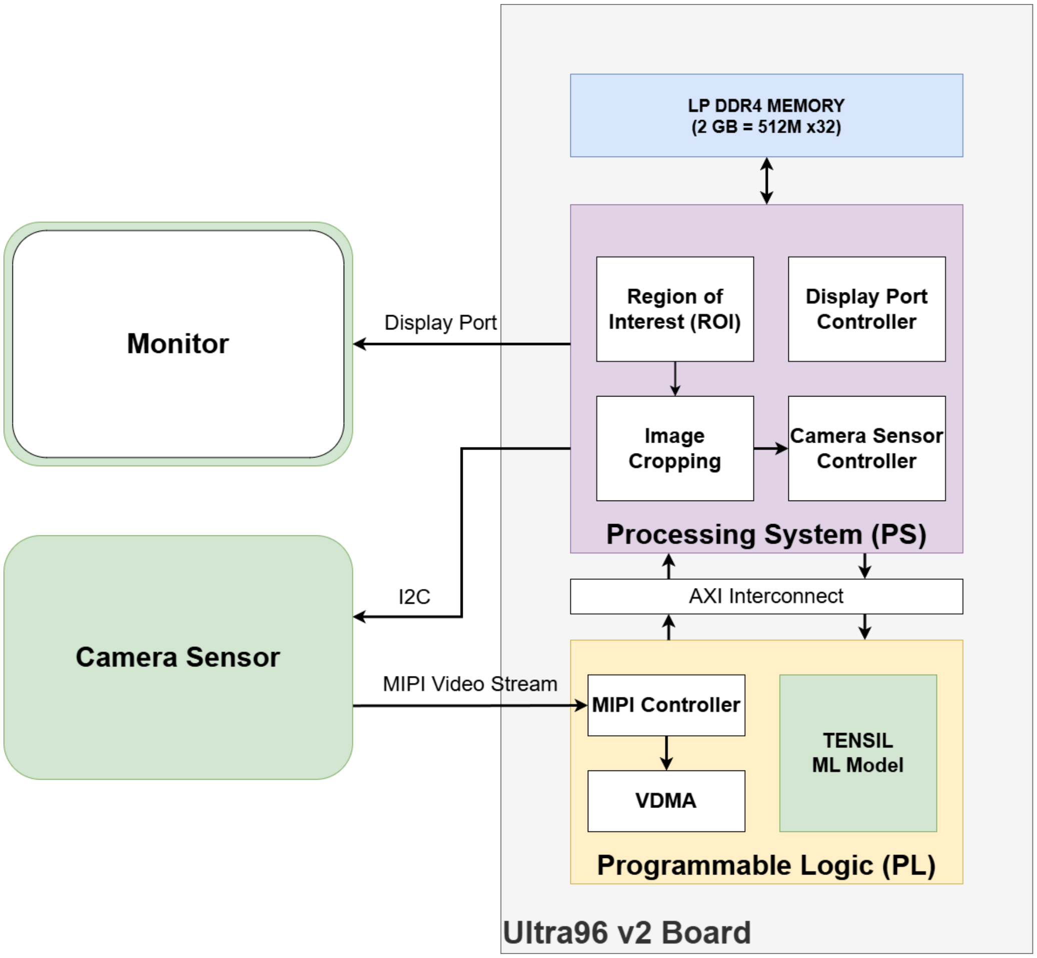


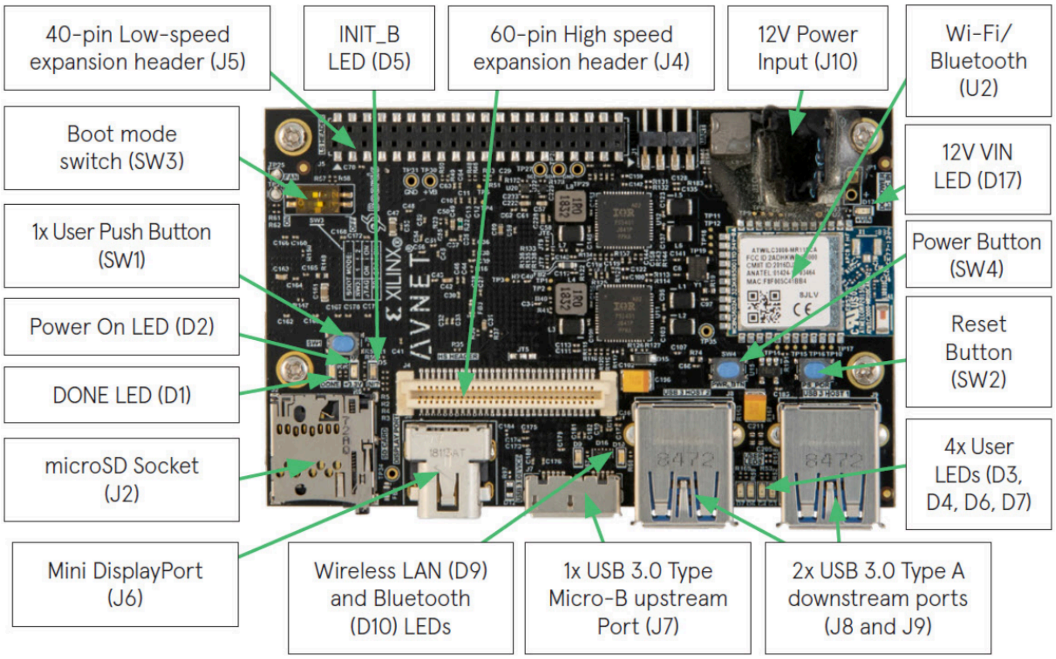
Figure 5.2: High-level Architecture

Above is a high-level block diagram of the system architecture, showcasing each of these primary components:

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 then be sent to the Ultra96 v2 board.

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)
4x User LEDs (D3, D4, D6, D7)
microSD Socket (J2)
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 5.3: Ultra 96 v2 Topology

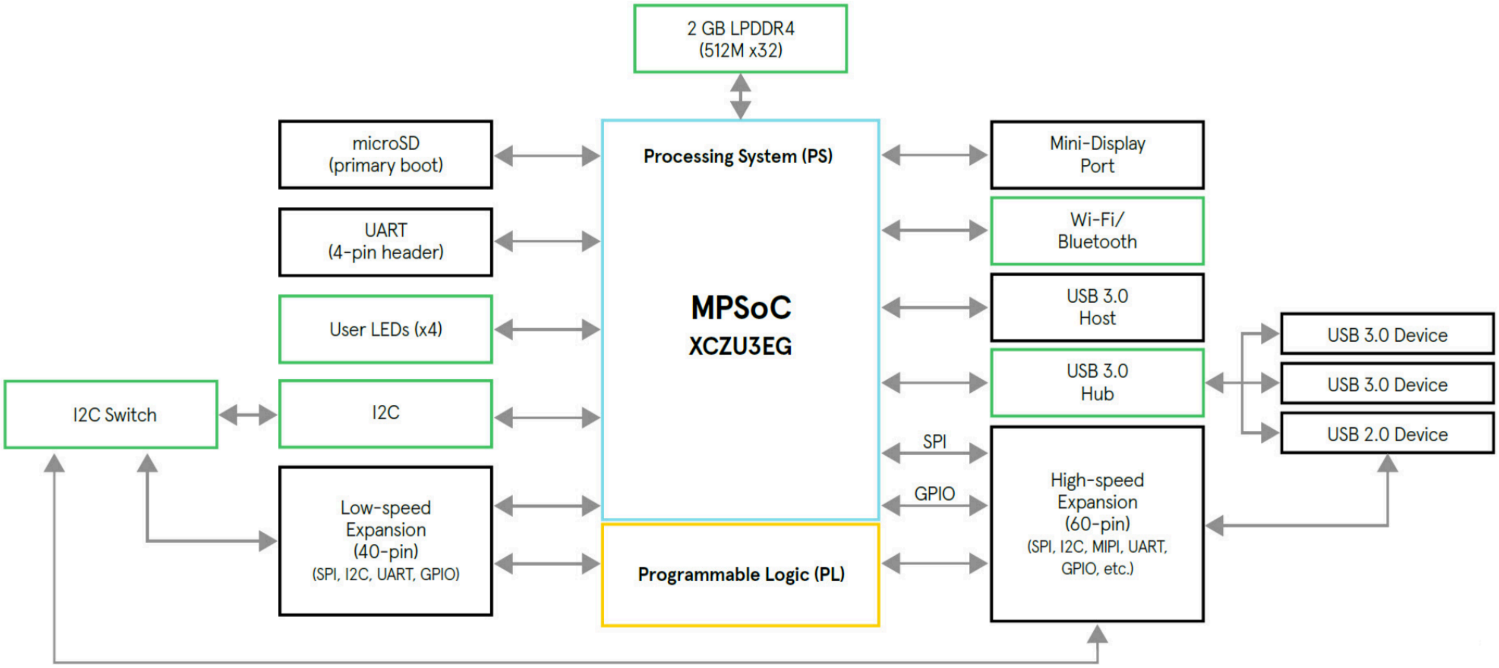


Figure 5.4: Ultra96 v2 Block Diagram

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 Tensil.

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

ALGORITHM	# COMPUTATIONS
	x = image resolution
		TIME (ms)	NOTES

Haar Like Features	x^3	11500	
YOLO	x^2 + 4x	1,850	80% accuracy
Generic Eye	x^2	123,500	
Eigen Face-Eye		12,100	

Figure 5.5: ROI Algorithm Comparisons

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.

Semantic Segmentation ML Model

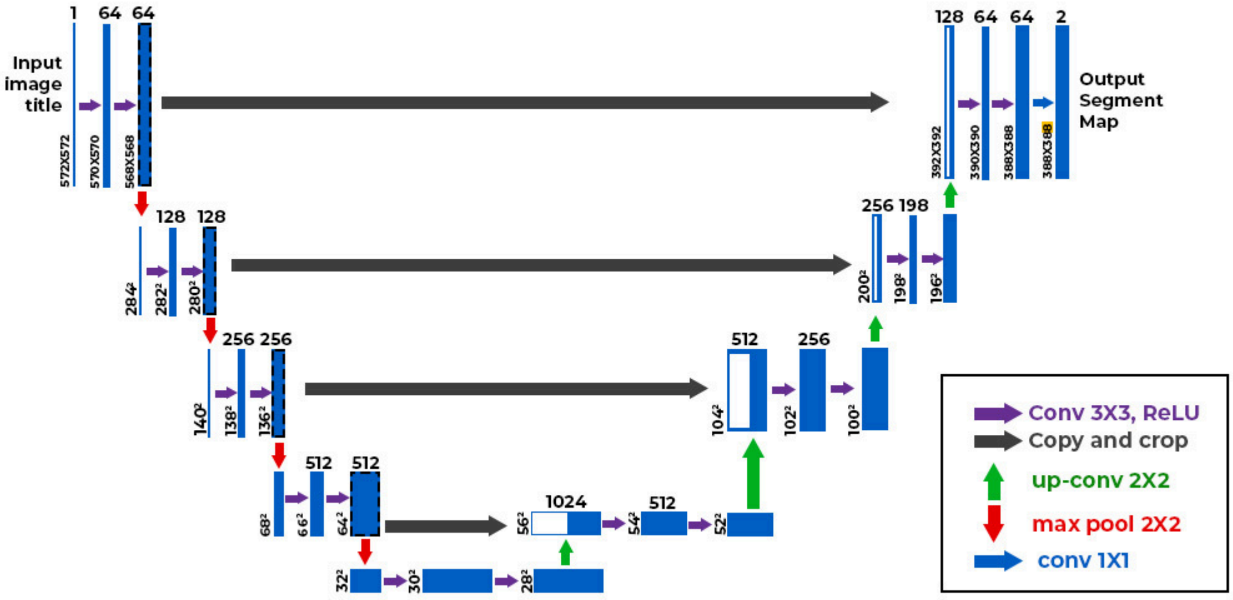


Figure 5.6: U-NET Convolutional Model Diagram

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).

Tensil

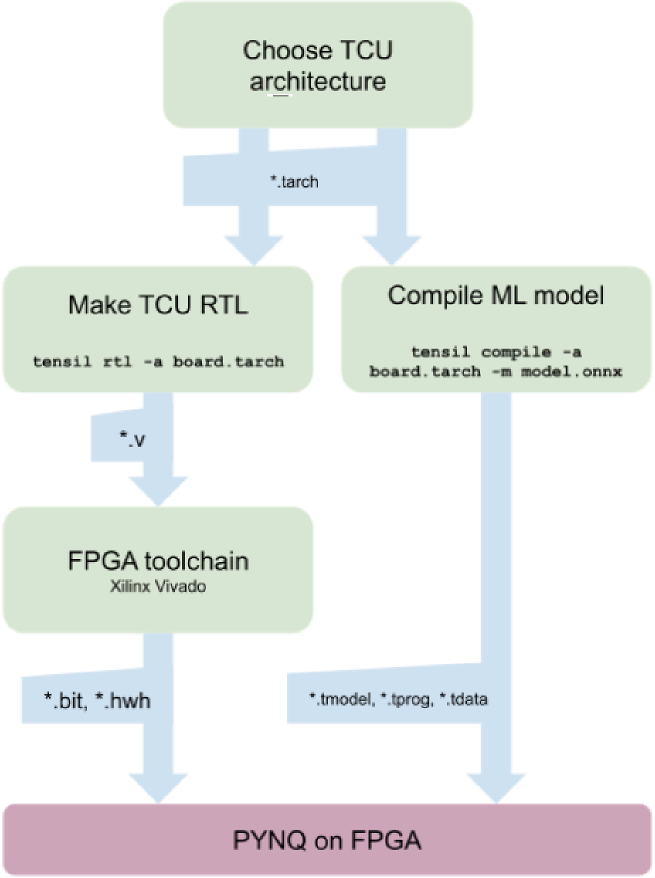


Figure 5.7: Tensil-AI Architecture Flow Diagram

Tensil is a set of tools that allows machine learning models to run on custom accelerator architectures. It includes an RTL generator, a model compiler, and a set of drivers. The semantic segmentation model will be compiled into a custom accelerator that will run on the Ultra96 v2 FPGA platform.

The ML model is trained and generated in PKL format. To compile with Tensil, the trained model will be converted to ONNX format. Once compiled on Tensil, the Tmodel, Tprog, and Tdata files will be ported into the PYNQ environment on the Ultra96 v2.

5.3 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.

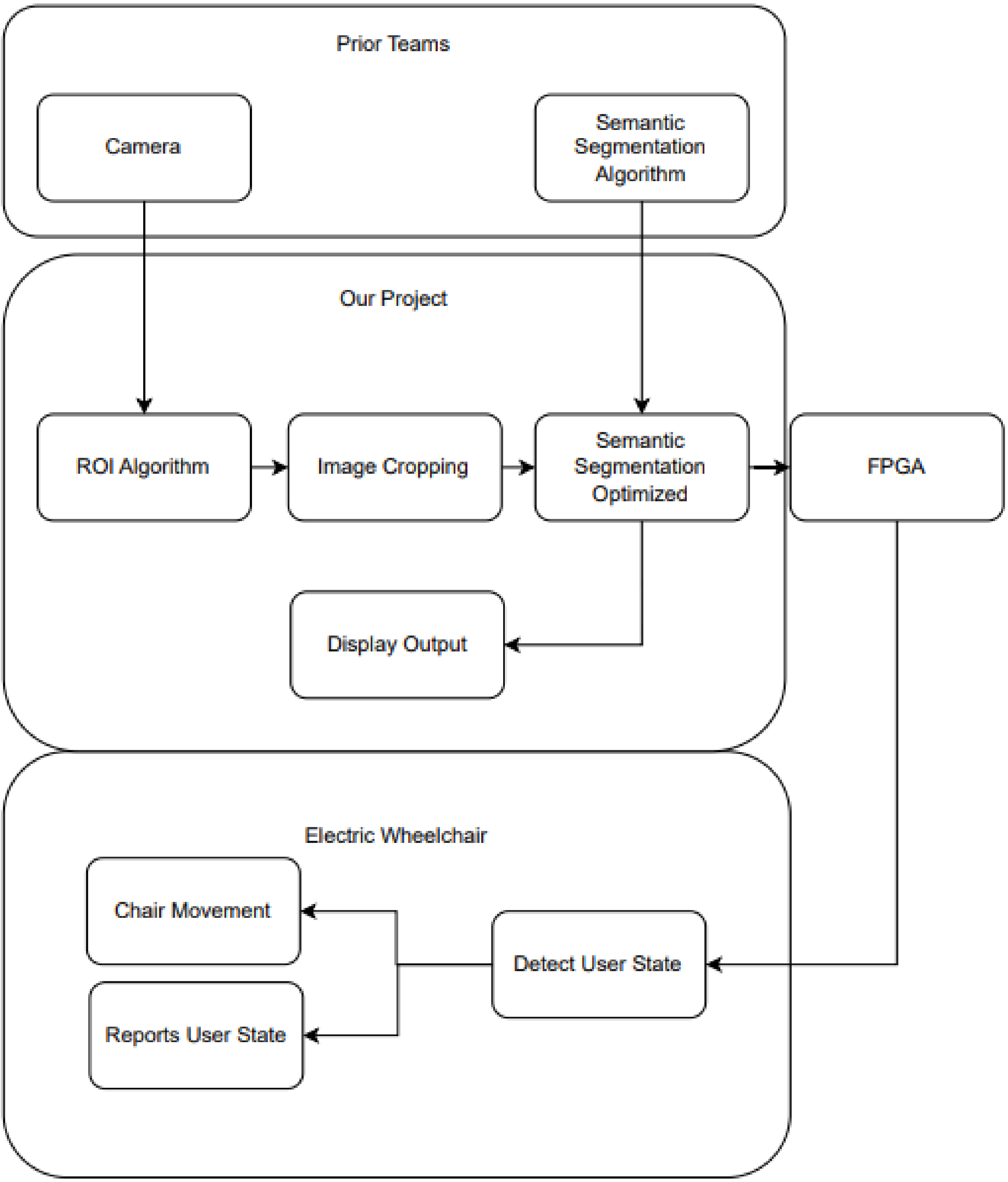


Figure 5.8: Functionality Diagram Timeline

Areas of Concern and Development

Our current design is well-aligned with the client's requirements, particularly in meeting real-time, accurate performance for assistive technology. However, our team has concerns with seamless integration of subsystems including the camera sensor, ROI algorithm, Tensil, and semantic segmentation model. It will be important for the team to smoothly transfer data between components without introducing latency, as any bottleneck could negatively affect performance.

Besides concerns for overall system integration, our team meets with the client on a weekly basis to address any concerns; they are typically resolved immediately, or at the very least possible next steps are discussed. We meet with our advisors on an as-needed basis, which is generally bi-weekly. During periods where we do not meet with advisors, we update them on our status via our weekly report. These recurring meetings mitigate many outstanding questions our team has.

5.4 Technology Considerations

Ultra96v2

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.

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.

Tensil-AI

Strengths: Accelerates AI inference on FPGA, integrates well with Ultra96v2, boosts performance for ML models on limited hardware.

Weaknesses: Limited flexibility for non-AI applications, requires optimized integration on Ultra96v2.

Trade-Offs: Balancing model performance with FPGA resource limits may require simplifying architecture.

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.

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.

Design Solutions & Alternatives

Based on the strengths, weaknesses, and trade-offs listed above, our team will work to find the line that maximizes technology strengths while minimizing weaknesses. Additionally, some of the weaknesses listed can potentially be strengths in our optimization efforts. For example, losing data that does not affect output and maintains accuracy would be a successful optimization, rather than a weakness. Alternatives may be discussed and elaborated upon given further research, however, based on current research these are the tools and technologies that will be utilized by our team.

5.5 Design Analysis

As this project is a continuation of several senior design projects throughout the country, there is a large amount of work SDMay25-01 is able to build off. During the first semester the main priority was gaining an understanding of the complex tools, technologies, algorithms, and models being utilized. Each team member has been assigned a role to be the team leader in that subject. This allows for specialization, while still encouraging collaboration between team members.

Through iterations of research, design, and prototyping we have our baseline model successfully ported to ONNX format and compiled in Tensil. Additionally, the Ultra96v2 has a setup PYNQ environment that can connect and run Jupyter Notebooks.

By the end of the semester our goal is to have a working (unoptimized) prototype. We plan to utilize the success of ISU's VPIPE senior design team to base our video pipeline off of (it is important to note this team is working with the same client).