HAML

Heterogeneous and Accelerated Computing for Machine Learning

DESIGN DOCUMENT

Senior Design Team sddec24-05

Client: JR Spidell

Advisor: Phillip Jones

Jonathan Tan – Kira Board Manager, Petalinux Developer

Josh Czarniak – DPU Manager Developer

Justin Wenzel – Multithreaded Program Developer

Kai Heng Gan – Semantic Segmentation Developer

Santiago Campoverde – Model Analyst

Team email: sddec24-05@iastate.edu

Team Webpage: https://sddec24-05.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

Circuit Design and Integration - Understanding the hardware components of the AMD Kria KV260 Vision AI Starter Kit and their interconnections.

Hardware Setup - Connect and configure the AMD Kria KV260 Vision AI Starter Kit with the workstation. Ensuring power requirements are met and managing power distribution if necessary.

Version Control - Using version control systems like Git for managing source code, hardware configurations, and project documentation.

Hardware Testing - Conduct thorough testing of the hardware setup to ensure proper functioning and compatibility.

Software Installation and Configuration - Installing the required software components such as TensorFlow, Docker, Xilinx Vitis, OpenCV, Python, and C/C++ compilers on the workstation. Ensuring compatibility and resolving any dependency issues.

Software Development Practices - Following best practices like modular design, documentation, and coding standards. Utilizing design patterns and frameworks appropriate for machine learning and computer vision tasks.

Containerization - Utilizing Docker for containerizing the software environment, ensuring consistency across different systems and simplifying deployment.

Development Environment Setup - Configuring the development environment for software development, including IDEs, text editors, and debugging tools.

Cross-Compilation - Setting up cross-compilation tools if targeting the AMD Kria KV260 Vision AI Starter Kit with different architectures.

Testing and Validation - Conducting rigorous testing of the software components, including unit, integration, and system tests. Validating the performance of machine learning models using appropriate metrics and datasets.

Optimization - Optimizing software performance for the target hardware platform, leveraging tools like Xilinx Vitis for profiling.

Security Considerations - Implementing security measures such as access controls, encryption, and secure communication protocols to protect sensitive data and systems.

Documentation - Maintaining comprehensive documentation covering hardware configurations, software setup, development workflows, and deployment procedures.

Continuous Integration and Deployment (CI/CD) - Implementing CI/CD pipelines for automated testing, build, and deployment processes to streamline development and ensure consistency.

Maintenance and Support - Establishing procedures for ongoing maintenance, updates, and support of the deployed system, including troubleshooting and bug fixing.

Standards:

- IEEE 3129-2023 - IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service
- IEEE 2802-2022 - IEEE Standard for Performance and Safety Evaluation of Artificial Intelligence Based Medical Devices: Terminology
- IEEE 7002-2022 - IEEE Standard for Data Privacy Process
- IEEE 3156-2023 - IEEE Standard for Requirements of Privacy-Preserving Computation Integrated Platforms
- IEEE 2842-2021 - IEEE Recommended Practice for Secure Multi-Party Computation
- IEEE 2952-2023 - IEEE Standard for Secure Computing Based on Trusted Execution Environment
- IEEE 1484.1-2003 - IEEE Standard for Learning Technology - Learning Technology Systems Architecture (LTSA)

Summary of Requirements

Hardware Requirement:

- AMD Kria KV260 Vision AI Starter Kit
- A workstation with a native Linux OS

Software Requirement:

- TensorFlow
- Docker
- Xilinx Vitis
- OpenCV
- Python
- C/C++

Collaborative Tools/Documentations Requirement:

- Version Control System - Git
- Project Management - Trello
- Microsoft Teams/Telegram
- Technical Documentation
- Group Documentation

Applicable Courses from Iowa State University Curriculum

CPR E 185 – Introduction to Computer Engineering and Problem Solving I

CPR E 186 – Introduction to Computer Engineering and Problem Solving II

CPR E 281 – Digital Logic

CPR E 288 – Embedded Systems I: Introduction

CPR E 308 – Operating Systems: Principles and Practice

CPR E 381 – Computer Organization and Assembly Level Programming

CPR E 414 – Introduction to Software Systems for Big Data Analytics

CPR E 424 – Introduction to High-Performance Computing

CPR E 425 – High-Performance Computing for Scientific and Engineering Applications

CPR E 487 – Hardware Design for Machine Learning

CPR E 488 – Embedded Systems Design

CPR E 527 – High-Performance Deep Learning

New Skills/Knowledge acquired that was not taught in courses

- OpenCV
- CUDA
- Ritnet
- Vitis-AI
- Petalinux/Yocto

Table of Contents

List of figures

1 Introduction

1.1. Problem Statement

Machine learning offers a versatile toolkit for a wide range of applications. In the context of our project, our client, JR Spidell, wants to develop a system capable of monitoring the focal point of an individual's gaze. The current primary objective of this endeavor is to analyze the movement of a user's pupils. This initial foundation opens the door to other applications such as monitoring emotions, visual aid assistance, and within different health fields and further general development of computer vision applications.

To achieve this, three models must be run simultaneously: image preprocessing, blink detection, and eye tracking. All models must run simultaneously as each model depends on the other to monitor the user's current state of their eyes from frame to frame of input. This project has benefitted from the contributions of many teams; previous teams successfully executed some of the models independently on the Kria board, providing some results and proof of concept for implementation. Our challenge, however, is to design a comprehensive program capable of concurrently operating all three models.

This task is complicated by the hardware constraints of the Kria board, which, at the moment, is equipped with a single Deep Learning Processing Unit (DPU). Both the blink detection and eye tracking models rely on the DPU. On top of that, by running multiple resource-heavy models simultaneously, other resources, such as memory and processor cores, have to be managed delicately.

With prior teams building a foundational code base and implementing the ability for some of the models to perform inferences while individually running on the Kria board. Our objective is to design a system that optimizes the distribution of the resources on the board across the different models to achieve a target throughput of 200 frames per second (FPS) while all models are running. Our design will perform multiple inferences on different frames simultaneously while also passing the frames across the different models. This implementation will handle all necessary inferences required to determine the state of the users' eyes in one pass, utilizing different techniques such as multi-threading, memory management, and proper resource allocation to achieve the target throughput. 200 FPS is deemed the minimal requisite for the system to fulfill its intended function effectively.

1.2. Intended Users

The product we are developing finds its significance across multiple user demographics, each with distinct characteristics and expectations. These include our primary client, with a vision for advancing research and educational opportunities, Dr. Mark Johnson, focusing on enhancing patient treatment through eye tracking, and a hypothetical third client, a software development company specializing in educational tools.

Primary Client: The Researcher and Benefactor

- **Persona:** Our primary client, JR Spidel, is an academic researcher and alumni of our university, dedicated to the innovative field of emotion detection via eye tracking. He is motivated by the dual objectives of contributing to groundbreaking research and offering practical learning experiences to students.

- **Needs:** The client aims to harness pupil-tracking technology to decipher emotional states, thereby advancing the understanding and application of this technology in various fields.
- **Benefits:** By achieving this, the client not only furthers research in an emerging area but also enriches the academic community by providing students with a hands-on project that bridges theoretical knowledge and real-world application. This alignment with the project's goal of fostering an educational ecosystem while pushing the boundaries of machine learning applications.

Client 2: Dr. Mark Johnson, The Medical Researcher

- **Persona:** Dr. Mark Johnson is a forward-thinking doctor engaged in pioneering research integrating eye tracking with patient care. He is deeply invested in exploring the correlation between eye movements and emotional states to improve both physical and mental treatment strategies.
- **Needs:** Dr. Johnson seeks innovative tools that can provide new insights into patient emotions, potentially unlocking more holistic treatment approaches.
- **Benefits:** The product promises to enhance Dr. Johnson's research by offering precise, real-time data on eye movement patterns, facilitating a deeper understanding of patient emotions. This could revolutionize the way mental and physical ailments are approached, directly linking our project to the broader aim of improving healthcare outcomes.

Client 3: EdTech Innovations, The Educational Software Company

- **Persona:** EdTech Innovations is a company at the forefront of developing educational software that incorporates cutting-edge technologies to create more engaging and effective learning experiences. Their team is composed of educators, developers, and researchers who are passionate about harnessing technology to enhance education.
- **Needs:** They are interested in exploring how eye tracking can be used to assess student engagement and comprehension in real time, tailoring educational content to individual needs for optimized learning outcomes.
- **Benefits:** Integrating our pupil-tracking technology could enable EdTech Innovations to develop applications that adapt to students' emotional and cognitive states, making learning more personalized and effective. This potential to revolutionize educational methodologies and tools reflects our overarching objective of leveraging machine learning to address complex, real-world problems in innovative ways.

Each of these user groups stands to gain substantially from our project. The researcher and benefactor, by achieving a groundbreaking stride in emotion detection; Dr. Mark Johnson, by opening new avenues in patient care; and EdTech Innovations, by pioneering personalized learning experiences. Together, they encapsulate the multifaceted impact of our work, highlighting its relevance to academic research, healthcare, and education.

2 Requirements, Constraints, and Standards

2.1. Requirements & Constraints

Functional Requirements

1. **Model Integration:** Successfully integrate and concurrently run the three specified models (image preprocessing, blink detection, and pupil tracking) on the Xilinx Kria KV260 board.
2. **System Throughput:** Achieve a system throughput of less than 200 FPS. This metric is crucial to ensure real-time processing and analysis capabilities, which are imperative for the system to function effectively in its intended applications.

User Interface (UI) Requirements

1. **Command Line Interface (CLI):** Design a user-friendly CLI for both technical and non-technical users.
    a. Outline clear commands to allow the user to interact and control configurations within the system.
    b. Help commands to assist the user and provide descriptions for different commands.
2. **Command Feedback:** Provide immediate and clear feedback based on each command, displaying the results or errors encountered during execution. Helping the user identify their input's impact on the system's state.
3. **Error Handling:** Provide detailed error handling and logging mechanisms. Errors will be clearly communicated to the user to assist with fixes, and logs will be detailed to assist the user in debugging and adjusting the system for expected performance.

Physical and Economic Requirements

1. **Hardware Compatibility:** All components for the time being will be compatible with the Xilinx Kria KV260 board, keeping hardware adjustments minimal for current costs.
2. **Cost-Effectiveness:** Design a system that is economical in its investment and for future maintenance and updates.

System Constraints

1. **Memory Limitations:** The Xilinx Kria KV260 board has 5GB of DDR memory. This finite resource must be allocated among the three models.
2. **FPGA Resource Allocation:** The available FPGA space for deploying our Deep Learning Processing Unit (DPU) is limited. Efficient use of FPGA resources is essential to accommodate the computational demands of our models without exceeding the board's capacity.
3. **Single DPU Utilization:** Given that the board houses only one DPU, there is a need for a strategy that allows the blink detection and pupil-tracking models to share the DPU effectively.

Additional Considerations

1. **Deployment Options:** During current development, deployment will be limited to a portable device, the Xilinx Kria KV260 board.

Data Handling and Privacy: Implement strict measures to protect user data privacy and security within the system.

2.2. ENGINEERING STANDARDS

- IEEE 3129-2023 - IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service[3]
    - This standard is essential as it provides guidelines for testing the robustness of AI-based image recognition services. Robustness testing ensures that these systems perform reliably under various conditions and can handle unexpected inputs or scenarios, which is critical for their real-world deployment.
- IEEE 2802-2022 - IEEE Standard for Performance and Safety Evaluation of Artificial Intelligence Based Medical Devices: Terminology[4]
    - With the increasing use of AI in medical devices, ensuring their performance and safety is paramount. This standard establishes terminology for evaluating the performance and safety of AI-based medical devices, providing clarity and consistency in assessing their effectiveness and reliability in clinical settings.
- IEEE 7002-2022 - IEEE Standard for Data Privacy Process[5]
    - In an era of increasing data breaches and privacy concerns, this standard sets forth processes for safeguarding data privacy. It outlines best practices and procedures for handling sensitive information, ensuring compliance with regulations, and fostering trust among users and stakeholders in data-driven applications and services.
- IEEE 3156-2023 - IEEE Standard for Requirements of Privacy-Preserving Computation Integrated Platforms [6]
    - Privacy-preserving computation is crucial for protecting sensitive data while still enabling useful computations. This standard defines requirements for platforms that facilitate privacy-preserving computation, ensuring that such systems adhere to established principles and practices for safeguarding privacy in data processing and analysis.
- IEEE 2842-2021 - IEEE Recommended Practice for Secure Multi-Party Computation[7]
    - Secure multi-party computation enables parties to jointly compute a function over their inputs while keeping them private. This recommended practice provides guidance for implementing secure multi-party computation protocols, promoting the development of robust and trustworthy systems for collaborative computing in sensitive applications such as finance, healthcare, and privacy-preserving analytics.
- IEEE 2952-2023 - IEEE Standard for Secure Computing Based on Trusted Execution Environment[8]
    - Trusted execution environments (TEEs) play a crucial role in securing sensitive computations and data on computing platforms. This standard defines requirements for secure computing based on TEEs, ensuring that systems leveraging these technologies adhere to established security principles and practices, thereby mitigating the risk of unauthorized access and tampering.
- IEEE 1484.1-2003 - IEEE Standard for Learning Technology - Learning Technology Systems Architecture (LTSA)[9]
    - As learning technology continues to evolve, having a standardized architecture is essential for interoperability and scalability. This standard defines the architecture for learning technology systems, providing a framework for designing, implementing, and integrating educational software and systems, thereby facilitating collaboration and innovation in the field of online learning and digital education.

3 Project Plan

3.1 Project Management/Tracking Procedures

We've chosen an Agile project management approach to maximize flexibility and adaptability as we work towards achieving our goal of achieving a throughput of <5ms (200 fps). This methodology allows us to continually reassess our implementations and decisions, ensuring we stay aligned with evolving project requirements.

To facilitate seamless collaboration among team members, we've leveraged platforms like GitHub and Trello. GitHub serves as our primary repository, to which our client also has access, enabling them to track our progress in real time. The source codes and documentation are primarily uploaded to our GitHub repository. Meanwhile, Trello enables us to organize tasks, issues, and milestones, ensuring every team member remains on track and focused on key deliverables.

By adopting Agile methodologies and utilizing these collaborative tools, we're not only ensuring transparency and accountability within our team but also fostering a dynamic environment where innovation and iteration thrive.

3.2 Task Decomposition

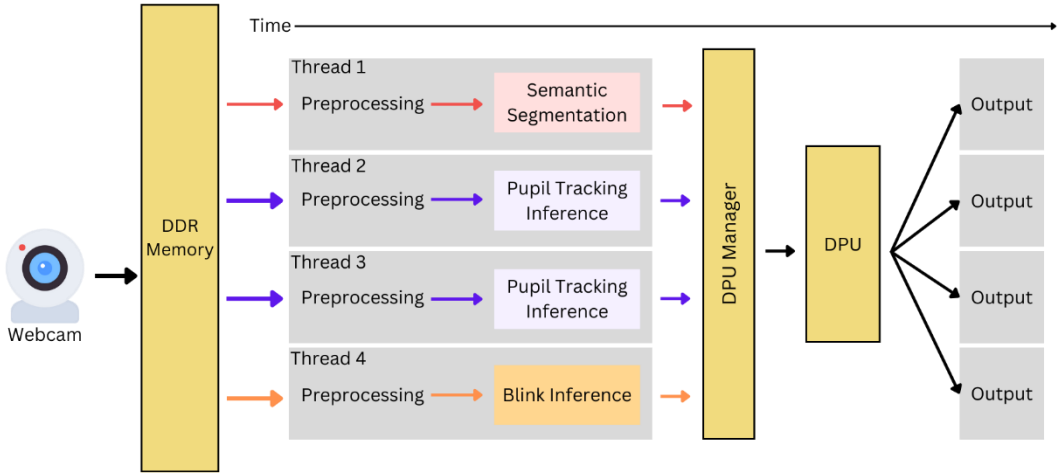


Figure 1 High level program flow

Design and Implementation (Updated)

Our ultimate goal is to achieve a throughput of ≤ 5ms (200 fps) for both blink detection and eye tracking models, while running all three of our core models—semantic segmentation, blink detection, and pupil tracking—in parallel. The diagram shown above represents our updated program design, which now focuses on a concurrent execution model to maximize efficiency and meet client performance requirements.

System Overview

Our program is designed to utilize four distinct threads running concurrently to handle the different models. We aim to leverage the Kria KV260 FPGA board's four DDR4 memory units (each 1 GB in size),

allocating one memory unit per thread to optimize data flow. The last memory unit is assigned to the CPU for other processing needs. Below is a summary of how we will distribute tasks across the four threads:

- Thread 1: Runs the Blink Detection model.
- Thread 2: Runs the Semantic Segmentation model.
- Threads 3 and 4: Both run the Eye Tracking model in parallel, with each thread processing a different set of frames to achieve the desired throughput.

Each thread processes frames concurrently, ensuring that we can achieve the desired performance for both blink detection and eye tracking, while semantic segmentation operates without the strict throughput requirement.

Thread Breakdown

Thread 1: Blink Detection

- This thread is dedicated solely to the blink detection model. The blink detection inference is applied directly to the input frames, making use of the Deep Learning Processing Unit (DPU) for accelerated processing.
- The blink detection is relatively lightweight, involving the identification of blinking status ("blink" or "no blink"). The DPU helps achieve the desired inference speed to meet the 5ms throughput requirement.

Thread 2: Semantic Segmentation

- The semantic segmentation model, previously part of pre-processing, is now executed independently in Thread 2. The goal of this component is to remove reflection or imperfection from the input frames using an image segmentation approach.
- Unlike the other models, semantic segmentation does not have a strict throughput requirement of 5ms. This allows us to prioritize computational resources towards blink detection and eye tracking.

Threads 3 & 4: Eye Tracking

- Threads 3 and 4 are both assigned the task of running the eye tracking model. This parallel approach is necessary to ensure that the eye tracking model meets the 5ms throughput requirement, given that eye tracking typically requires more extensive computation than blink detection.
- Each thread processes different input frames, applying the eye tracking inference using the DPU. The output consists of pupil coordinates (x-axis and y-axis), which are then collected and associated with the corresponding frame number.

Component Descriptions

Blink Detection

- This component is responsible for determining whether a blink has occurred. By directly utilizing the DPU for inference, we can efficiently achieve the required throughput of 5ms.

Semantic Segmentation

- This component now operates independently in Thread 2. The purpose of this segmentation is to pre-process each frame by removing imperfections such as reflections, ensuring the frame is suitable for further analysis by the other models.

- Given that semantic segmentation does not need to meet the ≤ 5ms throughput target, it allows more flexibility in its processing time, as long as it does not cause a bottleneck for the other threads.

Pupil Tracking

- Eye tracking is performed in parallel by two threads (Threads 3 and 4). This approach helps ensure that we can maintain a consistent throughput of ≤ 5ms per frame.
- The eye tracking model uses the DPU to infer the pupil coordinates, which are then aggregated with the final output for each frame.

Deep Learning Processing Unit (DPU)

- The DPU model, DPUCZDX8G_ISA1_B4096, is leveraged to perform the inference for all three models, helping us achieve the desired levels of performance.
- As a specialized machine learning accelerator embedded in the FPGA, the DPU enhances the efficiency of our design by executing the inference steps significantly faster compared to a general-purpose CPU or GPU.

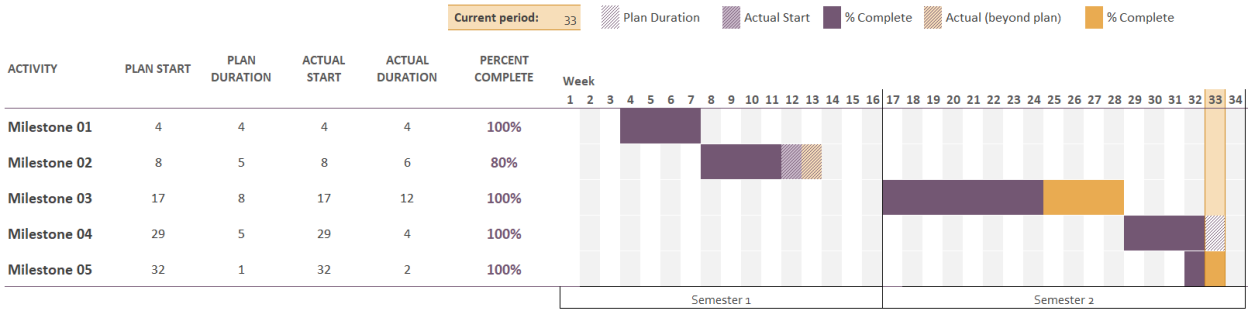
Concurrency Considerations

- By running each model in a separate thread, we are able to meet the diverse requirements of our client, particularly the ≤ 5ms throughput goal for blink detection and eye tracking.
- Thread synchronization and memory management are crucial in this design. Each thread is allocated its own DDR4 memory unit, with careful coordination to avoid race conditions and ensure smooth data flow.
- The output from each thread is stored in an array based on frame number, allowing for easy aggregation and post-processing.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Key Milestones	Evaluation Criteria
1. Understand previous team's code.	Fully understand previous team's code.
2. Combine image pre-processing, blink detection model, pupil-tracking model, into one program (serially).	Fully implement and test each model for accuracy.
3. Implement parallelism into program.	Achieve throughput of <5ms (200 fps).
4. Implement semantic segmentation for image pre-processing and retrain model.	Achieve model accuracy of top-5 95% (while maintaining <5ms throughput).
5. Implement eye tracking model with the pre-processed image	Achieve the accuracy of output from 86% to 92%.
6. Implement blink detection model (new) with the pre-processed image	Achieve the accuracy of output 99%.
7. Run three threads concurrently by initializing each thread to a memory	Successfully run three threads concurrently without fighting for memory.

3.4 Project Timeline/Schedule



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Milestone 01	4	4	4	4	100%
Milestone 02	8	5	8	6	80%
Milestone 03	17	8	17	12	100%
Milestone 04	29	5	29	4	100%
Milestone 05	32	1	32	2	100%

Figure 2 Project grand chart showing our milestones and progress

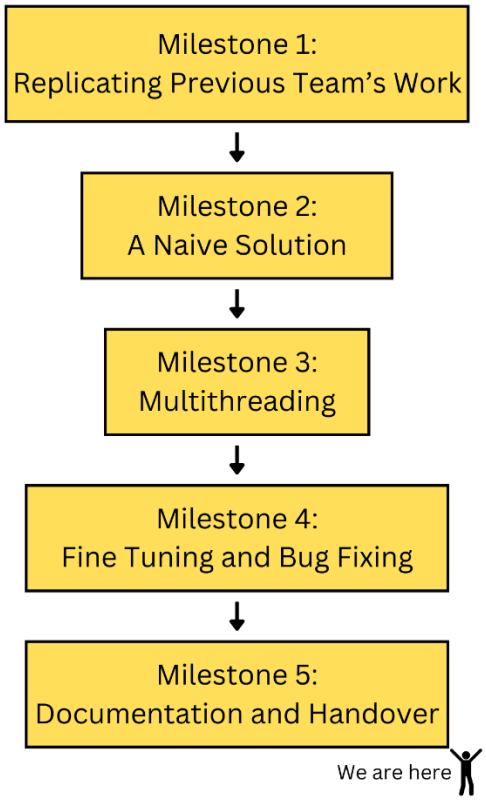


Figure 3 Milestone names

<u>Milestones Description</u>

Milestone 1:

- Replicate previous team work
- Perform simple analysis on previous team code

Milestone 2:

- Implementing a single threaded approach
- Identify bottlenecks and overheads through profiling

Milestone 3:

- Implement a multithreaded approach
- Measure model accuracy
- Perform timing analysis

Milestone 4:

- Fine tune milestone 3 code, fix bugs
- Refining semantic segmentation model

3.5 Risks and Risk Management/Mitigation

Risk	Probability	Mitigation Strategies
Not being able to complete the project in time	10%	• Regularly review and adjust project timelines during sprint retrospectives. • Implement a robust risk management plan to identify potential delays early on. • Break down project tasks into smaller, manageable chunks to track progress more effectively. • Allocate additional resources or adjust team priorities as needed to meet deadlines.
Combining multiple models into one program can increase the complexity of the software	20%	• Conduct thorough code reviews to ensure clarity, efficiency, and maintainability of the integrated models. • Implement modular design principles to encapsulate individual models, making the codebase more manageable. • Utilize comprehensive testing methodologies to validate the integration of models and identify any potential conflicts or performance bottlenecks.
Implement parallelism into the program could lead to synchronization overhead	30%	• Employ effective parallel programming paradigms such as task-based parallelism or data parallelism to minimize synchronization overhead. • Utilize synchronization primitives like locks, semaphores, or atomic operations judiciously to avoid contention and bottlenecks.

		• Profile and optimize critical sections of code to reduce serialization and maximize parallel execution.
Image semantic segmentation could take a longer time	10%	• Continuously optimize the machine learning models used for semantic segmentation to improve inference speed without compromising accuracy. Techniques such as model pruning, quantization, and architecture optimization can be explored. • Apply data augmentation techniques and preprocessing steps to the input images to reduce computational complexity without sacrificing model performance. Techniques like image resizing, cropping, and normalization can streamline inference. • Investigate model compression techniques such as knowledge distillation or weight pruning to reduce the computational requirements of the segmentation model while preserving its accuracy. This can lead to faster inference times on resource-constrained hardware.

3.6 Personnel Effort Requirements

Team member	Task	Subtask	Description	Estimated hours
Jonathan Tan	DPU Management, Petalinux Development, Timing analysis	Implement DPU sharing	Because there is only one DPU on board and multiple models share the resource, a proper DPU sharing mechanism needs to be implemented.	5
		Timing Analysis	Perform timing analysis on program	10
		Petalinux	Setup Petalinux OS with necessary packages/libraries for project	50
Josh Czarniak	Pupil tracking model, DPU Management	Modify the previous team's pupil-tracking code	The previous team's pupil-tracking code relies on the RPU; we need to remove that portion of the code and ensure that it still runs.	10
		Manage memory and	Due to the limited DDR memory on board, each eye tracking thread is only allowed 1GB of memory.	10

		limit to 1GB per thread		
		Implement DPU sharing	Implemented a proper DPU sharing mechanism for multiple models to use the DPU.	10
		Testing	Testing and debugging.	50
Justin Wenzel	Blink detection model, multithreaded application	Write the blink model	Using the blink detection model, write the program that implements the blink detection model.	10
		Manage memory and limit to 1GB per thread	Due to the limited DDR memory on board, each eye tracking thread is only allowed 1GB of memory.	10
		Multithreaded program	Developed multithreaded program framework, modular design allows for hot swapping of xmodels and datatypes	50
Kai Heng Gan	Image processing, semantic segmentation model	Optimize semantic segmentation algorithm	Optimizing the semantic segmentation algorithm for deploying the optimized model on kv260 FPGA board.	30
		Write a program for deploying semantic segmentation model on the board	Using the optimized semantic segmentation model, write a program to deploy semantic segmentation model by utilizing DPU.	30
		Testing	Testing and debugging the program to avoid errors occurred in the main function.	10
Santiago Campoverde	Profiling, model accuracy analysis	Research vaitrace profiler	Perform detail research on vaitrace profiler, which will allow team to look under the hood regarding timing analysis, and resource usage	20
		Developing testing scripts	Develop accuracy testing scripts for all three models	30
		Accuracy analysis	Perform model accuracy analysis on all three models based on testing results	20

3.7 Other Resource Requirements

Hardware Resources -

- **Xilinx Kria Evaluation Board** – For development and executing program, utilizing built-in DPU for model inferences.
- **ETG Provided Development Computer** – Provides a native Linux OS for the team to develop and test programs, including installation of other Xilinx development tools. Allows team to SSH into board from remote locations.

Software Resources -

- **TensorFlow** – An open-source machine learning library developed by Google, allowing developers to easily build and deploy machine learning models/applications.

- **Docker** – An open-source platform creating an environment that easily allows developers to develop, share, and run applications by packaging everything needed to run the software within a unit called a container.
- **Xilinx Vitis** – A software platform that assists in the development and launch of embedded software on Xilinx's different hardware, including FPGAs, in this project.
- **OpenCV** – An open-source computer vision and machine learning library. Used for many different cases but assists the project through image preprocessing.
- Python – A favored language of choice for its vast support and libraries to assist in machine learning and development.
- **C/C++** - Used within the embedded system for implementing low-level functionality and interfacing of the and between different hardware components.

Collaborative Tools/Documentations

- **Version Control System** – Using Git with the platform GitHub for code sharing among group members.
- **Project Management** – Using Trello for task management and progress checks for group members.
- **Microsoft Teams/Telegram** - Enables collaborative communication channels, and group and client communication.
- **Technical Documentation** – Utilizing different technical documents to further understanding of hardware and software tools (DPU, TensorFlow, Docker, OpenCV, etc.).
- **Group Documentation** – Utilizing specific documentation created by previous groups during development and testing, to assist with previous project understanding and provided code base.

4 Design

4.1 Design Context

4.1.1 Broader Context

In this section we explore the design of the project and its relationship to a broader context that ties the meaningful impact we are targeting. In the client's overall vision, this project is a multi-team, multi-year project which is being developed modularly.

Our team's role within this project is to design a system that incorporates multiple machine-learning models, focusing on tracking eye movement, into a single hardware device. One of the main impacts of our broader project would fall under a public health, safety, and welfare solution. Our client confirmed his goal is to create a healthcare device, a wheelchair system with eye detection. This device will benefit people with various disabilities by helping them with day-to-day life. However, this issue is more relevant to the end goal of the high-level project.

Our group's project is more focused on the development of the system. The impact of our group is more focused on the environmental aspect. The reason is that the current phase in the broader project is focused on developing models and hardware system optimization. Models, especially machine learning models, require a large amount of energy to perform operations. In our phase, we are responsible for minimizing the environmental impact by designing our models and hardware system around this fact.

Area	Description	Examples

Public health, safety, and welfare	With the implementation of the overall project, it will bring an accessibility solution to handicapped groups of people.	It will provide an alternative solution for helping individuals express and interact with the world through a device that will interpret eye gestures.
Environmental	The goal impact on the environment would be to diminish the consumption of energy by optimizing resources and the process the hardware goes through.	The optimization of the models and a well-planned schedule of the resources available allows for maximizing energy efficiency.
Global	Pupil-tracking technology has a wide global impact because of its usage across different applications and sectors worldwide.	The implementation can be used in healthcare for early diagnosis of neurological disorders. It can be used across different technology to assist people with mobility issues, assist with technology usage, and enhance interaction with virtual reality or other technology areas.
Cultural	Considering different cultural considerations involves recognizing the different ways people accept and interpret technology based on their norms and values. This will ultimately influence our design and implementation to accommodate many people.	Education is treated differently around the world with a wide range of curriculum and teaching standards. Some regions may embrace the ability to use pupil-tracking technology in education, while others will be skeptical of its usage in education.
Social	Consider how pupil-tracking technology will affect the ways in which people interact with each other and technology itself. It also covers the accessibility and ethical considerations involved in deploying technology.	Pupil-tracking technology can impact the way people with disabilities interact with technology or applications, offering independence through gaze-based communication tools. It also raises the concern for privacy and potential surveillance which creates an ethical debate.
Economic	The economic factors involve the cost of deployment, development, and the potential market demand which will influence future production costs and scale.	Developing an pupil-tracking system requires a significant investment in research and development. This can lead to high initial costs for development and initial deployment but can lead to economic benefits through niche markets and applications.

When it comes to the impact of global, cultural, and social factors, the impact is less relevant to our project, at least in the current phase. The goal of our project is more towards making a change in the healthcare and machine vision industry.

4.1.2 Prior Work/Solutions

Doing a market investigation on similar products regarding eye tracking, we found higher-end products and other types of uses for this technology.

- Apple: Apple Vision Pro headset
- Eyeware: Pupil-tracking software for gamers

- Tobii: Eye and head tracking hardware for gamers

Some of the issues with current pupil-tracking products are that they target a specific audience instead of showcasing their capabilities for other groups. Apple Vision Pro is a high-end product targeting professionals and encourages improved media devices. Eyeware developed software compatible with webcams to track eye movements for an optimized gaming experience. Tobii offers hardware with a similar goal of improving gamers' gaming experience through eye and head tracking.

Although these products provide an excellent example of pupil-tracking technology, they do not emphasize the possibilities this technology can work towards in health and wellness. Our project aims to showcase the potential pupil-tracking technology that can impact a specific audience while also keeping the open functionality of modularity regarding its implementation on a larger scale project.

4.1.3 Technical Complexity

Our project's technical complexity is evident through the need to integrate multiple components and subsystems, each with separate challenges.

FPGA Board (Xilinx Kria KV260):

- Engineering Principle: Utilizes field-programmable gate array (FPGA) technology, which allows hardware to be configured by the user post-manufacturing. This technology is crucial for high-speed, parallel processing capabilities and ideal for real-time image processing and machine-learning tasks.

Deep Learning Processing Unit (DPU):

- Scientific Principle: The DPU is specialized in deep learning inference, leveraging data reuse in neural network architectures to analyze visual data and interpret complex patterns such as eye movements and blinks.
- Engineering Principle: Optimizes computational efficiency and power consumption, crucial for deploying AI models in embedded systems.

Image Preprocessing, Blink Detection, and Eye Tracking Models:

- Mathematical Principles: These models utilize advanced image processing techniques, including geometric transformations, filtering, and edge detection, which are foundational in computer vision.
- Scientific Principles: Employ statistical methods and machine learning models that analyze temporal and spatial data to accurately predict eye positions and movements.

Memory Management Interface:

- Engineering Principle: Implements an efficient memory management system that ensures high data throughput and minimal latency, crucial for real-time processing applications. Techniques such as memory segmentation and caching are used to optimize memory access.

Parallel Processing and DPU Resource Sharing Mechanism:

- Engineering Principle: Ensures concurrency and synchronization across multiple processing units, a complex issue in computer architecture and operating systems.

4.2 Design Exploration

4.2.1 Design Decisions

In this section, we elaborate on the critical choices that shape our solution's framework and execution. These decisions, driven by our goal to efficiently utilize the Xilinx Kria KV260 board, revolve around optimizing DPU resource sharing, managing memory effectively, and implementing parallel processing to meet our throughput requirements. Each decision is instrumental in ensuring the success and efficiency of our project.

1. DPU Resource Sharing

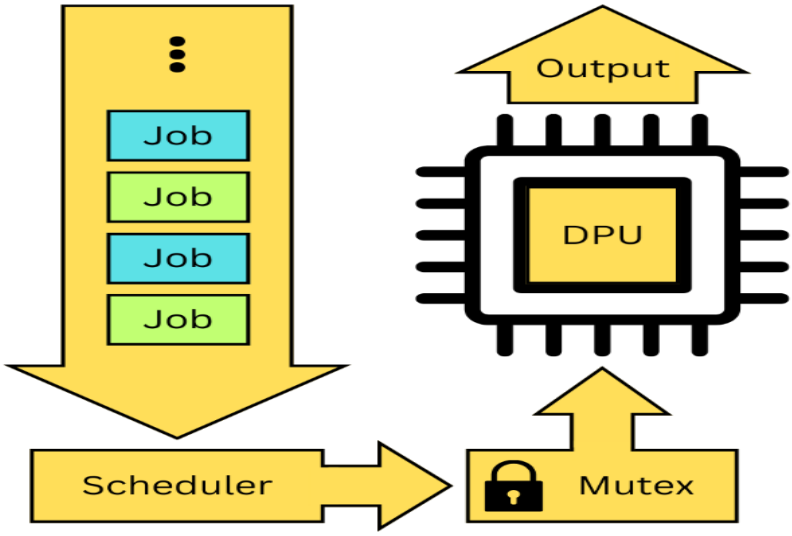


Figure 4 DPU Manager Flow

- Decision: To implement two methods for DPU resource sharing: a software-only approach and an approach using the Zynq UltraScale+'s onboard hardware mutex for enhanced control.
- Software Approach: We will design a function, use_DPU(), to manage DPU calls via execute_async(), incorporating a queuing system with priority scheduling. This ensures organized access to the DPU, prioritizing tasks based on their urgency and importance, which may vary as we analyze performance outcomes.
- Hardware Mutex Integration: In conjunction with our software strategy, leveraging the Zynq UltraScale+'s hardware mutex provides a robust mechanism for mutual exclusion at the hardware level, offering a failsafe against potential software bottlenecks.
- Importance: Efficient DPU resource sharing is crucial for maintaining high performance and responsiveness of our system, particularly when running multiple models that depend on this shared resource.

2. Memory Management

- Decision: Adopting a memory affinity approach to divide the available 5GB of DDR memory into distinct banks for dedicated process usage.
- Importance: This strategy ensures that each model can access its required memory resources without interference, reducing contention and latency. It's a pivotal decision for optimizing memory utilization and performance, given the physical memory constraints of our FPGA board.

3. Parallel Processing Implementation

- Decision: To run the blink detection, eye tracking, and semantic segmentation models on separate threads, with eye tracking further split across two threads to handle different frames simultaneously.

Importance: This approach is aimed at maximizing throughput and minimizing latency. By employing parallel processing, we can address the inherently slow nature of the eye tracking model and ensure that the system meets our throughput requirement of <5ms. Parallelism allows us to leverage the full computational potential of the FPGA board, ensuring that each component operates efficiently and contributes to the overall speed and reliability of the system.

4.2.2 Ideation

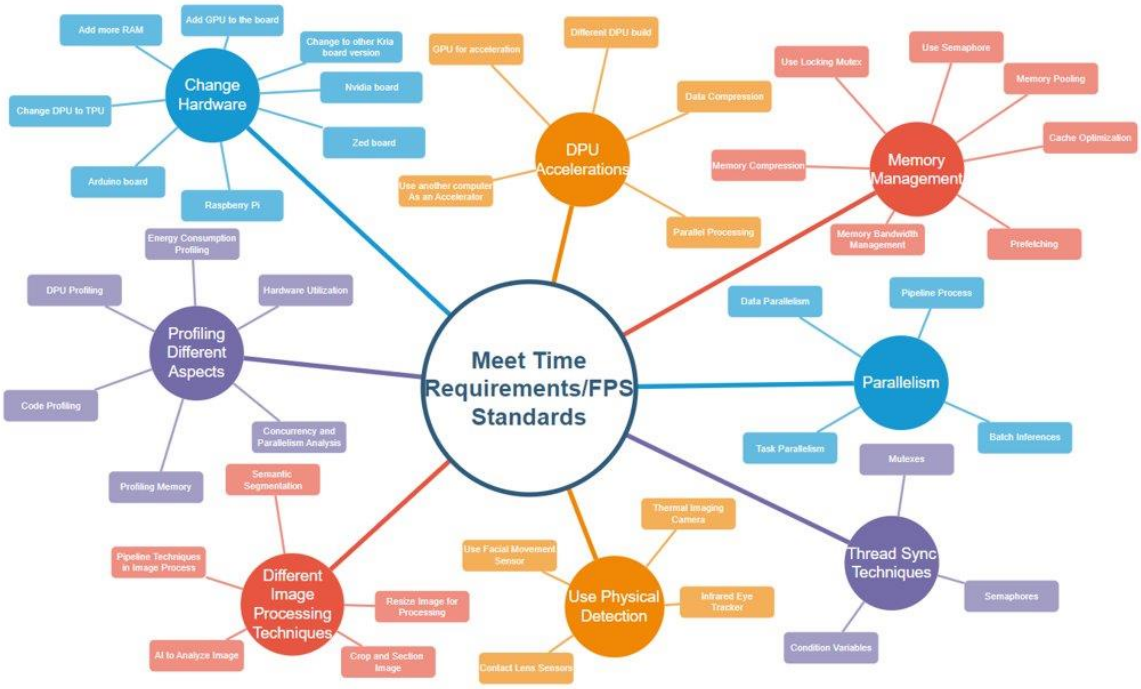


Figure 4 Lotus blossom diagram showing project components

1. Change Hardware

- One of the ideas for changing hardware could be increasing RAM on the board. Having more RAM allows each thread to have a larger memory allocation without contention for resources. This can prevent slowdowns caused by memory thrashing and improve overall system responsiveness.
- Instead of using an AMD Kria board, we could switch to NVIDIA Jetson AGX Xavier. It features an integrated NVIDIA Volta GPU with 512 CUDA cores, along with an eight-core ARMv8.2 CPU complex, making it well-suited for both compute-intensive and graphics-intensive tasks. NVIDIA GPUs are equipped with CUDA cores and Tensor Cores, which are highly parallelized processing units optimized for different types of workloads. Depending on your application's requirements, these specialized cores can accelerate computations and speed up processing tasks.

2. Parallelism

- Parallelism is one way for us to process multiple tasks together. By executing multiple tasks concurrently, parallelism allows for overlapping computation, communication, and I/O operations.

This can lead to improved overall throughput and reduced latency, as tasks can progress simultaneously rather than waiting for one another to complete.

- Due to the architecture of the Kria board of having four separated DDR4 memory, data can be processed independently across different subsets or partitions of the dataset. Parallelism enables data parallelism by distributing these data partitions across multiple processing units, allowing for simultaneous processing of different parts of the dataset and reducing overall processing time.

3. Memory Management

- In a multi-threaded environment, concurrent access to shared resources without proper synchronization can lead to data corruption or inconsistent states. By using locking mechanisms such as mutexes (mutual exclusion locks), semaphores, or read-write locks, developers can ensure that only one thread or process accesses a shared resource at a time, preventing data corruption and maintaining data consistency.
- Memory compression algorithms compress data stored in RAM, allowing more data to fit within the available memory capacity. This can reduce the frequency of paging (swapping memory contents between RAM and disk), which is a costly operation in terms of performance. By keeping more data in RAM and reducing the need for frequent disk accesses, memory compression can lead to faster overall performance.

4. DPU Acceleration

- DPUs are often used to accelerate machine learning inference tasks, such as image classification, object detection, and natural language processing. By leveraging dedicated hardware resources optimized for matrix operations and neural network inference, DPUs can significantly speed up the process of executing machine learning models, leading to faster inference times and improved overall performance.
- Instead of using DPU to infer machine learning models, ASICs could be used in this application. ASICs are custom-designed hardware accelerators optimized for specific tasks or models. ASICs can offer superior performance and energy efficiency compared to general-purpose CPUs or GPUs for specialized tasks like blink detection and eye tracking. Designing and manufacturing ASICs can be expensive and time-consuming, but they can provide significant performance benefits for specific workloads.

5. Different Image Processing Techniques

- Instead of processing the entire image, focus on extracting and processing only relevant features or regions of interest. Techniques such as region-based processing, saliency detection, and object detection can help narrow down the processing scope, leading to faster execution times.
- Down sampling involves reducing the resolution of an image, typically by averaging or subsampling pixels. This technique can significantly reduce the computational complexity of subsequent image-processing tasks while maintaining essential information.

4.2.3 Decision-Making and Trade-Off

DPU Resource Sharing Options

Options:

1. Software-only approach: Utilize a software-based queue and priority scheduling system.
2. Combined Software and Hardware Mutex approach: Integrate the software solution with the hardware Mutex provided by Zynq UltraScale+ for added efficiency and reliability.

Analysis:

1. The software-only approach offers simplicity and flexibility in implementation but might face scalability and efficiency issues under high demand.
2. The combined approach adds complexity but promises improved reliability and control, ensuring that DPU access conflicts are minimized.

Decision:

We opted for the combined approach, as the added control and reliability from the hardware mutex significantly outweigh the increased complexity. This choice was driven by our priority for system robustness and performance under varying loads.

Memory Management Options

Options:

1. Unified Memory Pool: A single, shared memory pool accessible by all processes.
2. Memory Affinity: Dividing the available memory into dedicated banks for specific processes.

Analysis:

1. A unified memory pool simplifies memory management but can lead to contention and inefficient use of resources.
2. Implementing memory affinity introduces complexity in management but ensures dedicated resources, reducing contention and potentially increasing performance.

Decision:

The memory affinity strategy was selected for its direct benefits in reducing latency and increasing the predictability of memory access times, which is crucial for meeting our throughput targets.

Parallel Processing Strategies

Options:

1. Sequential Processing: Running models one after another, focusing on simplicity.
2. Concurrent Execution with Limited Parallelism: Introducing basic parallelism while maintaining some sequential operations.
3. Full Parallel Processing: Maximizing parallel execution across all models and processes.

Analysis:

1. Sequential processing simplifies development but does not utilize the full capabilities of our hardware, leading to potential bottlenecks.
2. Limited parallelism offers a balance but may still underutilize available resources.
3. Full parallel processing maximizes hardware utilization but requires sophisticated control mechanisms to manage resource sharing effectively.

Decision:

We embraced full parallel processing, accepting the challenge of complexity for the sake of maximizing throughput and efficiency. This strategy aligns with our goal to leverage the FPGA board's capabilities fully, ensuring that each component contributes optimally to the system's overall performance.

Decision-Making Tools

To facilitate our decision-making, we employed a weighted decision matrix, assigning values to key criteria such as performance efficiency, complexity, scalability, and reliability. This quantitative analysis supported our qualitative assessments, guiding us toward choices that best align with our project goals and constraints.

4.3 PROPOSED DESIGN

4.3.1 Overview

Our project involves creating a system on the Kria KV260 board that can understand where someone is looking and infer their emotions from their eye movements. Imagine it as a smart camera that doesn't just see you but tries to understand how you're feeling by paying close attention to your eyes.

Key Components of Our Design

DPU: This part of the FPGA is like an accelerator for machine learning applications.

Memory Management: Because memory is limited, we've set up a system where the memory is divided into sections, each dedicated to a specific task. This way, everything runs smoothly without any hold-ups, ensuring that the system can keep up with real-time analysis without any lag.

Parallel Processing: To make the system faster, we can run multiple models at the same time, i.e., multitasking. It can run several tasks at the same time (like watching for blinks and tracking eye movements) without getting mixed up. This is key to making the system fast and responsive.

How It All Comes Together

All these parts work together like a well-coordinated orchestra. The DPU is the main processing unit, and the DPU scheduling unit ensures that all models share the resources as needed. The memory management unit also ensures that all model has sufficient memory to run. Finally, parallel processing allows the system to process multiple operations at the same time, achieving a higher throughput.

The result? A system that can quickly and accurately understand where you're looking and how you're feeling just by watching your eyes. This technology could have many uses, from helping doctors understand their patients better to making computers and gadgets more responsive to our needs and emotions.

4.3.2 Detailed Design and Visual(s)

High-Level Overview

Our design is centered around a highly efficient FPGA-based pupil-tracking system, specifically utilizing the Kria KV260 FPGA board. Below is a detailed overview of our system:

- FPGA Board (Xilinx Kria KV260): Serves as the core platform, integrating all components and models.
- Deep Learning Processing Unit (DPU): A specialized processing IP within the FPGA for executing deep learning models.
- Memory Banks: The FPGA's memory is strategically divided into separate banks, each allocated to different processes (e.g., image preprocessing, blink detection, eye tracking) to optimize memory usage and system performance.

- Image Preprocessing: This subsystem prepares raw eye images for analysis, improving the accuracy of subsequent detection, tracking, and semantic segmentation models.
- Blink Detection and Eye Tracking Models: Core models that analyze preprocessed images to detect blinks and track eye movements, respectively. These models are critical for determining the user's gaze direction and emotional state.
- Semantic Segmentation Model: An image processing technique that would remove glare or noise from the preprocessed images. This model is important for increasing the accuracy of blink detection and eye tracking models.
- DPU Resource Sharing Mechanism: Incorporates a software-based queue with priority scheduling and hardware mutex to manage access to the DPU among different models.
- Parallel Processing Threads: To enhance system throughput and responsiveness, blink detection and eye tracking models are executed on separate threads, with eye tracking further split to handle different frames simultaneously.

Sub-System and Component Descriptions

Image Preprocessing

- Operation: Enhances image quality for accurate analysis, involving noise reduction, contrast adjustment, and scaling. This step is crucial for the reliable performance of downstream models.
- Technical Requirement: Must process images within milliseconds to ensure real-time performance, integrating seamlessly with the eye tracking and blink detection models.

Blink Detection and Eye Tracking

- Operation: Utilize machine learning models to interpret preprocessed images, identifying blinks and tracking eye movements. Blink detection prioritizes rapid identification of eye closures, while eye tracking focuses on determining gaze direction.
- Integration: Both models request DPU access via the resource-sharing mechanism, with scheduling managed to balance urgency and computational load.

Semantic Segmentation

- Operation: Utilize machine learning model to interpret preprocessed images, processing the eye images to 4-classes segmented images.
- Integration: This model request DPU access via the resource-sharing mechanism, with scheduling managed to balance urgency and computational load.

DPU Resource Sharing Mechanism

- Software Queue with Priority Scheduling: Algorithms queue for DPU access, with priority given based on predefined criteria, ensuring critical tasks receive timely processing.
- Hardware Mutex: Enhances control over DPU access, preventing conflicts and ensuring smooth operation across concurrent tasks.

Memory Management

- Affinity Approach: Allocates separate memory banks for each major process, reducing contention and speeding up access times, which is critical for maintaining high throughput and system responsiveness.

Parallel Processing Threads

- Implementation: Designed to execute multiple models in parallel, significantly reducing processing time and increasing system throughput. Special consideration is given to the pupil-tracking model, which is split across threads to handle different frames, addressing its computationally intensive nature.

This technical description, accompanied by the detailed block diagram, provides a comprehensive understanding of our design. It outlines the critical components, their functions, and how they are integrated to create a high-performance pupil-tracking system. This information should enable another senior design team to grasp the intricacies of our solution and consider its implementation.

4.3.3 Functionality

In a Healthcare Setting

User Action: A patient sits in front of a diagnostic monitor equipped with our pupil-tracking system during a mental health assessment.

System Response: As the patient views various stimuli on the screen, the system analyzes their gaze and blinking patterns. It provides real-time feedback to healthcare professionals about the patient's emotional state and engagement, assisting in diagnosing or tailoring treatment plans.

In an Educational Environment

User Action: A student interacts with an educational software application on a tablet that incorporates our pupil-tracking technology.

System Response: The system monitors the student's eye movements to gauge where their attention is focused and how they react to different educational content. This information helps the software adapt in real-time, offering a personalized learning experience by emphasizing topics that captivate the student's interest or revisiting areas where their attention wanes.

In Personal Computing

User Action: A user navigates a website using a computer equipped with our pupil-tracking system.

System Response: The system detects the user's gaze direction, allowing for hands-free navigation based on where the user is looking. It could also adjust the content display based on the user's emotional response to different elements, enhancing the browsing experience.

4.3.4 Areas of Concern and Development

1. High Throughput and Low Latency: The system's capability to process data at a high rate, with a target throughput of <5ms, ensures real-time responsiveness. This is crucial for applications requiring immediate feedback based on eye movement analysis, such as adaptive learning software or patient emotional state monitoring in healthcare settings.

2. Efficient Resource Utilization: Through intelligent design choices like DPU resource sharing, memory affinity, and parallel processing, our system optimizes the limited resources of the FPGA board. This ensures that the device operates smoothly, even under the demand of running multiple complex models simultaneously.

3. Scalability and Flexibility: The modular approach to model implementation and resource management allows for flexibility in system deployment across various contexts, from healthcare to personal computing. This adaptability ensures that our design can evolve to meet emerging user needs and technological advancements.

Primary Concerns for Delivering a Product/System

1. Model Efficiency and Accuracy: The effectiveness of the pupil-tracking and blink models heavily depends on the accuracy and efficiency of the underlying software system. Ensuring these models can perform under real-world conditions without significant errors or delays is crucial.

2. Hardware Limitations: While we've designed our system to optimize resource utilization on the Kria KV260 board, physical constraints such as memory capacity and DPU availability remain a challenge.

4.4 Technology Considerations

We are using the AMD Kria KV260, a development platform for Kria KV260 SoMs, in our design. The Kria KV260 System-on-Module (SoM) by AMD offers a compact and integrated solution for embedded applications, providing a balance of performance, power efficiency, and versatility.

Strengths

- The KV260 is equipped with high-performance interfaces tailored for robotics and industrial applications. These interfaces likely include GPIO, UART, I2C, SPI, and CAN ports, facilitating seamless integration with various sensors, actuators, and control systems.

- The KV260's support for Kria KV260 SoMs offers versatility, allowing developers to choose the appropriate SoM configuration based on their application requirements. This scalability ensures that the platform can address a wide range of robotics and industrial use cases.

- Leveraging the capabilities of the Kria KV260 SoMs, the KV260 platform offers scalability in terms of processing power, memory, and connectivity options. This scalability enables developers to scale their robotic systems to meet evolving performance demands.

Weaknesses

- The advanced features and capabilities of the KV260 platform come with a higher upfront cost compared to simpler development platforms. This could be a limiting factor for developers with budget constraints or hobbyists exploring robotics projects.

- Developing applications for robotics and industrial applications can be complex, requiring expertise in hardware, software, and system integration. While the KV260 platform aims to simplify development with native ROS 2 support and high-performance interfaces, there may still be a learning curve for developers new to robotics.

Trade-offs

Performance vs. Power Consumption

The high-performance interfaces and processing capabilities of the KV260 platform may result in higher power consumption, especially in battery-powered robotics applications. Developers must balance performance requirements with power efficiency to ensure optimal system operation and longevity.

Versatility vs. Specialization

While the KV260 platform offers versatility in supporting various robotics and industrial applications, it may be optimized for specific use cases within these domains. Developers should assess whether the platform's features align with their project requirements or if a more specialized development platform would be more suitable.

Solutions

1. NVIDIA Jetson Series

- NVIDIA Jetson series of embedded platforms, such as Jetson Nano, Jetson Xavier NX, and Jetson AGX Xavier, offer high-performance GPU acceleration suitable for running multiple machine learning models concurrently.
- These platforms feature CUDA support, allowing developers to leverage GPU parallelism for accelerated inference tasks.
- The Jetson series also provides support for popular machine learning frameworks like TensorFlow, PyTorch, and ONNX, facilitating easy deployment of machine learning models.

2. Google Coral Dev Board

- The Google Coral Dev Board is another option for accelerating machine learning inference tasks at the edge.
- It features Google's Edge TPU (Tensor Processing Unit) for high-performance AI acceleration with low power consumption.
- The Coral Dev Board supports TensorFlow Lite and TensorFlow Lite Micro, making it suitable for running multiple machine learning models concurrently in resource-constrained environments.

Design Alternatives

1. Distributed Computing Architecture

- Instead of relying on a single powerful device, distribute the computational load across multiple edge devices interconnected in a network. Each edge device can be responsible for processing a subset of the input data or running specific machine-learning models. Use communication protocols such as MQTT or gRPC for inter-device communication and coordination. This approach allows for scalability, fault tolerance, and efficient utilization of resources across the network.

2. Custom ASIC-based Solution

- Design custom Application-Specific Integrated Circuits (ASICs) tailored to the specific requirements of your machine-learning models. Develop dedicated hardware accelerators optimized for parallel execution of inference tasks. Leverage the high performance and power efficiency of ASICs to achieve the desired throughput and latency targets. This approach requires significant upfront investment in ASIC design and fabrication but offers the potential for unparalleled performance and energy efficiency.

4.5 Design Analysis

Our team has deployed three machine learning models, such as pupil-tracking, blink detection, and semantic segmentation models on the KV260 FPGA board. Our team developed a multi-threaded application to achieve parallelism to ensure our inputs could be processed with a throughput of 200 FPS.

Here's a breakdown of the situation and potential plans for future design and implementation work:

1. Validation of Target Hardware: The first step would be to validate the design on the Xilinx Kria KV260 evaluation board. This involves deploying the multithreaded application that can run the various models in parallel on the board and testing their performance in a real-world environment.

2. Identifying Challenges: Once the design is tested on the board, it's crucial to identify any challenges or discrepancies between the expected and observed performance. This could include issues related to resource constraints, hardware limitations, or unexpected behavior.

3. Iterative Optimization: Following validation and identifying challenges, an iterative optimization process would be necessary to address any issues and fine-tune the implementation. This may involve optimizing models for hardware acceleration, adjusting resource allocation strategies, or refining parallel processing techniques.

4. Testing and Validation: Rigorous testing and validation on the target hardware platform are essential to ensure that the system meets performance requirements and functions reliably under various conditions.

5. Documentation and Reporting: Comprehensive documentation of the implementation process, including challenges faced and solutions developed, should be maintained. This documentation will serve as a valuable resource for future iterations of the project and knowledge transfer.

6. Feasibility Assessment: Throughout the implementation and optimization process, ongoing feasibility assessments should be conducted to evaluate whether the design goals are achievable within the constraints of the hardware platform.

7. Addressing Build Issues: If any build issues are encountered during the implementation process, they should be addressed promptly through troubleshooting, debugging, and potentially revising the design or implementation approach.

5 Testing

Testing Strategy Overview

Testing is crucial in our project, directly impacting the effectiveness and reliability of our system. Our testing strategy is intricately designed to validate each component and its integration, ensuring that our system not only meets the design specifications but also adheres to the requirements for accuracy and performance.

- Testing Philosophy: Our team adopts the idea of "test early, test often," which involves integrating testing throughout the development cycle rather than at the end. This approach helps us identify issues early, allowing for timely corrections that are crucial given the complexity of our system.
- Unique Testing Challenges: One of the unique challenges in our project is the need to make sure that the accuracy and throughput of the pupil-tracking and blink-detection models on the FPGA hardware is inherently different from traditional software environments. The limited resources on the FPGA and the need for real-time processing add complexity to testing these models effectively.

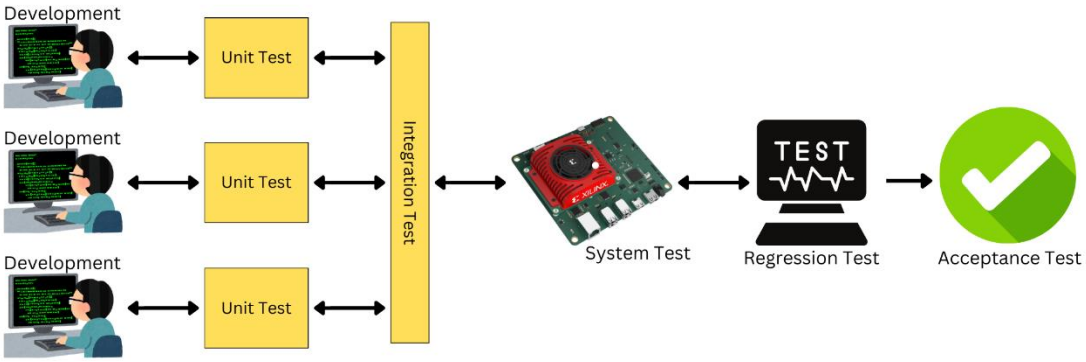


Figure 5 Project testing flow

Testing Schedule

- Continuous Integration Testing: As components of the system are developed and integrated, they will undergo continuous testing to ensure compatibility and performance standards are met throughout the development process.
- Pre-Deployment Testing: Before the system is fully deployed, it will undergo comprehensive testing to ensure all components function seamlessly together and meet the project requirements.
- Post-Deployment Testing: After deployment, the system will be monitored and tested under real-world conditions to ensure it continues to perform as expected and adjusts to practical challenges.

5.1 Unit Testing

Accuracy Testing of Models

- Blink Detection Accuracy: We will test the blink detection model to ensure it accurately identifies actual blinks. This involves comparing the model's output against a set of ground truth data where images had been manually tagged. The comparison results are then cataloged in a Confusion Matrix to establish an accuracy metric on True positives/negatives and False positives/negatives.
- Pupil Tracking Accuracy: The pupil-tracking model will be tested for its ability to accurately track the pupil coordinates against known positions. The model output set is analyzed against a set of ground truth data, performing Root Mean Squared Error (RMSE) equation to get the standard deviation of the results. We take into consideration the range of positions to come up with a Relative RMSE, which is used to represent an accuracy metric.

- Semantic Segmentation Accuracy: The semantic segmentation model's performance will be evaluated on its ability to accurately segment the validation dataset. The model's predictions will be compared against a manually labeled ground truth dataset, represented as NumPy arrays. These ground truth labels serve as a benchmark for validating the segmentation accuracy. Key performance metric such as mean Intersection over Union (mIoU) will be calculated to quantify the alignment between the model's output and the ground truth annotations.
- Benchmarking on Laptop: Prior to deployment on the FPGA, both models will be benchmarked on a laptop using the h5/pb format. This step ensures that the models behave as expected in a controlled environment before testing them using the FPGA-specific xmodels format.

5.2 Interface Testing

- Hardware-Software Interface: This interface ensures that the software algorithms (image preprocessing, blink detection, and pupil tracking models) correctly interact with the FPGA hardware, particularly the DPU. This test will be achieved using an on-board GDB debugger.
- Inter-Model Interface: This involves the data exchange between the image preprocessing, blink detection, and eye tracking models. This test will be done using an on-board GDB debugger.
- Memory Management Interface: The interface between the memory banks and each processing model ensures that data is appropriately allocated and accessed without conflicts. We can probe memory components (DDR, BRAM) using Vitis' memory monitoring tool and Vivado Hardware Manager.

5.3 Integration Testing

Benchmarking of Integrated System:

- Use of Timing Analysis: We will use C RAII profiling library to analyze the integrated system's performance. These tools provide detailed timing metrics of individual function's execution time, which are critical for assessing the system's efficiency and pinpointing bottlenecks.
- Real-Time Performance Testing: Since the system is designed for real-time applications, testing will also focus on ensuring that it meets the required throughput of 200 FPS. This involves repeated trials under varying conditions to verify consistent performance.

5.4 System Testing

Continuous Input of Labeled Frames:

- Description: To test accuracy and performance, we continuously feed the system labeled frames that represent various pupil-tracking scenarios. This extensive input helps validate the system's ability to accurately interpret eye positions and movements against known benchmarks.
- Tools: We employ logging tools within our testing framework to record the system's outputs and compare them with expected results. These tools help monitor the accuracy and responsiveness of the system, providing real-time feedback on performance metrics.

Performance and Throughput Testing:

- Description: The system must adhere to performance requirements, specifically maintaining a throughput of <5ms as specified. This testing ensures that the system can handle the required frame rate without delay, maintaining high accuracy under operational load.
- Tools: We utilize the C RAII profiling library to measure and analyze individual function's execution time. This tool provides detailed insights that help identify performance bottlenecks and areas for optimization.

5.5 Regression Testing

Performance and Memory Monitoring

- Analysis with C RAII profiling library: We leverage C RAII profiling library to analyze individual function's execution time. This tool is instrumental in identifying any performance regressions introduced by recent changes.
- Memory Logs: We will pay attention to memory logs, which track memory allocation and usage over time. This helps ensure that memory management remains efficient and that updates do not introduce memory leaks or excessive consumption.

Automated Regression Tests

We will implement an automated regression test that run after each significant update or periodically as needed. These tests include a suite of predefined tasks that the system must perform, comparing the outputs with historical data to ensure consistency.

5.6 Acceptance Testing

Functional Requirements

- Accuracy Testing: Conduct tests to verify that the pupil-tracking and blink-detection models perform with high accuracy under various conditions.
- Throughput Testing: Evaluate the processing speed to ensure the system meets <5ms requirement for image processing and inferencing.

Non-functional Requirements

- Reliability Testing: Run the system for extended periods to check for any signs of failure or for degraded performance over time.
- Stress Testing: Simulate different load conditions to ensure the system can handle different levels of traffic and inputs without performance loss.
- Resource Management Testing: Monitor the usage of FPGA resources and verify the system utilizes the available memory and processor cores optimally.

Client Involvement

- Client Review Sessions: Perform regular meetings with clients to review testing progress, results, and adjustments in testing strategy for future tests based on results and client feedback.
- Demo Days: Provide "demo days" to the client where the team will show the system in action. Allowing the client to provide immediate feedback on the system's current implementation and abilities.

5.7 Results

As we progress toward the testing phase of our system, we anticipate obtaining a series of detailed results that will demonstrate the system's compliance with both functional and non-functional requirements. Below, we outline the expected types of data and results from our comprehensive testing strategy.

1. Labeled Frame Analysis Results

- Expected Data: A list of labeled frames, where each frame includes both ground truth data (annotated eye positions and blink status) and predictions made by the system (annotated). These frames are expected to be in PNG format, allowing for easy visual comparison.
- Analysis Process: The labeled frames will be visually inspected to assess the accuracy of the pupil-tracking and blink-detection models. Discrepancies between ground truth and predictions will be quantitatively analyzed to calculate the accuracy percentage of the system.
- Expected Results: We aim to achieve an accuracy rate that meets or exceeds the project's specifications. Specifically, the blink detection should correctly identify blinks with high precision, and the pupil-tracking model should accurately track the gaze within a minimal margin of error.

2. Performance Metrics from Timing Analysis

- Expected Data:
    - Throughput: A critical metric indicating how many frames per second the system can process. This number should meet the requirement of <5ms per frame to ensure real-time performance.
- Analysis Process: Throughput times and other performance metrics will be analyzed using the Timing Analysis. The data will help identify any performance bottlenecks and verify that the system utilizes hardware resources efficiently without exceeding capacity.
- Expected Results: The throughput should consistently fall below the 200FPS threshold under normal operating conditions, confirming the system's capability to perform real-time processing.

6 Implementation

1. Implementing Parallelism:
    a. Research and explore techniques for implementing parallelism in the program to achieve the target throughput of <5ms (200 fps). This involves studying various parallel computing paradigms, such as multithreading, multiprocessing, and GPU acceleration, to identify the most suitable approach for the application.
    b. Experiment with multithreading or multiprocessing approaches to distribute computational tasks and optimize resource utilization. Investigate how these parallelization techniques can be integrated into the existing codebase to leverage the available hardware resources efficiently.
    c. Explore task-based parallelism frameworks such as OpenMP or CUDA to parallelize specific computational tasks within the program. Evaluate the feasibility and performance impact of using these frameworks in conjunction with the existing models and hardware architecture.
    d. Investigate data parallelism techniques such as SIMD (Single Instruction, Multiple Data) or vectorization to exploit parallelism at the instruction level and accelerate the processing of large datasets. Explore how these techniques can be applied to optimize performance while maintaining model accuracy.
2. Enhancing Image Pre-processing Speed by Altering Semantic Segmentation Model:
    a. Refine the training sequence by eliminating redundant steps to reduce the model's parameter count and computational complexity. This process involves analyzing the training pipeline to identify and remove unnecessary operations or layers that do not contribute significantly to the model's performance.
    b. Conduct iterative training and retraining of the segmentation model to achieve a top-5 accuracy of 95%. This involves fine-tuning the model architecture, hyperparameters, and training data to optimize performance while maintaining the desired level of accuracy.

    c. Continuously evaluate the model's performance through rigorous testing and validation procedures. This includes assessing both quantitative metrics, such as accuracy, and qualitative aspects, such as visual inspection of segmentation results to ensure consistency and reliability.
    d. Implement optimization techniques such as model pruning, quantization, and architecture adjustments to further enhance inference speed without compromising accuracy. Experiment with different optimization strategies to find the most effective combination for achieving the desired throughput while meeting accuracy requirements.
3. Improving Eye Tracking and Blink Detection Accuracy:
    a. Refine the pupil-tracking and blink-detection models to enhance accuracy and reliability. This involves revisiting the existing models to identify areas for improvement, such as feature selection, model architecture, or parameter tuning, to achieve better performance.
    b. Conduct extensive testing and validation to achieve the desired accuracy levels of 86% to 92% for eye tracking and 99% for blink detection. Develop comprehensive testing protocols and datasets to evaluate the performance of the models under various conditions, including different lighting conditions, head movements, and user demographics.
    c. Explore advanced machine learning techniques such as deep learning or ensemble methods to improve the robustness and generalization capabilities of the pupil-tracking and blink detection models. Investigate state-of-the-art algorithms and architectures in the literature and adapt them to the specific requirements of the application.
    d. Implement real-time feedback mechanisms or adaptive learning models to dynamically adjust the pupil-tracking and blink detection models based on user feedback or changing environmental conditions. This adaptive approach can help improve accuracy and adaptability in real-world scenarios.
4. Implementing Multithreading for Concurrent Execution:
    a. Develop and implement multithreading capabilities to enable concurrent execution of blink detection and pupil-tracking models. This involves designing a multithreaded architecture that can effectively distribute computational tasks across multiple threads, allowing different components of the system to run simultaneously.
    b. Ensure proper initialization and management of memory resources to prevent contention and optimize performance. Implement memory isolation techniques such as memory partitioning or memory pools to allocate memory resources dynamically and efficiently among different threads. This helps prevent memory conflicts and ensures that each thread has access to the necessary resources without contention.
    c. Implement synchronization mechanisms such as locks, semaphores, or barriers to coordinate access to shared resources and ensure thread safety. Proper synchronization is essential to prevent race conditions and ensure that data integrity is maintained during concurrent execution.
    d. Explore thread scheduling strategies to optimize resource utilization and minimize thread contention. Techniques such as task prioritization, load balancing, or thread affinity can help maximize throughput and minimize latency by efficiently scheduling threads based on system resources and workload characteristics.
5. Profiling and Optimization:
    a. Utilize the PetaLinux kernel for profiling, leveraging tools such as Xilinx Vitis AI to gather comprehensive performance data. This involves instrumenting the system to collect metrics related to CPU usage, memory utilization, I/O operations, and other relevant performance indicators.

b. Analyze profiling data to identify performance bottlenecks and areas for optimization. Use visualization tools and performance analysis techniques to identify hotspots in the code, such as functions or code segments with high execution times or resource utilization.
c. Prioritize optimization efforts based on the profiling results, focusing on critical sections of the code that contribute most significantly to overall system performance degradation. This may involve modelic optimizations, code refactoring, or architectural improvements to eliminate inefficiencies and reduce computational overhead.
d. Implement optimizations to enhance overall system performance, addressing identified bottlenecks and inefficiencies. This may include model optimizations, data structure improvements, or parallelization techniques to exploit multi-core architectures and accelerate computation.
e. Conduct iterative testing and validation to assess the impact of optimizations on system performance. Develop benchmarks and performance metrics to quantify the effectiveness of optimization efforts and ensure that performance targets are met or exceeded.
f. Continuously monitor system performance and revisit optimization strategies as needed to maintain optimal performance levels. Regularly profile the system to identify new bottlenecks or performance issues that may arise due to changes in workload, data characteristics, or system configuration.

6. Documentation and Reporting:
    a. Document the implementation process thoroughly, detailing each step taken, methodologies used, and any modifications made to the original design plan. Include descriptions of challenges encountered during implementation and the corresponding solutions devised to overcome them.
    b. Maintain detailed records of experimental results, performance metrics, and validation tests conducted throughout the implementation process. Document any anomalies or unexpected behavior observed during testing and the troubleshooting steps taken to resolve them.
    c. Prepare comprehensive reports and presentations summarizing the progress and outcomes of the implementation plan. These reports should provide a comprehensive overview of the project, including objectives, methodologies, results, and conclusions.
    d. Include visual aids such as charts, graphs, and diagrams to illustrate key findings and trends observed during testing and validation. Use clear and concise language to communicate complex technical concepts effectively to both technical and non-technical stakeholders.
    e. Incorporate feedback from project stakeholders and team members into the documentation and reporting process. Solicit input and suggestions for improvement to ensure that the final reports accurately reflect the project's progress and outcomes.
    f. Ensure that all documentation and reports are organized, well-structured, and easily accessible to project stakeholders, team members, and other interested parties. Provide appropriate version control and access permissions to maintain the integrity and security of project documentation.

7 Professional Responsibility

7.1 Areas of Responsibility

Area of Responsibility	Definition	NSPE Canon Code of Ethics	IEEE Code of Ethics
Work Competence	Perform work of high quality, integrity,	Perform services only in areas of their competence;	Requires maintaining and updating technical

	timeliness, and professional competence.	Avoid deceptive acts.	competence; this complements NSPE's focus on practicing within one's area of competencies.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	Honesty in data-based claims can be related to NSPE's commitment to providing faithful results to entities.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	Honesty and realistic representations in all technical information, copying NSPE's area for objective and truthful reporting.
Health, Safety, Well-Being	Minimize risks to the safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	Mandates prioritizing safety, health, and welfare of the public, similar to NSPE, but adds a clear imperative to protect the environment.
Property Ownership	Respect the property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	Emphasizes crediting others' work such as their intellectual property, as NSPE's outlines acting as faithful agents regarding clients' ideas and information.
Sustainability	Protect the environment and natural resources locally and globally.	N/A	Integrates environmental protection with public safety and welfare, as NSPE focuses on local and global resource protection.

Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	Encourage contributions to society through technology understanding and professional development, expanding on NSPE's more general benefit through ethical practices.

7.2 Project Specific Professional Responsibility Areas

Area of Responsibility	Rating (High/Medium/Low)	Relevance/Reasoning
Work Competence	Medium	Our team possesses most of the required technical skills, but many technical aspects are new to every member. We improve understanding through research, application, and discussion within the team by sharing resources.
Financial Responsibility	High	A private client funds the project with no current profit-making objective. The team respects the client's resources and aims to hit client standards.
Communication Honesty	Medium	We strive for transparency with our clients, but current challenges have held the team back from meeting goals. Current communication creates the challenge of being able to easily present results.
Health, Safety, Well-Being	Low	While we recognize the importance of health, safety, and well-being, it has not been a primary focus during this development stage of the project. Is expected to increase with project development.
Property Ownership	High	We respect our client's intellectual property from previous teams and check proper permissions for all third-party resources used.
Sustainability	Medium	We recognize the resource consumption our project can create and aim to develop our implementation to limit

		resource usage. This will grow when as the project develops and profiling of system resource usage.
Social Responsibility	Medium	Our project's current direct engagement with communities is limited, as we focus more on laying the groundwork for future applications rather than immediate social outcomes.

7.3 Most Applicable Professional Responsibility Area

In our project, Property Ownership emerges as the most applicable area of professional responsibility. This is shown in our commitment to respect the intellectual property of previous teams that developed our existing code base and other features used within our project provided by our client. While also making sure that our third-party resources are acquired properly or potentially offered as open source. Our current work to uphold this area is shown by our checks to prevent infringement and by utilizing a thorough documentation process, which maintains transparency and accountability in the use of external contributions on our project. The teams adhering to Property Ownership is important in the current innovation of our project that involves the inclusion of different intellectual assets. This adherence wont only last for one phase of the project but also sets a precedent for future development of our project.

8 Closing Material

8.1 Discussion

As we wrap up this project, reflecting on potential enhancements and alternative strategies sparks interesting possibilities for future development. While we have achieved significant milestones, exploring new methods could lead to further performance improvements. Here are some avenues that we or future teams might consider:

Expanding DPU Capabilities:

- Parallel Model Inference: Currently, our system operates with a single-core DPU, which naturally limits the parallel processing capabilities of our pupil-tracking system. One potential upgrade could involve expanding the hardware to include multiple DPUs. This change would allow simultaneous inference across several models, potentially doubling or even tripling the throughput capabilities. Such an enhancement would be particularly beneficial in scenarios requiring extensive real-time processing, such as in environments with multiple simultaneous users.
- DPU Optimization: Beyond adding more DPUs, optimizing the existing DPU configuration for better utilization could also yield significant gains. Techniques such as model pruning, quantization, or employing more efficient machine learning models could reduce the computational load on the DPU, enhancing the speed without necessitating hardware expansion.

Enhancing CPU and Memory Utilization:

- Complete CPU and Memory Affinity: Implementing full CPU and memory affinity could be a game-changer. By dedicating specific cores and memory banks to particular tasks or models, we can minimize context switching and memory swapping overheads. This approach ensures that each

component operates within its optimal resource environment, potentially smoothing out any bottlenecks in data processing and speed.

- Advanced Scheduling Algorithms: Developing more sophisticated scheduling algorithms could improve the efficiency of resource allocation. For example, employing dynamic scheduling based on real-time system load or predictive modeling based on usage patterns could ensure that resources are allocated not just efficiently but also proactively.

Incorporating any of these suggestions would not only push the boundaries of what our pupil-tracking system can achieve but also provide valuable learning opportunities for future teams. Each of these enhancements comes with its own set of challenges and requirements, making them worthy candidates for continued research and development.

8.2 Conclusion

Our group has successfully developed a sophisticated pupil-tracking system that runs pupil-tracking, blink detection, and semantic segmentation models in parallel on the Xilinx Kria KV260. By setting and achieving multiple milestones, we have not only met but exceeded the client's requirements, notably achieving a system throughput of less than 5ms, which marks a significant improvement in processing speed.

Throughout this project, each team member played a crucial role by taking on specific tasks that collectively created a cohesive and functional system. These roles ranged from model development, system integration, testing, and optimization, to documentation and setup of development environments. This structured division of labor ensured that all aspects of the project were meticulously managed and executed, leading to a successful outcome that aligned perfectly with the client's expectations.

In addition to the development of the main program, our group has established a robust foundation for future development teams. We have implemented several key infrastructural elements to support ongoing and future projects, including:

- Extensive Documentation: Comprehensive guides and detailed documentation have been prepared, covering everything from system architecture and code documentation to user manuals. This documentation ensures that future teams can easily understand and build upon the work already done without the need for extensive reverse engineering.
- Portable Development Environment: We have set up a Docker-based environment, which encapsulates all necessary development tools and libraries in a container. This approach not only simplifies the setup process for new developers but also ensures consistency across various development setups, reducing the "it works on my machine" syndrome.
- Version Control System: A GitHub repository has been meticulously maintained throughout the project, providing version control and source code management. This repository serves as a central hub for all project-related code, allowing for collaborative development and historical tracking of changes, which is essential for troubleshooting and understanding the evolution of the project codebase.

The infrastructure and resources established by our team are designed to empower future teams, enabling them to innovate and extend the system with new features or improvements efficiently and effectively. With these tools in place, the project is well-positioned to evolve in line with technological advancements and changing user needs.

Overall, the successful completion of this project represents a significant milestone in the application of embedded AI for real-time image processing and analysis. Our team is proud to deliver a system that not

only meets the technical specifications and performance requirements set forth by the client but also provides a scalable and maintainable platform for future innovation and development.

8.3 References

[1] "AMD Technical Information Portal," *docs.amd.com*. https://docs.amd.com/r/en-US/ds986-kv260-starter-kit (accessed Apr. 17, 2024).

[2] Chaudhary, A. K., Kothari, R., Acharya, M., Dangi, S., Nair, N., Bailey, R., Kanan, C., Diaz, G., & Pelz, J. B. (2019). RITnet: Real-time semantic segmentation of the eye for gaze tracking. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. https://doi.org/10.1109/iccvw.2019.00568

[3] IEEE Standard for Robustness Testing and Evaluation of Artificial Intelligence (AI)-based Image Recognition Service, IEEE 3129-2023, 2023

[4] IEEE Standard for Performance and Safety Evaluation of Artificial Intelligence Based Medical Devices: Terminology, IEEE 2802-2022, 2022

[5] IEEE Standard for Data Privacy Process, IEEE 7002-2022, 2022

[6] IEEE Standard for Requirements of Privacy-Preserving Computation Integrated Platforms, IEEE 3156-2023, 2023

[7] IEEE Recommended Practice for Secure Multi-Party Computation, IEEE 2842-2021, 2021

[8] IEEE Standard for Secure Computing Based on Trusted Execution Environment, IEEE 2952-2023, 2023

[9] IEEE Standard for Learning Technology - Learning Technology Systems Architecture (LTSA), IEEE 1484.1-2003, 2003

[10] "Jetson modules, support, ecosystem, and lineup," NVIDIA Developer, https://developer.nvidia.com/embedded/jetson-modules (accessed Apr. 16, 2024).

[11] "Dev Board," Coral, https://coral.ai/products/dev-board/ (accessed Apr. 16, 2024).

[12] Patel, N. (2024, January 30). Apple Vision Pro Review: Magic, until it's not. The Verge. https://www.theverge.com/24054862/apple-vision-pro-review-vr-ar-headset-features-price

[13] Beam Eye Tracker Reviews. Steambase. (n.d.). https://steambase.io/games/beam-eye-tracker/reviews (accessed Apr. 16, 2024).

[14] Tobii Gaming. "Tobii Eye Tracker 5 | Next Generation of Head and Eye Tracking." Tobii Gaming, 1 Apr. 2024, gaming.tobii.com/product/eye-tracker-5 (accessed Apr. 16, 2024).

8.4 Appendices

AMD Kria KV260 Vision AI Starter Kit: https://docs.amd.com/r/en-US/ds986-kv260-starter-kit

9 Team

9.1 Team Members

- Jonathan Tan – Senior Computer Engineering
- Josh Czarniak – Senior Computer Engineering
- Justin Wenzel – Senior Computer Engineering
- Kai Heng Gan – Senior Cyber Security Engineering
- Santiago Campoverde – Senior Software Engineering

9.2 Required Skill Sets for Your Project

Technical Skills:

- Machine Learning and Computer Vision
- Embedded System and Hardware Optimization
- Software Programs
    - C/C++
    - Python
- Data Management
- System Profiling and Benchmarking

Additional Skills:

- Project Management
- Communication
- Testing and Quality Assurance
- Documentation

9.3 Skill Sets Covered by the Team

Technical Skills:

- Machine Learning and Computer Vision : All Members
- Embedded System and Hardware Optimization : Jonathan, Justin, Josh
- Software Programs : All Members
    - C/C++
    - Python
- Data Management : Jonathan, Kai
- System Profiling and Benchmarking : Santiago

Additional Skills:

- Project Management : Santiago
- Communication : All Members
- Testing and Quality Assurance : Jonathan, Santiago
- Documentation : All Members

9.4 Project Management Style Adopted by the Team

We use Agile as our project management style throughout our development process. Agile has provided us with flexibility and adaptability. Our implementation does not have only one approach. Agile provides us the ability to check many different solutions, allowing us to reevaluate our implementations and design decisions. The group sets up individual milestones and reevaluates our implementations after each milestone is reached to help outline the next milestone we will achieve.

9.5 Initial Project Management Roles

- Jonathan Tan – DPU Management, Kria Board Manager
- Josh Czarniak – Pupil Center Location Model
- Justin Wenzel – Blink/No-Blink Model
- Kai Heng Gan – Semantic Segmentation Algorithm
- Santiago Campoverde – Data Profile/Model Analytics

9.6 Team Contract

Members

1. Justin Wenzel
2. Jonathan Tan
3. Josh Czarniak
4. Kai Heng Gan
5. Santiago Campoverde

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

   Regular Team Meetings (w/o client and advisor) : Thursday 2:10p-3p @ In-person

   Client Meetings : Wednesday 7p-8p @ Telegram Call

   Advisor Meetings : Thursday (Bi-weekly) 1p-2p @ In-person

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
   Teams chat : Microsoft Teams chat group
   Team updates : Microsoft Teams team
   Client chat : Telegram group
   Source control : GitHub

3. Decision-making policy (e.g., consensus, majority vote):
   Majority vote

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):
   Meeting minutes saved in Microsoft Team Files.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
    a. Members are expected to attend all meetings on time unless due to unforeseen circumstances that were previously communicated.
    b. Procedure for Absences: In the case of an unforeseen circumstance, the team member(s) involved should inform the team one day in advance, if possible, in the team's Microsoft Teams.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
    a. Members are expected to complete assigned tasks in a timely manner; timelines might be modified accordingly by vote or approval of advisor and/or client.
    b. Accountability Measures: Steps to be taken if a member consistently fails to meet deadlines or complete assignments.
        i. Initial Warning: If a team member fails to meet a deadline or complete an assignment, they will first receive a private, verbal warning from the team leader.
        ii. Written Warning: Continued issues will result in a written warning, outlining the specific concerns and expectations for improvement. This warning will be documented and shared with the team advisor.
        iii. Meeting with Advisor: If the problem persists, a meeting will be arranged with the team member, the team leader, and the advisor to discuss the issues and potential solutions.
        iv. Reassignment of Tasks: As a last resort, if there is no improvement, the member's tasks may be reassigned to other team members, and the situation will be reported to the course instructor or program coordinator for further action.
        v. Regular Performance Reviews: To prevent issues, regular performance reviews will be conducted to provide feedback and identify any potential problems early on.
    c. Flexibility in Roles:
        i. Open Communication: Team members are encouraged to openly communicate if they feel overwhelmed or unable to complete their assigned tasks due to unforeseen circumstances.
        ii. Task Redistribution: In such cases, the team will collectively discuss and redistribute tasks to ensure that the project remains on track without overburdening any single member.
        iii. Temporary Assistance: Other team members may temporarily assist the overwhelmed member until they are able to resume their normal workload.
        iv. Role Adjustment: If necessary, roles within the team can be reevaluated and adjusted to better align with each member's strengths and current capabilities.
        v. Regular Role Review: The team will regularly review the distribution of roles and responsibilities to ensure that it remains effective and equitable for all members.

3. Expected level of communication with other team members:
    a. Members are expected to check Microsoft Teams team daily.
    b. Any team related questions/confusions should be addressed to the Microsoft Teams team "General" channel.
    c. Members are expected to help each other to ensure everyone is on the same page.

4. Expected level of commitment to team decisions and tasks:
    a. Members are expected to participate in team discussions and voting.
    b. Members are expected to complete weekly tasks on time and document them into weekly status reports.
    c. Members are free to speak up their mind during team meetings.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):
    a. **Team Organization Leader**: Santiago Campoverde
    b. **Team/Advisor Meeting Leader**: On a rotating basis picked using pickerwheel.com but shall not be the same as client meeting leader.
    c. **Kria Board Manager, Petalinux/Profiling:** Jonathan Tan
    d. **DPU Management and Pupil Tracking Location Model:** Josh Czarniak
    e. **Multithreaded Application:** Justin Wenzel
    f. **Semantic Segmentation Model:** Kai Heng Gan
    g. **Data Profile/Model Analytics:** Santiago Campoverde

    Role description:
    a. **Team organization leader**: Responsible for managing timelines.
    b. **Team/Advisor meeting leader**: Responsible for leading team/advisor meeting discussions.
    c. **DPU Management, and Kria Board Manager:** Manages hardware resources and interaction with hardware for team members and their models.
    d. **Pupil Tracking Location Model:** Implements and manages pupil tracking machine learning model and its execution on the Kria board.
    e. Blink/No-Blink Model: Implements and manages blink/no-blink machine learning model and its execution on the Kria board.
    f. **Semantic Segmentation Model:** Implements and manages semantic segmentation machine learning model and its execution on the Kria board.
    g. **Data Profile/Model Analytics:** Profiles the program execution including the different models/model's performance and resource usage across the Kria board.

2. Strategies for supporting and guiding the work of all team members:
    a. Members are expected to help other team members' needs, including communication, technical, and other challenges.
    b. Conduct regular one-on-one check-ins to discuss progress, challenges, and any support needed.

3. Strategies for recognizing the contributions of all team members:
    a. **Transparent Progress Tracking**: We shall use a shared platform or tool to track and display the progress of projects. This ensures transparency in who is contributing what, allowing for a clear understanding of each member's involvement.
    b. **Regular Feedback Mechanisms**: Our team will establish a structured feedback system where team members can give and receive constructive feedback. These discussions shall be conducted weekly during team meetings, which helps in recognizing areas where members excel and where they need support.

Collaboration and Inclusion

1. Skills, expertise, and unique perspectives each team member brings to the team.
    - Justin Wenzel
        - Top 3 languages/skills/etc.:
            - C
            - Hardware Design
            - Debugging
        - Past Projects/Internship:

        - MIPS single and pipeline processor (CPRE 381)
        - Implemented Deep Neural Network Layer C++ Implementation (CPRE 487)
        - Multi-threaded request handling system (CPRE 308)
- Jonathan Tan
    - Top 3 languages/skills/etc.:
        - C
        - RTL Design
        - Debugging (soft/hardware)
    - Past Projects/Internship:
        - Research with Berk on Microarchitectural Side Channel Attack
        - Custom Hardware Convolutional Machine Learning Accelerator (CPRE 487 & personal project)
        - Firmware Intern @ Eaton working on the Smart Breaker 2.0 Communication
- Josh Czarniak
    - Top 3 languages/skills/etc.:
        - Java
        - C/C++
        - VHDL
    - Past Projects/Internship:
        - MIPS Processor
        - Chess boxing App
        - Coded the game of battleship.
- Kai Heng Gan
    - Top 3 languages/skills/etc.:
        - C/C++
        - Python
        - VHDL
    - Past Projects/Internship:
        - REMADE Prototype System Development
        - MIPS Processor
        - Manual Sorting Circuity
- Santi Campoverde
    - Top 3 languages/skills/etc.:
        - C/C++
        - Linux/Unix
        - Project Management
    - Past Projects/Internship:
        - Event Manager Mobile Application
        - Terminal Pokémon Rougelike
        - Customized Dynamic Reports using Power BI

2. Strategies for encouraging and supporting contributions and ideas from all team members.
    a. **Regular Brainstorming Sessions**: The team shall hold brainstorming sessions whenever needed, ensuring a safe and open forum for all members to contribute ideas. All ideas will be documented and considered without immediate judgment or criticism.
    b. **Rotation of Meeting Leadership**: Meeting leadership will rotate among all team members on a lucky draw basis (using pickerwheel.com). This is to ensure diverse leadership styles and give each member the opportunity to facilitate discussions on team progress and ideas.
    c. **Active Listening and Acknowledgment**: All team members commit to practicing active listening during discussions, valuing each contribution, and providing constructive feedback. This includes acknowledging and building upon others' ideas during team meetings.

d. **Respectful Feedback Mechanism**: Feedback on ideas will be given respectfully and constructively, focusing on the idea's content and viability rather than the individual presenting it. This ensures a positive and productive environment for idea development.
e. **Idea Implementation and Recognition**: When a team member's idea is selected for implementation, the team will acknowledge and celebrate this contribution, promoting a culture of recognition and motivation for innovative thinking.
f. **Encouraging Participation from All Members**: During meetings, efforts will be made to encourage input from quieter members, ensuring a balanced contribution from all team members. This may include direct solicitation of their opinions on specific topics.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)
   a. **Establish a Reporting Channel**: Designate a specific channel or person for team members to report collaboration or inclusion issues.
   b. **Confidentiality Assurance**: Assure team members that their concerns will be handled confidentially and with sensitivity.
   c. **Prompt Action**: Ensure that reported issues are addressed promptly to prevent escalation.
   d. **Feedback Loop**: Provide feedback to the team member who reported the issue on the actions taken and the resolution process.
   e. **Continuous Improvement**: Use reported issues as opportunities for continuous improvement in team dynamics and collaboration.

Goal setting, Planning, and Execution

1. Team goals for 491:
   a. Provide detailed documentation of the project setup and implementation for the client to easily replicate.
   b. Implement a single threaded implementation that involves every team member role, ultimately processing at least a single frame.
   c. Profile the single threaded implementation, analyzing its accuracy, and performance both software and hardware based.
   d. Present results to client and discuss steps for further implementation to speed up the process through a multi-threaded approach using the profiled results for decision making.

2. Team goals for 492:
   a. Develop multithreaded program that can run all three models in parallel.
   b. Optimize the semantic segmentation model to be run on the Xilinx Kria KV260 board.
   c. Profile multithreaded program performance, and model accuracy.
   d. Create documentation so that project can be hand down to future teams.

3. Strategies for planning and assigning individual and teamwork:
   a. **Skill-Based Task Allocation with Growth Opportunities**: We will assign tasks primarily based on each member's current skills while also considering their personal growth goals. This ensures that each member is working in areas they are proficient in, while also having the opportunity to develop in areas they wish to improve.
   b. **Flexible and Collaborative Assignment Process**: Maintain flexibility in task assignments, allowing for adjustments based on team members' feedback and project requirements. Encourage open discussions for task allocation, ensuring that members have a say in what they work on, aligning with their interests and development goals.

c. **Regular Review and Support**: Conduct regular reviews of task assignments to ensure they align with individual skills and growth aspirations. Provide necessary support and resources for tasks that are aligned with members' learning objectives, fostering an environment of continuous learning and improvement.

4. Strategies for keeping on task:
   a. **Active Oversight by Team Leader**: The team organization leader takes an active role in monitoring the schedules and progress of all team members. This includes keeping track of deadlines and ensuring that everyone is aware of their responsibilities and timelines.
   b. **Regular Progress Reporting in Meetings**: During each team meeting, members are required to report on their progress. This practice promotes accountability and provides a platform for discussing any challenges or delays in tasks.
   c. **Support and Task Reassignment for Struggling Members**: If a team member is significantly behind, the team leader will offer encouragement and support, which may include additional resources or guidance to help them catch up. In cases where the workload is found to be imbalanced, the leader has the discretion to reassign tasks to ensure equitable distribution of work and maintain project momentum.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
    a. In the event of any infractions or breaches of team contract obligations, immediate attention will be given to addressing the issues.
    b. A dedicated meeting will be promptly scheduled to address present conflicts or infractions. This meeting will serve as a platform for open communication, discussion, and resolution.
    c. The emphasis during the meeting will be on collaborative problem-solving. Team members will be encouraged to express their perspectives, and efforts will be made to find mutually agreeable solutions.

2. What will your team do if the infractions continue?
    a. Bring up the infractions with the advisor Dr. Jones and try to resolve the infractions being committed within the team. If infractions are not resolved or are severe, continue with the next bullet point for the next course of action.
    b. Bring up the infractions to Dr. Shannon/Fila, and request that action be taken upon the team member either by making adjustments within the team or removing the person from the team.

****

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) Santiago Campoverde DATE 1/30/24

2) Jonathan Tan DATE 1/30/24

3) Justin Wenzel DATE 1/30/24

4) Kai Heng Gan DATE 1/30/24

5) Josh Czarniak DATE 1/30/24