Heterogeneous Machine Learning

DESIGN DOCUMENT

Team Number
ssdec23-02
Client
JR Spidell
Advisors
Diane Rover
Team Members/Roles
Sandro Panchame, Rudolph Nahra, Alek Comstock, Jeffrey Kasper
Team Email
sddec23-02@iastate.edu
Team Website

Revised: Dec 10, 2023

Executive Summary

Development Standards & Practices Used

7001-2021 - IEEE Standard for Transparency of Autonomous Systems.

- Iso/iec/ieee international standard - software and systems engineering - software testing -- part 2: test processes - redline

1532-2002 - IEEE Standard for In-System Configuration of Programmable Devices

Summary of Requirements

- Our system takes, as input, a series of images that are frames of a video that was taken of a human eye at close range
- For each frame, system will use our machine learning model to determine: whether the person is blinking, and, if not blinking, the location of the pupil of the eye in the image.
- Our system must be able to process frames faster than they are received on average, such that minimal storage is needed for incoming frames.
- Our system will be implemented on a single Kria SOM 260 board

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents applied to your project.

CPRE308 - Operating Systems

CPRE288 - Embedded Systems

COM S 474/574 - Introduction to Machine Learning

CPRE381 - Computer Organization

CPRE487/587 - Hardware design for Machine Learning

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Team

1.1 Team Members

Sandro Panchame, Alek Comstock, Jeffery Kasper, Rudolph Nahra

1.2 Required Skill Sets for Your Project

(if feasible – tie them to the requirements)

- Embedded systems knowledge
- Experience working on a linux operating system
- General knowledge on the python language
- Knowledge of neural networks
- Skills in analyzing neural networks

1.3 Skill Sets covered by the Team

(for each skill, state which team member(s) cover it)

- general/base knowledge of embedded systems: Alek C., Rudolph Nahra, Jeffery K.
- Knowledge of Python Language: Sandro P., Rudolph Nahra, Jeffery K.
- Experience with linux system: Sandro P., Rudolph N., Alek C., Jeffery K.
- Knowledge of neural networks: Rudolph Nahra

1.4 Project Management Style Adopted by the team

Waterfall Project management: Each week, we meet with our client and he breaks down what has been accomplished each week, and what needs to get done. We are starting with small goals to establish our understanding of the technology, and building up to the final project. We will then find a person best suited for each week's goal and assign it - whether done by self-volunteer, best suited for the role, or elimination.

1.5 Initial Project Management Roles

(Enumerate which team member plays what role)

Alek Comstock - Embedded systems design

Jeffery Kasper - Embedded system design

Sandro Panchame - Neural network analysis and optimization

Rudy Nahra - Embedded system design

2 Introduction

2.1 Problem Statement

Eye tracking can require very high frame rates (200+ FPS) to correctly capture some eye movement patterns, such as the saccade.

It is difficult to achieve these frame rates in an embedded system.

We will achieve accurate and fast eye tracking using:

Heterogeneous hardware specialized for machine learning (ML)

Implemented on a field programmable gate array system on chip (FPGA SOC)

A custom ML model.

2.2 Requirements & Constraints

Functional Requirements:

- The system takes in a series of images of a human eye and, for each frame, outputs the position of the pupil in the image, whether they are blinking, and the eye movement pattern they are exhibiting
- The system processes a series of images of eye movement and blinks, and outputs the position of the pupil, and outputs if the eye is open or closed.
- The system runs on only a Kria SOM KV260 (constraint)

Functional Requirements:

- Process each frame of a video feed with enough throughput to keep up with incoming images
- The Root Mean Squared Error (RMSE) of pupil position estimation must be within 3 pixels of the ground truth
- Must make use of Real-Time Processing Units (RPU) to enable response to hard time constraints

The resource requirements:

2 Kria KV260 board
3 Development computer capable of running Vitis, Vivado, and Petalinux tools
4 Large dataset to train ML model

2.3 Engineering Standards

https://ieeexplore.ieee.org/document/9726144

7001-2021 - IEEE Standard for Transparency of Autonomous Systems.

This standard is important for our project because we need to analyze our deep learning model to ensure its output is safe, as our project could be employed in safety-critical applications. The standard will help us measure how safe they are.

https://standards.ieee.org/ieee/29119-2/7498/

29119-2-2021 - Iso/iec/ieee international standard - software and systems engineering - software testing -- part 2: test processes - redline

This standard is useful to use as a form of ensuring we meet many possible problems our project could run into. It ensures that each step along the way, any updates or changes to the project, should meet our testing standards.

https://ieeexplore.ieee.org/document/1176958

1532-2002 - IEEE Standard for In-System Configuration of Programmable Devices

This standard applies to our project because we will utilize an FPGA board. An FPGA board is a type of Programmable Device. We will need to utilize these standards to effectively configure the board.

2.4 Intended Users and Uses

Possible use cases:

- As a part of a larger system to track the eye movement of an operator of a large vehicle or aircraft and determine the operator's state to determine if they're fit to continue operation.
- To aid medical professionals in finding possible eye-related illnesses.
- As a proof-of-concept for other Machine Learning Algorithms to be used on an embedded system, information on this project could help develop other kinds of tools.

Intended users:

- Engineers
- System designers

3 Design

3.1 Design Context

3.1.2 User Needs

List each of your user groups. For each user group, list a needs statement in the form of:

User group needs (a way t0) do something (i.e., a task to accomplish, a practice to implement, a way to be) because some insight or detail about the user group.

- Engineers may need a subsystem to take in a person's eyes and output information based on the eye such as position and movement patterns.
- Doctors need a way to cheaply and non-invasively detect irregularities in a patient's eyes because rapid detection of eye diseases may improve patient outcomes.

3.1.3 Prior Work/Solutions

https://arxiv.org/pdf/2206.00877.pdf

3.2 Design Exploration

3.2.1 Design Decisions

- We decided to use real-time processors as our controller processor. This is because when integrated with a larger system, our system will need to respond quickly to incoming images and control signals, so we control everything from RPU because the RPU can guarantee that it responds to these stimuli in a timely manner.
- We decided to use our four APU cores to perform preprocessing on our images. This is because in our profiling, we discovered that preprocessing the images took much, much longer than inference on the DPU.
- We decided to have one large DPU instead of several smaller ones, as we reasoned that the DPU is already able to support very high parallelism, so it would not be necessary to run 3 DPUs in parallel.

3.2.2 Ideation

For one design decision, describe how you ideated or identified potential options (e.g., lotus blossom technique). List at least five options that you considered.

The project has been pretty linear, some decisions will have to be made that we haven't gotten to yet. Possible future decisions could be:

- Improving accuracy of the machine learning model
    - Training on a larger dataset
    - Try different model architectures
- Determine how many (APU and DPU) processing units we will use.
    - The time it takes to process the images, and classify the movements will determine how we go about this.
    - Times we were working with were assumed, may not be the case.

3.2.3 Decision-Making and Trade-Off

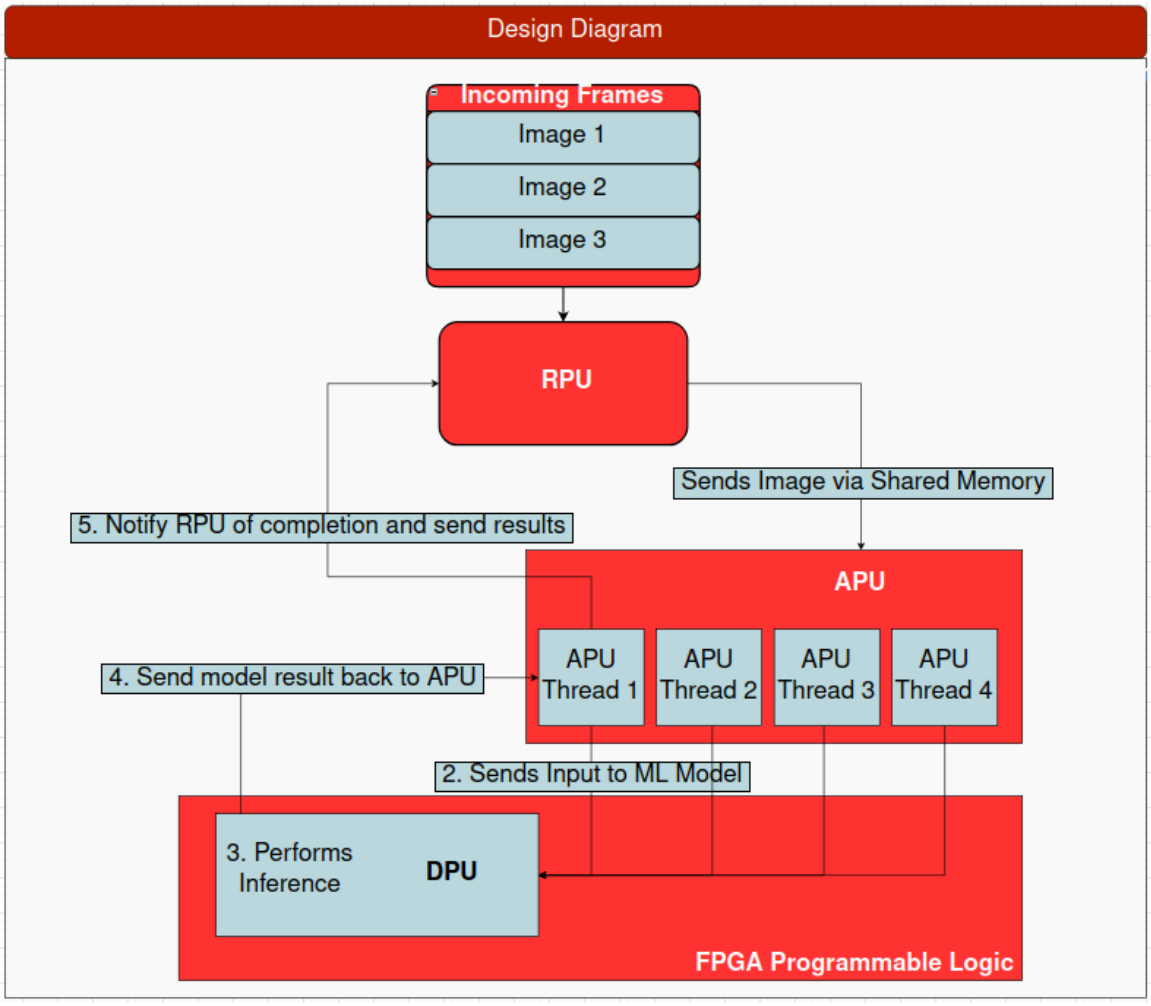
Demonstrate the process you used to identify the pros and cons or trade-offs between each of your ideated options. You may wish you include a weighted decision matrix or other relevant tool. Describe the option you chose and why you chose it.

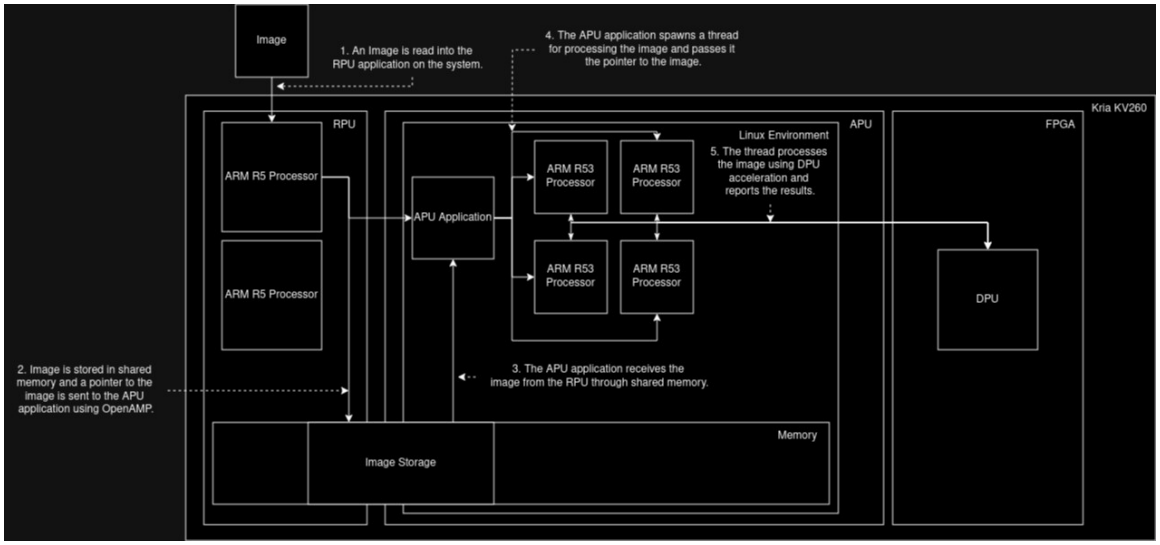
- The number of processing units was largely predetermined; part of our challenge was to get the algorithms working on the specified board, which came with its own processing units. There is the opportunity to add one or 2 more, but it's not necessary until we determine we don't have enough processing power.

- While there may exist other ways to reach our goals, there are reasons as to why we haven't explored those avenues. This is mostly due to time constraints, with how much time we put into research and getting things to work on our current plan, we may not have time to explore other options.

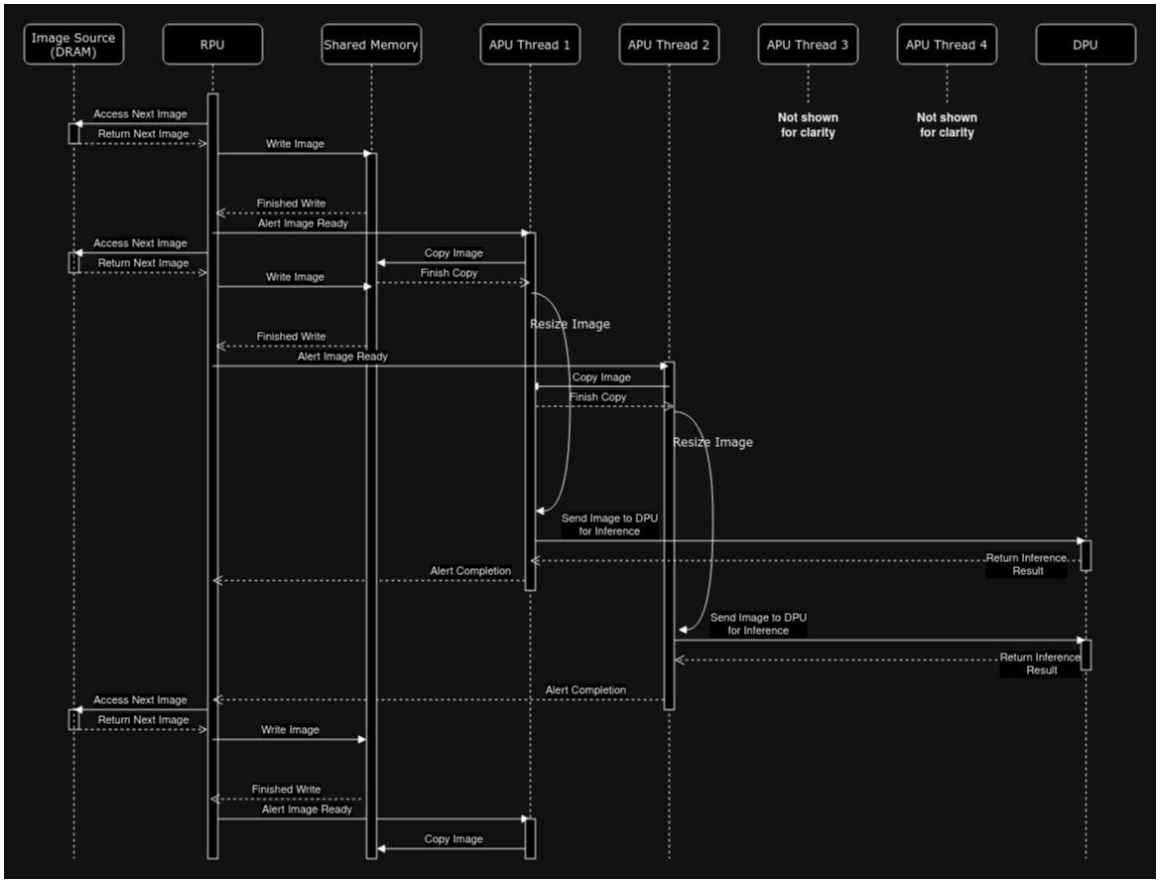
3.3 Updated Design

3.3.1 Design Visual and Description

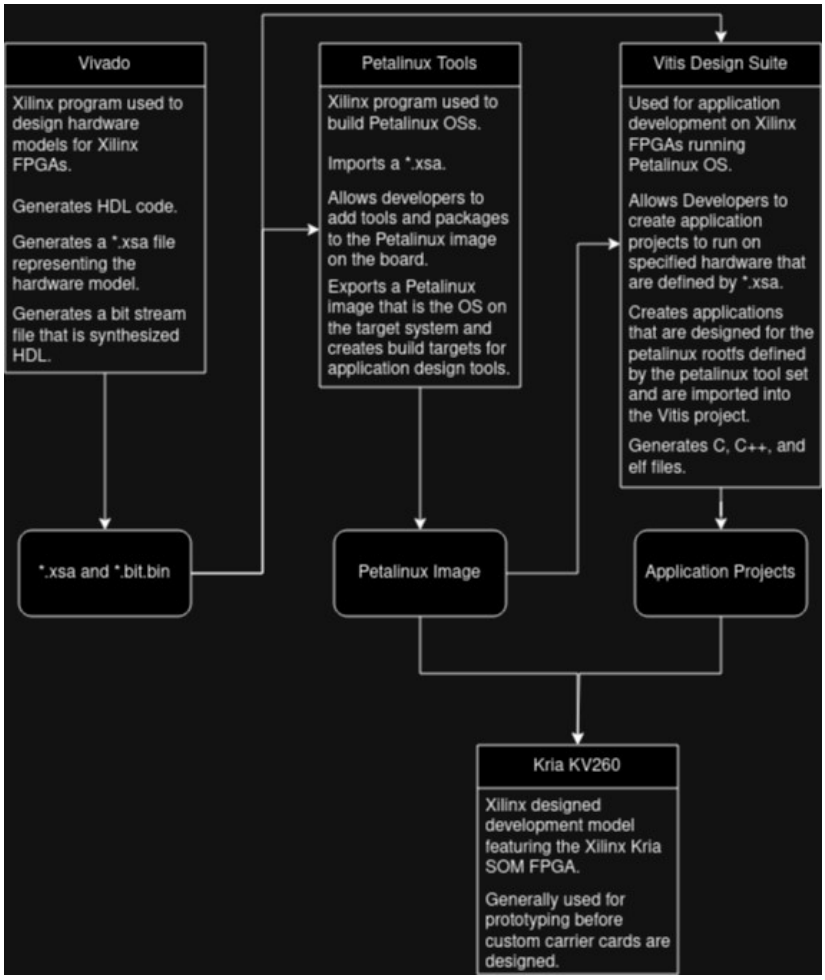

Design Diagram
Incoming Frames
Image 1
Image 2
Image 3
RPU
Sends Image via Shared Memory
5. Notify RPU of completion and send results
APU
APU Thread 1
APU Thread 2
APU Thread 3
APU Thread 4
4. Send model result back to APU
2. Sends Input to ML Model
3. Performs Inference
DPU
FPGA Programmable Logic



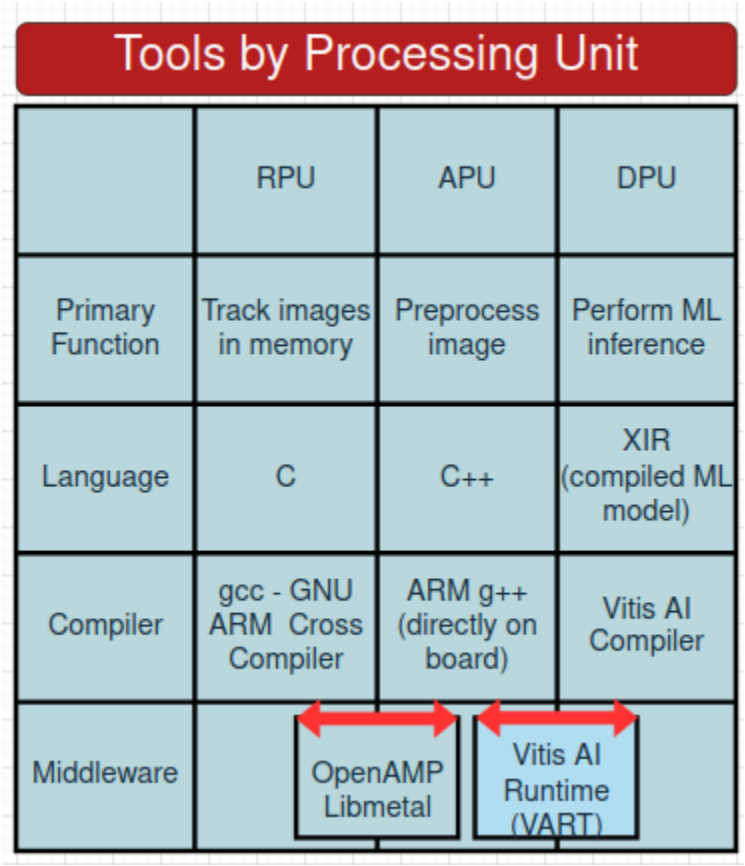
The above images describe how an image will flow through the different heterogeneous computing elements within our system. There are three different regions of the Kria KV260 shown. The RPU is responsible for taking in images in real time and sending them via shared memory to the APU. The APU performs preprocessing in a multithreaded fashion, then sends the images to the DPU. The DPU performs the inference on the image, then sends the result back to the APU. The APU finally notifies the RPU of the completion.



This diagram depicts a timeline of our multithreaded processing, as well as the interaction between each heterogenous element.



The task that an APU must run for each frame of recorded eye movement is as follows: rescale the image, then use the rescaled image as the input to a neural network. The output of the neural network is then sent via bus to an external device.



3.5 Previous Design Analysis

- Neural network verification through Marabou proved to be a challenge to work with and was cut from the project.
    - The verification of a model takes O(2^n) time to run
    - To test it against the model required a lot of RAM
        - a model taking in a 128x128 image and containing one dense hidden layer required 24GB of RAM
        - A model taking a 64x64 image and containing one convolutional layer would crash with 24GB of RAM and 50GB of swap space
    - More applicable to models with a small number of inputs.
- Movement classification through remodnav had, quite ironically, the opposite issue.
    - Remodnav is meant to work on a time series, where portions of the data are chunked and clssifications can begin.

    - With only two positions to work with, classification was often incorrect or did not occur at all.
- For Neural Network verification, our task may need to be split into smaller portions
    - if the output of each layer is known, perhaps the verification can be done layer by layer
    - Maybe the frames can be sectioned off into four quadrants, with respect to the pupil center, and verification can be performed on those sections.
- For movement classification, perhaps it is best to train an additional model.
    - For classification, the model would most likely need to take in information such as the distance of pupil centers from the subsequent frames, and the time between the frames.

3.7 Evolution since Senior Design 1

Many changes were made from our design in the first semester.

- Most importantly, we have added the use of the Xilinx Deep Learning Processing Unit (DPU) within the FPGA fabric of our board. This processor is able to perform inference of the model much, much faster than the APU can. This also adds significant complexity to the APU.
- We have decided that our APU cores should run in Symmetric Multiprocessing (SMP) mode, with access to the petalinux operating system, rather than standalone. This is because they need access to the Vitis AI Runtime (VART) and OpenAMP libraries to communicate with the other heterogenous computing elements on the board.
- We decided to train our own model, rather than use one provided by the client.

Marabou -

Marabou was a framework for the verification of neural networks. Our initial plan was to use it to with our machine learning model; the problem is Marabou is optimized for small neural networks, so it would use large amounts of RAM (>30Gb) before eventually crashing.

4 Testing

4.1 Unit Testing

- The neural network is tested by separating our training data into two parts: training data, and testing data. After training, the model is evaluated based on its performance on the testing data.
- Developing a workflow to feed the neural network frame data and classifying movement as the video runs to visualize performance

- Unit testing our individual processing elements would not have any meaning, as each of them MUST interact with another element to perform even part of its job.

4.3 Integration Testing

- We have many different parts to integrate into our design. These are:
    - RPU
    - APU
    - DPU
    - OpenAMP memory region
    - Libmetal memory region
- To test communication between the RPU and APU via OpenAMP, we send a predetermined series of bytes from the APU to the RPU, and the RPU simply sends them back to the APU. The APU ensures that all the bytes are the same value that it sent.
- To test communication between the RPU and APU via Libmetal, we placed an image into the shared memory region on the RPU side then from the APU side, we read the image and verified that all data was the same.
- To test integration between the APU and DPU, we performed inference on images loaded from the petalinux filesystem.

4.4 System Testing

- We tested the system by loading images from the petalinux filesystem and performing inference on them using the DPU. Importantly, we recorded the time that it took to load the images, the time it took to preprocess the images, and the time that it took to perform inference.

4.5 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

- We demonstrate that the system meets the requirements by using the time taken by our system test over an inference of 1000 frames. We also measure the RMSE of inference performed by our model on the board.

4.6 Security Testing (if applicable)

- This does not apply to the project because it is concerned with finding the center of a pupil. If, say, identifying information of the people (name, age, race, etc.) involved were stored somewhere, there would be a security concern. Access to the embedded system would be the concern of the larger system in which it is integrated

4.7 Results

- Our CNN achieved an RMSE of 2.54 pixels in images of dimension 384x288. For comparison, the average radius of the pupil in these images is about 35 pixels.

- In our testing of throughput and latency, we were able to process 220 frames per second, with the most time being spent in preprocessing of the image.

5 Implementation

- We have a trained neural network that has RMSE value of 2.54 on testing data
- Simulated real-time performance of feeding the model video frames and deriving movement classifications with the use of a circular buffer.
- Application of the structural similarity index in an attempt to reduce redundancy in the dataset.
- Utilize OpenAMP to design an algorithm to take in real-time events (video frames) and facilitate computational cores to run the neural network on the events

5.1 The Machine Learning Model

The machine learning model uses an architecture that was provided by the client. Originally trained on a different dataset, however, the predictions made by the model did not reflect a possible position within the frame. This may have been due to a transformation concerning projection. The provided architecture was used for the model and trained on a new, larger dataset.

Different architectures, with fewer layers, were experimented with, however, they did not prove as accurate as the provided architecture in training.

5.2 Frame Processing Algorithm

Our project utilizes three applications to process frames.

5.2.1 The RPU

The RPU is responsible for receiving a frame and transferring the frame to the APU in real-time. After the frames have been processed the results will be sent to the RPU. The RPU will report processing results in frame order. To accomplish its task, the RPU uses two libraries: OpenAMP and Libmetal. First, the RPU loads an image into a memory region that is shared between APU and RPU using Libmetal. Once the image is in memory, the RPU sends a small message to the APU via OpenAMP as a notification that the image is now in memory, as well as the specific location and the frame number.

5.2.2 The APU

The APU receives frames from the RPU and performs preprocessing on them. The APU is multi-threaded allowing the APU to preprocess multiple frames at once. After the APU has finished preprocessing on a frame it will facilitate inferring from the DPU. Once the DPU has returned the results of the inference the APU will forward the results to the RPU. The APU uses the Vitis AI Runtime (VART) library to facilitate interaction with the DPU. This library allows us to send an input to the DPU as an array of bytes, and receive the result asynchronously.

5.2.3 The DPU

The Deep Learning Processing Unit or DPU is responsible for inferring results from data that is sent to it. The DPU is Xilinx intellectual property and a soft core that we programmed into the FPGA logic of the Kria K26 SOM chip. Utilizing Xilinx tools we programmed the DPU with our Machine learning model so the DPU could utilize the model to perform inferences. The DPU performs inference much faster than any other step in this process.

6 Closing Material

6.1 Discussion

Discuss the main results of your project – for a product, discuss if the requirements are met, for an experiment-oriented project – what are the results of the experiment, if you were validating a hypothesis – did it work?

- Since the neural network is being trained on frame data from videos, and considering the similarity of images from frame to frame. An attempt to reduce redundancy in the dataset was made
    - Through the application of the Structural Similarity index, and varying thresholds of .94, .92, and .90. Models of the same architecture were trained on subsets of the training set.
    - Compared to the control model, which had an RMSE of 2.54, the other models had RMSE values of 2.62, 2.95, and 4.54. Respective to the previously mentioned thresholds.
- The acquisition of the larger dataset proved to be a considerable boon.
    - It provided insight on the general workflow in working with a considerably large dataset
    - It illustrated the need for appropriate system requirements. Often, on the virtual machine acquired from the University's ETG service, requests for additional storage space and RAM were made.
    - Often, assumptions regarding the dataset had to be revisited and changes needed to be made accordingly

- To demonstrate the performance of the model and classification of eye movement, a circular buffer, OpenCV, and remodnav were used.
    - Visualization of the predicted against the true values against a video was achieved
    - The distance between those values was displayed
    - Classifications of the eye movements against the classifications in the dataset were also displayed
    - Classifications other than fixation could not be observed.
- In our testing of throughput and latency, we were able to process 220 frames per second, with the most time being spent in preprocessing of the image.

6.2 Conclusion

Summarize the work you have done so far. Briefly reiterate your goals. Then, reiterate the best plan of action (or solution) to achieving your goals. What constrained you from achieving these goals (if something did)? What could be done differently in a future design/implementation iteration to achieve these goals?

On the hardware, we have had success installing and interacting with the PetaLinux operating system. In order to run this operating system we needed to update the firmware. We have begun writing code to utilize the functionality provided by the OpenAMP framework. This code will be used to run the neural network in a system similar to multi-threading. This system will need to finish fast enough so that when a new frame is provided a processor can begin running the neural network against the frame.

On the machine-learning side, when obtaining a larger dataset for training, data organization, wrangling, and frame extraction needed to be performed. Furthermore, on movement classification, Remodnav was more geared towards a time series of eye movement velocities, and for proper classification, more frames were needed instead of simply two.

Before training an idea was proposed to remove extremely similar images from the dataset for training. The Structural similarity index was used as a metric for removal. Four models, trained on the full portion of training data and trimmed portions of varying thresholds, were compared. Ultimately, a model trained on data trimmed with a threshold of .94 had a similar performance to a model trained on the complete training data.

6.3 References

List technical references and related work/market survey references. Do professional citation style (ex. IEEE).

[1]

J. Grese, C. S. Păsăreanu, and Erfan Pakdamanian, “Formal Analysis of a Neural Network Predictor inShared-Control Autonomous Driving,” *AIAA Scitech 2021 Forum*, Jan. 2021, doi: https://doi.org/10.2514/6.2021-1580.

[2]

“Kria KV260 and PetaLinux 2022.1: Part 01-Getting Started,” *Hackster.io*. https://www.hackster.io/mohammad-hosseinabady2/kria-kv260-and-petalinux-2022-1-part-01-getting-started-ed5a25 (accessed Apr. 24, 2023).

[3]

G. Katz *et al.*, “Con sis te n t * Comple te * W ell Docu m e n te d * Easy to R e us e * The Marabou Framework for Verification and Analysis of Deep Neural Networks.” Accessed: Apr. 24, 2023. [Online]. Available: https://aisafety.stanford.edu/marabou/MarabouCAV2019.pdf

7 Appendix

7.1 Appendix I

Operation Manual

Notice: The source code of our project, as well as the grand majority of files mentioned in this manual, have been kept private at the request of our client.

To operate the system, perform the following steps:

1. Obtain a WIC image of our custom petalinux build
2. Burn WIC image to Kria KV260’s microSD card
3. Obtain user home zip file, containing our source code as well as several scripts to facilitate build and development.
4. Unzip home zip file, and place contents in the /home/petalinux directory on the SD card.
5. Plug in power to Kria KV260.
6. Load the DPU into PL by running the ./load_dpu.sh script
7. Move the RPU firmware in the user home directory to the /lib/firmware directory
8. Load the RPU firmware and start the RPU by running the ./start_rpu.sh script

9. Images for inference reside in the images directory in the user home folder. These images can be replaced with new ones.
10. Run the ‘sd-inference’ program and observe results. The program can be modified and rebuilt by running the ./build.sh script.