Machine Learning & Heterogeneous Computing for Real-time Eye Tracking

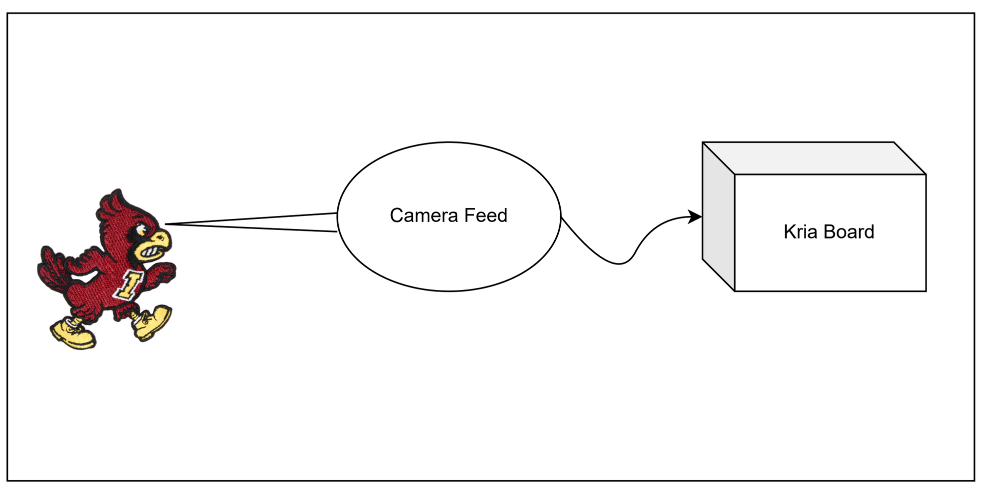
Members: Rudolph Nahra, Sandro Panchame, Jefferey Kasper, Alek Comstock
Advisor: Diane Rover, Client: JR Spidell

Introduction

- Problem
Eye tracking of humans can require very high frame rates to correctly capture eye movement patterns. It is difficult to achieve these frame rates in an embedded system.
- Solution
Accurate and fast eye tracking can be achieved using hardware using heterogeneous hardware specialized for machine learning (ML) on a field programmable gate array system-on-chip (FPGA SOC), and a custom ML model.

Intended Uses/Users

- Uses
Track fast human eye movements for use in larger systems
- Users
Designers & engineers of larger systems
ex: Disability assistance systems, auto focusing, health evaluation



Tools by Processing Unit

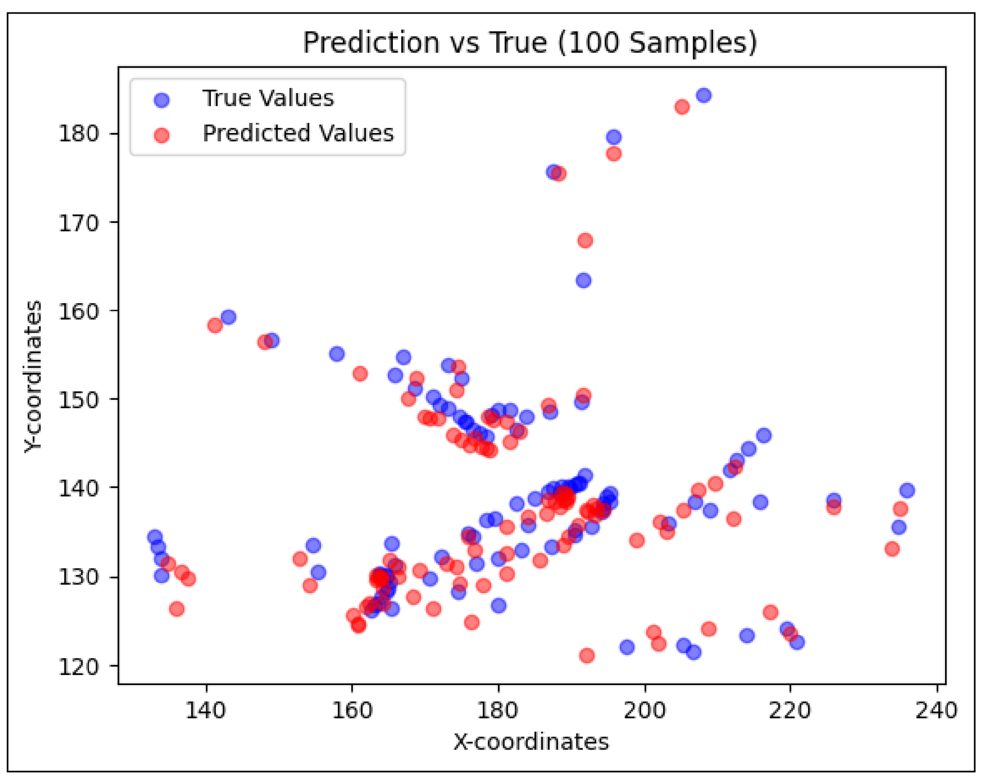
	RPU	APU	DPU
Primary Function	Track images in memory	Preprocess image	Perform ML inference
Language	C	C++	XIR (compiled ML model)
Compiler	gcc - GNU ARM Cross Compiler	ARM g++ (directly on board)	Vitis AI Compiler
Communi-cation Library	OpenAMP Libmetal		Vitis AI Runtime (VART)

Machine Learning Details

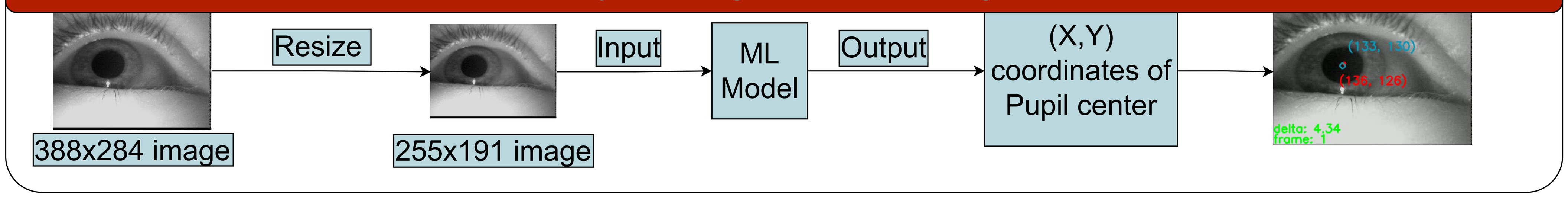
The machine learning models were trained using the TEyeD dataset, which contained videos of eye movement and annotated data. Roughly 260,000 frames were used in training, with an output of xy coordinates.

Tools used: Python 3.10.0, tensorflow, pandas, numpy, opencv-python, scikit-image, matplotlib, jupyter

Model Predicted Outputs VS Actual Values



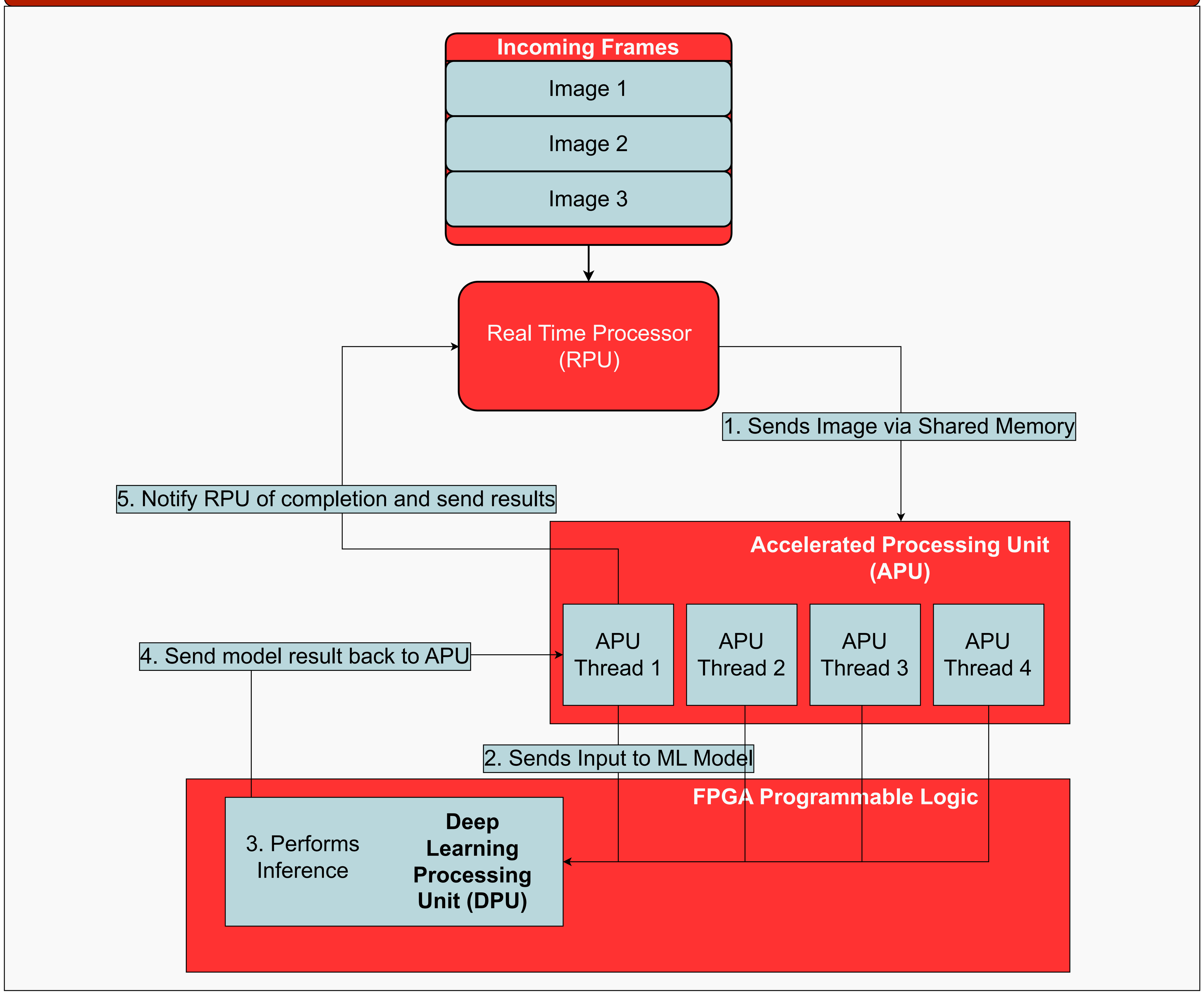
Eye Image Processing



Steps: 1 -> 2 Accelerated Processing Unit (APU)

Steps: 2 -> 3 Deep learning Processing Unit (DPU)

Design Diagram



Design Requirements

<u>Functional Requirements:</u>
- Take in images from video feed of eye
- Output position of pupil, open/close, and eye movement pattern

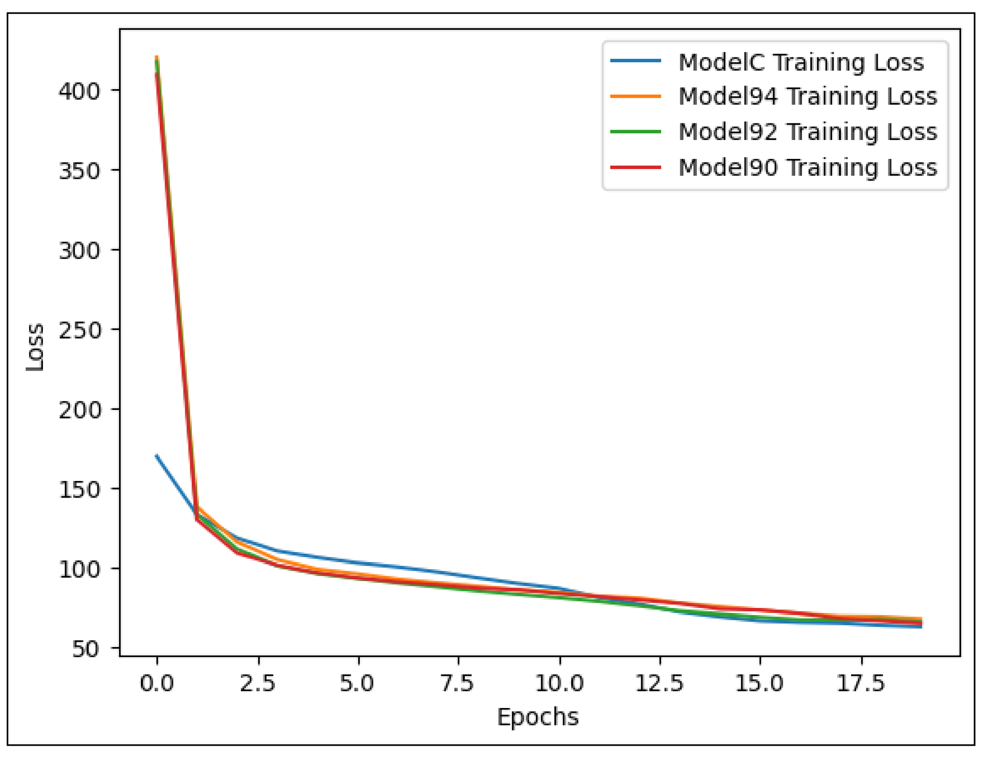
<u>Nonfunctional Requirements:</u>
- Process each frame of a video feed in less than 1/60th of a second per frame.
- Root Mean Squared Error (RMSE) of pupil position estimation be within the radius of the pupil
- Usage of the Real-time Processing Units to enable response to hard time constraints

<u>Resource requirements/constraints:</u>
- Restricted to the Kria KV260 platform

<u>Standards</u>
- IEEE 3161-2022: IEEE Standard for Digital Retina Systems
- IEEE 2671-2022: IEEE Standard for General Requirements of Online Detection Based on Machine Vision in Intelligent Manufacturing

Image of Kria KV260 board

Training Loss of All Models



Testing and Results

Test Type	RPU	APU	DPU	ML Model
Unit Test	N/A	N/A	N/A	Evaluation of root mean squared error on validation dataset with non-quantized model
Integration Test	Measured ability and speed of message passing between APU and RPU. Result: ~7300 messages/sec		Performed ML inference on DPU with images loaded from disk. Measured throughput of inferences including time to load and resize images. Result: 205 frames per second	
Throughput Test	Full system throughput test. Measured time start to finish of inference of 1000 frames. Includes RPU reading frames from memory, sending to APU, which sends to DPU for inference and sends result back to RPU.			
Correctness Test	As part of integration test, confirmed that message contents are not corrupted in messaging system.		Compared output of DPU inference with TensorFlow inference results to confirm negligible quantization error	Calculated RMSE of 2.54 pixels, compared to pupil with average diameter 70 pixels and image size 384 by 288