EyeMatic

–

Eye Anatomy Recognition ECE Capstone

Final Design Review

Overview

1. Problem
2. EyeMatic
3. Team
4. Diagrams
5. Demo Videos
6. Components
7. Machine Learning
8. Completed Tasks
9. Model Progress
10. Issues
11. Future Improvements
12. Acknowledgements
13. Conclusion + Q&A

Problem

- Cataract surgery is very common
    - ⅙ Americans develop cataracts by age 40
    - More than half of Americans develop cataracts by 80
- Patients' eyes usually rotate or move around when patients lie down during surgery
    - Makes surgery confusing and complicated at times
    - Surgeon’s attention primarily focused on camera display

What is EyeMatic? - The Solution

- EyeMatic is a camera system that utilizes machine learning in order to detect eye anatomy on a patient's eye.
- Differentiate different anatomy of eye anterior → makes system smarter
    - Guide other surgical devices to do eye alignment + registration
- Allows surgeons to focus more on the surgery at hand

Team

Kenya Aridomi

EYE MATIC

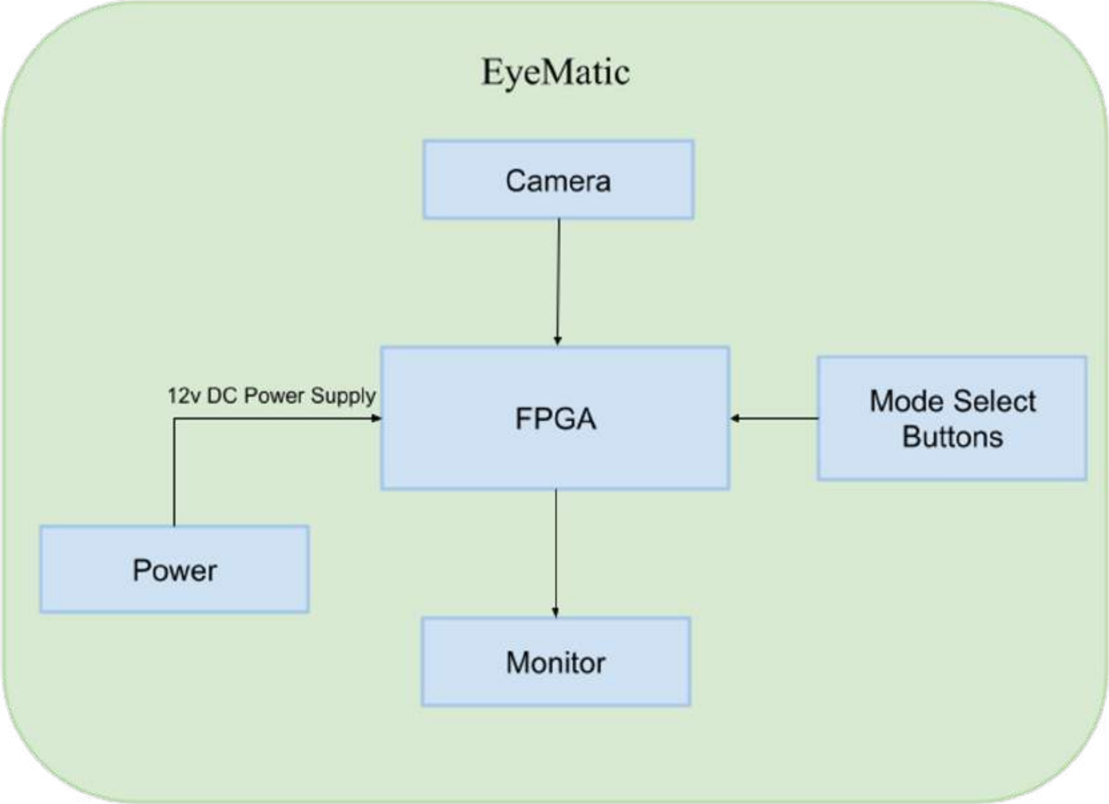
Andrew Chen

Michelle Ly

Ethan Nguyen

Marco Wong

Block Diagram



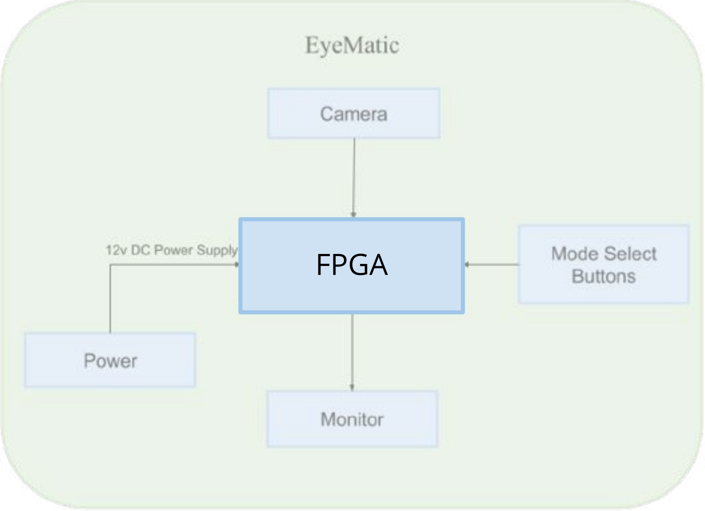
Workflow

1. Obtain images for tagging and training the neural network. Images are sourced from different datasets.
2. Tag the images.
3. Train the model, Tiny YOLOv3 (based on the Darknet framework) to generate the .weights file.*
4. Run the customized .sh script to convert the Darknet config file, weights, and labels to the open-source ONNX format. Output is network in .hex.
5. Configure the post-processing C code in SoftConsole to align with the neural network's configuration. Clean, build, and generate the firmware.
6. Load the neural network model to an available address slot and the firmware onto the FPGA using Libero.
7. Generate the bitstream and program/load it onto the FPGA (takes about 15-20 minutes).
8. Power cycle the board, then wait for the firmware to be read from the SPI flash storage.
9. Test the system to ensure appropriate functionality.
10. The process is complete once testing is successful.

*Modifying the configuration file as necessary.

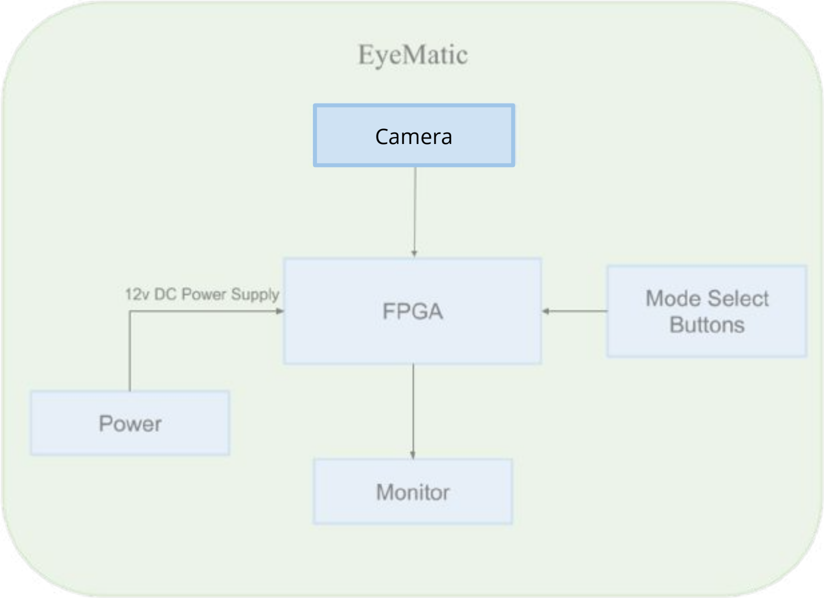
Components - FPGA

- PolarFire FPGA MPF300-VIDEO-KIT - manufactured by Microchip Technology
- 4GB DDR4 x32 RAM
- Embedded programming and debugging using SPI and JTAG
- 300K Logic Elements
- 1x 1Gb SPI Flash Memory
- USB to UART interface
- HDMI 2.0 RX and TX
- HDMI 1.4 TX



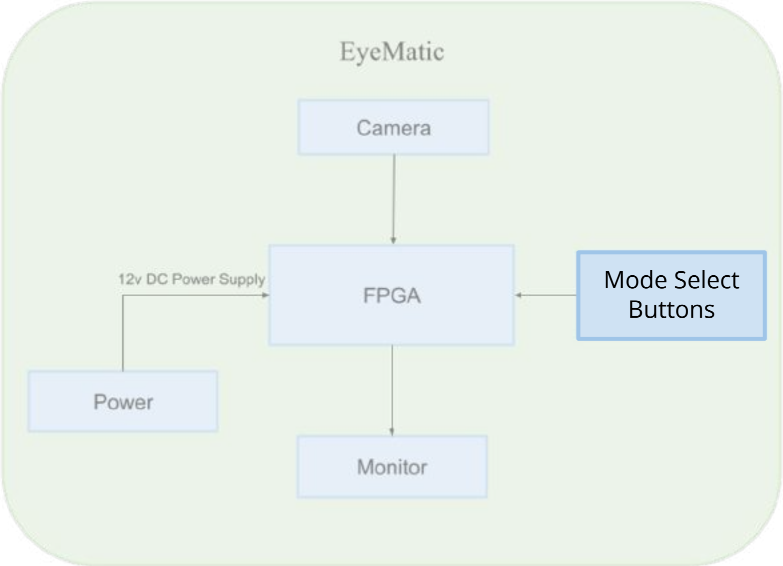
Components - Camera

- Sony Dual Camera Sensor over Amphenol FCI connector
- 60FPS RGB
- 8.42M pixels resolution



Components - Mode Select Buttons

- Allows user to pick and choose from a list of programmed models
  - Flashed and stored in SPI memory

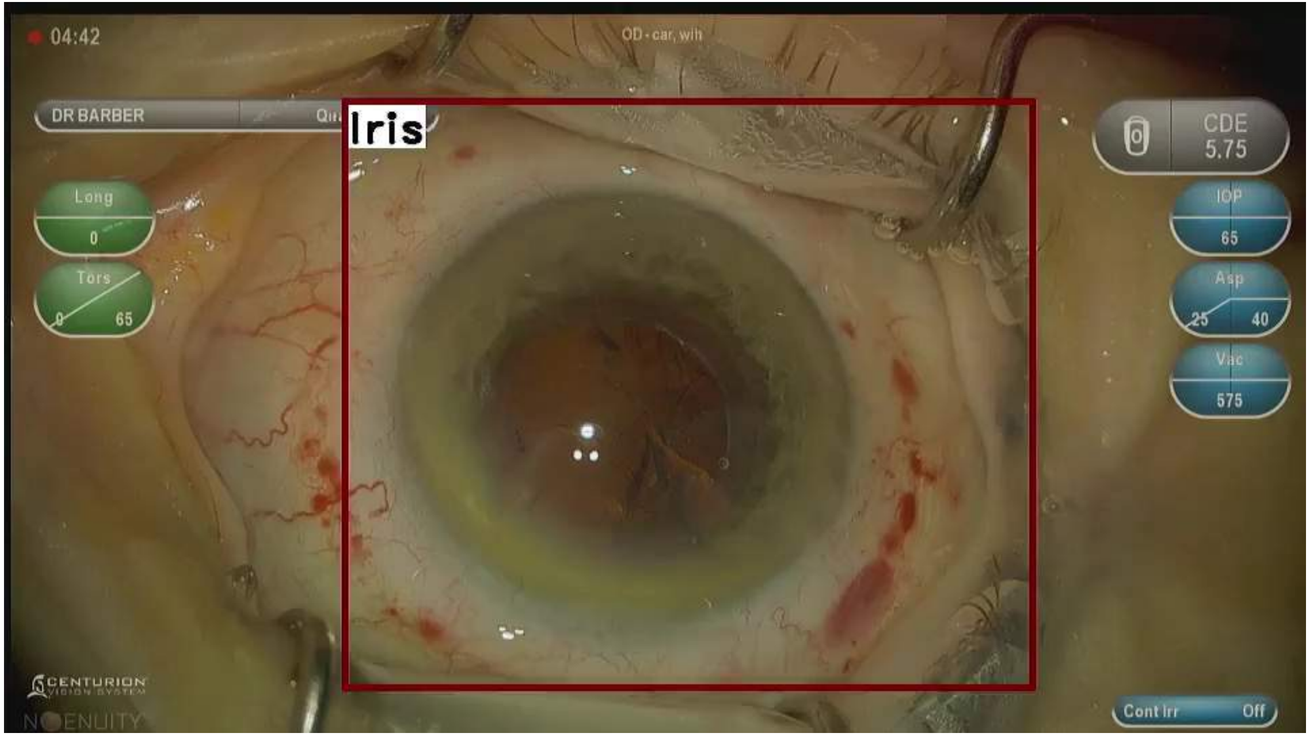


Components - Microscope

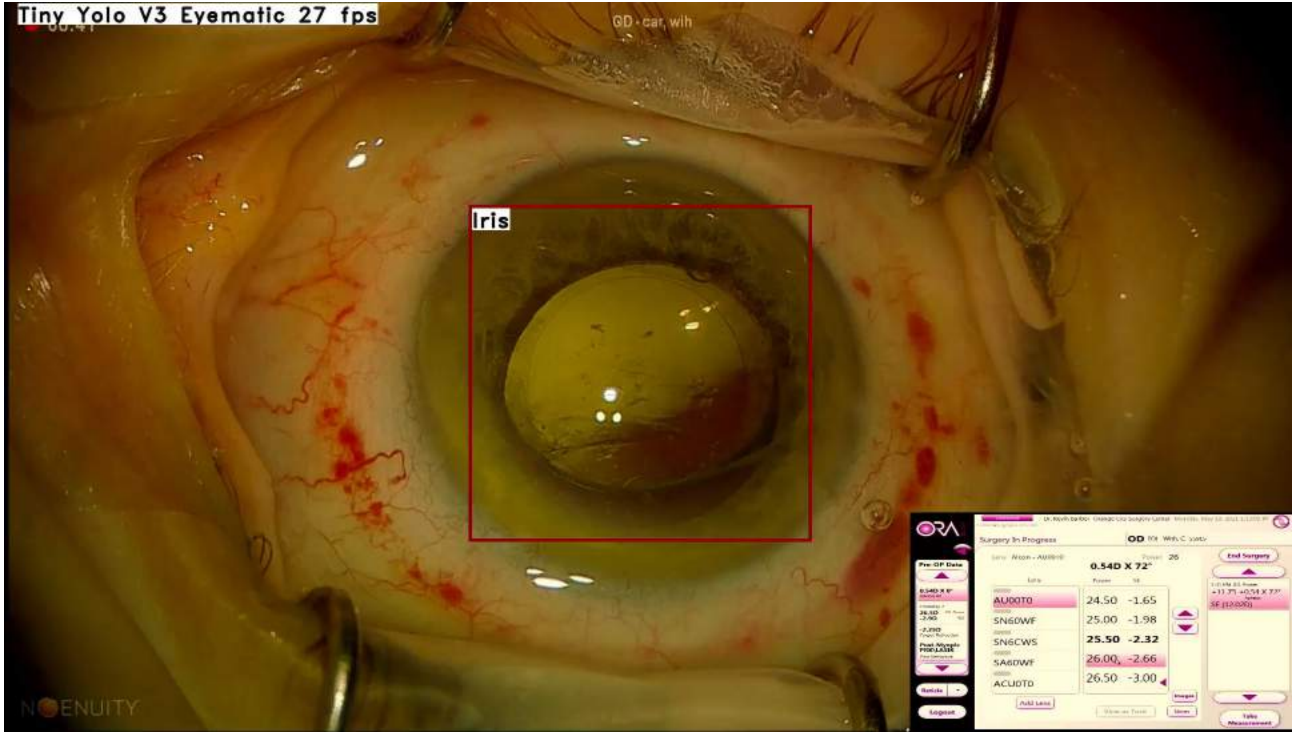
- TrueVision, 3D Surgical Camera
  - Works similarly to a microscope
  - Can zoom in, change lights
    - Utilized with:
      - Glass eye
      - Physical tools
  - Uses FPGA camera attached to FPGA for machine learning detection

VIEWER DISCRETION ADVISED:
CLOSE UP OF EYES, SHARP OBJECTS

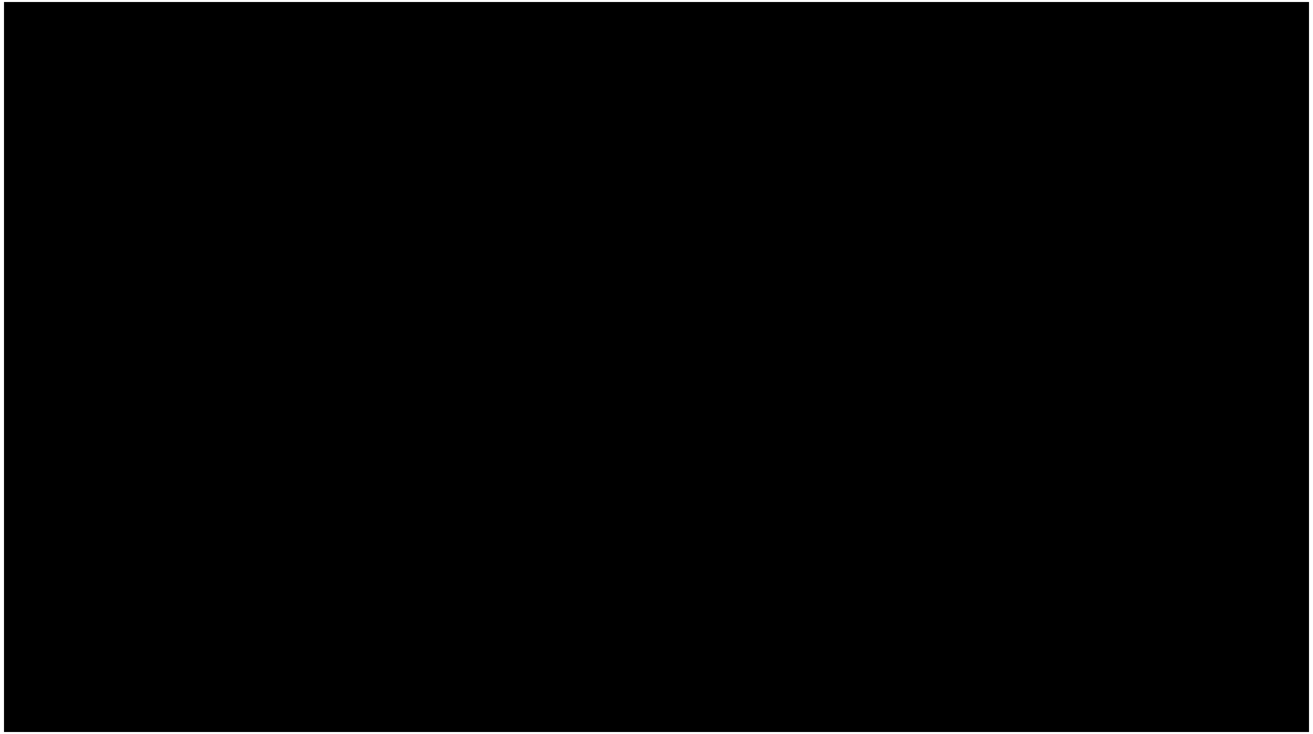
Demo Video (2 Classes, Model)



cont. 3 Classes, Flashed Model

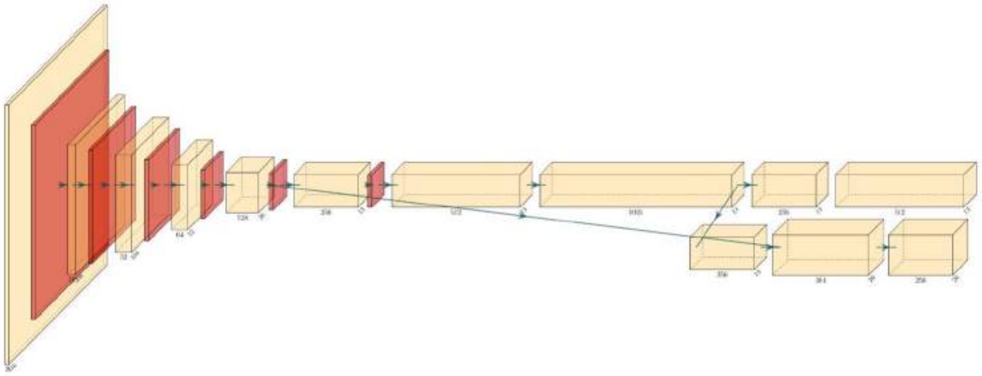


cont. 3 Classes, Flashed Model, using Microscope



Machine Learning

- Convolutional Neural Net (Tiny YOLOv3)
  - Detection task
    - Iris, Pupil, Instrument
  - Darknet architecture for fast GPU training
  - Small and easy to train/run
- Datasets
  - Footage from Alcon split into stills
  - Separate additional datasets for diversity (IEEE, openEDS)



Completed Tasks

Finished retagging images

New category: instruments

Found more images for more accurate training

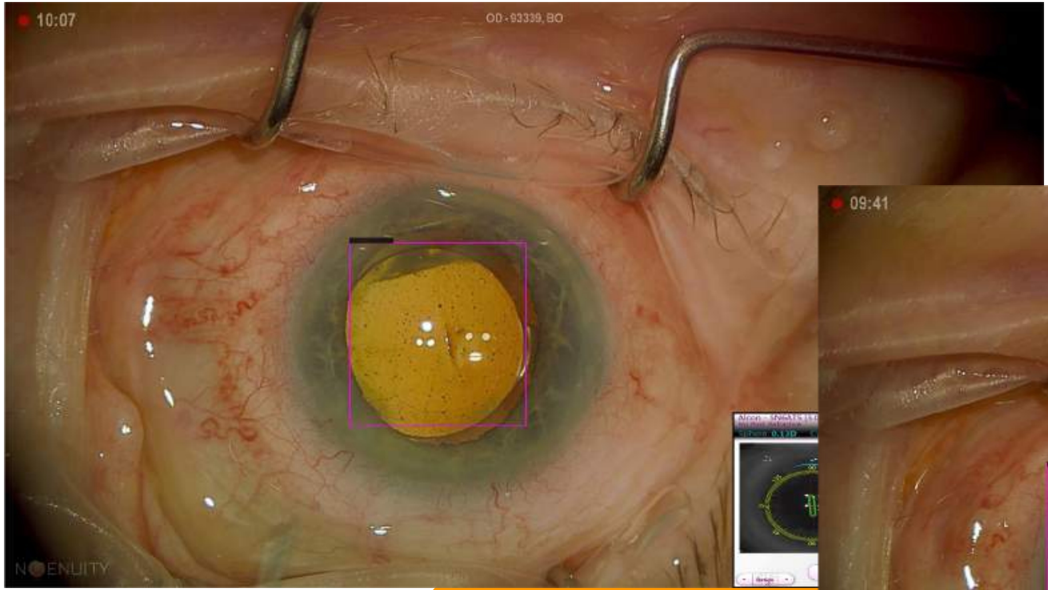
(Eyes without cataracts)

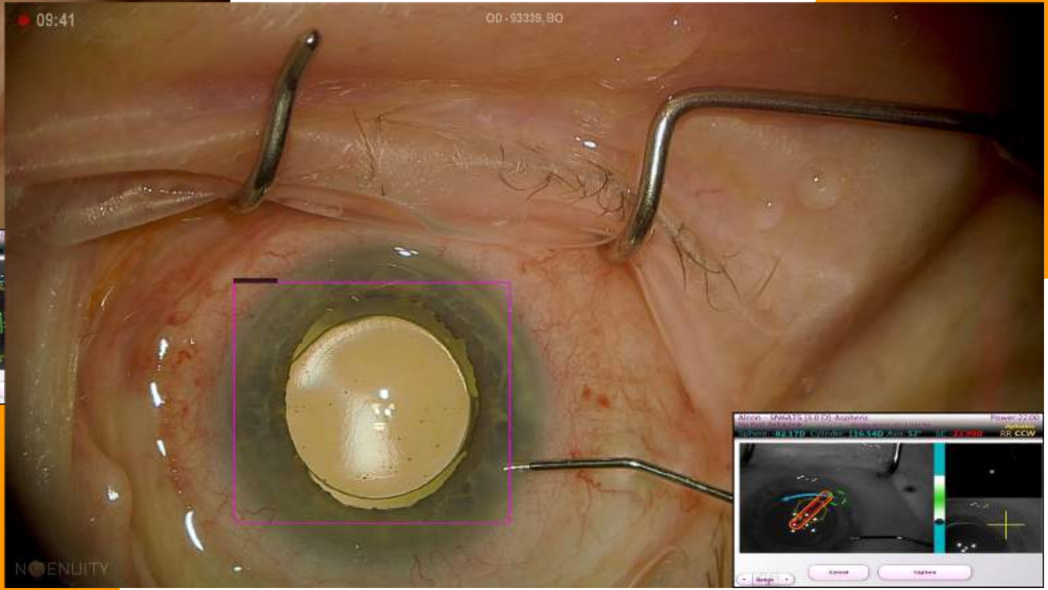
Trained new model

Uploaded new model

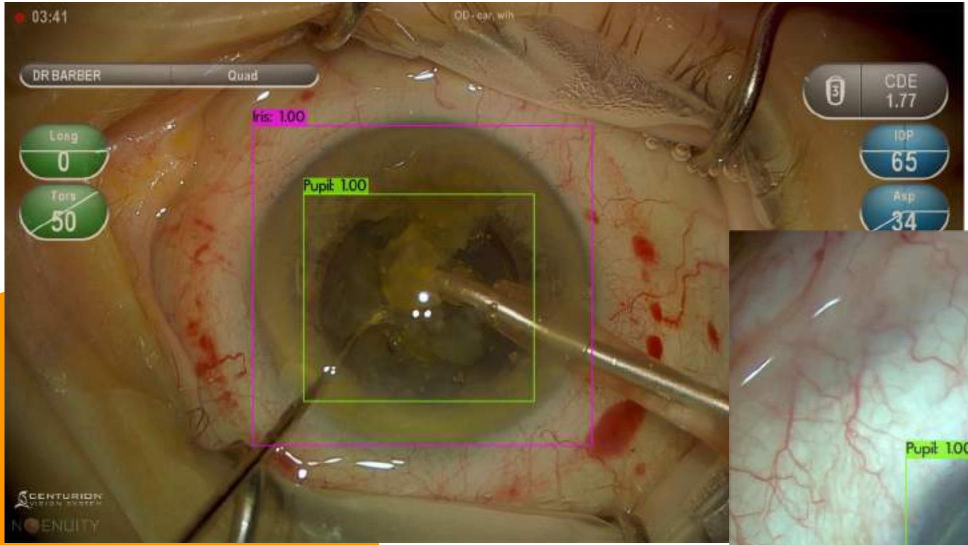
VIEWER DISCRETION ADVISED:
CLOSE UP OF EYES, SHARP OBJECTS

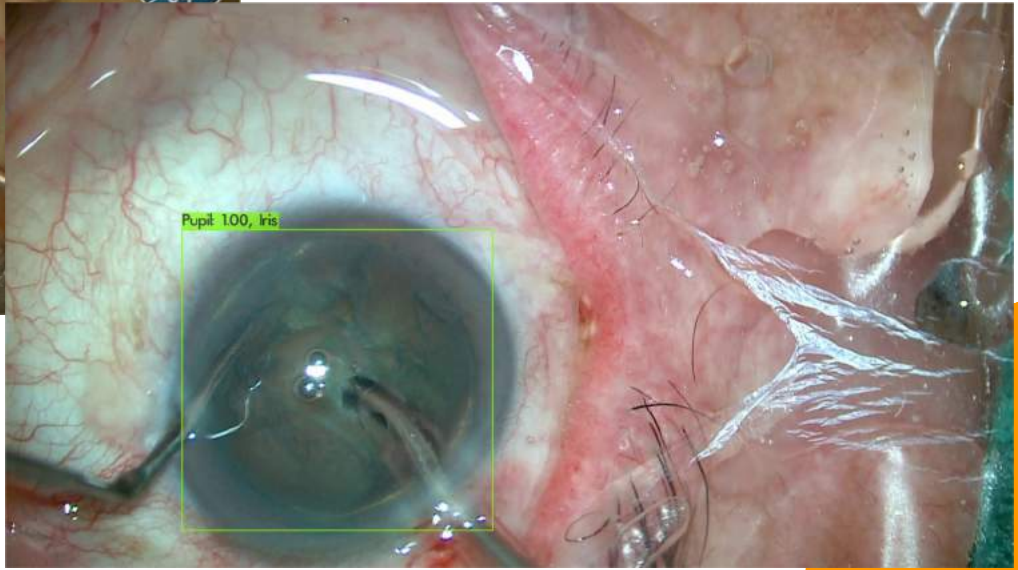
Model Progress (FALL)



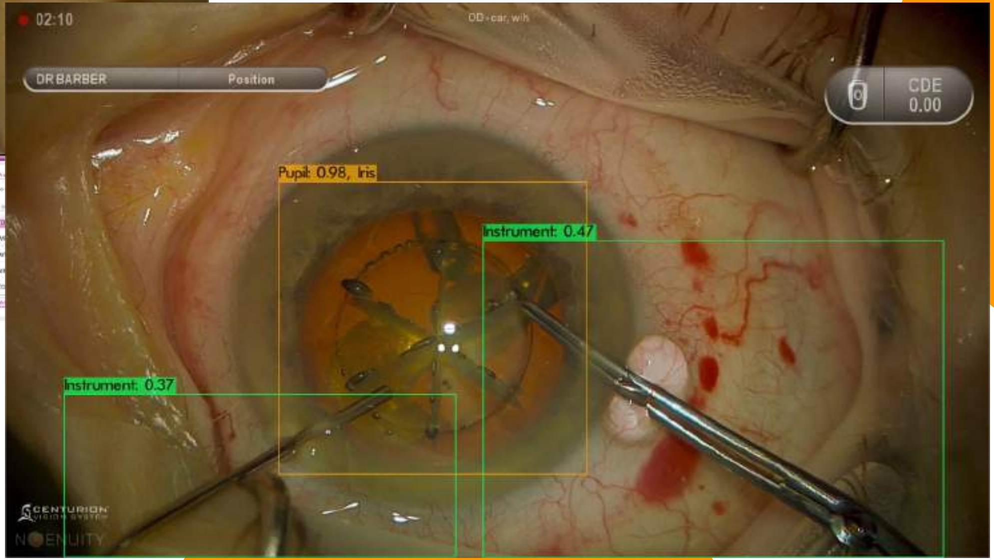


Model Progress (WINTER)





Model Progress (SPRING)



Issues

1. Out of date firmware: We had two conflicting versions of pre-existing firmware that was given to us. This led to the board being unable to flash or even display anything.

2. Hard coded C code with poor documentation: A lot of the code was hardcoded and was not mentioned in their demo guide. Led to many issues that could have been resolved had this been documented (i.e. random bounding boxes, or models not loading in).

3. Bad Video Drivers: There was a problem with the HDMI output from the main PC and some laptops that didn’t allow for us to receive video output from the board. This issue was seemingly random and out of our control.

Future Improvements

- Accuracy with model, some images where the pupil and iris look too similar have issues
    - Increase data diversity
    - Train using more tagged images, being cautious with overfitting
- Stable setup to mount model for the microscope
    - Build case and corresponding adapters to ensure consistent camera feed
- Scalability
    - Costs significantly less relative to surgical cameras
    - Minimize fixed costs such as component price, protective case and adapters

Acknowledgements

Dr. Yogananda Isukapalli (Capstone Lead)

Alex Lai (TA)

Yuepei Hu (Alcon)

Ky Nguyen (Alcon)

Garo Janir (Microchip)

Thank you!

Questions?