

# Machine Learning for Human Biometrics

---

sddec22-14: Nathanael Morris, Ritvik Maripally, Ron Mei Hang Teoh, Yee Shen Teoh, Zi-Jan Wong

# Problem Statement

- Surgeons tend to get tired if operate for long periods of time
- They also get stressed when something goes wrong in the surgery
- Our project aims to create a device to identify any abnormalities in the surgeon's eye movement, and provide appropriate advice to the surgeon to prevent potential human error



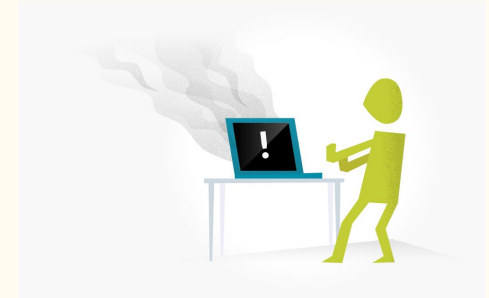
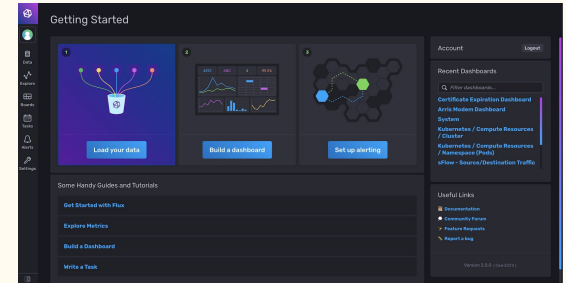
# Unit Testing

Machine Learning - Convolution Neural Network (CNN)  
eye detection testing.

InfluxDB - Test the boundary  
values of the code.

Ultra96 - Smoke test and stress test to check if  
HW is able to support the software.

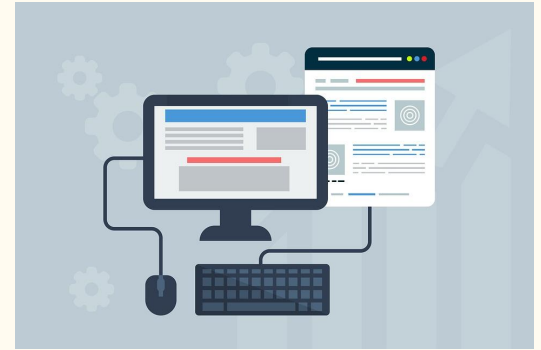
Database Storage - Perform a penetration test to figure out  
bugs and safety measures of system



# Interface Testing

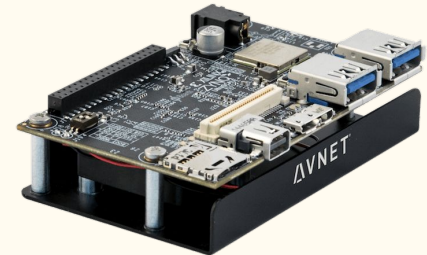
We have three interfaces

- Pupil Detection Interface
- Eye Movement Classifier Interface
  - We have data with the correct movement classifications and eye movement locations that we can use to test our project.
  - After we get the pupil locations from the pupil detection component, when we send it to the eye movement classifier component, we can ensure that the locations are accurate by comparing it to correct data that was manually gathered.
- Database Interface
  - We can use unit tests to ensure that the data being sent from the eye movement classification component to the database is being received in the correct format and order, and to make sure none of the data got corrupted.



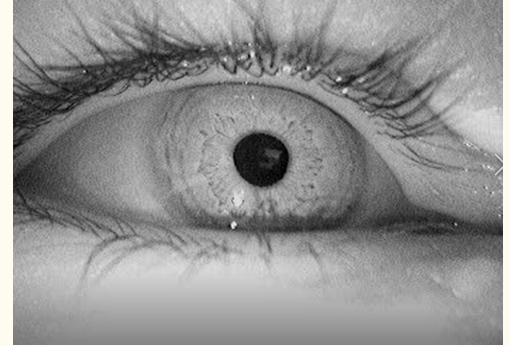
# Integration Testing

- Machine Learning Algorithm - no problems taking in the generated input, and generates a correct output format.
- Ultra96 Board - communicate with the camera to generate input and store it in the database.
- InfluxDB - receive input from the camera and is able to perform read/write operations.



# System Testing

- provide a video footage of eye movements manually
- confirm that the system acts appropriately
- all interface and integration tests will be applicable to the system test.



# Regression Testing

- Create a duplicate in case the it breaks while adding additional features.
- Run more test to ensure the new feature does not break anything.
- Need to ensure we do not break the CNN model, REMoDNaV classification algorithm, tables in InfluxDB, Ultra96, and security.



# Acceptance Testing

- Run a few tests using sample video provided by client.
- Record any observation that can be made (run time, output data, etc).
- Verify with client if the design requirements are satisfied.





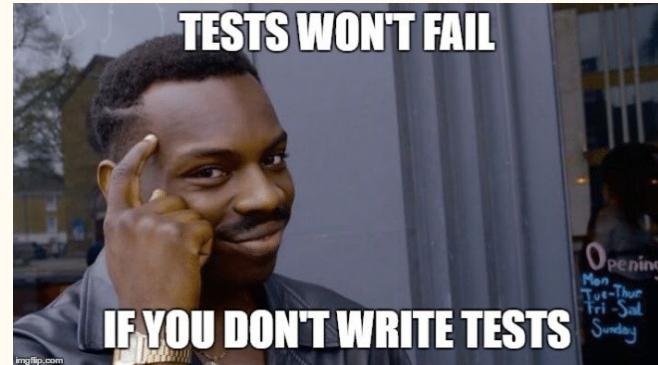
# Security Testing

- First do a risk assessment on what needs to be fixed and risk factors for our project
- Once everything has been assessed, fix the problems by prioritizing which is the most important
- After everything has been addressed, penetration test will occur which basically is meant to test against your own project to figure out other problems and if safety measures that were implemented works.
- Cycle through steps 2-3 till everything works the way it should



# Results

- Have finalized our component yet, so not much testing is done.
- We can observe and record data from each tests (data such as run time, process time, etc).
- Those data can verify if our implementation meets the requirements.
- Start building up more test once components are finalized.
- Will be more familiar with our design and understand what kinds of test are important.



Thank You