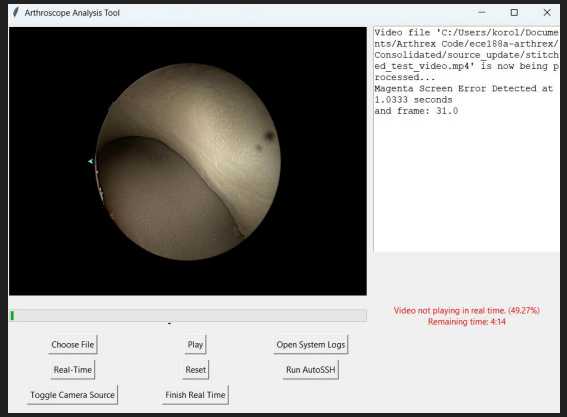
Arthrex: Arthroscopic Video Error Detection



AccuScope
Spotting errors before they matter.

05.29.2024

UCSB

design review + halfway point of our project

agenda (Will)

introduction of our goals (Jing)
- show type of video we are working with
- the types of errors we hope to detect with our scripts
- problem statement

scripts + gui
- fall quarter post processing (Will)
- winter quarter real time processing + gui work (Will)
- go into error detection methods
  - green screen + dropout + magenta lines (Zion)
  - frozen frame/lag (sidney)
  - shimmer
    - histogram analysis (Zion)
    - frame differencing (Jing)
  - Pano-to-70 (Tony)
  - Unknown error discovery (Tony)
- design overview
  - Gui tour (Sidney)
- SSH +logging (?)

results (Will)
- hundreds of videos captured (lots of pano-to-70)

future directions
- Gui improvements
- Processing improvements
- Tuning error scripts
- Interpreting error logs
- Analyzing data + exe packaging

Design review ultimately for
- identifying potential problems
- Getting feedback
- Validating design choices
- Alignment with requirements

Make sure to tie back to the purpose of the product

Showcase problem addressed (problem statement)
Our solution (scripts)
Key takeaways
Project’s significance (can tie back to problem statement)
Process used
Challenges faced

AGENDA

01 INTRODUCTION OF GOALS

02 SCRIPTS & GUI OVERVIEW

03 RESULTS

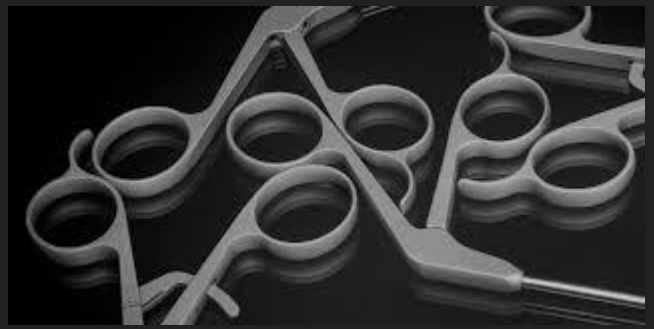
04 FUTURE DIRECTIONS

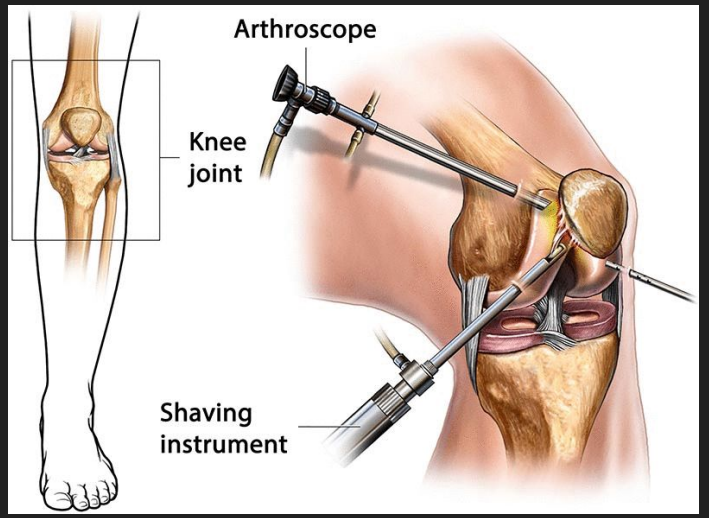
Introduction & Background

1

Arthrex & Arthroscopy

- Global leader in minimally invasive surgical technology, research, manufacturing, and medical education





https://www.healthdirect.gov.au/arthroscopy

Example Video Footage as Seen on Synergy Monitor

Alternate view with viewing angle label

Main view

Cavity is illuminated by LED source in controller and light guide

Circular video feed is captured by the camera head through the arthroscope lens assembly

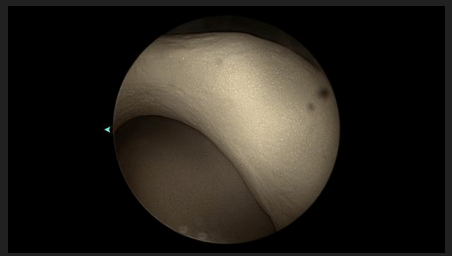
Problem Description (Glitches)

Freeze-Frame/Lag

Green Flash

Shimmer

Pano-to-70 Glitch



Frame Dropout

Magenta Lines

- Disclaimer: Frame Dropout and Magenta lines are generated errors (not from system)

Problem Statement

Government Guidelines	Current Arthrex Standards	Our Ideal Product
Surgical footage must not be interrupted for longer than 5 seconds at a time for safe operations	Surgical footage is not interrupted for longer than 5 seconds, but very rarely anomalies occur	Surgical footage experiences anomalies and interruption exceedingly rarely, and any anomalies can be easily diagnosed and addressed

Product Pipeline

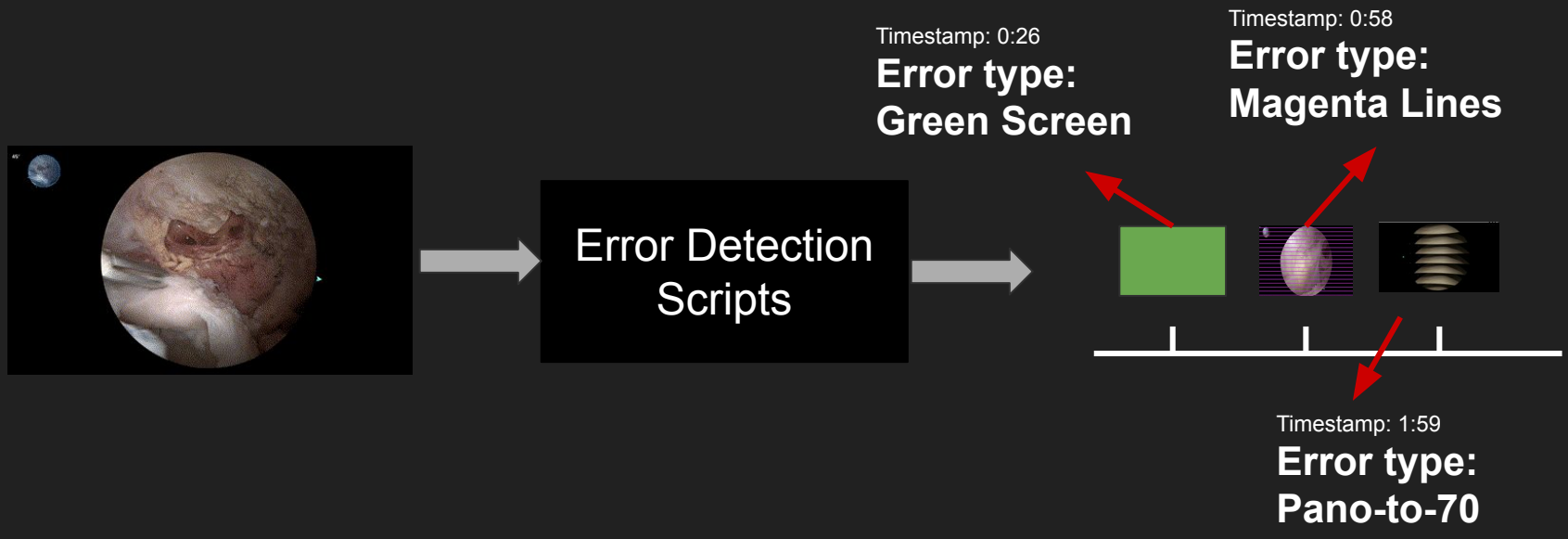
1. Detect when the error occurs.

2. Classify the error.

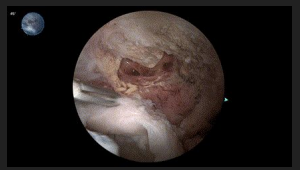
Error Detection Scripts

Output:
Timestamp: 0:26
Error type: Green Screen

3. Provide intuitive error summary.



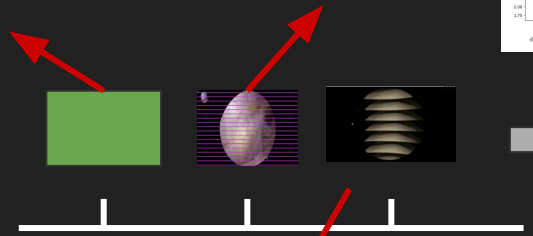
4. Provide System Information



Error Detection Scripts

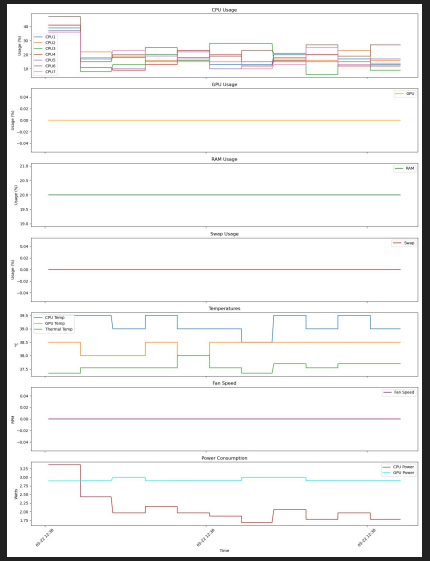
Timestamp: 0:26
Error type: Green Screen

Timestamp: 0:58
Error type: Magenta Lines



Timestamp: 1:59
Error type: Pano-to-70

System Logs




```
Magenta Screen Error Detected at 1.0333 seconds
and frame: 31.0
Magenta Screen Error Detected at 5.0333 seconds
and frame: 151.0
Magenta Screen Error Detected at 10.0333 seconds
and frame: 301.0
Magenta Screen Error Detected at 15.0333 seconds
and frame: 451.0
Pano-70 Error Detected at 15.2 seconds
and frame: 456.0
Magenta Screen Error Detected at 20.0333 seconds
and frame: 601.0
Dropout Error Detected at 27.9333 seconds
and frame: 838.0
Frozen Frame Detected at 51.1 seconds
```

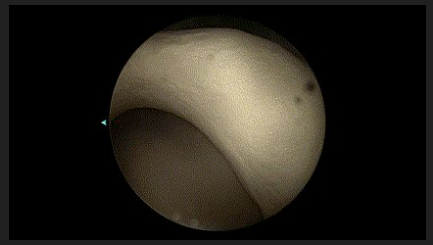
Scripts & GUI

2

```
current_dir = os.path.dirn
video_path = os.path.join(c


codeStart = time.time()
frozen_frame_flags = []

video = cv2.VideoCapture(vi
if not video.isOpened():
    print("Video could not
    sys.exit('Load Video Pr

fps = int(video.get(cv2.CAP
frame_width = int(video.get
frame_height = int(video.ge
frame_size = (frame_width,
output_size = (frame_width

# Create an output stream t
out = cv2.VideoWriter('resu

# Create Masks
_, initial_frame = video.re
lmask, smask = create_mask(
video.set(cv2.CAP_PROP_POS_

plt.imshow(lmask)

# Create variables for the
```

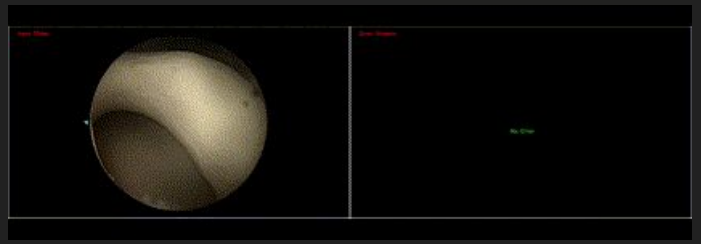
Post-Processing

Post-Processing Error Detection:



Saved Arthroscopic Video

Error Detection Scripts



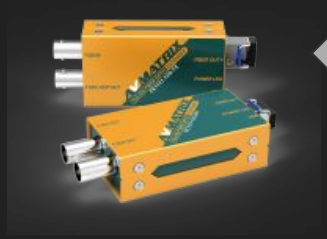
Generated Error Summary

- Time stamp of errors
- Classification of error
- Video playback of error

Real Time Processing

Real-Time Error Detection:

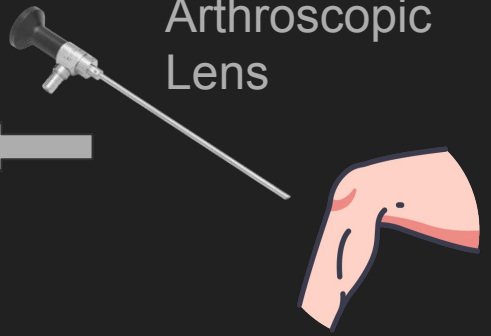
Frame Grabber

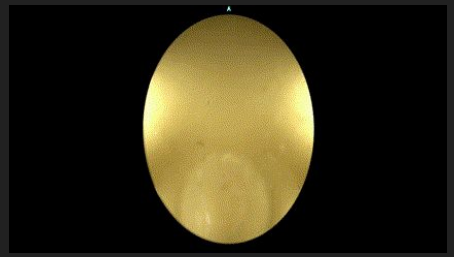


Controller

Camera Head

Arthroscopic Lens





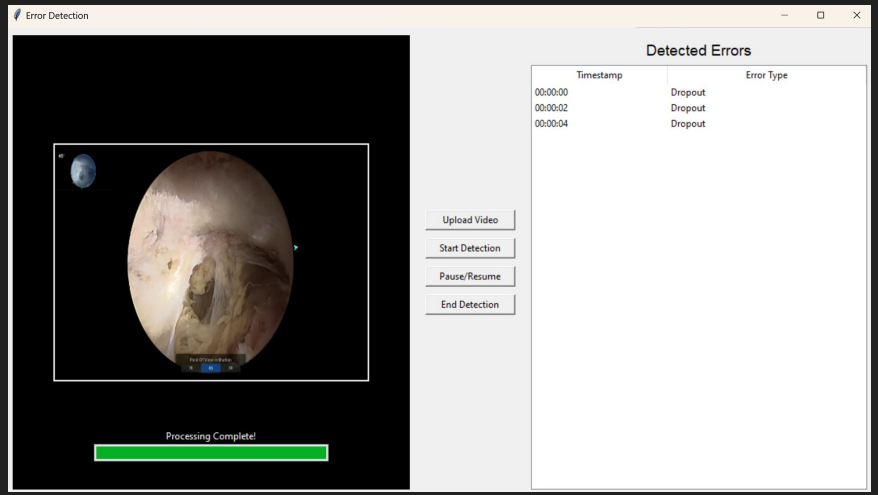
Real-Time Footage

Real-Time Error Detection Scripts

Generated Error Summary

Graphical User Interface:

Post-Processing



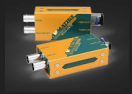
Real-Time

Design Topics

Part I: Error Detection Methods (specific techniques for each type of error)

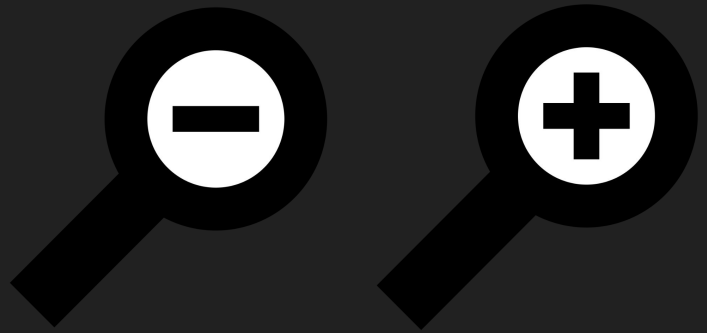



Part II: Design Considerations and Issues




Part III: GUI and Final Product

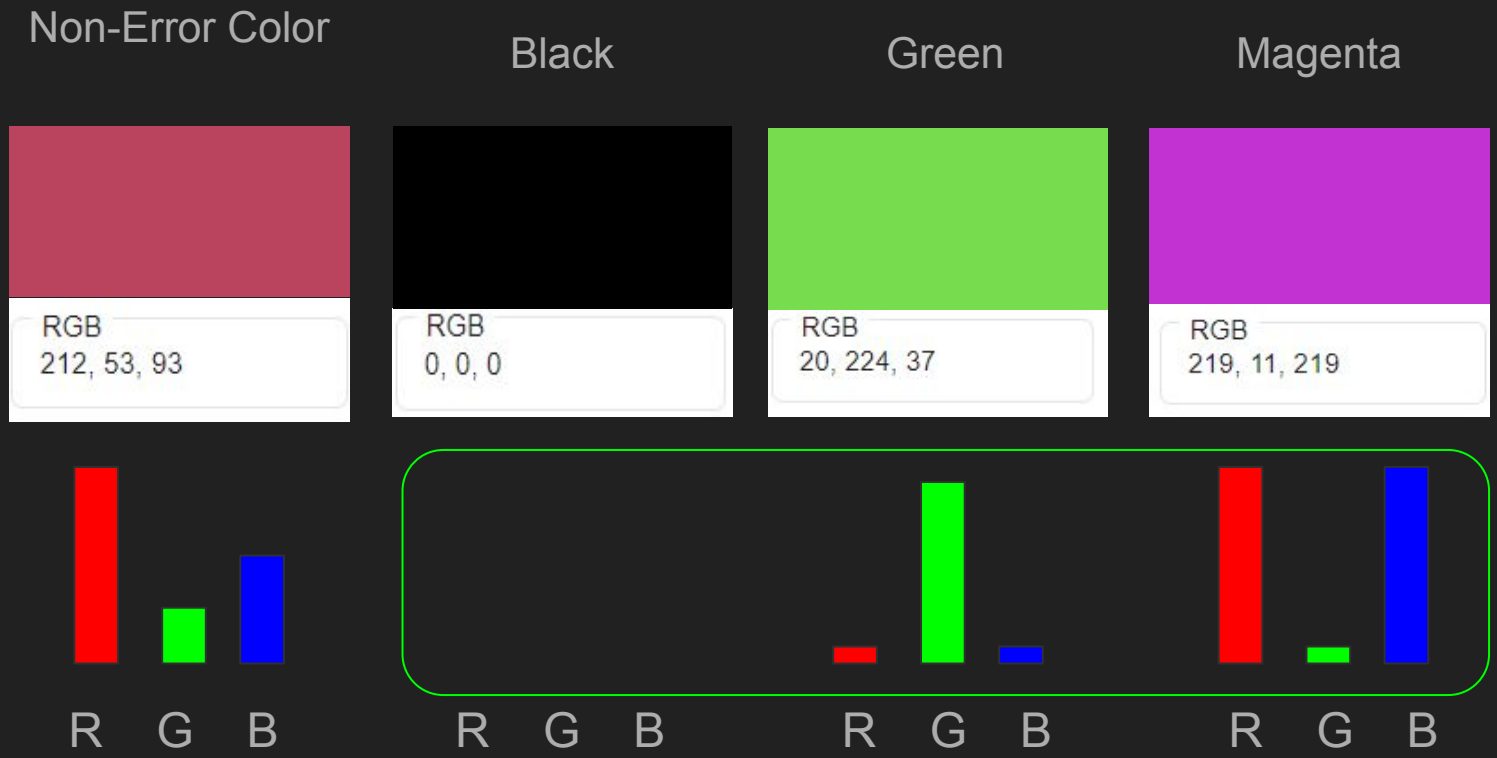
Part I: Error Detection Methods





Green-Screen, Dropout, Magenta-Lines

Using RGB to Set Thresholds



Conditional Thresholding

Green Flash

At Least 1 Pixel Satisfies:

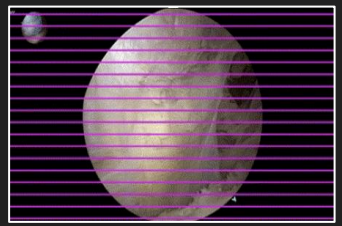
R	G	B
<50	>100	<50

Frame Dropout

All Pixels Satisfy:

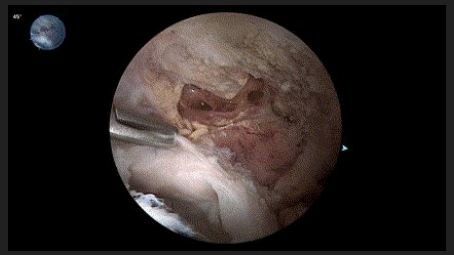
R	G	B
<20	<20	<20

Magenta Lines



At Least 1 Pixel Satisfies:

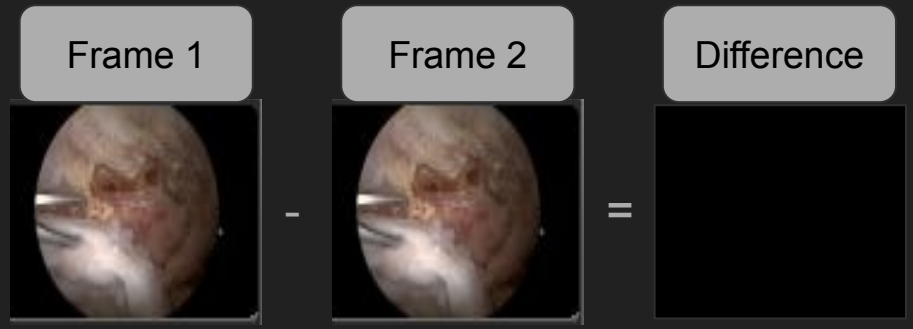
R	G	B
>100	<50	>100

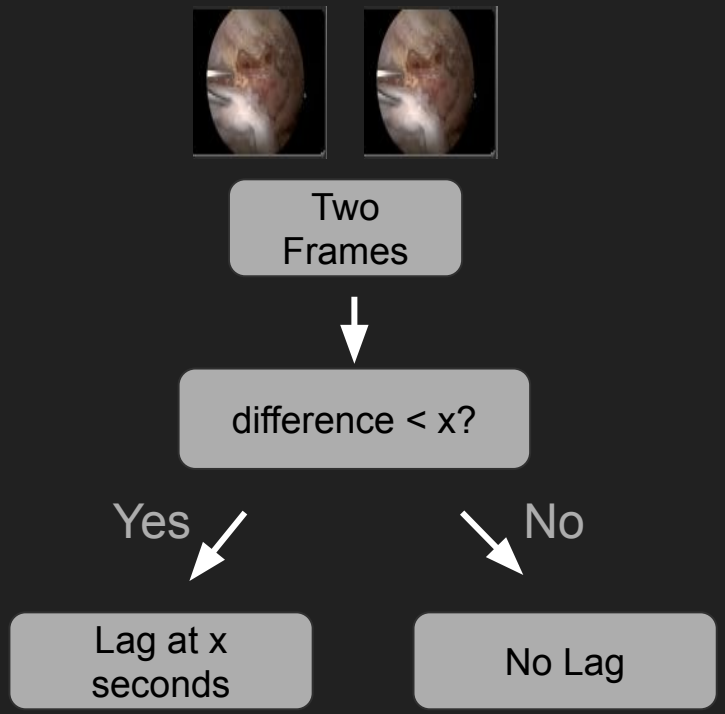


Frozen Frame/Lag

Lag Detection

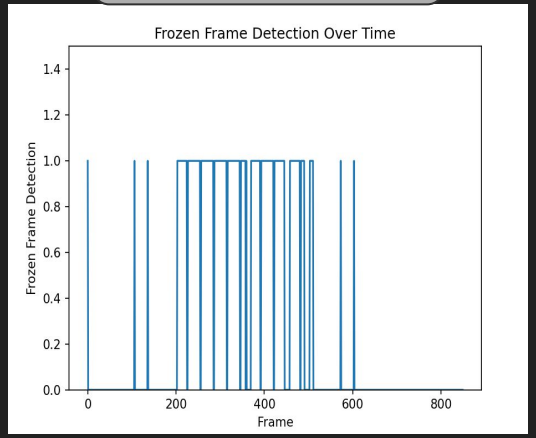
Frame Differencing:



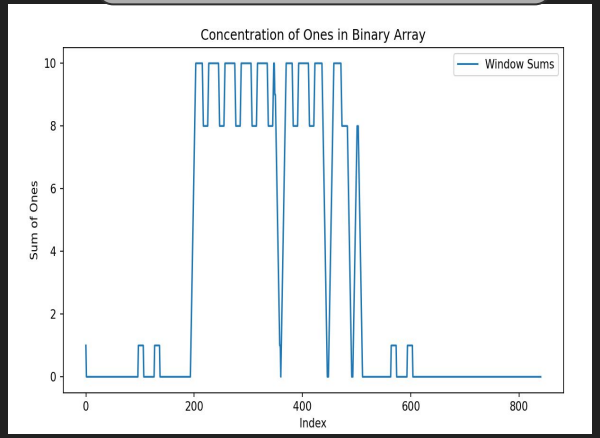


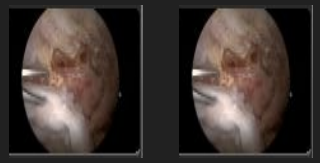
Lag Detection

All times with difference < x



Sliding Window Average (Lag Density)





Two Frames

difference < x?

Yes

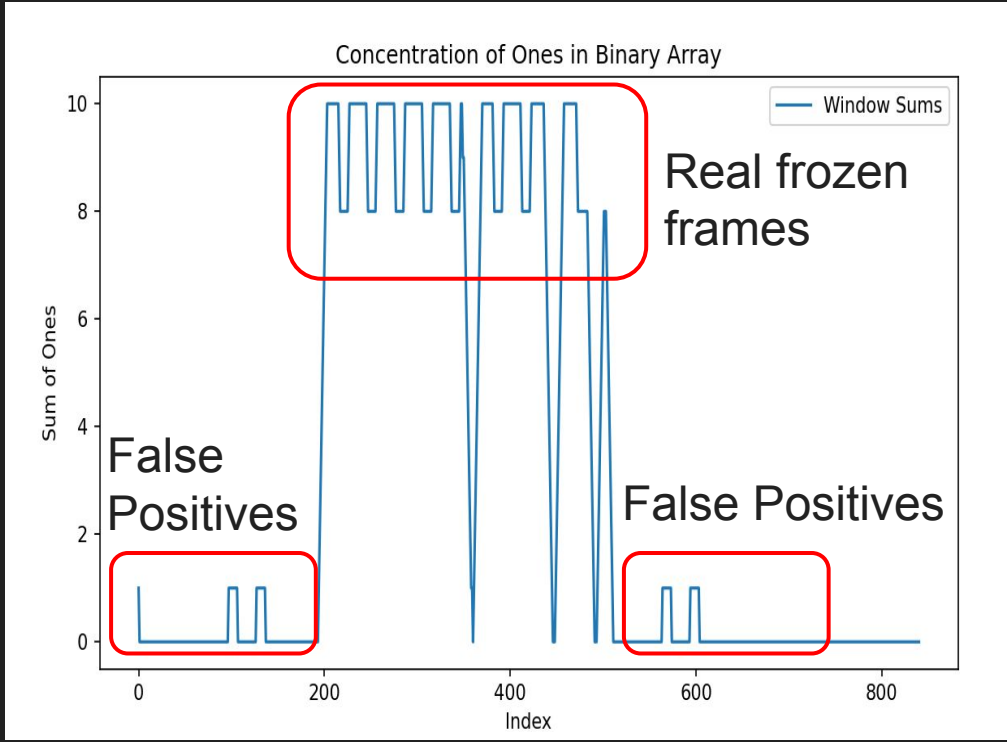
No

Lag at x seconds

False Positives!

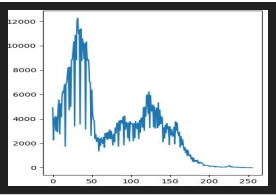
No Lag

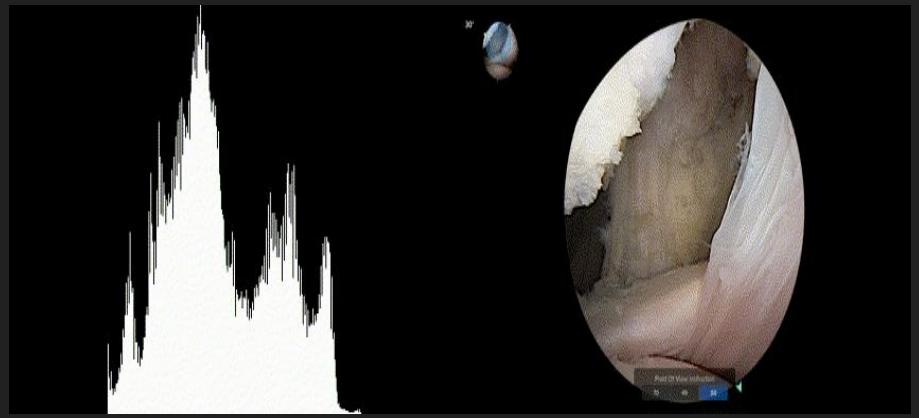
Lag Detection

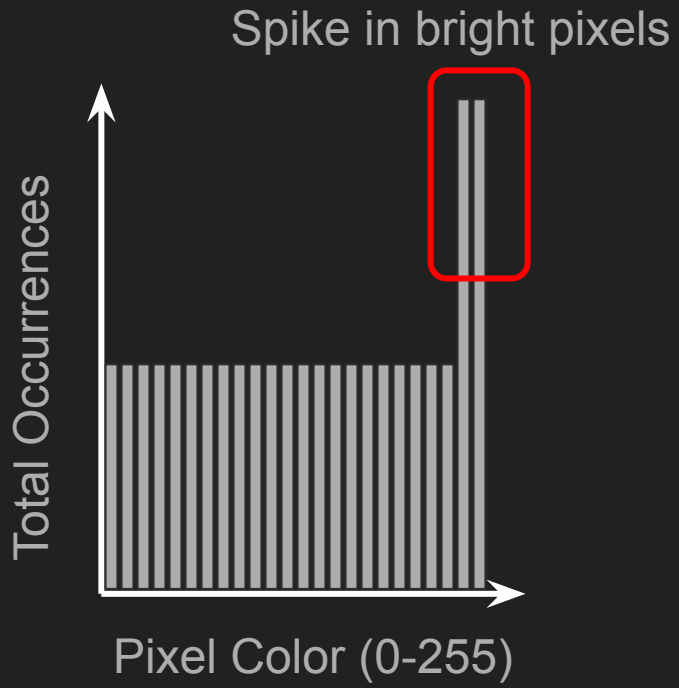


Shimmer

Histogram Statistics



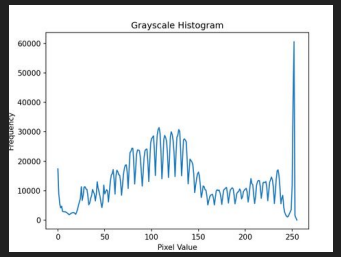




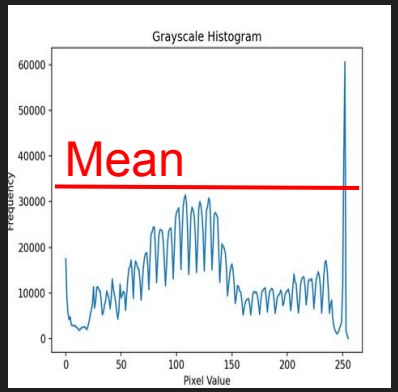
Shimmer Detection

Input Frames

Generate Histogram



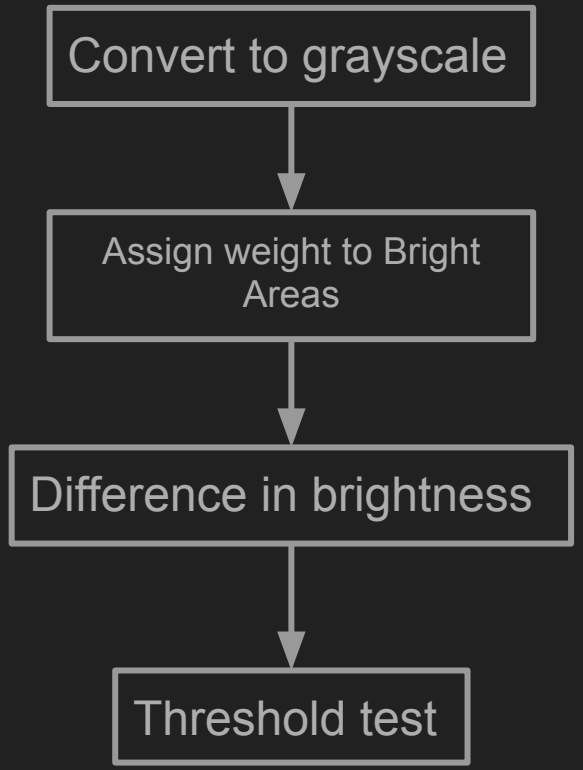
Compare Bright Pixels to Mean



- Calculate z-score for bright pixel values (see how far they are from the average count)
- If the amount of bright pixels is above a z-score of ‘3’, that is considered a **shimmer error**

Alternative to Histogram: Temporal Shimmer Detection

Shimmer Detection Method with Frame Differencing

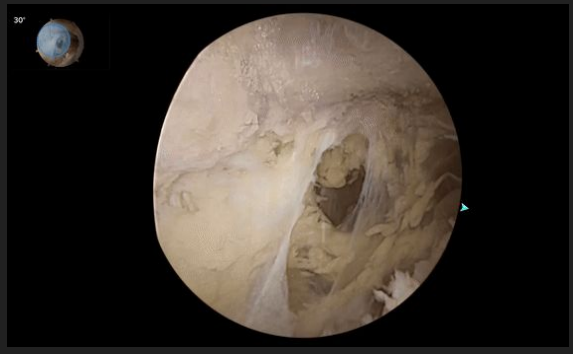


Instance of “shimmer”

Temporal Shimmer Detection Results

0.88s – 2.77s

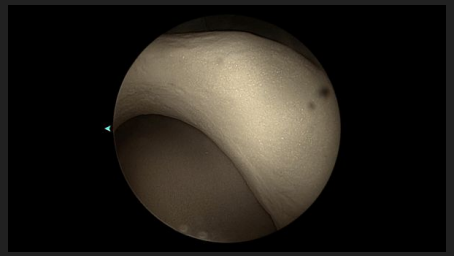
7.02s - 10.88s



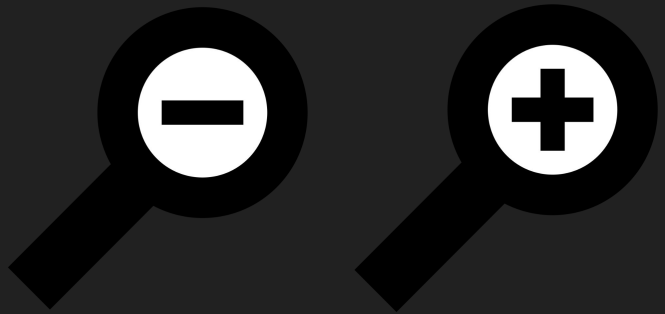
15.6s-17.45s

Eyeball shimmer timestamp: ~3s ~8s ~17s

Mostly accurate but still has unpredictable behavior because shimmer is hard to define



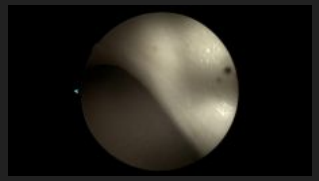
Pano-to-70 Detection

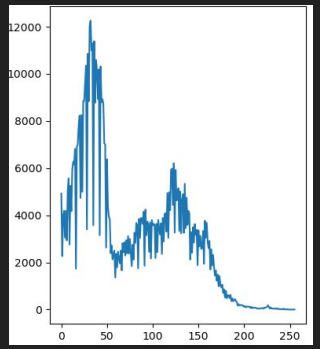


Pano to 70 Patch

- New software update fixed Pano to 70
- Regardless, we have made a lot of progress on our Pano to 70 Algorithm
- 1) Eliminating False Positives
- 2) Achieving perfect performance on hundreds of saved videos

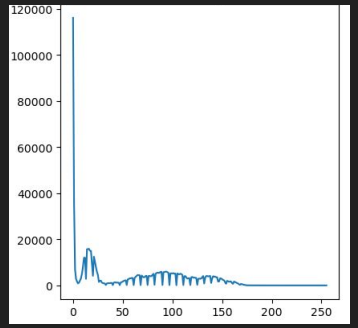
Histogram-Based Pano-to-70 Detection **(Obsolete)**





→ Balanced Histogram

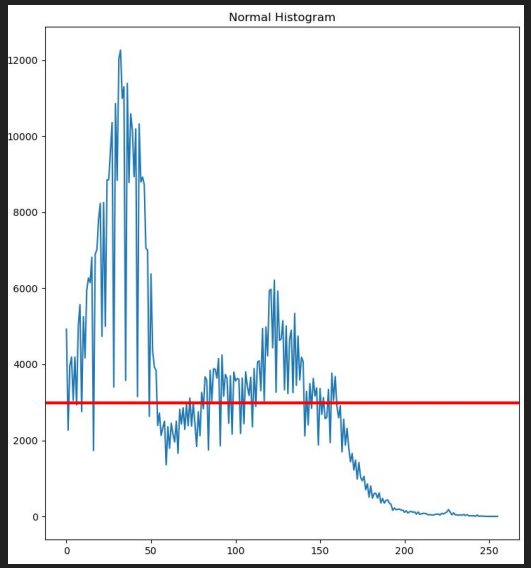
- Not too much variation in pixel count
- Wide intensity coverage

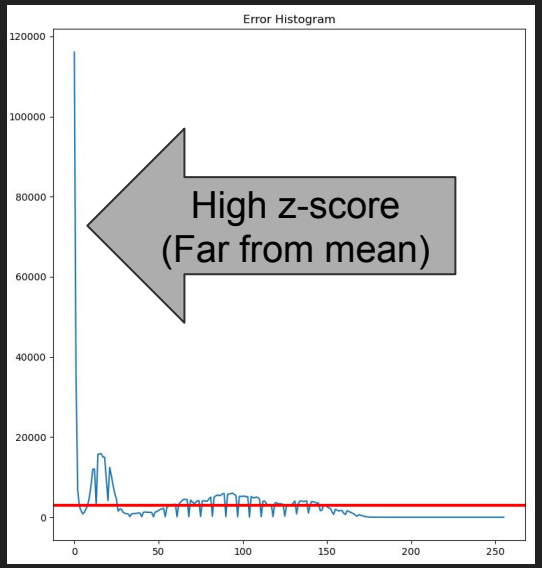


→ Un-Balanced Histogram

- Extreme spike in ‘0’ intensity count
- Loss of intensity coverage

Histogram-Based Pano-to-70 Detection **(Obsolete)**





Frame

Histogram with z-score > 5 in low intensity values?

Yes

No

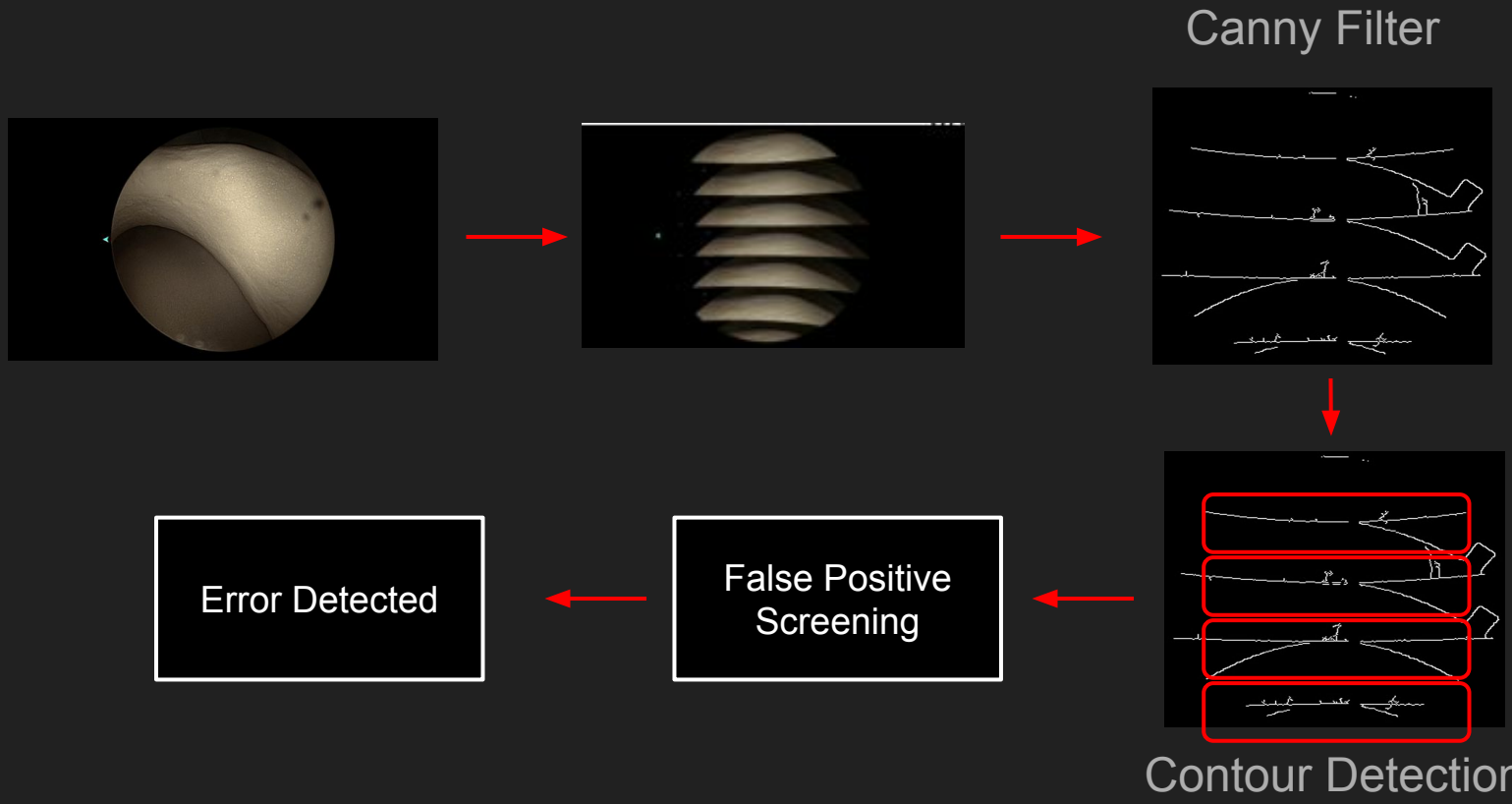
Pano to 70 at x seconds

No Pano to 70

Pano-to-70 Error Detection: Edge Detection **(Obsolete)**

Distinct edges!

Pano-to-70 Error Detection: Edge Detection **(Obsolete)**



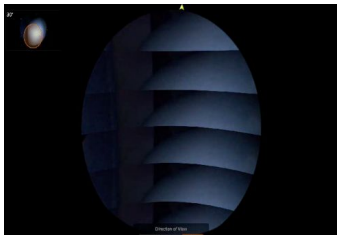
Edge Detection is not the Best Solution

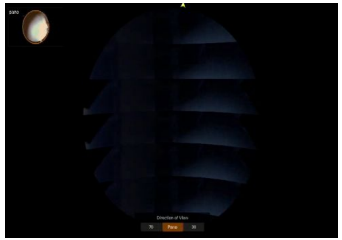
Not all Pano-to-70 Errors are created equally!

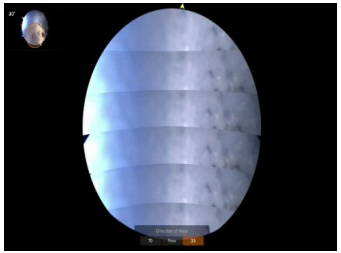
Some edges are very clear. Others aren't visible at all

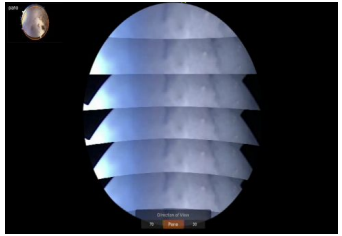
One thing in Common: All pano-to-70 frames have a vertically repeating pattern

* These frames are all taken from one video


37
Direction of View

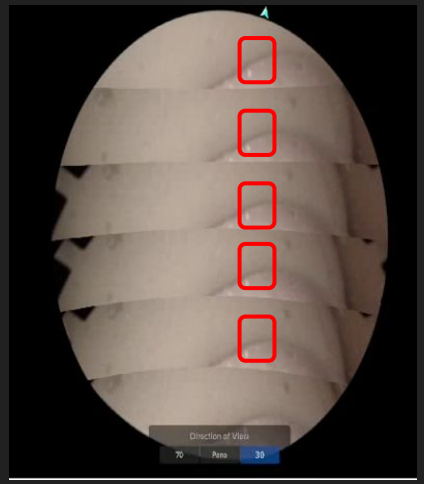

1970
Direction of View
70
Pure
30

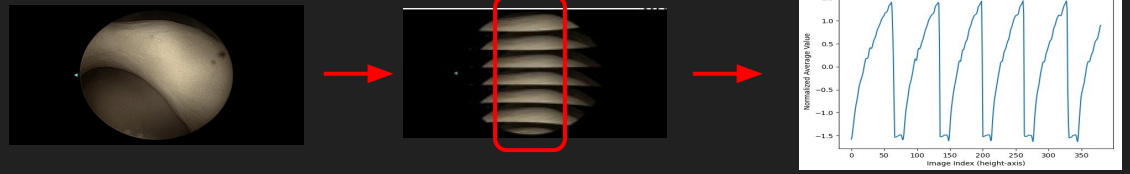

37
Direction of View
70
Pure
30


1970
Direction of View
70
Pure
30

Repeated Region Detection

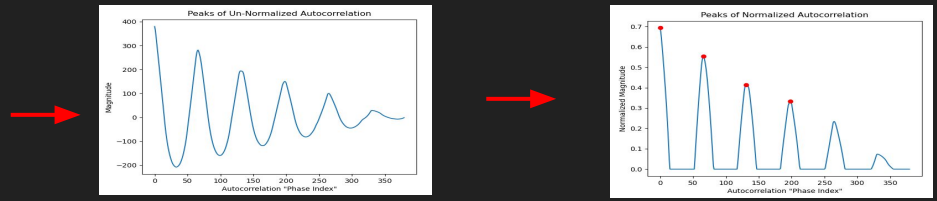
- **Pano to 70 errors all have repeating pixels:**
- By far the most accurate and effective method developed





Average over center columns (2D to 1D)

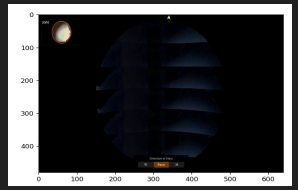
Perform Autocorrelation

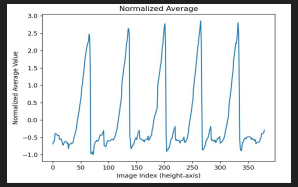


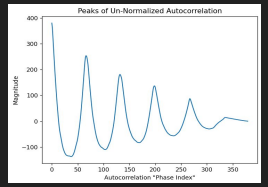
Normalize and count autocorrelation peaks

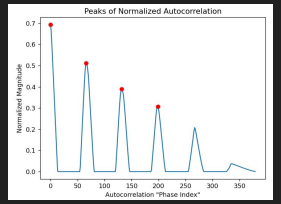
Count of peaks is our measure for repeated region detection

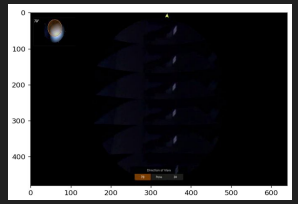
Repeated Region Examples

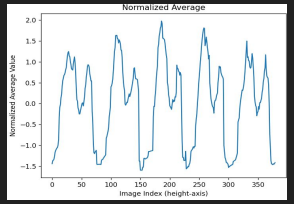


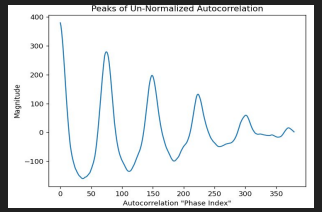


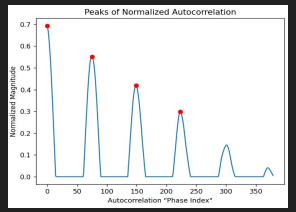




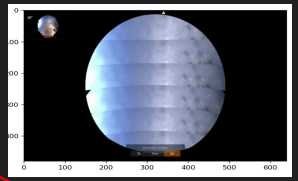


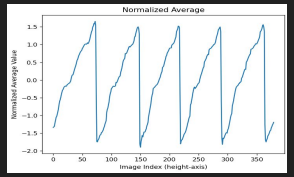


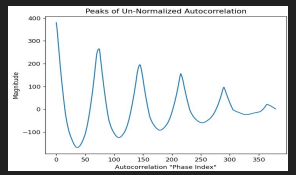


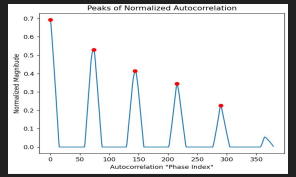


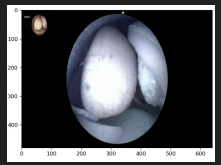
Pano-70 Errors

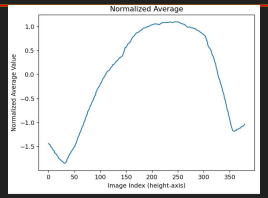


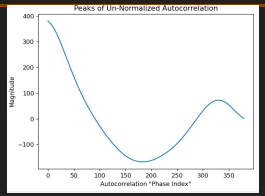


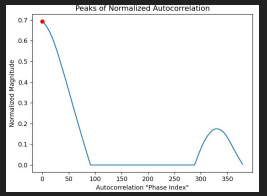










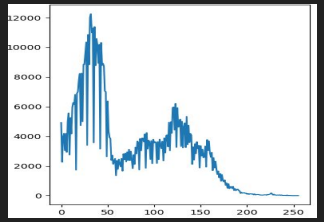


Random Frame

Unknown Error Discovery

New Error Discovery

- Weighted combination of Frame Differencing, and Histogram
- Flags parts of the video with rapid changes from frame to frame
- Sensitivity is adjustable to fit the surgical scene and expected magnitude of error



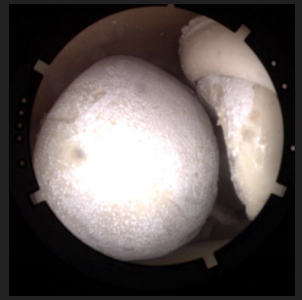
Methods for Fast Video Processing

- **Numba Pre-Compilation:**
  - Green Screen, Magenta, Dropout
  - Pre-compiles code into C++
  - Much faster than Numpy Vectorization in these cases

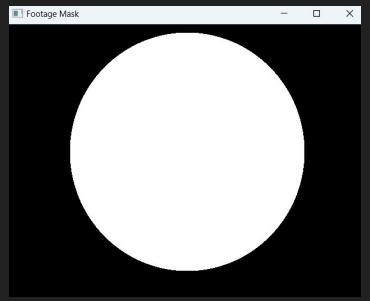
- **Numpy/Scipy Array Operations:**
  - Pano to 70
  - Efficient for 1D data
  - Faster than Numba for Autocorrelation

- **OpenCV Functions:**
  - Frozen Frame
  - Efficient built-in function for frame difference calculation

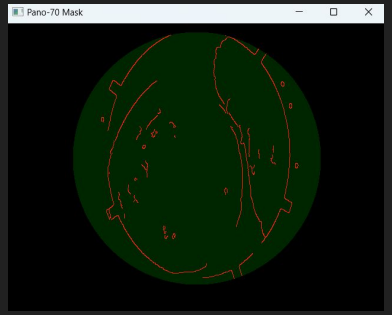
Masking for Unpredictable Real-Time Environment



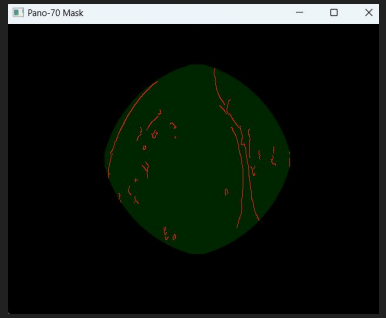
Original



Valid Region



Edges visible (Bad)



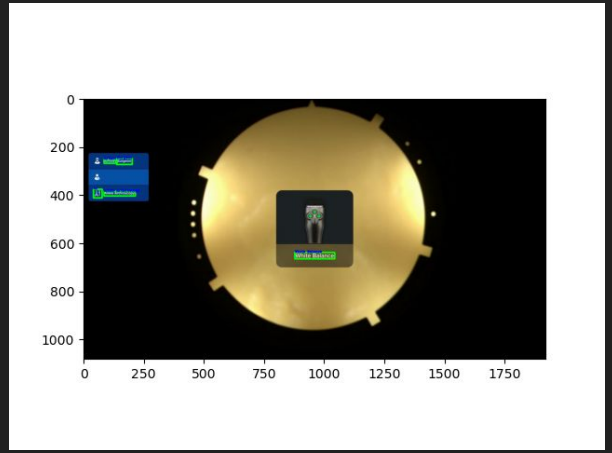
Edges Masked Out

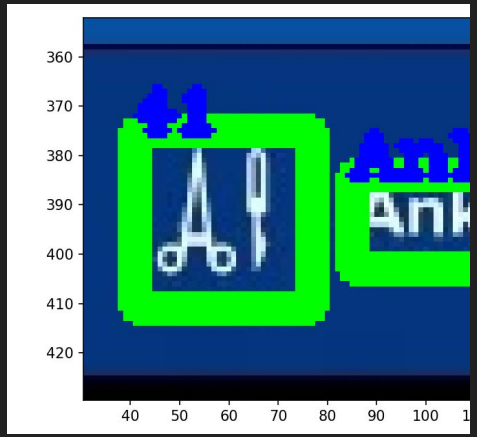
- Camera settings can influence error detection
  - Edges from surgical markers
  - Different zoom settings
  - Lighting

- Masking is only used in Dropout detection currently, but was previously useful for edge detection, and may be useful for future implementations so we kept it in the application for now

False Positives due to Menus

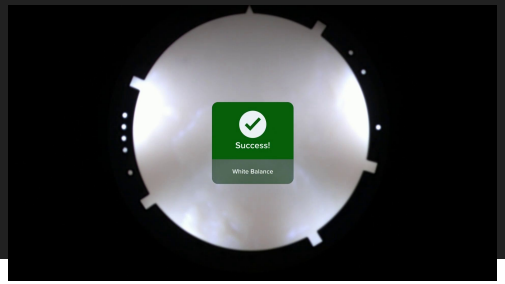
- Previously, we would get Pano-to-70 and green-screen errors from Menus
- Using Cross-Correlation with images of common menus, we were able to detect if a frame contains a menu
- Due to updates to our scripts, this is no longer strictly necessary



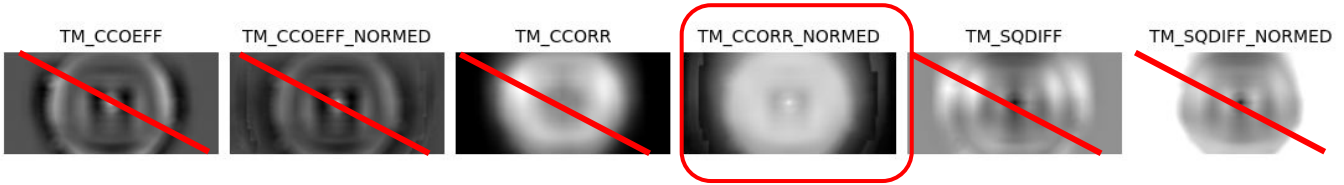


Kernel:

Frame:

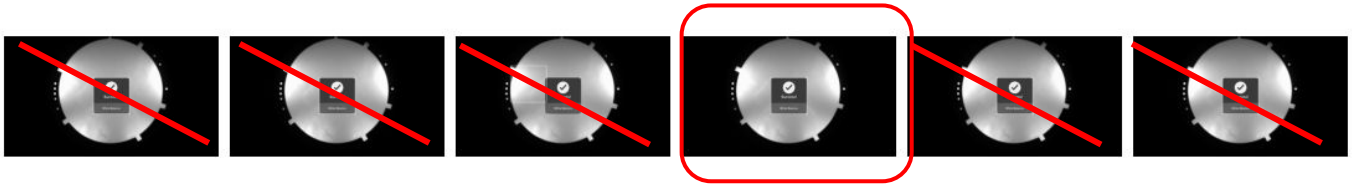


Matching Result



Normalized Cross Correlation is the best and simplest way to find menus in the frame

Detected Point



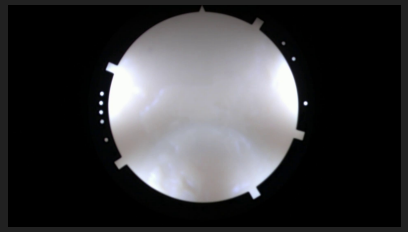
Menu Detection Confidence Intervals

Kernel:

Case 1:

Confidence: 0.999870777130127

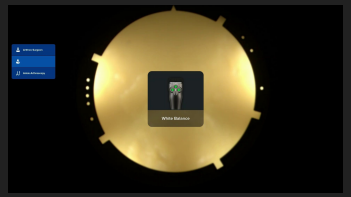
Case 2:



Confidence: 0.8562105298042297

Kernel:

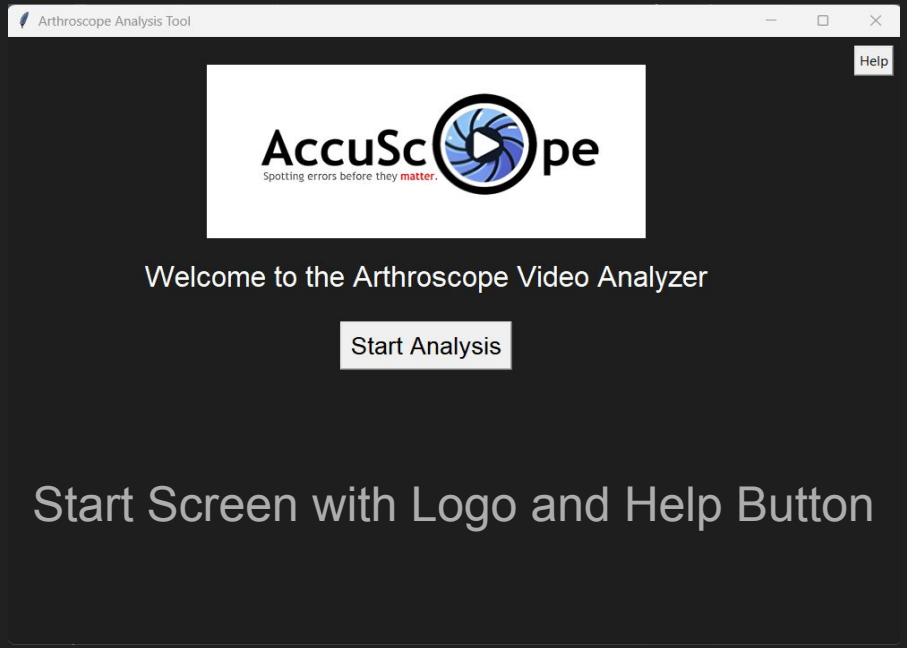
Case 3:



Confidence: 0.8597193360328674

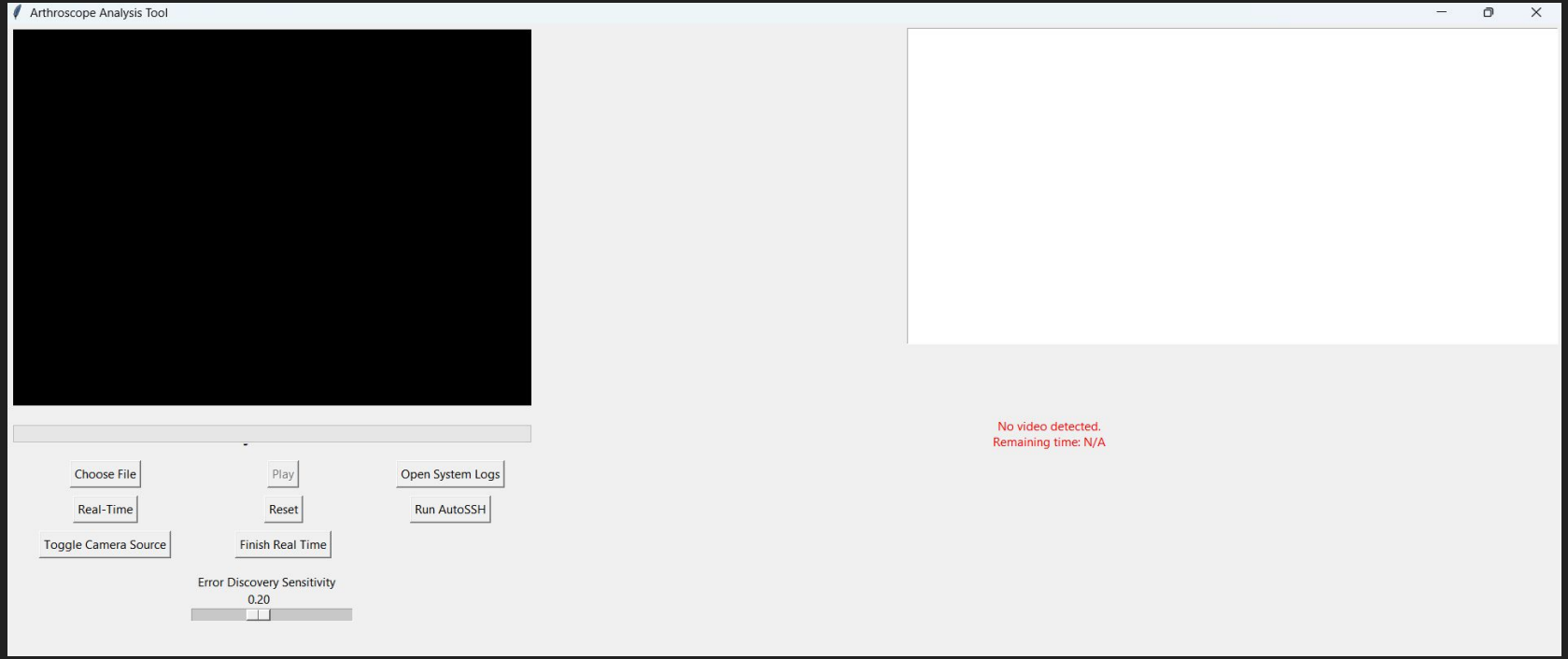
Part II: Design Overview and Roadmap

GUI Tour



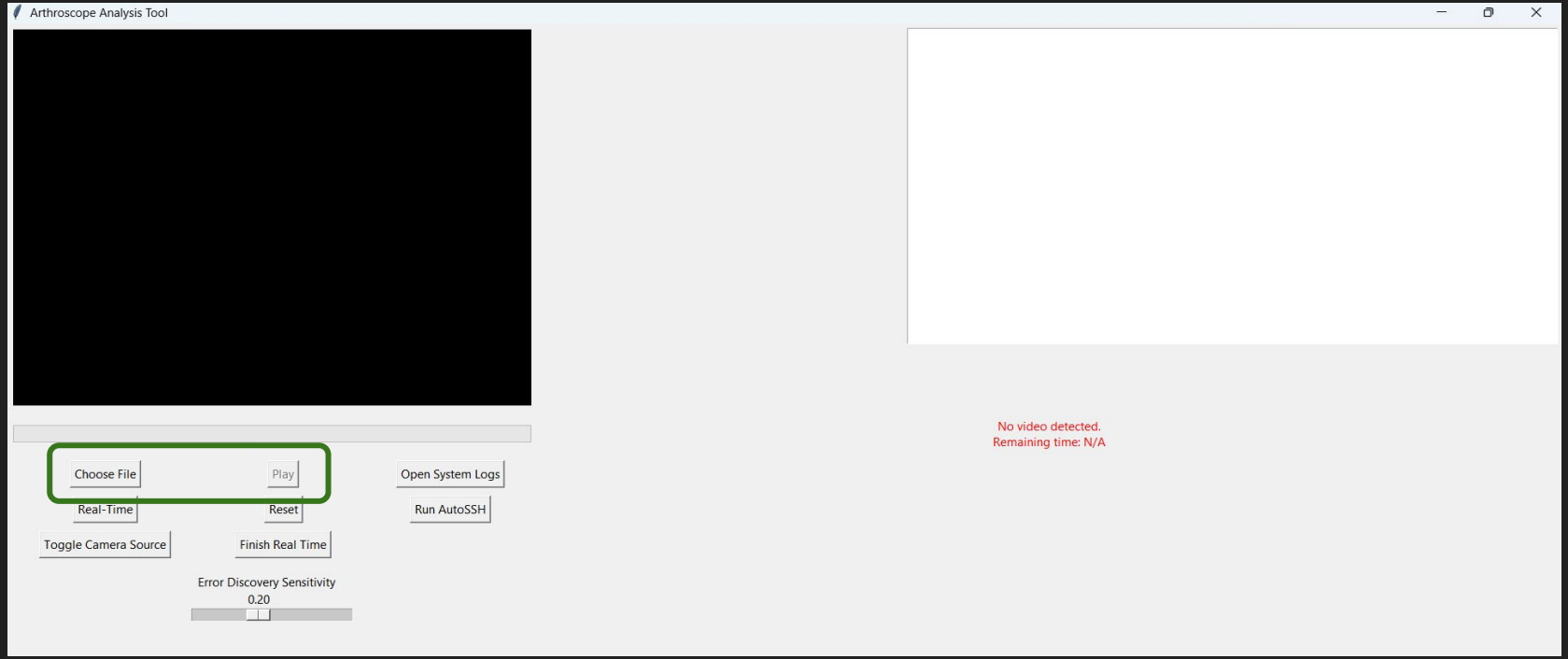
Start Screen with Logo and Help Button

GUI Tour



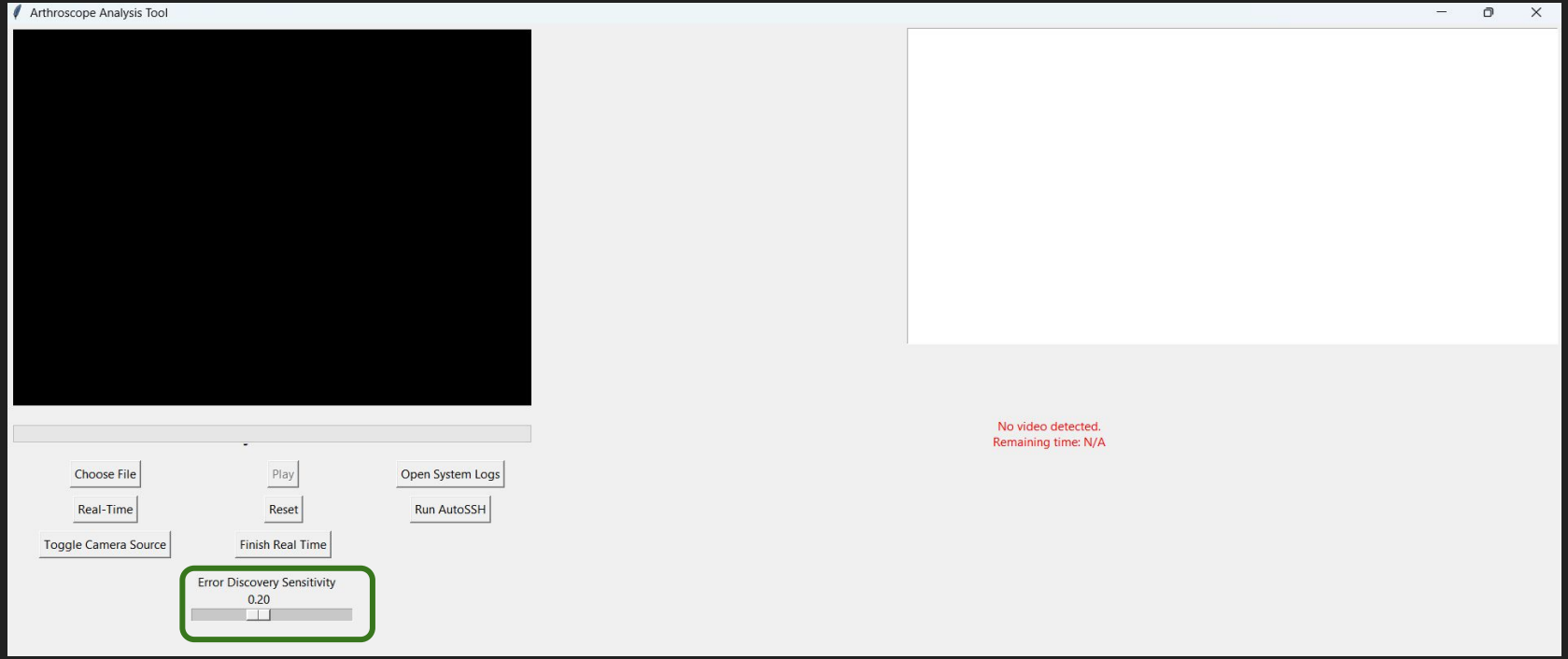
Main Screen with Video Analysis settings

Post Processing



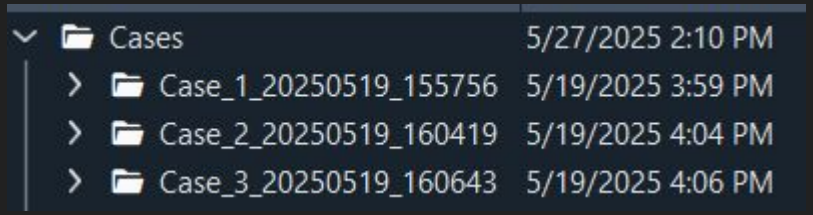
To load file from computer, click “Choose File”. To start processing, click “Play”. To reset and restart processing, click “Restart”

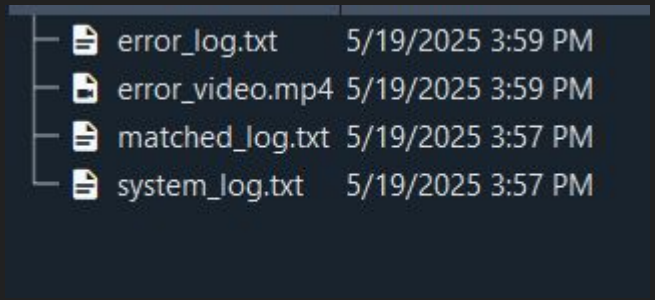
Post Processing: Error Discovery



If the saved video has a suspected ‘new’ error that does not have a dedicated detection script, set the error discovery sensitivity. Otherwise, keep it at 0.

Post Processing: Accessing results





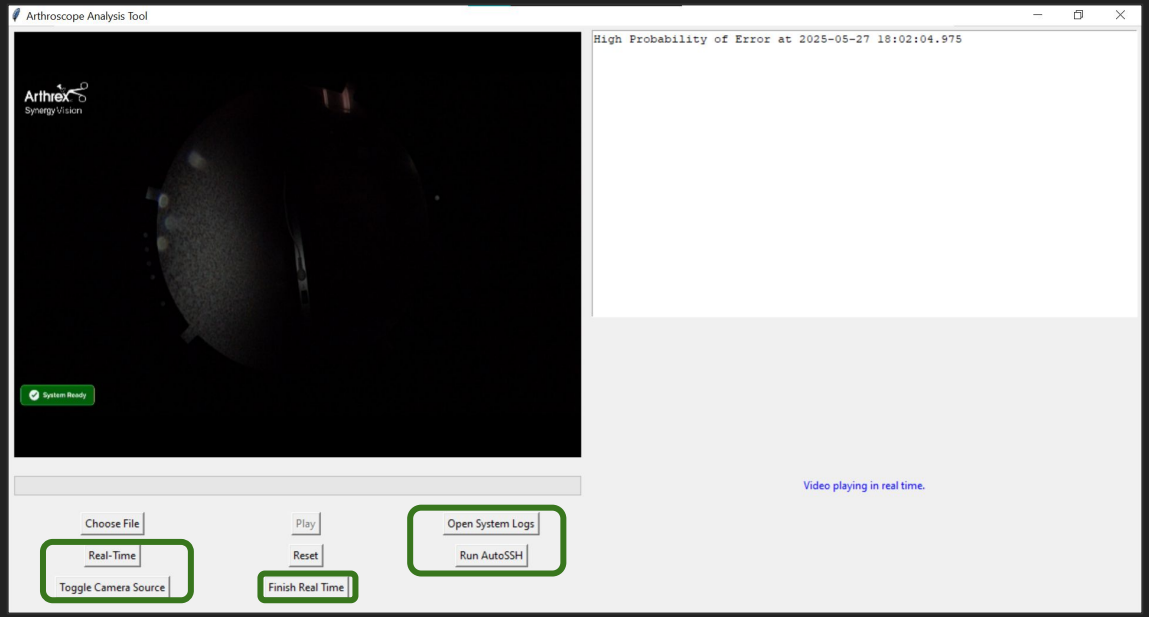
Post Processing: Accessing results

error_log.txt	5/19/2025 3:59 PM	
error_video.mp4	5/19/2025 3:59 PM	
matched_log.txt	5/19/2025 3:57 PM	
system_log.txt	5/19/2025 3:57 PM	

```
Video file 'C:/Users/[redacted]/Documents/Arthrex Code/ec
Pano-70 Error Detected at 10.2667 seconds
and frame: 616.0
Pano-70 Error Detected at 17.7333 seconds
and frame: 1064.0
Pano-70 Error Detected at 28.2167 seconds
and frame: 1693.0
Video file 'C:/Users/korol/Documents/Arthrex Code/ec
```

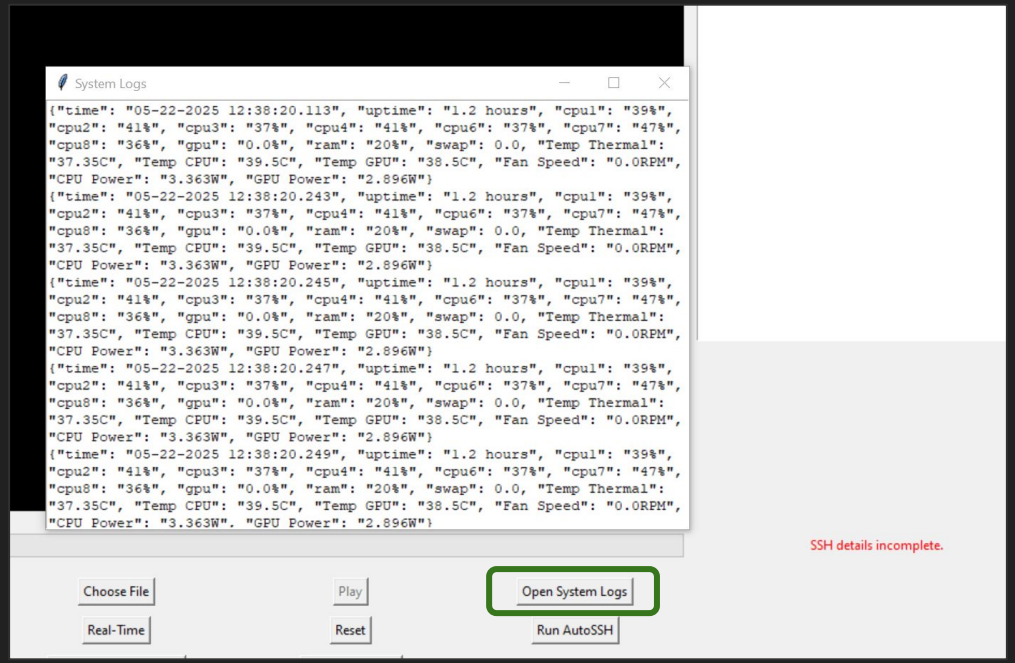
Empty for post-processing

Real-Time



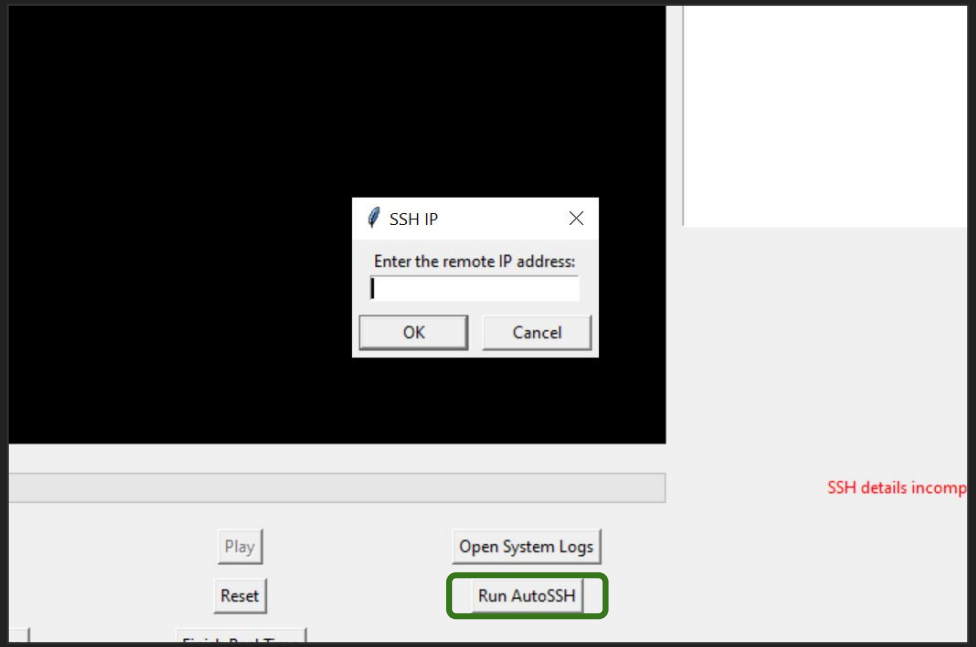
Real Time + Toggle Camera Source is used to get to the right live footage source, and End Real Time will clear the feed

Real-Time



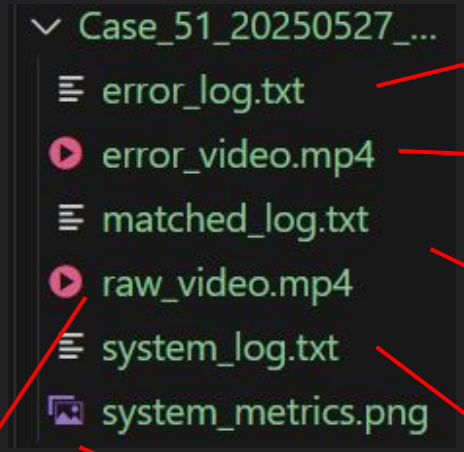
Latest system logs from when AutoSSH was ran

Real-Time



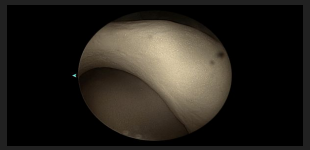
Run AutoSSH button will prompt the user for IP address, username, and password to access the system's information

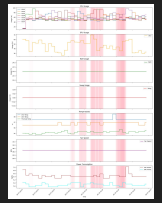
Real Time: Accessing results



Cases > Case_51_20250527_214919 > error_log.txt

```
1  High Probability of Error at 2025-05-27 21:49:55.382
2  High Probability of Error at 2025-05-27 21:49:55.529
3  Dropout Error at 2025-05-27 21:49:57.451
4  Dropout Error at 2025-05-27 21:49:57.659
5  High Probability of Error at 2025-05-27 21:49:57.710
6  High Probability of Error at 2025-05-27 21:49:57.778
7  Dropout Error at 2025-05-27 21:49:58.641
8  Dropout Error at 2025-05-27 21:49:58.829
9  High Probability of Error at 2025-05-27 21:49:58.881
10 High Probability of Error at 2025-05-27 21:49:58.947
11 High Probability of Error at 2025-05-27 21:49:59.812
```





Cases > Case_51_20250527_214919 > matched_log.txt

```
1  High Probability of Error at 2025-05-27 21:49:55.382
2  System Info: {'time': '05-27-2025 21:49:55.382', 'uptime': '250.38 hours', 'cpu1': '71%', 'cpu2': '56%',
3
4  High Probability of Error at 2025-05-27 21:49:55.529
5  System Info: {'time': '05-27-2025 21:49:55.529', 'uptime': '250.38 hours', 'cpu1': '71%', 'cpu2': '56%',
6
7  Dropout Error at 2025-05-27 21:49:57.451
8  System Info: {'time': '05-27-2025 21:49:57.452', 'uptime': '250.41 hours', 'cpu1': '55%', 'cpu2': '50%',
9
10 Dropout Error at 2025-05-27 21:49:57.659
11 System Info: {'time': '05-27-2025 21:49:57.659', 'uptime': '250.43 hours', 'cpu1': '60%', 'cpu2': '49%',
12
13 High Probability of Error at 2025-05-27 21:49:57.710
14 System Info: {'time': '05-27-2025 21:49:57.710', 'uptime': '250.43 hours', 'cpu1': '60%', 'cpu2': '49%',
```

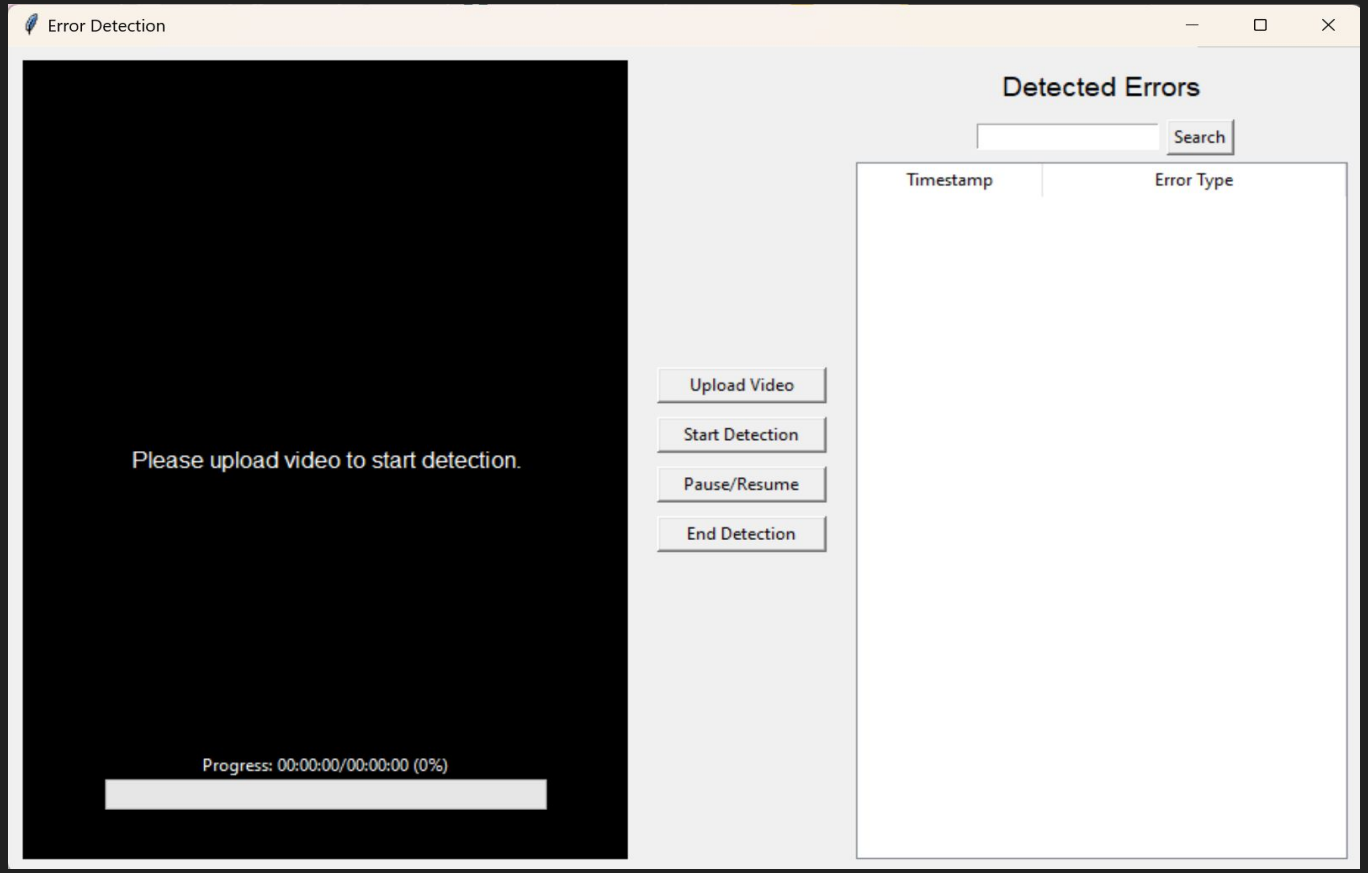
Cases > Case_51_20250527_214919 > system_log.txt

```
1  System Log for case 51
2
3  --- Remote Monitoring Output ---
4  {"time": "05-27-2025 21:49:49.579", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
5  {"time": "05-27-2025 21:49:49.720", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
6  {"time": "05-27-2025 21:49:49.724", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
7  {"time": "05-27-2025 21:49:49.727", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
8  {"time": "05-27-2025 21:49:49.729", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
9  {"time": "05-27-2025 21:49:49.731", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
10 {"time": "05-27-2025 21:49:49.734", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
11 {"time": "05-27-2025 21:49:49.737", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
12 {"time": "05-27-2025 21:49:49.739", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
13 {"time": "05-27-2025 21:49:49.741", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
14 {"time": "05-27-2025 21:49:49.743", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
```

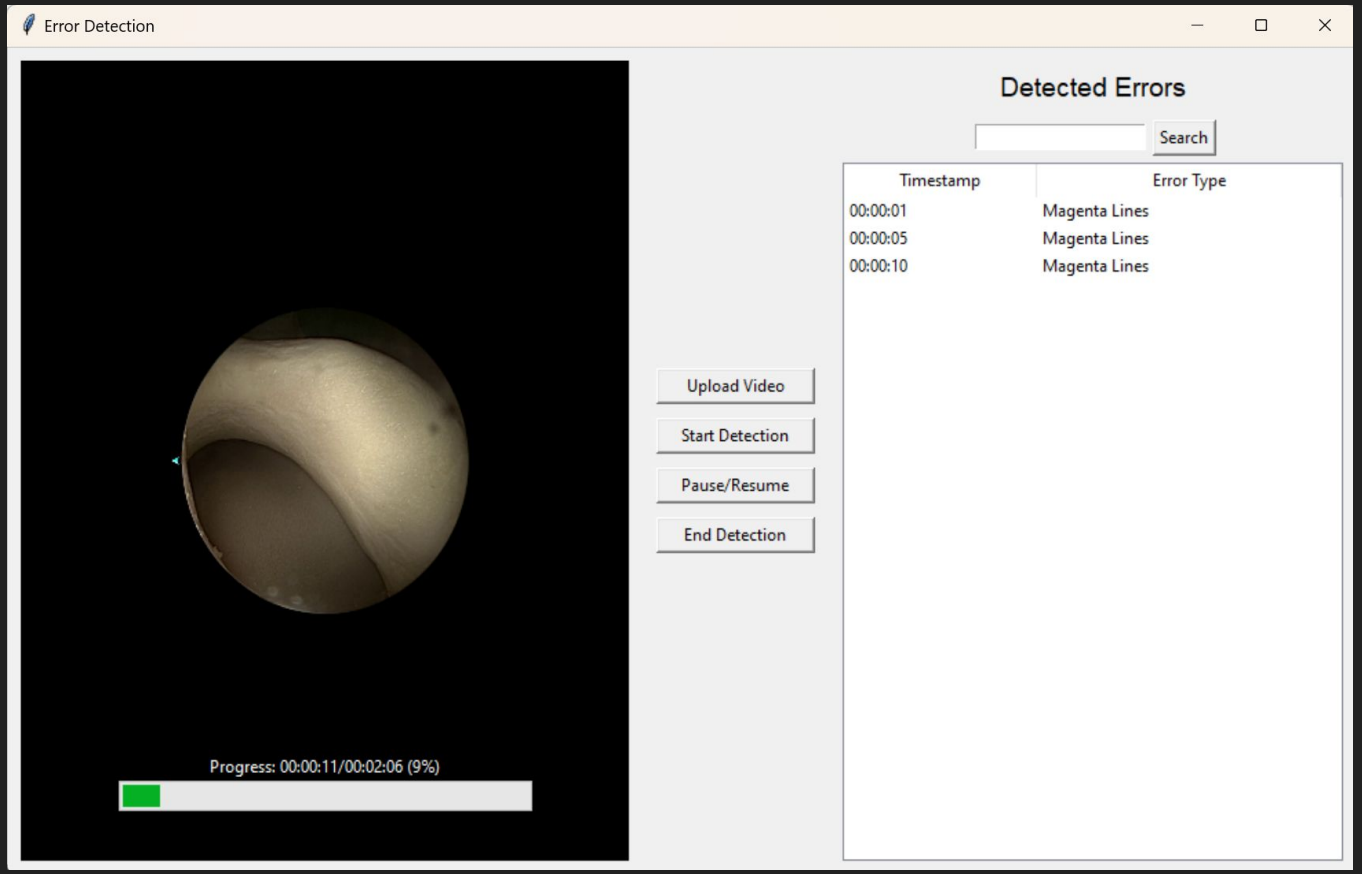
Part III: GUI Progress

Real Time GUI

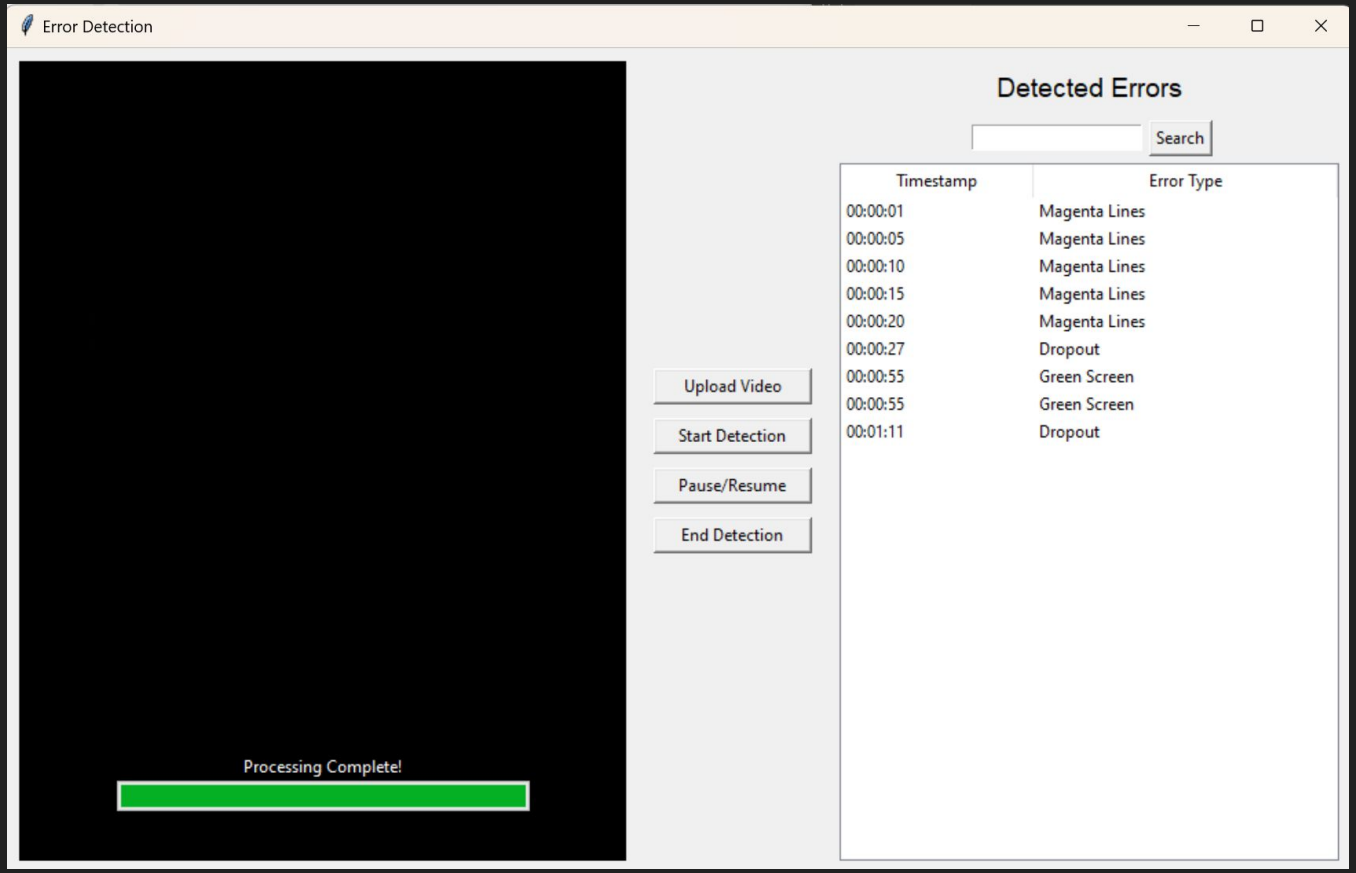
Post-Processing GUI - Appearance



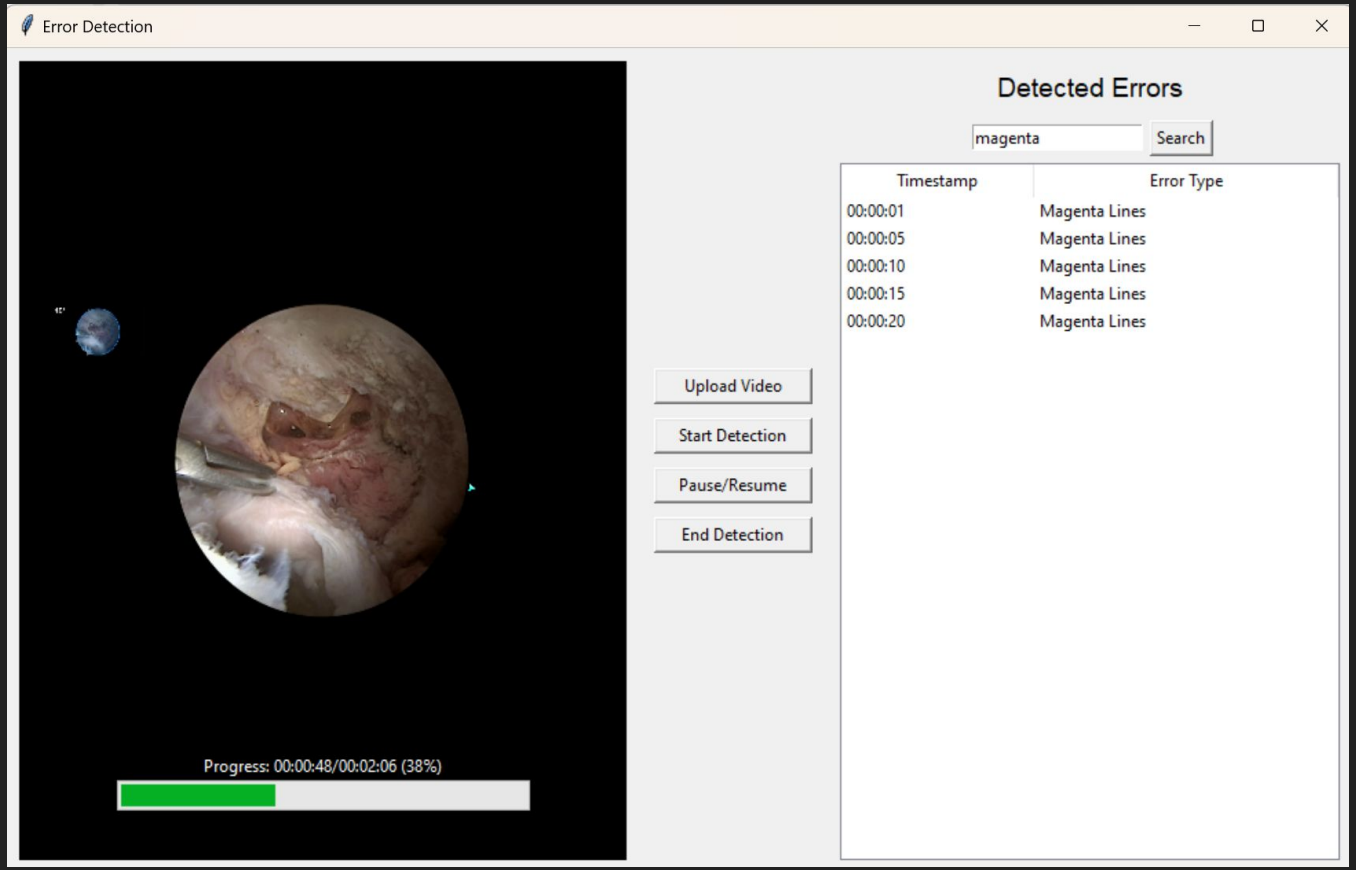
Post-Processing GUI - Processing



Post-Processing GUI - Final results



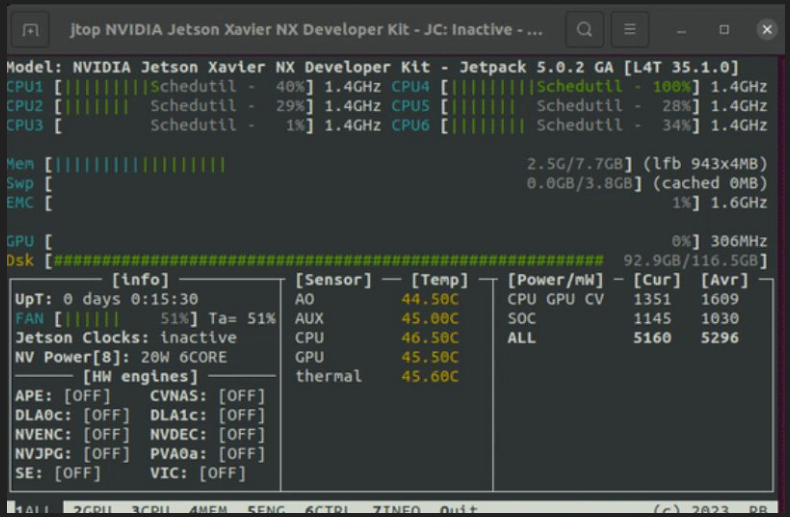
Post-Processing GUI - Search Function



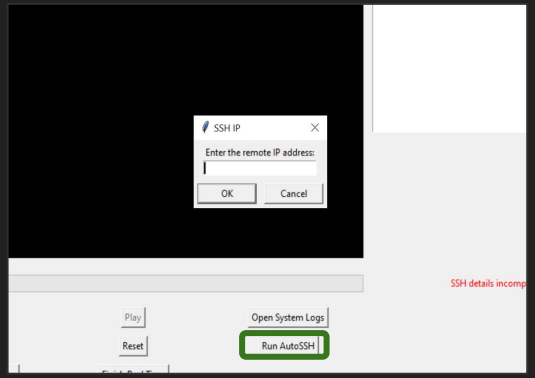
Part III: Auto-SSH Connection

Original Setup

- PowerShell
  - Access the device using linux commands
  - Access jtop
- WinSCP
  - Access the files in a GUI
- Wanted to find a more robust and user friendly solution
  - Python Fabric



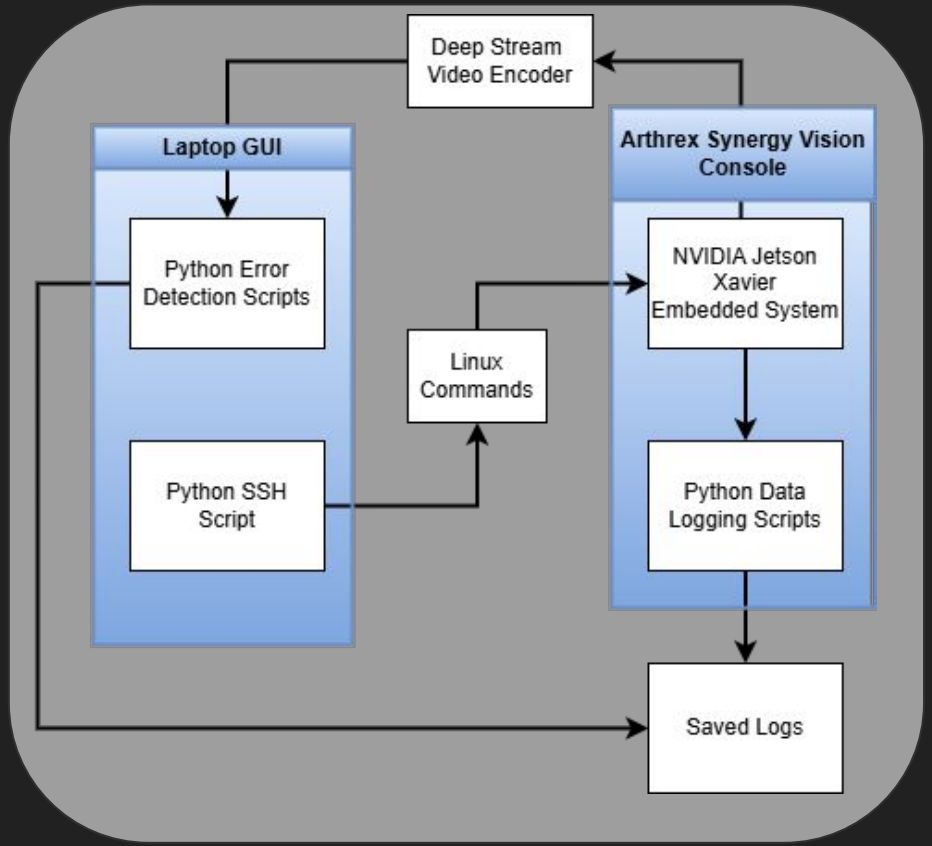
Automated SSH From GUI



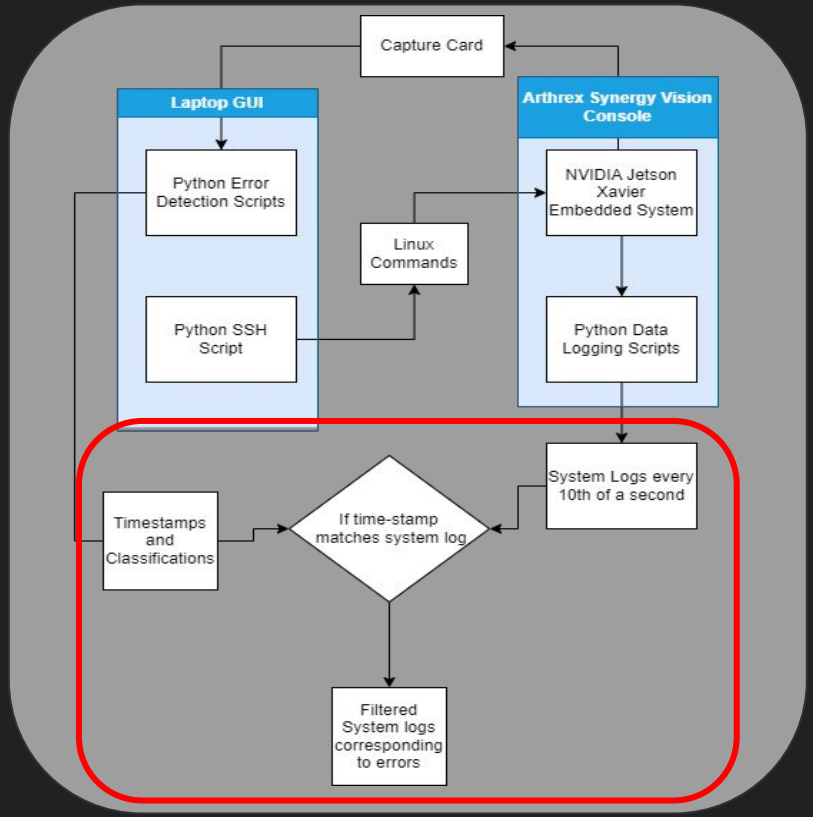
Fabric (Python)

Fabric: Python Library allowing us to sign into SSH from GUI
Argv used to prompt for user inputs.
Subprocess.Popen: Built-in method that allows us to run system logging in background while doing error detection

Block Diagram of Integrated System



SSH Logging in More Detail



Real-Time Logging Results

Full System Log

Cases > Case_51_20250527_214919 > system_log.txt

```
System Log for case 51

--- Remote Monitoring Output ---
{"time": "05-27-2025 21:49:49.579", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.720", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.724", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.727", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.729", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.731", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.734", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.737", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.739", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.741", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
{"time": "05-27-2025 21:49:49.743", "uptime": "250.29 hours", "cpu1": "26%", "cpu2": "25%", "cpu3": "25%",
```

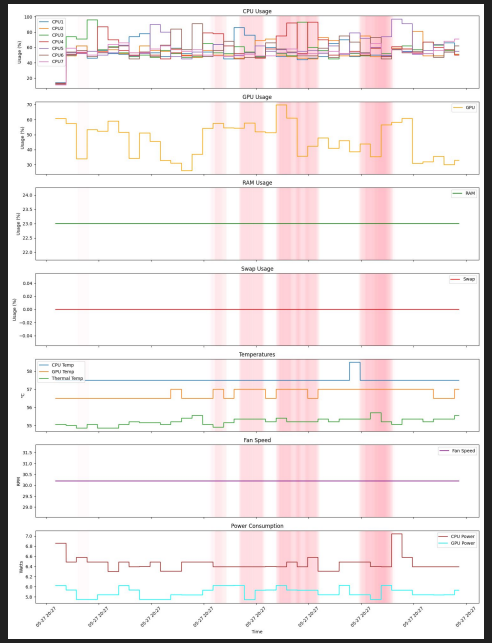
+

Cases > Case_51_20250527_214919 > error_log.txt

```
High Probability of Error at 2025-05-27 21:49:55.382
High Probability of Error at 2025-05-27 21:49:55.529
Dropout Error at 2025-05-27 21:49:57.451
Dropout Error at 2025-05-27 21:49:57.659
High Probability of Error at 2025-05-27 21:49:57.710
High Probability of Error at 2025-05-27 21:49:57.778
Dropout Error at 2025-05-27 21:49:58.641
Dropout Error at 2025-05-27 21:49:58.829
High Probability of Error at 2025-05-27 21:49:58.881
High Probability of Error at 2025-05-27 21:49:58.947
High Probability of Error at 2025-05-27 21:49:59.812
```

Script Error Log

Filtered Log, system information graph, and Error Density Visualization



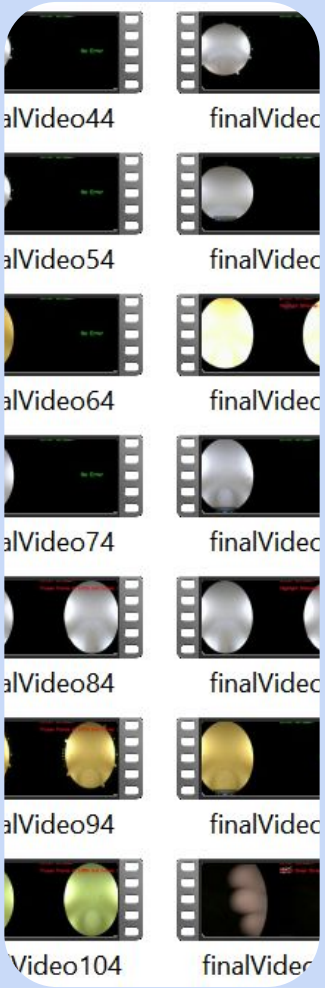
Data collected: CPU, GPU, RAM, Board Temps, Time

Cases > Case_51_20250527_214919 > matched_log.txt

```
High Probability of Error at 2025-05-27 21:49:55.382
System Info: {'time': '05-27-2025 21:49:55.382', 'uptime': '250.38 hours', 'cpu1': '71%', 'cpu2': '56%',

High Probability of Error at 2025-05-27 21:49:55.529
System Info: {'time': '05-27-2025 21:49:55.529', 'uptime': '250.38 hours', 'cpu1': '71%', 'cpu2': '56%',

Dropout Error at 2025-05-27 21:49:57.451
System Info: {'time': '05-27-2025 21:49:57.452', 'uptime': '250.41 hours', 'cpu1': '55%', 'cpu2': '50%',

Dropout Error at 2025-05-27 21:49:57.659
System Info: {'time': '05-27-2025 21:49:57.659', 'uptime': '250.43 hours', 'cpu1': '60%', 'cpu2': '49%',

High Probability of Error at 2025-05-27 21:49:57.710
System Info: {'time': '05-27-2025 21:49:57.710', 'uptime': '250.43 hours', 'cpu1': '60%', 'cpu2': '49%',
```

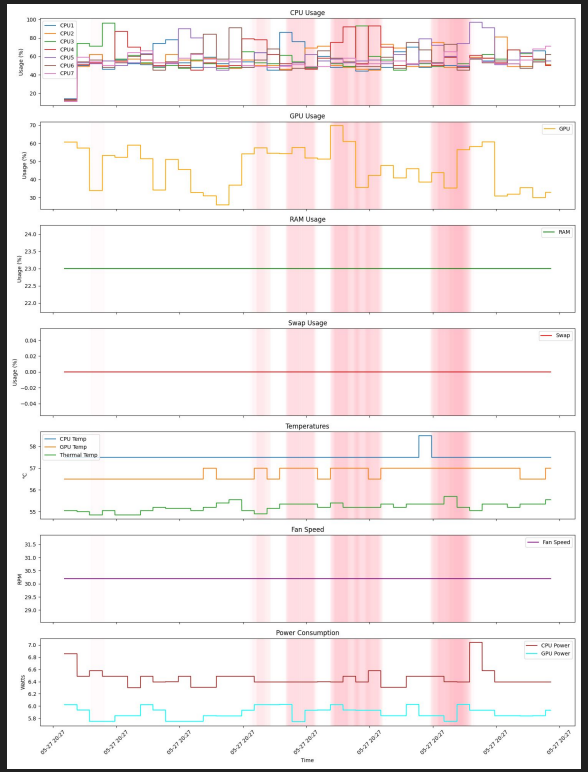
Results

3


alVideo44
finalVidec
alVideo54
finalVidec
alVideo64
finalVidec
alVideo74
finalVidec
alVideo84
finalVidec
alVideo94
finalVidec
Video104
finalVidec

Hundreds of Videos Captured

Real-Time Error Detection, Classification, and System Info



Real-Time Video Error Detection

Real-Time Video Error Detection

Real-Time Video Error Detection

Future Improvements

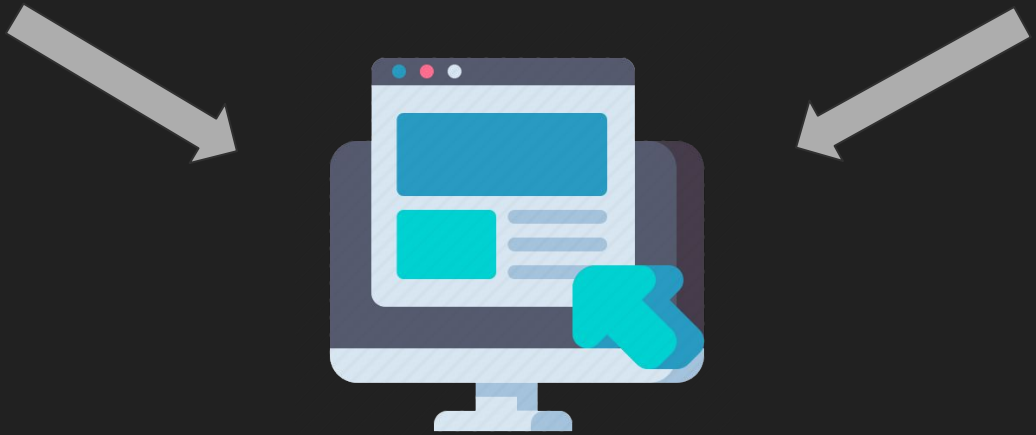
4

1) Proof and Verification of Detection Scripts

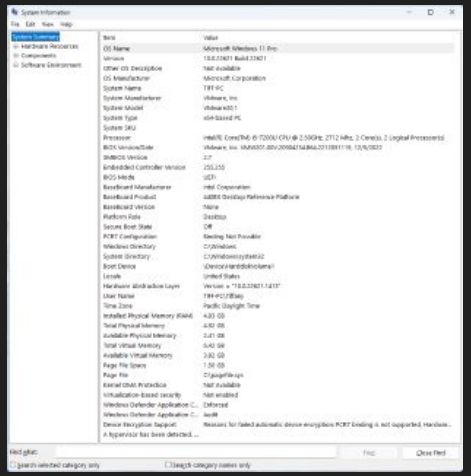
- Collect more footage
- Try more system settings
- Try on different laptops/operating systems
- Iterate solutions if problems are found

...

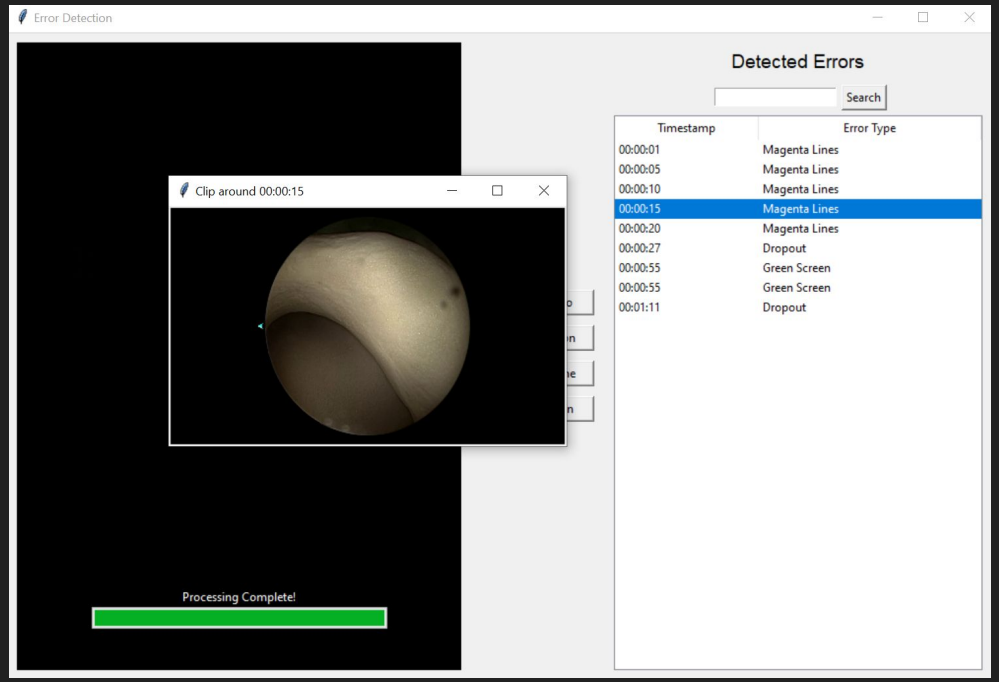
2) Integration of Scripts and GUI



3) Checking System Status

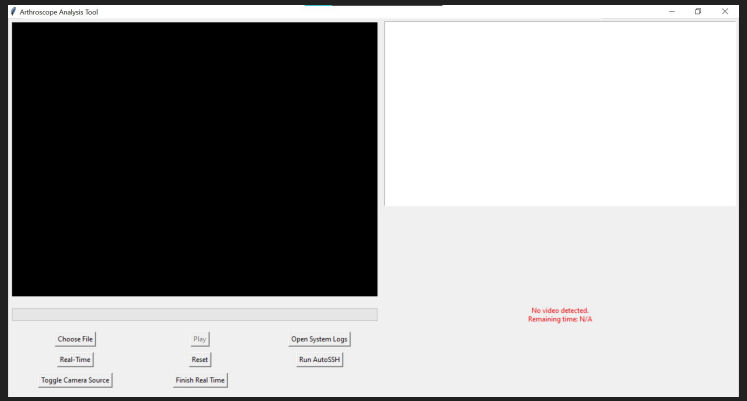


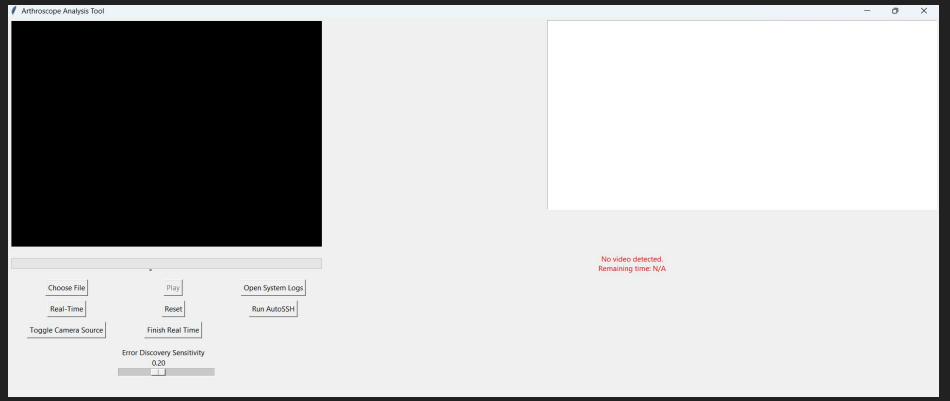
Minor GUI Improvements



Note: different GUI used here with clip showcase feature

Minor GUI Improvements





Different screen sizes & resolutions may limit what can be seen

Speeding up Video Processing

- Continue to pursue faster-than-real-time speeds for post-processing
- Reduce hardware strain: Could achieve efficient error detection on smaller devices
- Reduces probability of frame drops or skipping in extreme cases

Tuning Scripts for New Error Discovery

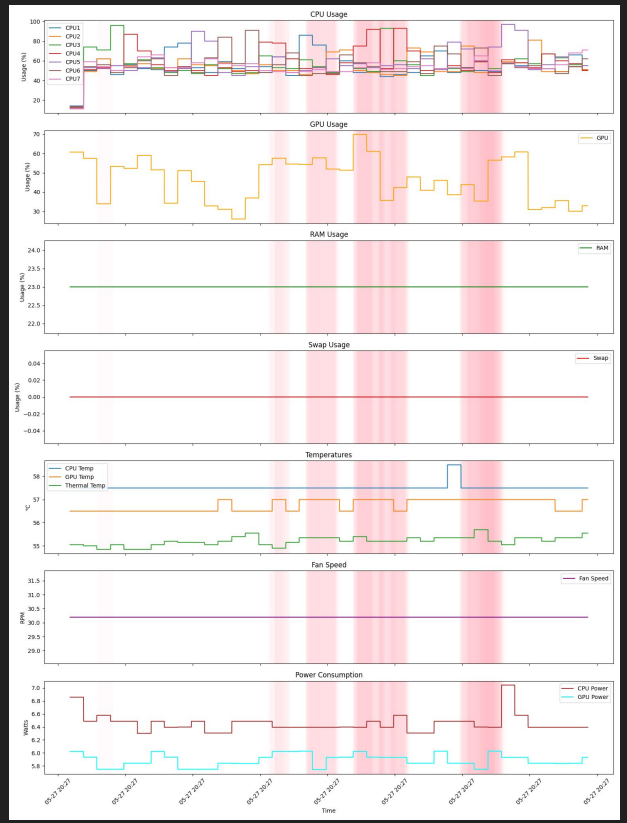
What if the system is updated or new errors are encountered?

Approaches	Biggest Benefits	Biggest Costs
Differencing (current script)	Easy to implement and use	False positives and manual sensitivity thresholding
Machine Learning (Next Frame Prediction, Contextual Video Captioning)	Automated and smart, No false positives, potentially more detailed error description	Not robust to new environments, not real-time, huge hardware burden, large amounts of data and labels needed

Interpreting Error Logs

What can we do with this new information?

- Take note of system status when errors occur
- Create dataset with system status corresponding to errors
- Many techniques exist to model system-status to error correlation (ML, Statistics, State-Space prediction, etc.)



Future Improvements

What can be improved?

- Analyze system data using statistical methods
- Analyze system data with Machine Learning methods
- Try packaging the files into executables

Thank you!
Any questions?