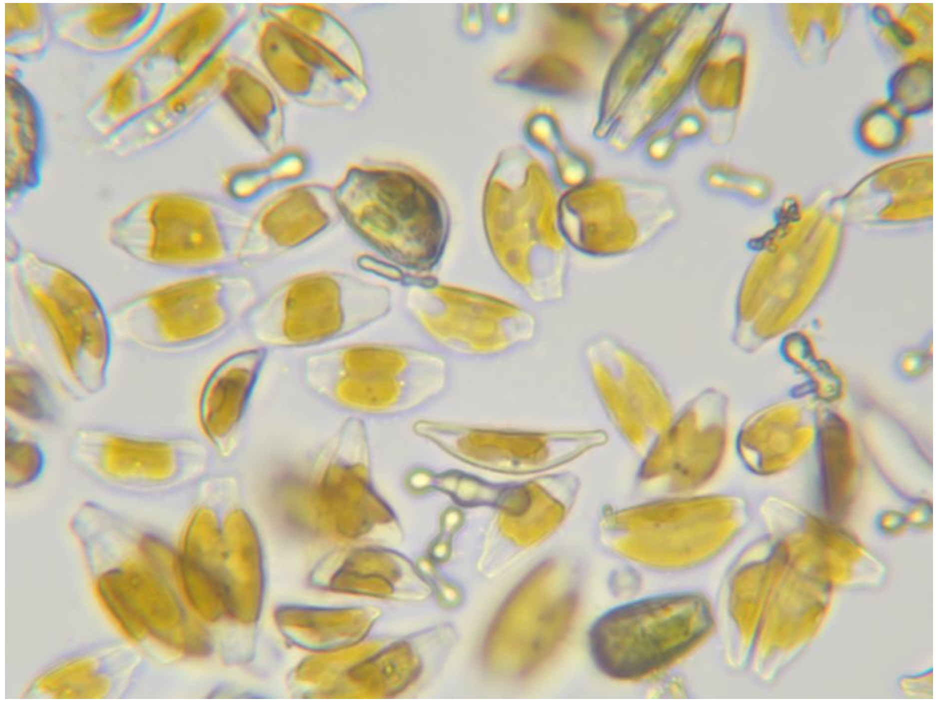
Applying Computer Vision to Measuring Algal Diversity



Emma Clow, M.S. Aquatic Resources

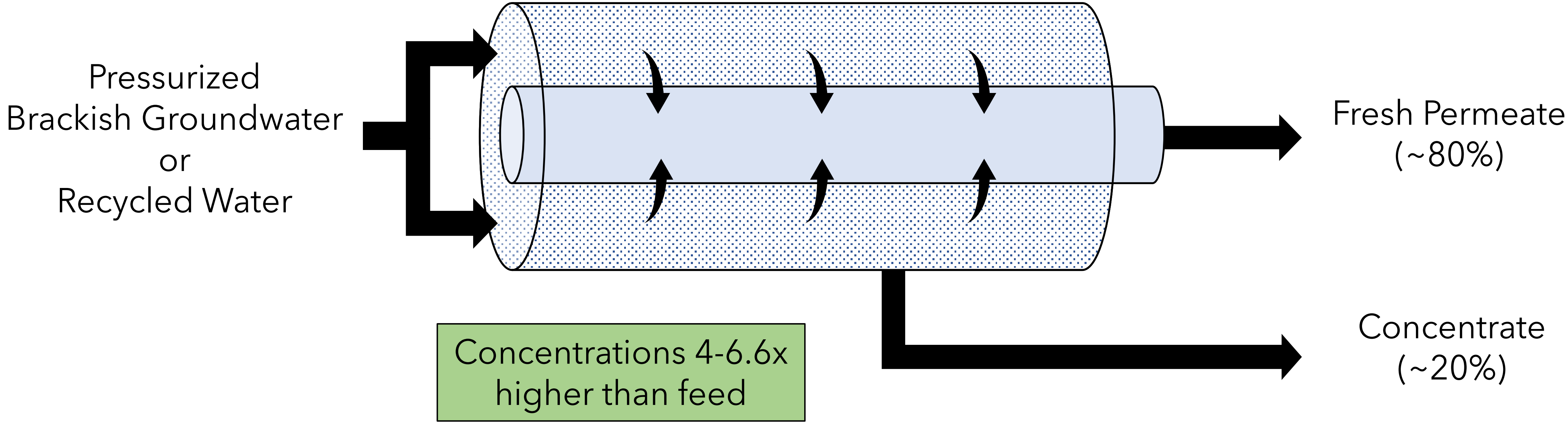
Tristan Pedro, B.S. Computer Science

DataLab12.github.io

TEXAS STATE UNIVERSITY

Problem Statement

- ➢Rejection of dissolved constituents → RO concentrate (ROC)
- ➢Freshwater recovery limited by solubility limits



How to Reduce Waste and Increase Freshwater Recovery? Diatoms!

Aqueous silica removal

↓

Decreased membrane scaling

↓

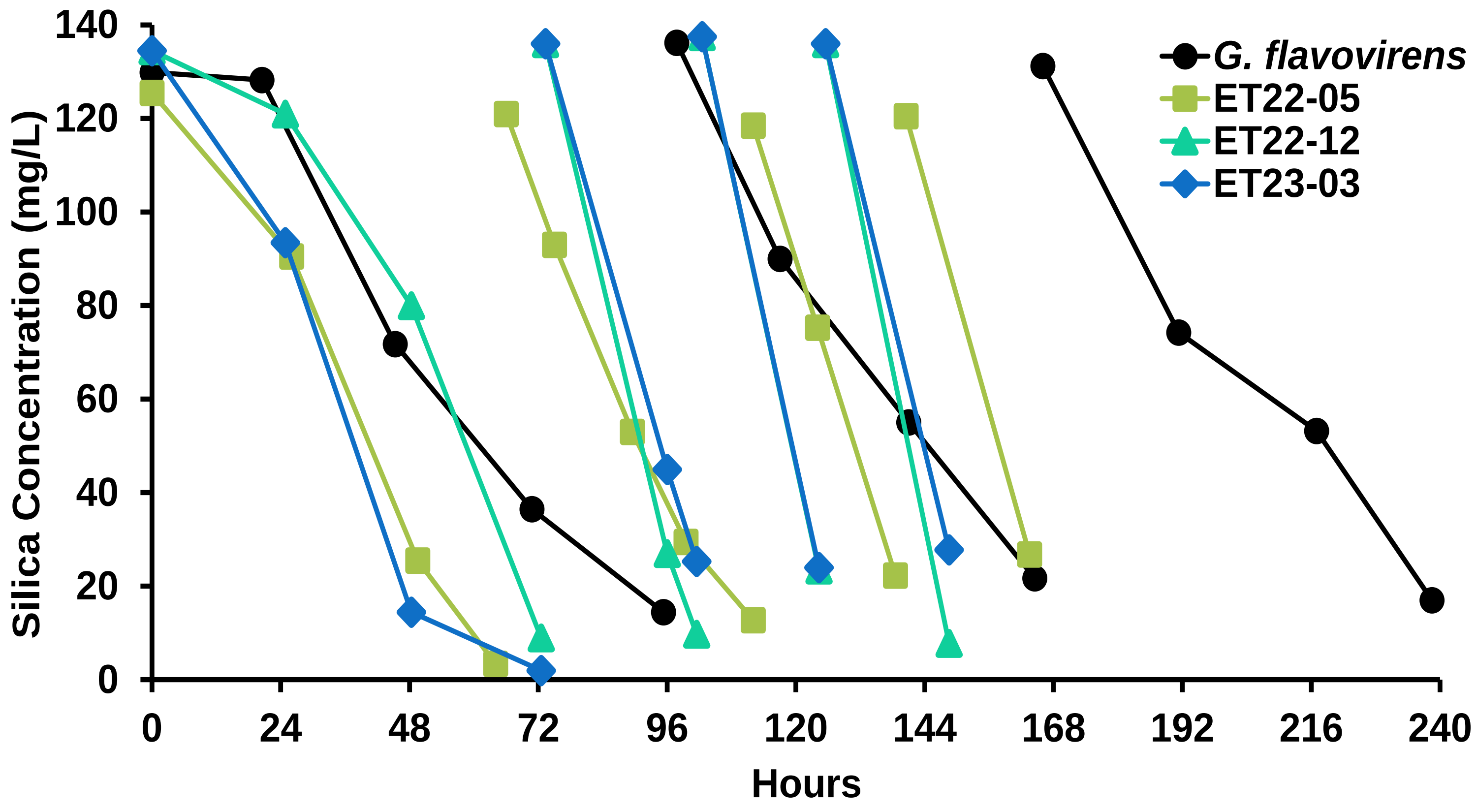
Secondary RO

↓

Freshwater recovery ~95%

Research Objectives

- Previous research focused on unialgal culture *G. flavovirens*
- Mixed algal cultures show faster silica removal from ROC

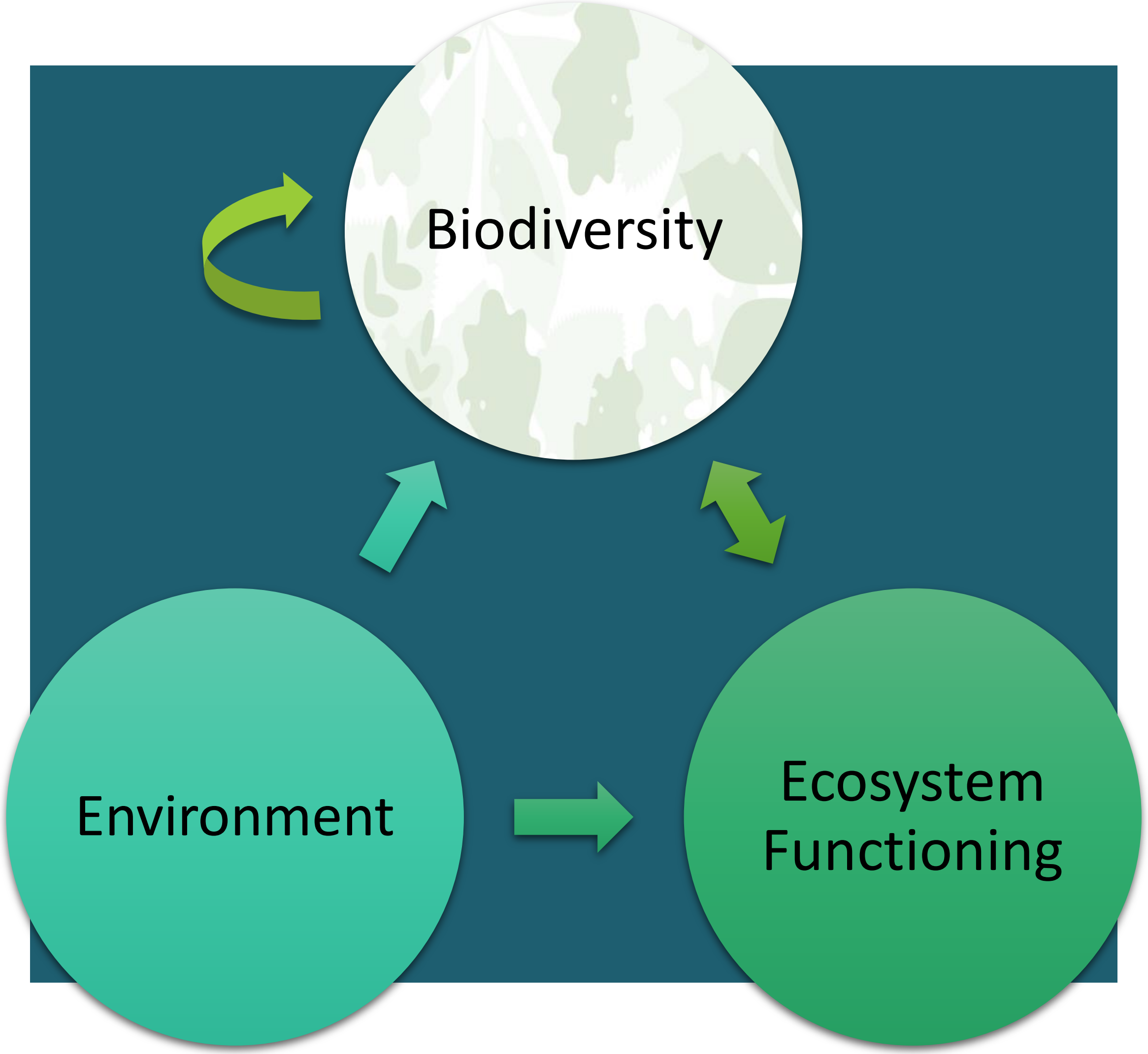


DataLab12.github.io

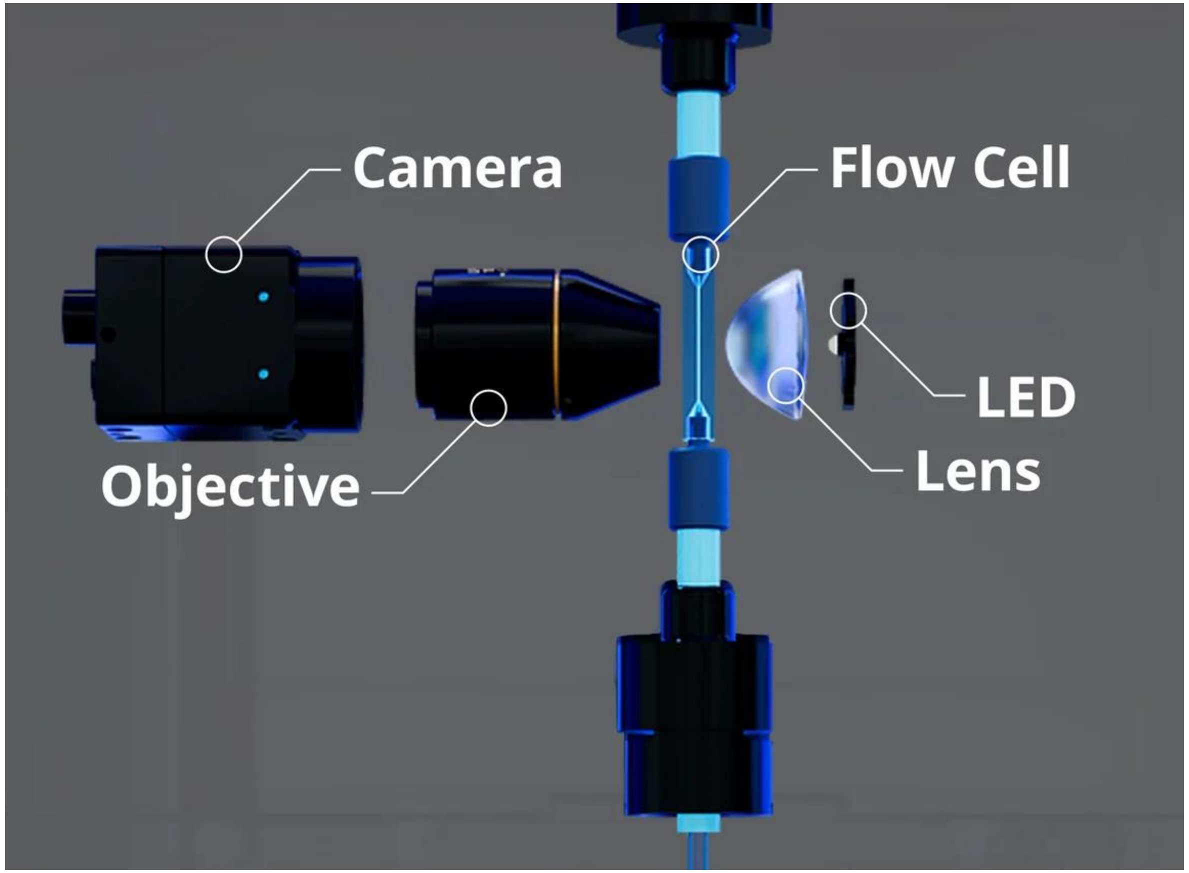
TEXAS STATE UNIVERSITY

Research Objectives

- Investigate effect of diversity in diatom cultures on ROC treatability performance
- Biodiversity-Ecosystem Function (BEF) Theory
- Measuring diversity:
  - Metagenomics
  - Microscopy
  - FlowCam



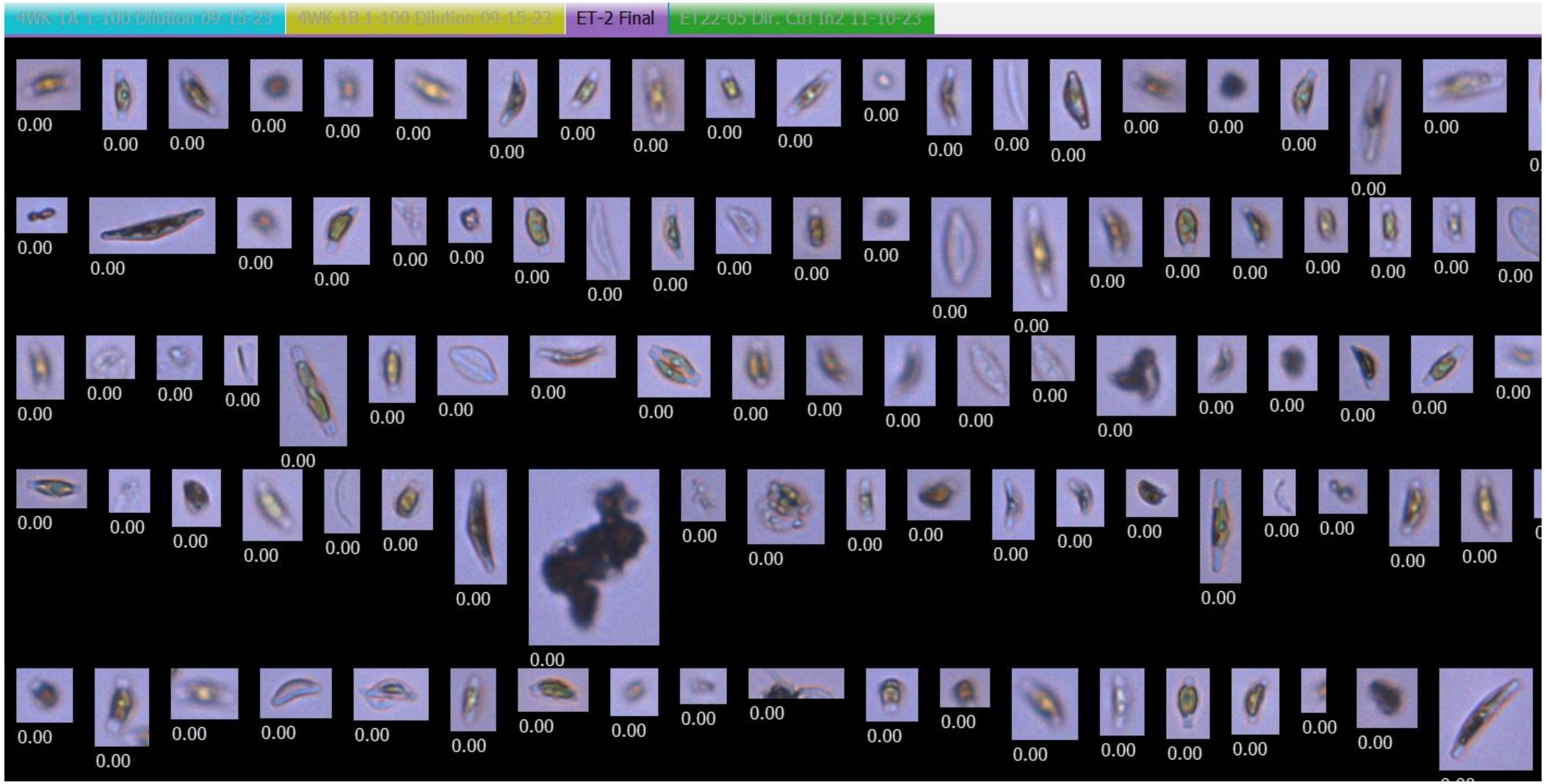
FlowCam: Flow Imaging Microscopy



Particles move through a flow cell as images are captured in real-time by a high-speed camera (https://www.fluidimaging.com/)

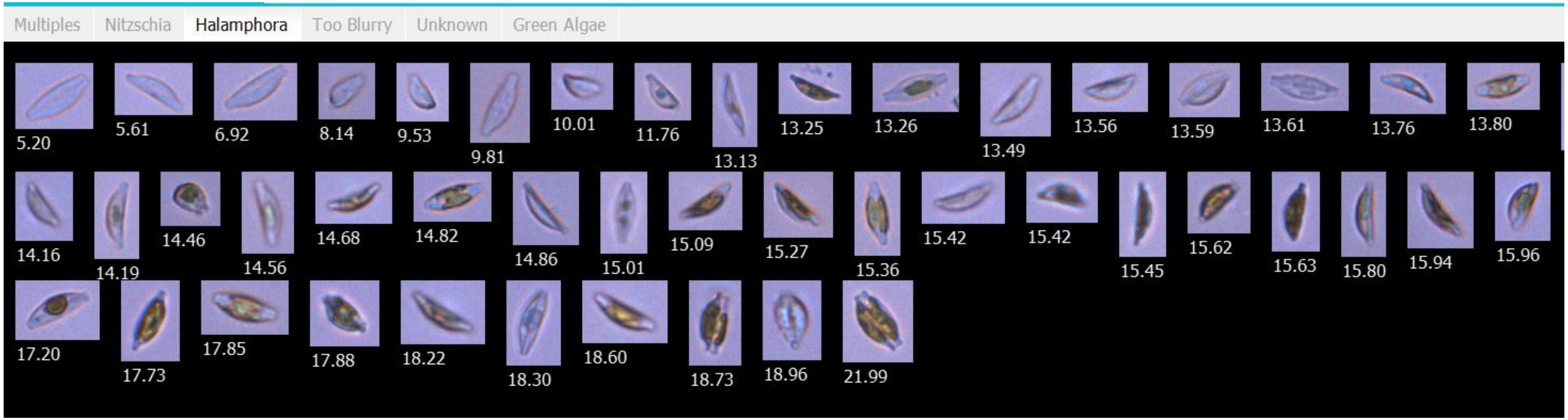
Raw Data

- Images
- Binary images
- .lst file



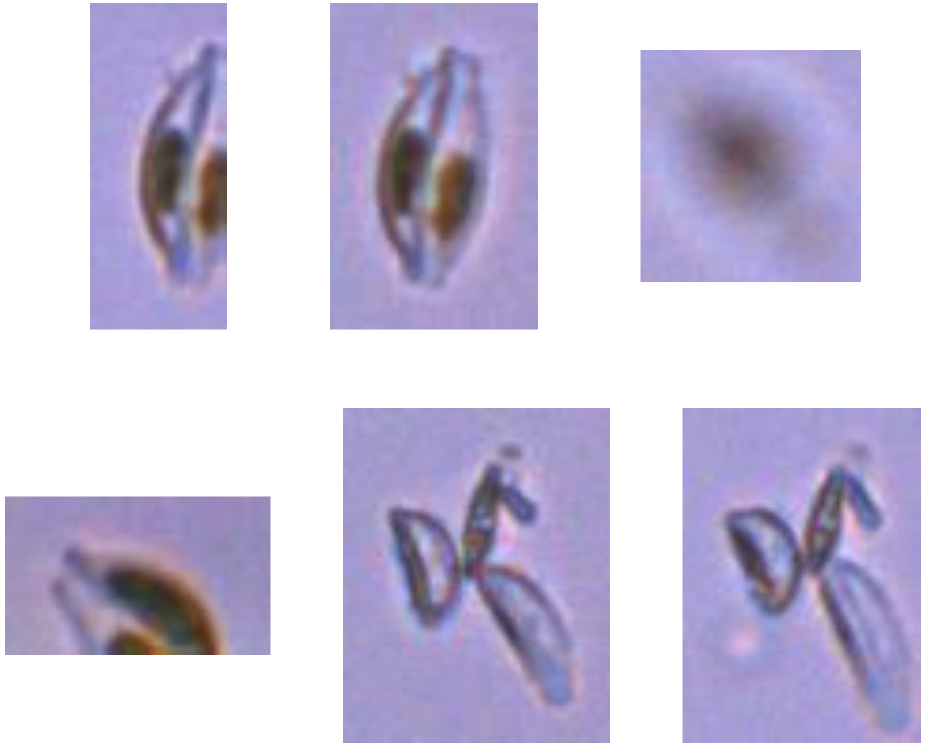
Current Labeling Process: Manual

- FlowCam software statistical analysis vs. machine learning
- Bottleneck: **TIME**



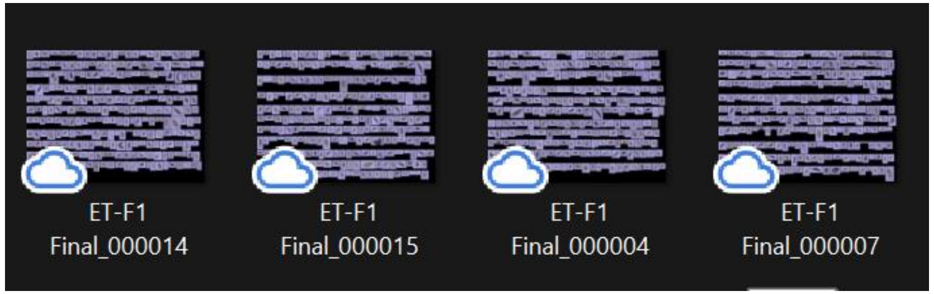
Anticipated Outcome

- Expedite classification and eliminate images that can't be classified
  - Empty
  - Blurry
  - Partial
  - Duplicate
- *Genus species* counts based on size and morphology

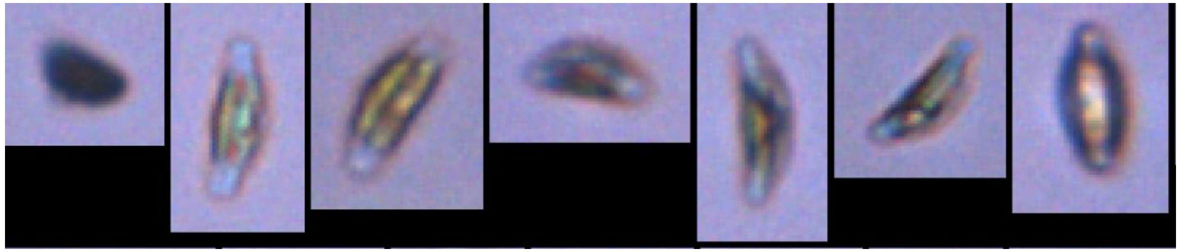


Data Preprocessing: Step 1

1. Sheet of Images from FlowCam



2. Looking Inside One of the Sheets

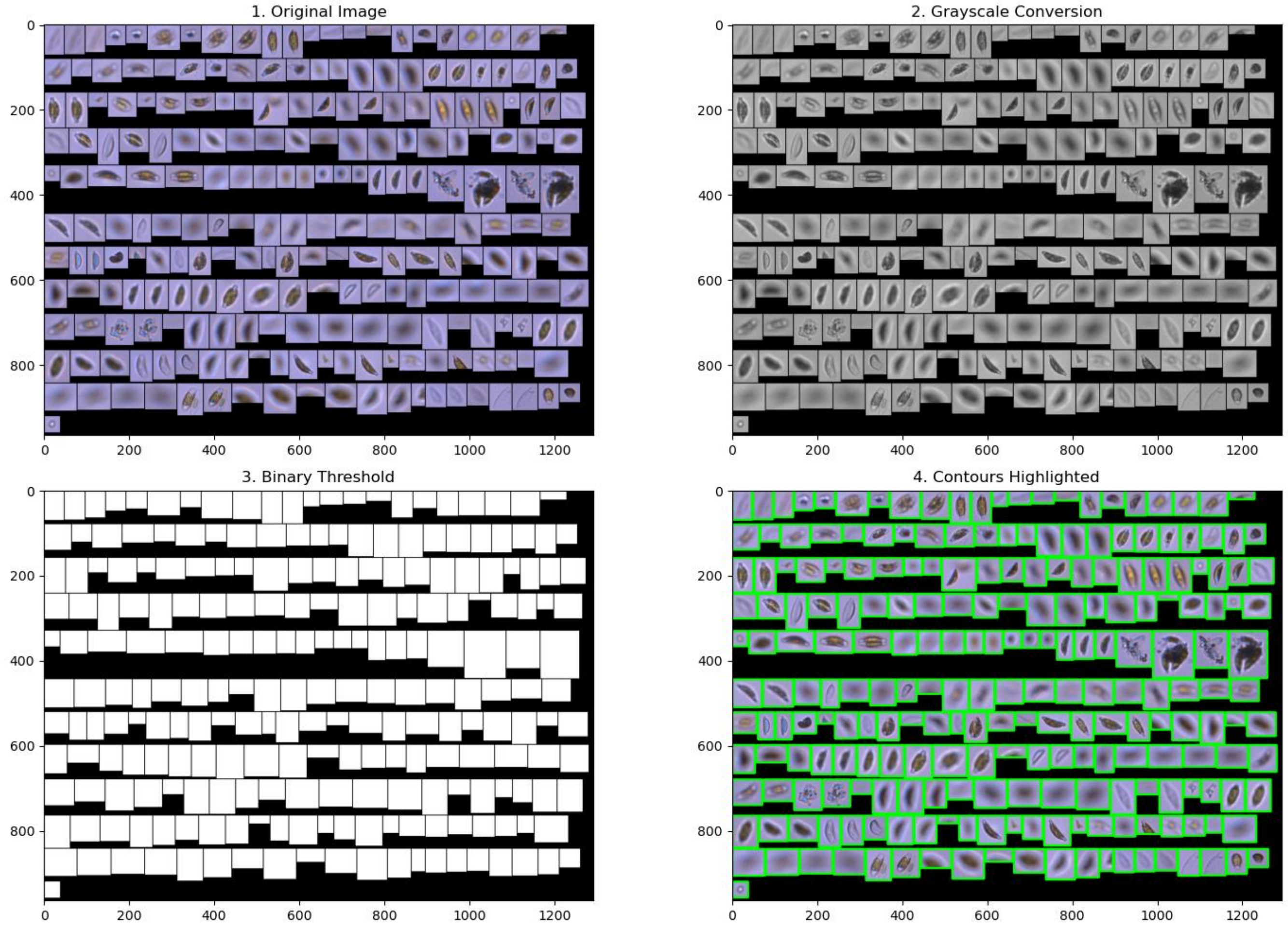


3. Individual Images in Directory

TLDR: Images from FlowCam to Individual Processable Images

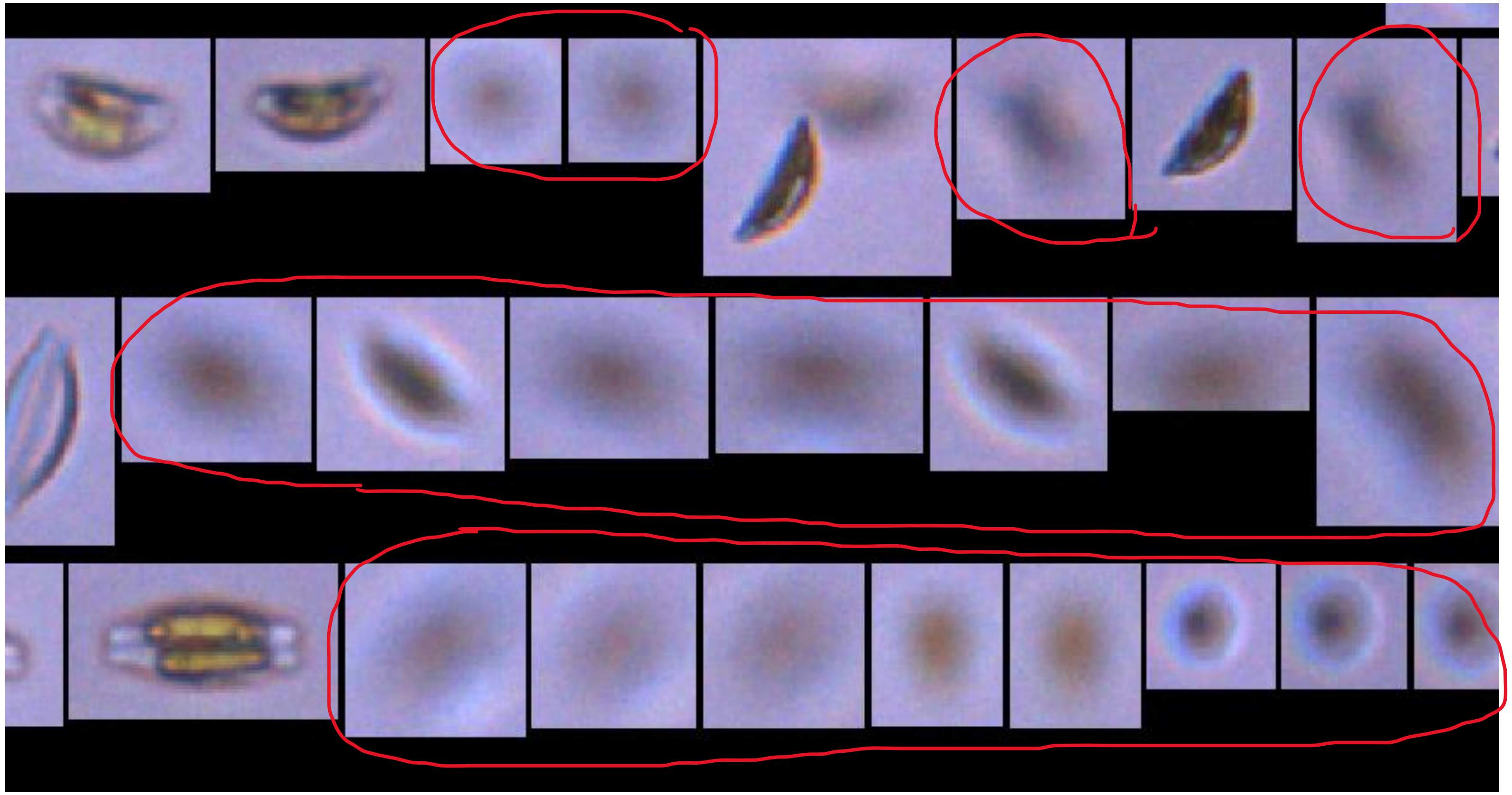
Solution to Step 1: Contours!

- Load Sheet of Images from FlowCam
- Convert to Grayscale & Binarization
- Contour Detection:
  - Contours are continuous lines that represent boundaries of objects in an image.
- Use Each Contour as a Bounding Box to Extract Each Image
  - Valid contours are sorted based on the position of their bounding box's top-left corner



Data Preprocessing: Step 2

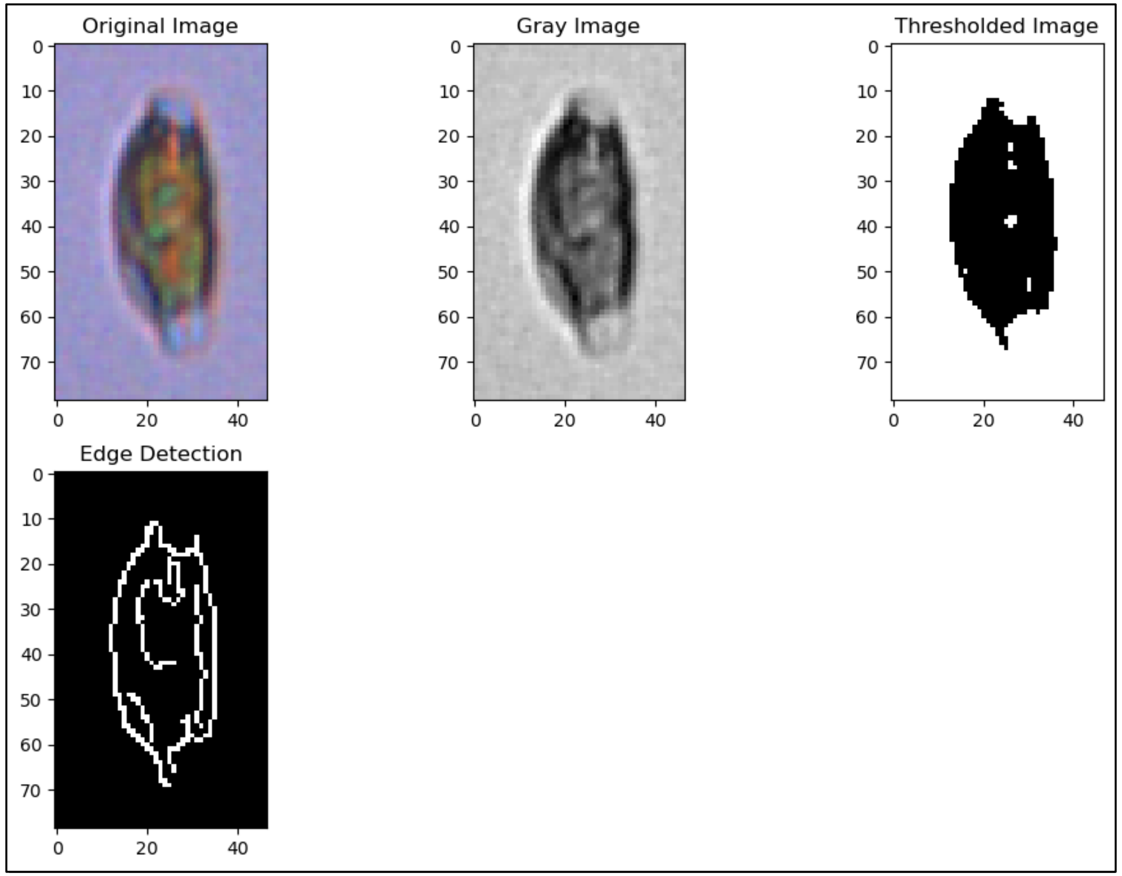
➢ Filtering Out Empty/Blurred Cell Images



“Empty” cells in circled in red.

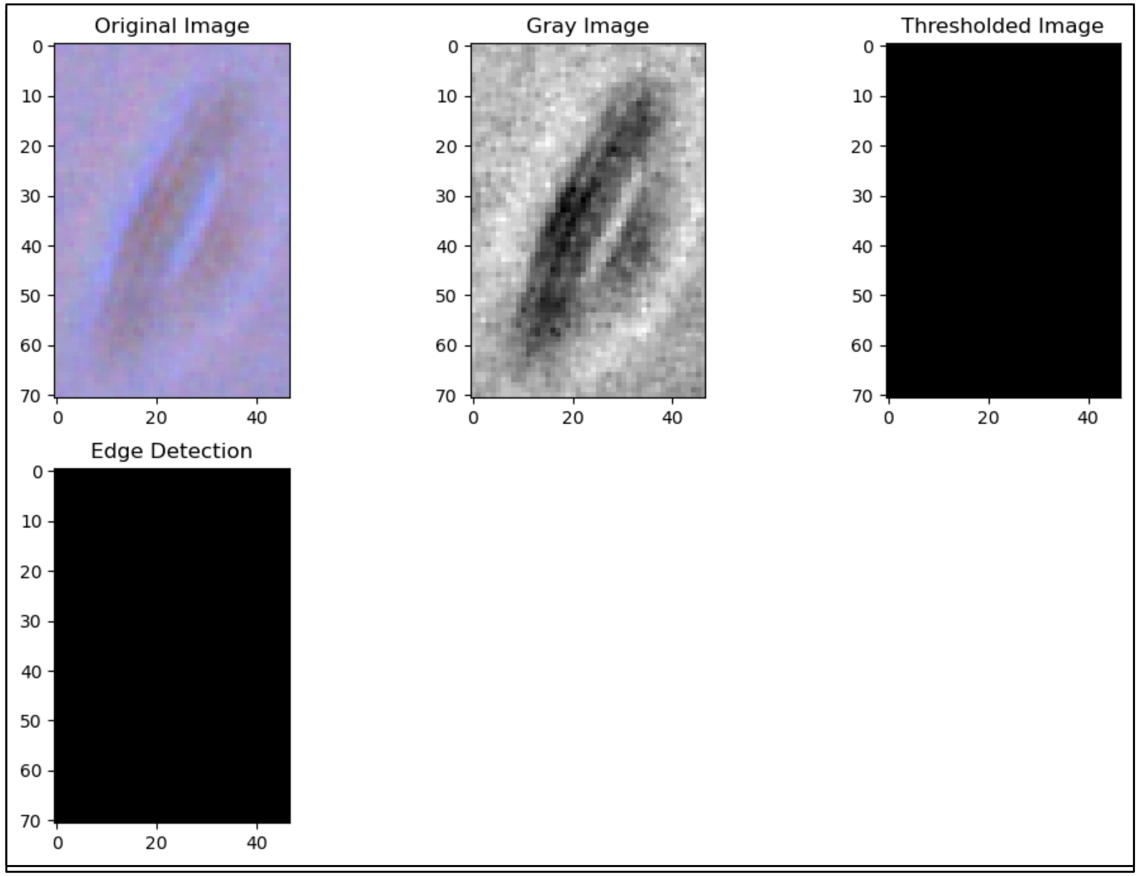
Solution to Step 2: Edge Detection

Normal Cell Image



VS

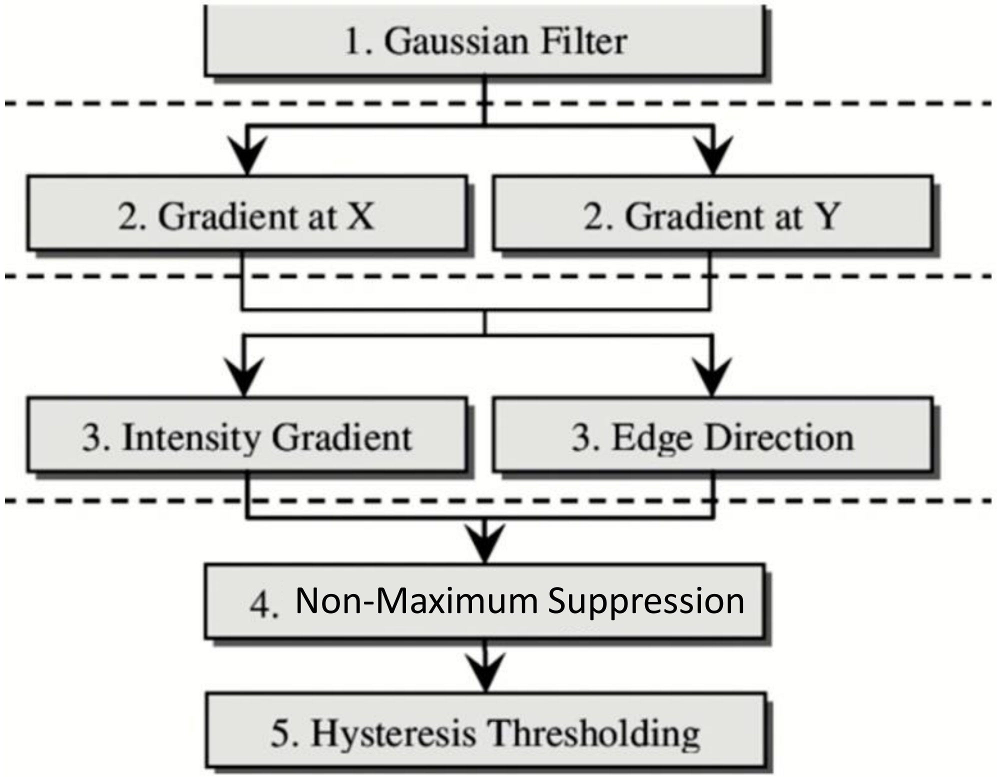
"Empty/Blurred" Cell Image



Edge Detection Algorithm Explained

- Gaussian Filter:
  - Smoothens (blurs) the image to reduce noise.
- Gradient at X and Y:
  - Computes the gradient (rate of change) of pixel intensity in both horizontal and vertical directions
- Intensity Gradient
  - Combines the gradients to determine the overall intensity gradient at each pixel
- Edge Direction:
  - Calculates the direction of edges based on the gradient
- **Non-Maximum Suppression**
  - Suppresses non-maximum edge responses to preserve only the strongest edge pixels
- **Hysteresis Thresholding**
  - Applies high and low thresholds to identify and link edges, creating continuous edge contours

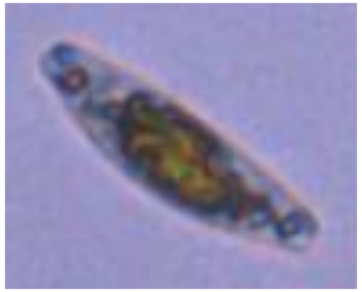
OpenCV's Canny Edge Detector Algorithm Pipeline



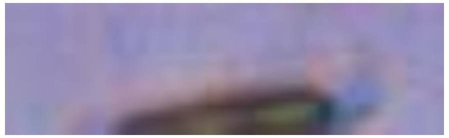
Data Preprocessing: Step 3

- Filtering out the images that have partial cells

Normal Cell Images



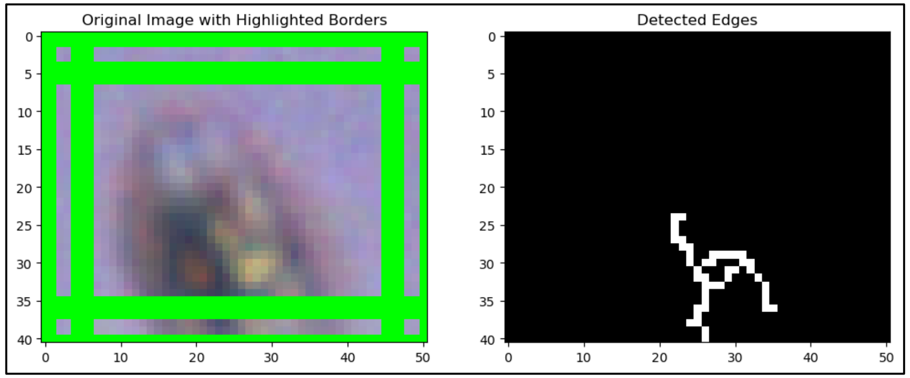
Partial Cell Images



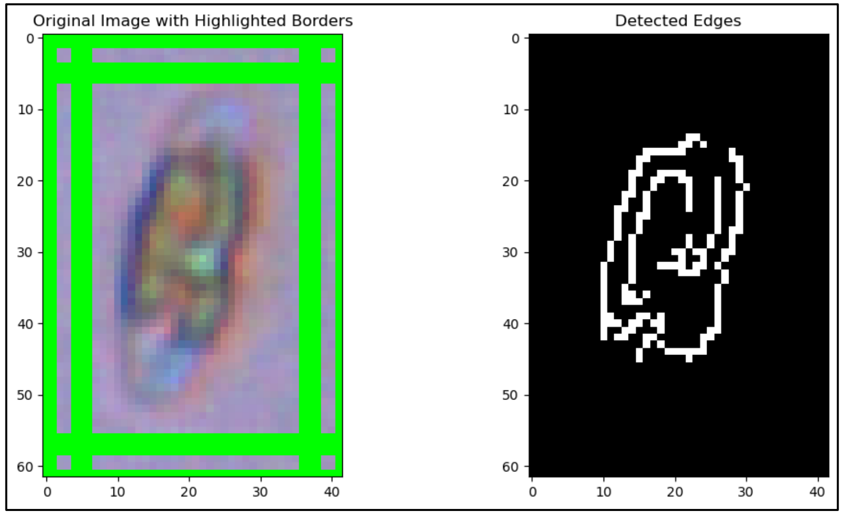
Solution to Step 3: Edge Detection Again!

- Utilize OpenCV's Canny Edge Detection Algorithm Again
- This time, we focus on detecting edges near image borders to find partial cells
- Visualized results to fine-tune the border size parameter

Partial Cell Image



Normal Cell Image



Data Preprocessing: Step 4

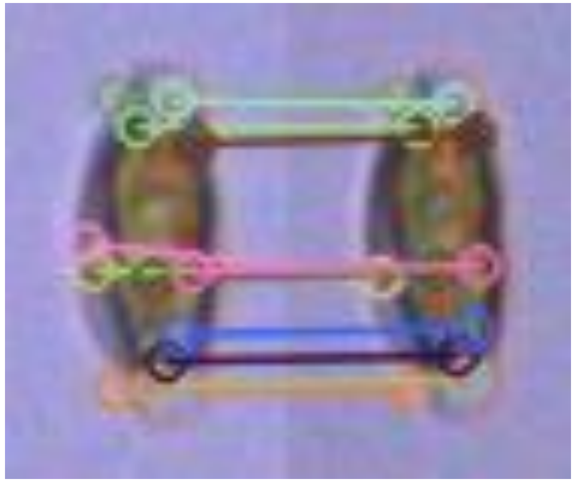
- Filtering out the duplicate cell images

Raw Data Subset

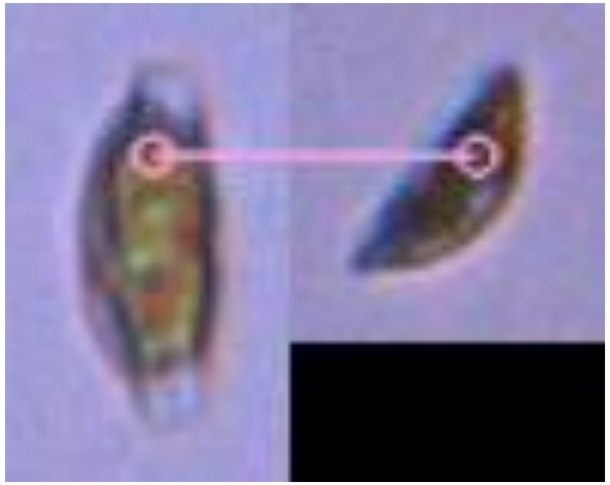
Solution to Step 4: Feature Matching

- Scale-Invariant Feature Transform (SIFT)
  - Algorithm detects and matches key points invariant to scale and rotation
- High number of consistent key points between images indicates a match
- Effective for filtering duplicates

Group of Same Cells



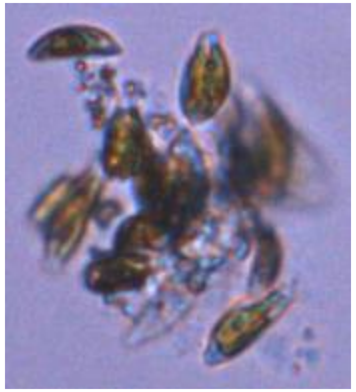
Group of Different Cells



Accuracy of Algorithms on a Manually Labeled Subset

- Algorithms stronger at filtering for Aggregates, most Halamphoras, and Multiples
- Weaker at filtering Nitzchia, and some of the Halamphoras, due to the edge detector not finding edges in a few of the manually classified images

Aggregate
(All Passed)

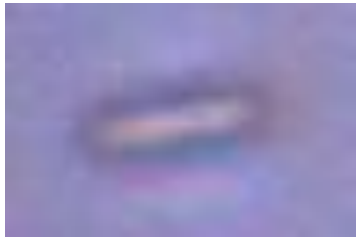


Multiple
(All Passed)

Halamphora
(Pass 145)

Halamphora
(Fail 4)

Nitzchia
(Pass 2)



Nitzchia
(Fail 2)

Data Preprocessing Still?: Step 5

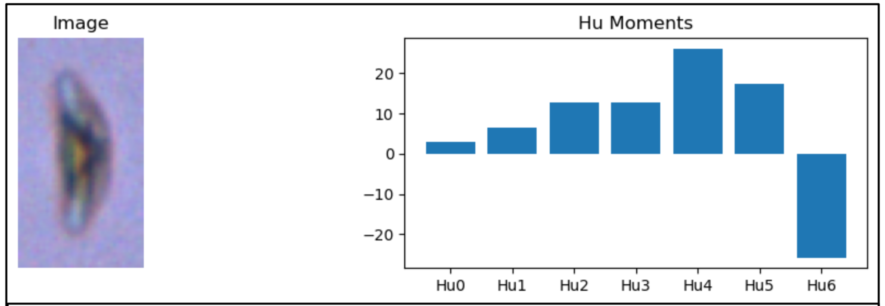
- Grouping Similar Cell Images
  - “*Genus species* counts based on size and morphology”

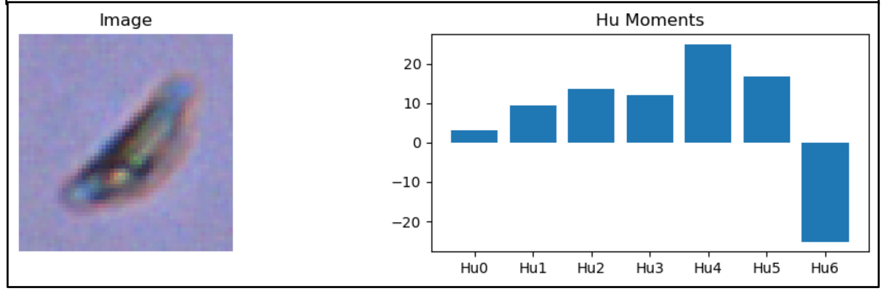
- Two Steps
  - Feature Extraction
    - Getting information from the image
  - Clustering
    - Finding relationships (grouping)

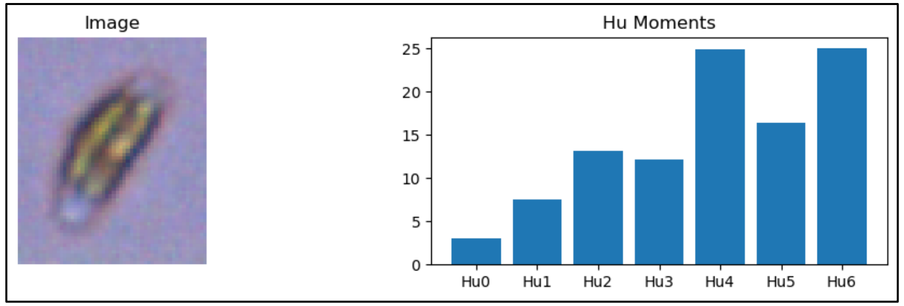
- A lot failed to show me anything meaningful…

Feature Extraction: Hu Moments

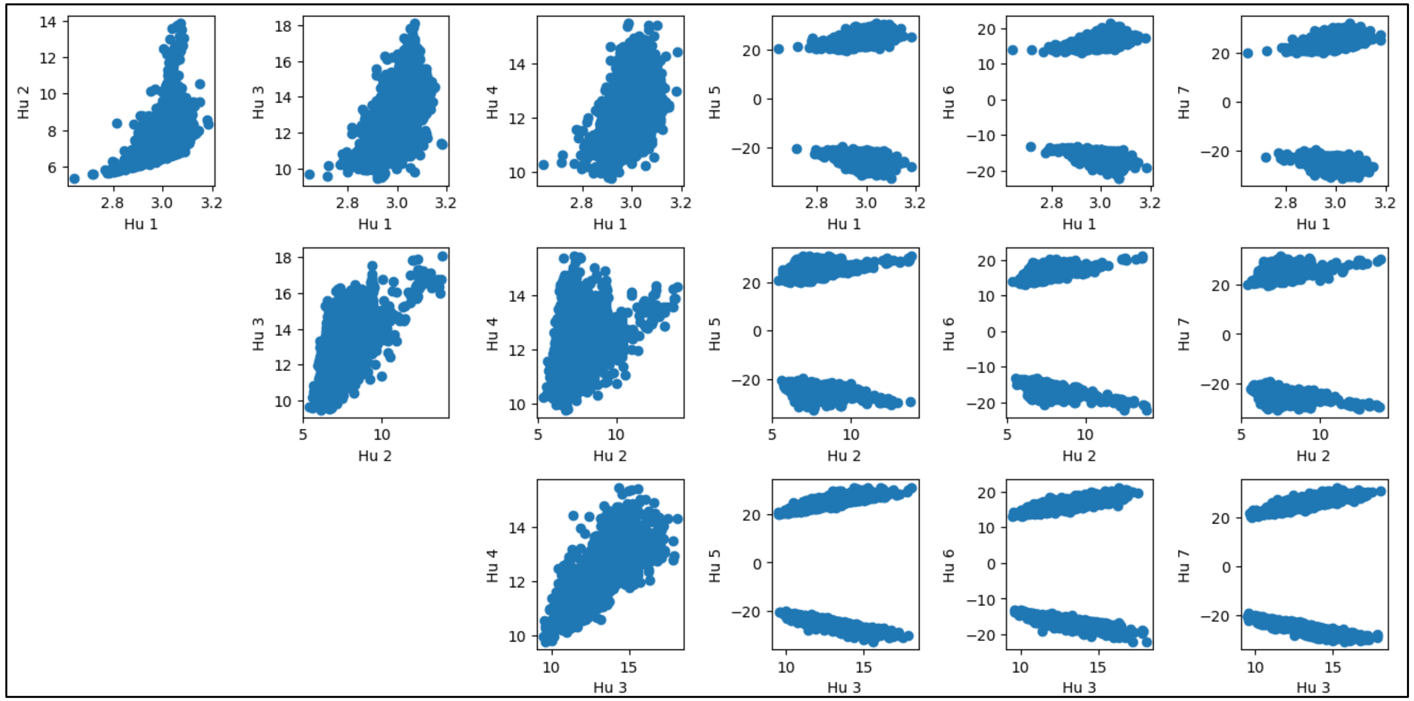
- What are Hu Moments?
  - Set of seven numbers derived from the moments of an image
- Why Hu Moments?
  - Invariant to Image Transformations
  - Robust for comparing image shapes
- Feature Extraction
  - Compact representation of an image's shape, allowing for analysis based on geometric features

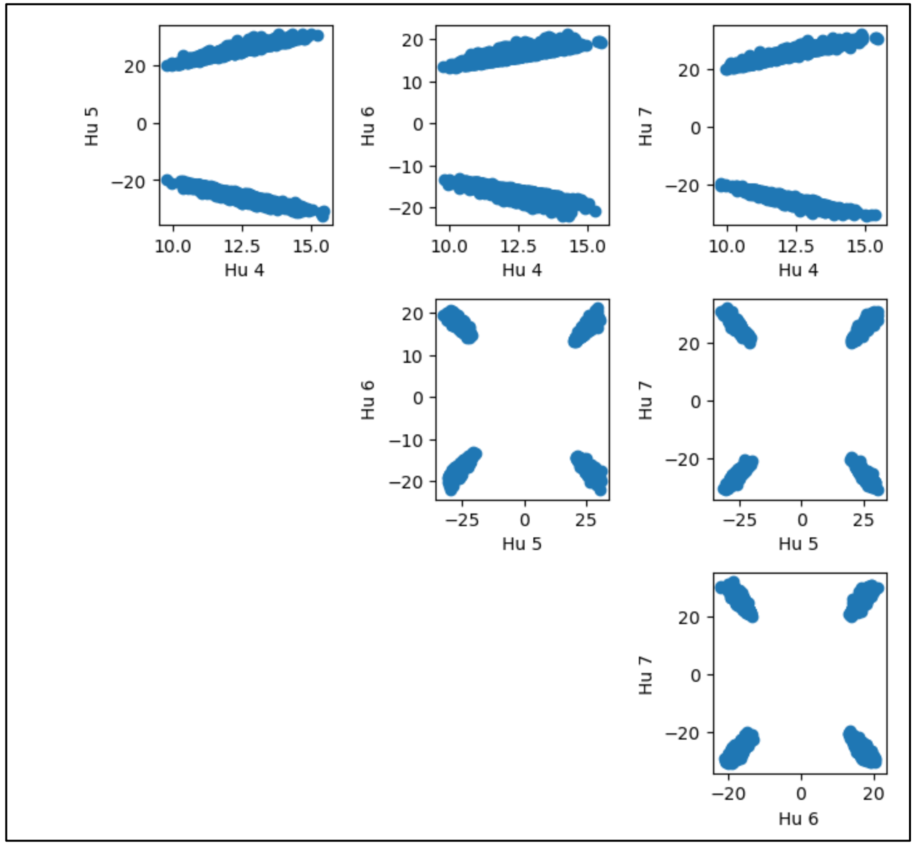






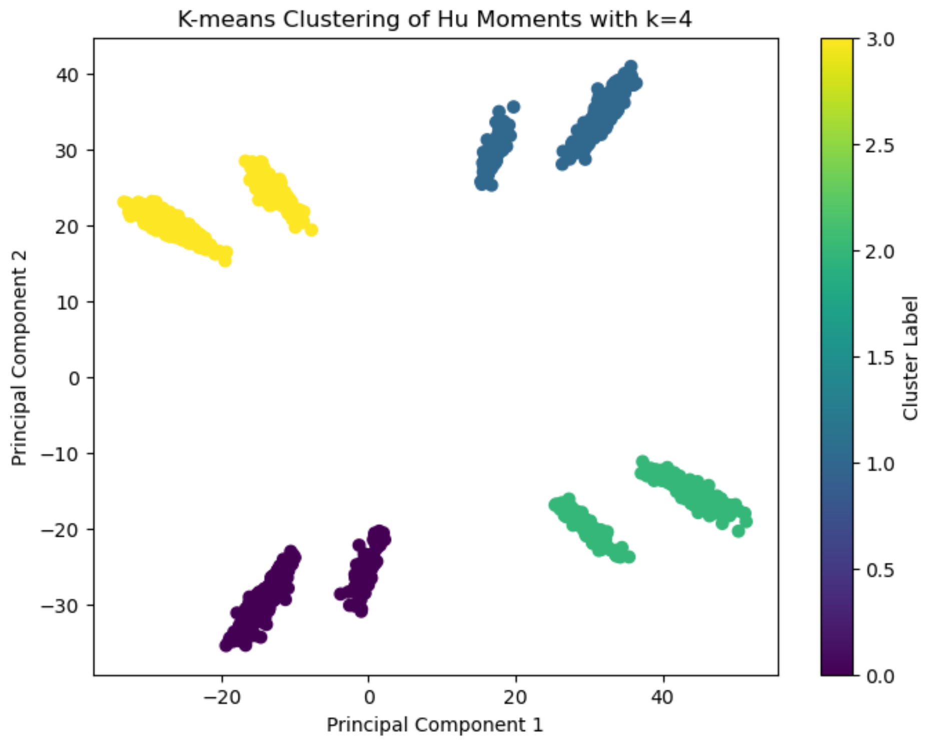
Scatter Plot of Hu Moments

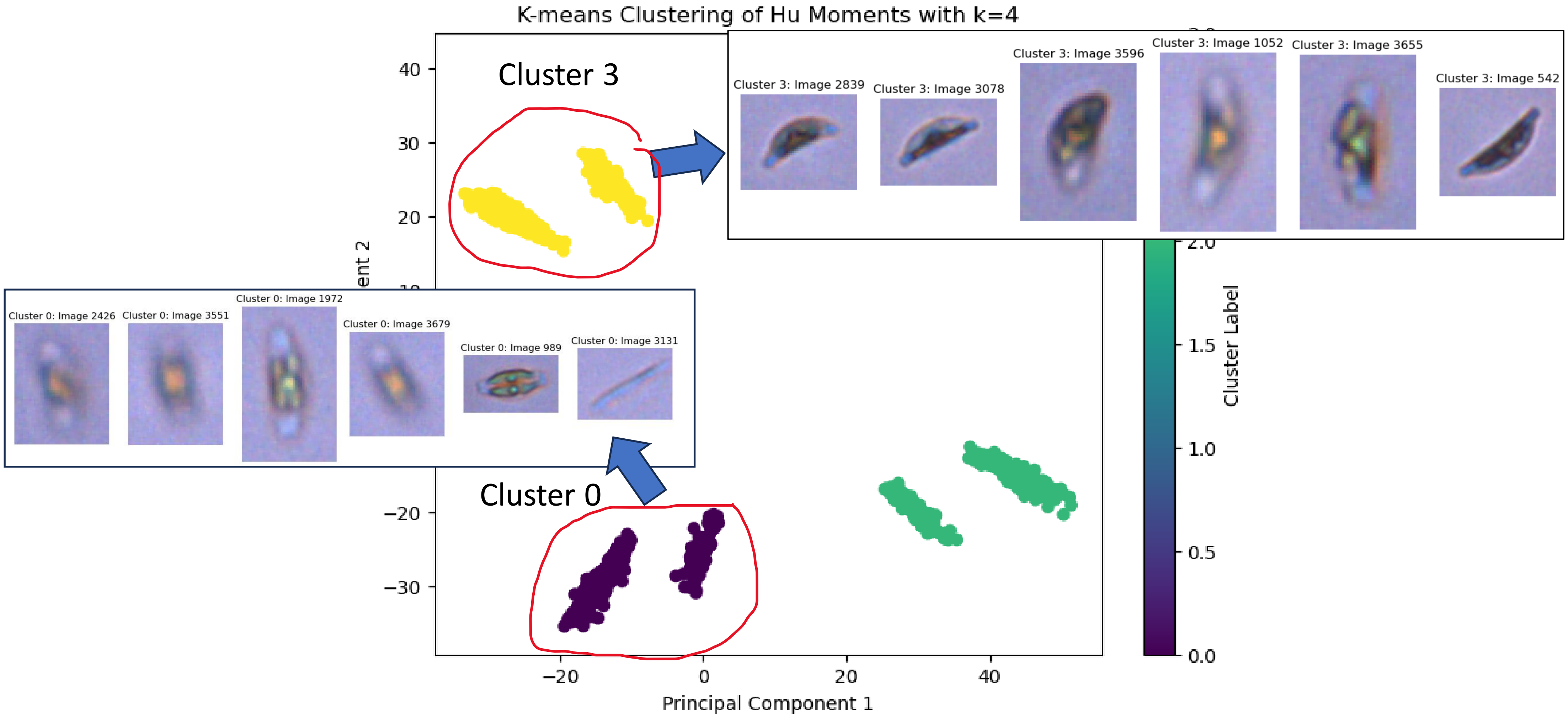




Hu Moments: Outcome

- Applying Principal Component Analysis (PCA)
  - Used PCA to go from 7 Hu Moments for each image (7D) into a 2D space
- Insightful Groupings
  - After applying K-means, k=4 achieved the best silhouette score (87%)
  - Silhouette score measures how similar an image is to its own cluster, compared to other clusters




K-means Clustering of Hu Moments with k=4
Cluster 3
Cluster 0
Cluster 3: Image 2839
Cluster 3: Image 3078
Cluster 3: Image 3596
Cluster 3: Image 1052
Cluster 3: Image 3655
Cluster 3: Image 542
Cluster 0: Image 2426
Cluster 0: Image 3551
Cluster 0: Image 1972
Cluster 0: Image 3679
Cluster 0: Image 989
Cluster 0: Image 3131
40
30
20
−20
−30
−20
0
20
40
Principal Component 1
Cluster Label
2.0
1.5
1.0
0.5
0.0

Future ML/DL Exploration Options

- Covered unsupervised to an <u>extent</u> I'm happy with for now

- Explore Supervised Methods (if we get labeled data)
  - Baseline (Decision Tree, etc.)
  - Possible Model Selections (CNN, SVM, etc.)
  - Methods requiring few labels (semi-supervised and active learning, etc.)

Questions?
or
Suggestions?