Scaling of Evolutionary Search of Algorithm Space to Speed-Up Scientific Image Understanding Workflows

Nicholas Grabill - University of Michigan
Kai Pinckard - Reed College
Dirk Colbry - Michigan State University

Outline

- What is Scientific Image Understanding
- Motivating the problem
- Our Solution and how it Scales

Visual Observations

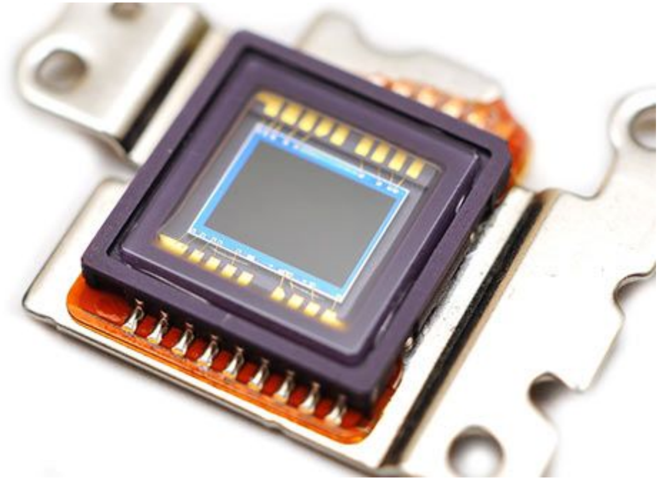
- Long history in Biology
- Traditionally done by hand

Photography

- Changes science
  - Scientists are able to record video without knowing what they will see
  - Cameras may see something the scientists missed
  - Different scientists can view the same data with different scientific questions in mind

Cameras Everywhere

- Transforming how scientists gather data
- Very affordable
- Data is becoming very cheap to gather, so there is a lot more of it



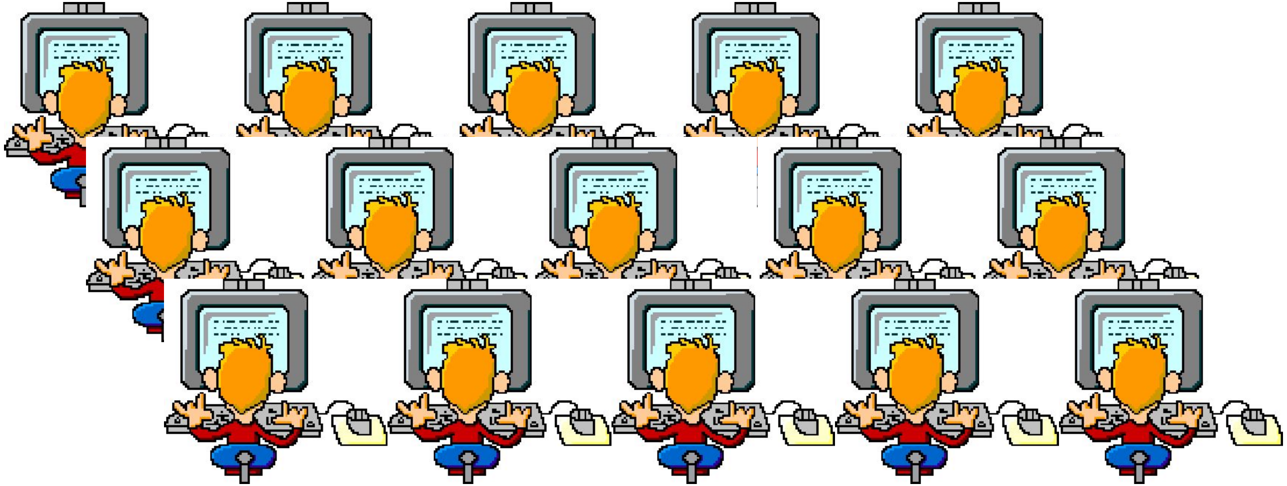
Charge-Coupled Device (CCD)

How are digital images analyzed?

Graduate students are cheap…

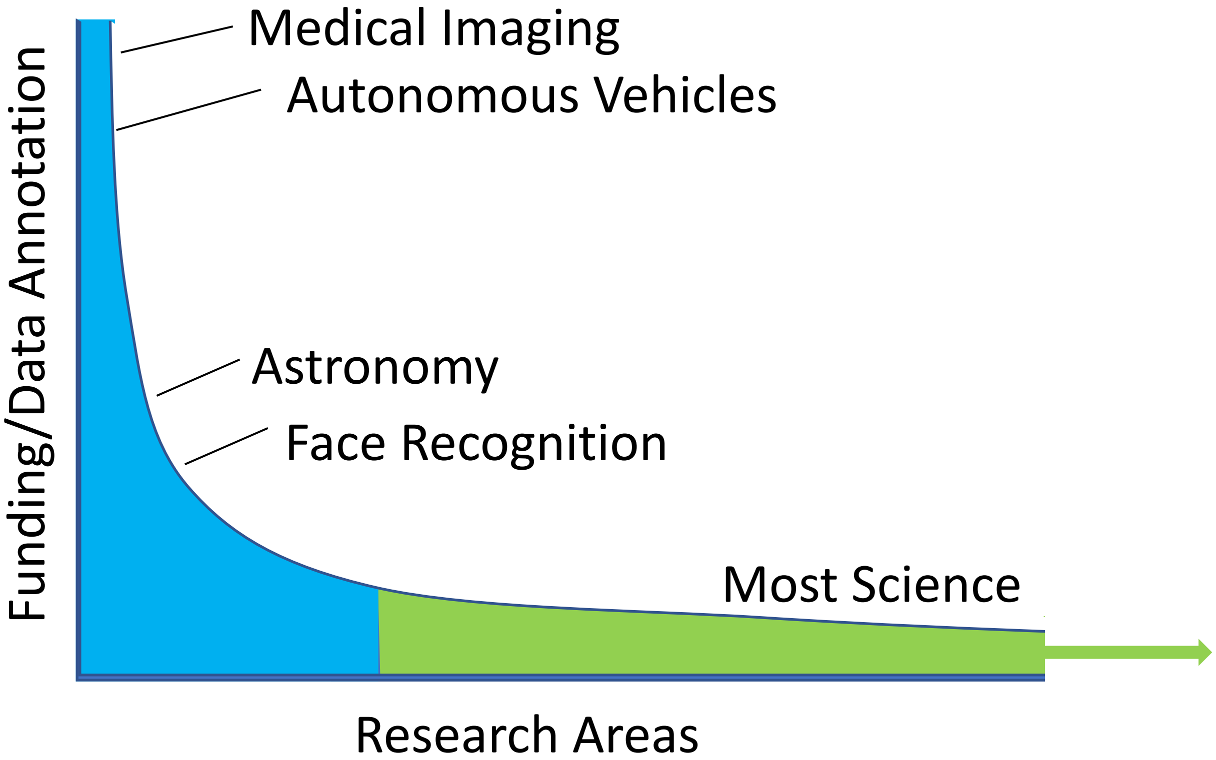
Undergraduates are even cheaper!

Also, easy to run in parallel



With enough data we can use machine learning

Serving the Long Tail of Scientific Imaging



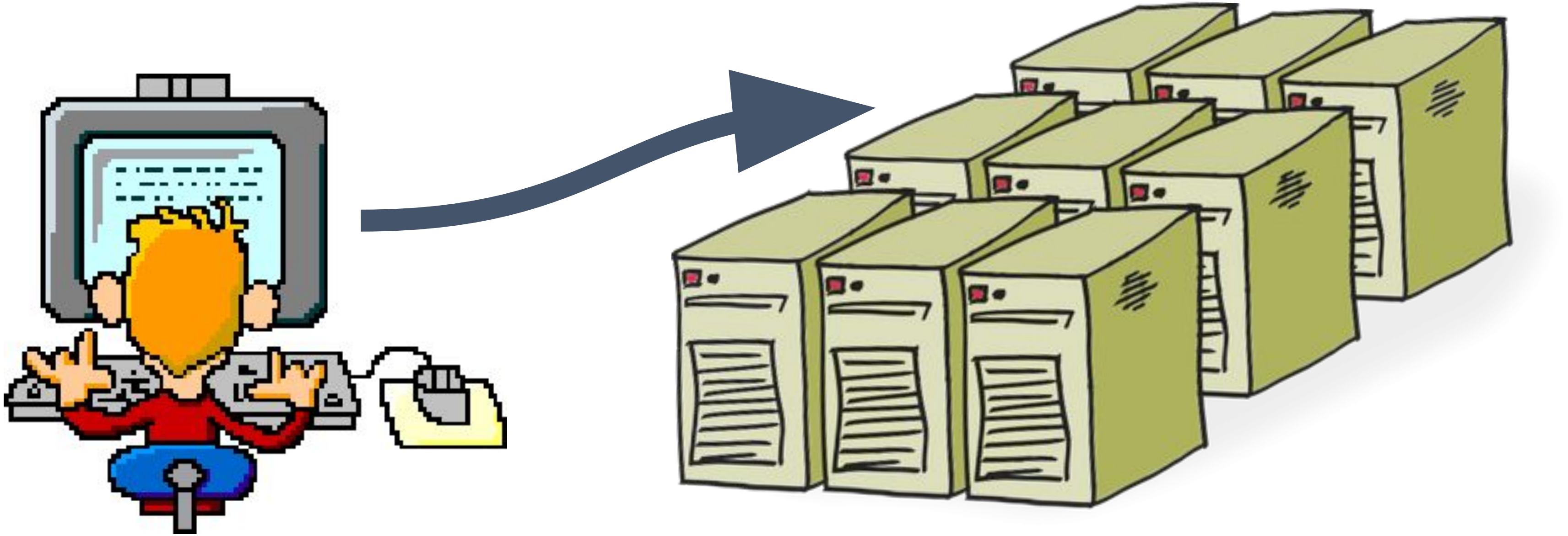
Why is automating the long tail of science hard?

- Data annotation can be expensive.
- Features change with every problem
- Projects can’t afford an engineer for every new idea
- Not everyone can be an expert in image analysis, so training every scientist doesn’t always work
- By the time you are done annotating a training set you may be done with the research!

While

Researcher Annotates Images
(in the foreground)

Computers search for solutions
(in the background)



Example: Image Segmentation

Original Image

Annotated Ground Truth Label

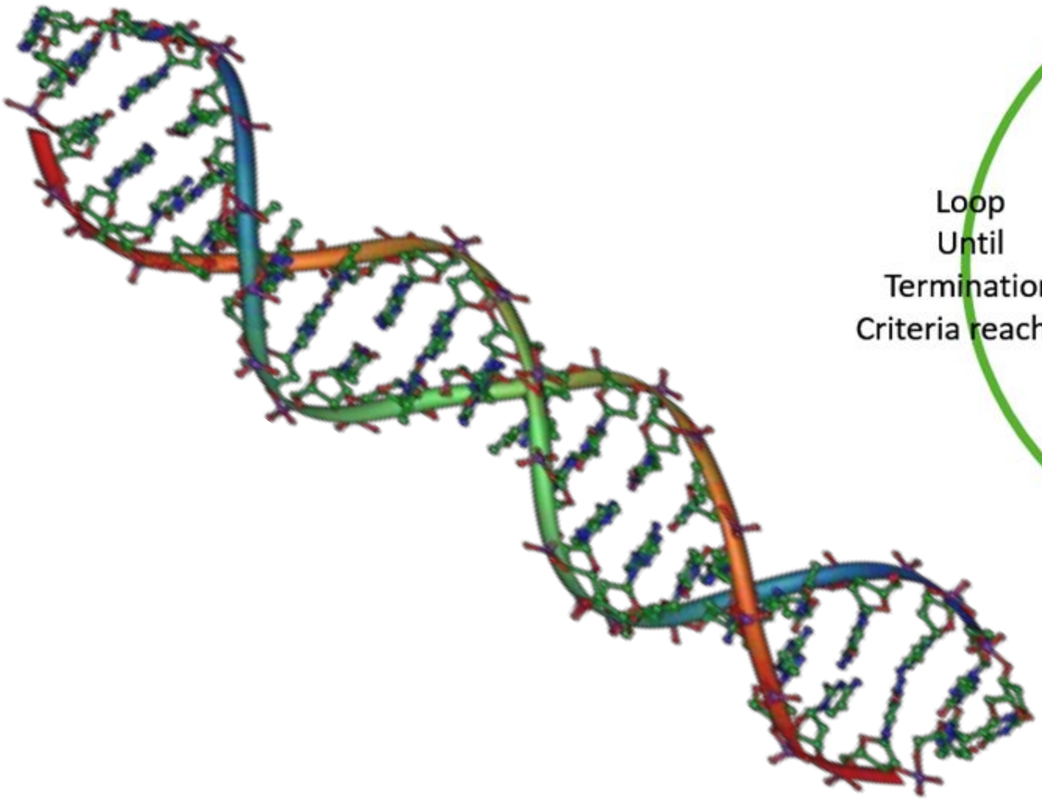
Example from KOMATSUNA plant dataset: http://limu.ait.kyushu-u.ac.jp/~agri/komatsuna/

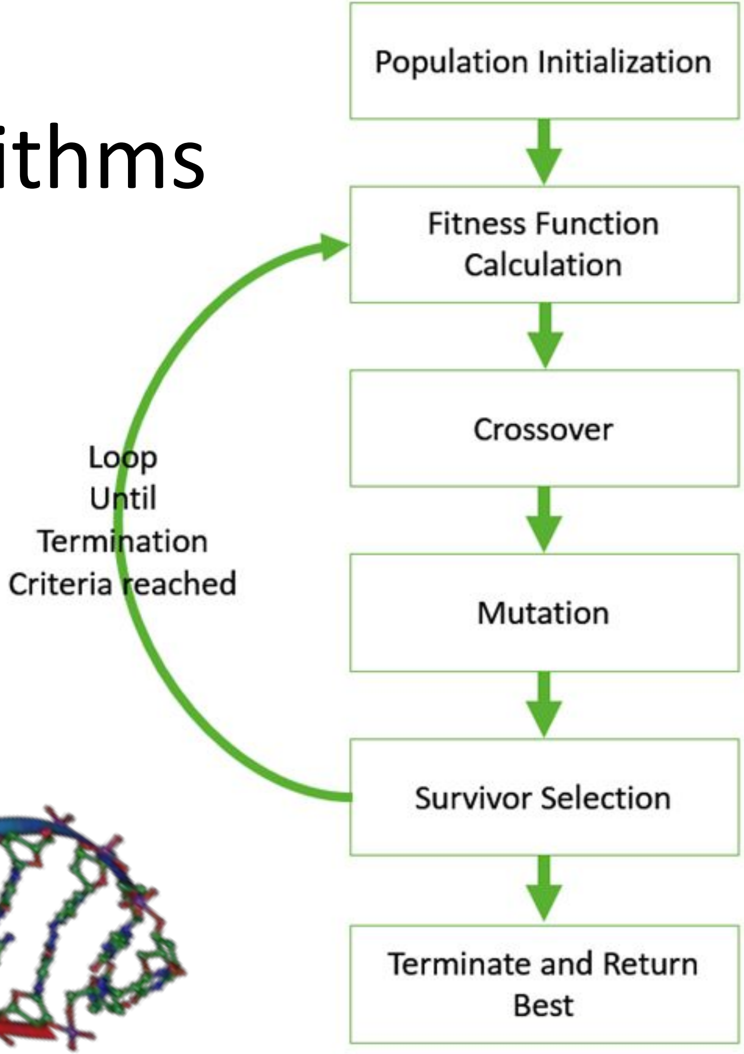
Python Skimage.Segmentation Library

1. thresholding(image[, …]) - Basic image thresholding (not part of skimage)
2. random_walker(data, labels) - Random walker algorithm for segmentation from markers.
3. active_contour(image, snake)- Active contour model.
4. felzenszwalb(image[, …]) - Computes Felsenszwalb's efficient graph based image segmentation.
5. slic(image[, …]) - Segments image using k-means clustering in Color-(x,y,z) space.
6. quickshift(image[, …]) - Segments image using quickshift clustering in Color-(x,y) space.
7. watershed(image[, …]) - Find watershed basins in image flooded from given markers.
8. chan_vese(image[, mu, …])- Chan-Vese segmentation algorithm.
9. morphological_geodesic_active_contour(…) - Morphological Geodesic Active Contours (MorphGAC).
10. morphological_chan_vese(…) - Morphological Active Contours without Edges (MorphACWE)
11. inverse_gaussian_gradient(image) - Inverse of gradient magnitude.
12. circle_level_set(…[, …]) - Create a circle level set with binary values.
13. checkerboard_level_set(…) - Create a checkerboard level set with binary values.

Which one is Best?

Basic Genetic Algorithms





Why Use GAs?

- Can search highly ***heterogeneous*** search space
- Can search ***non-differentiable*** search spaces
- ***Easy*** to seed search space with known engineered solutions
- Can ***scale*** easily (task level scaling)
- Output is ***human readable***

Part 1: Define Your Population Space

[Algorithm (1-13), option 1, option2, option3, …, optionN]

[5, 244, 0.44, 72, …, -28] …   [2, 10, 0.1, 1, …, 1035]

Part 2: Fitness Function

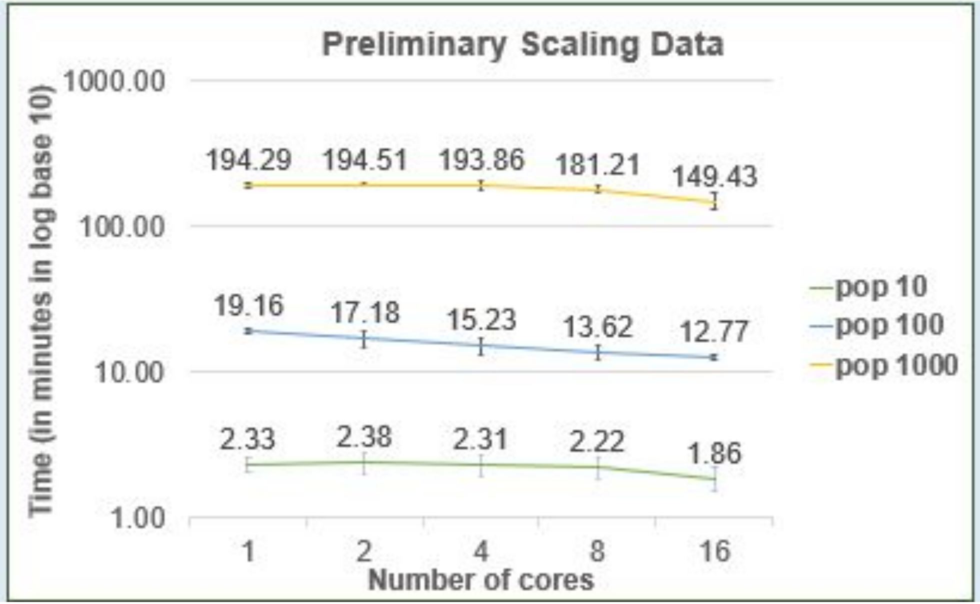
Original Image

Annotated Ground Truth

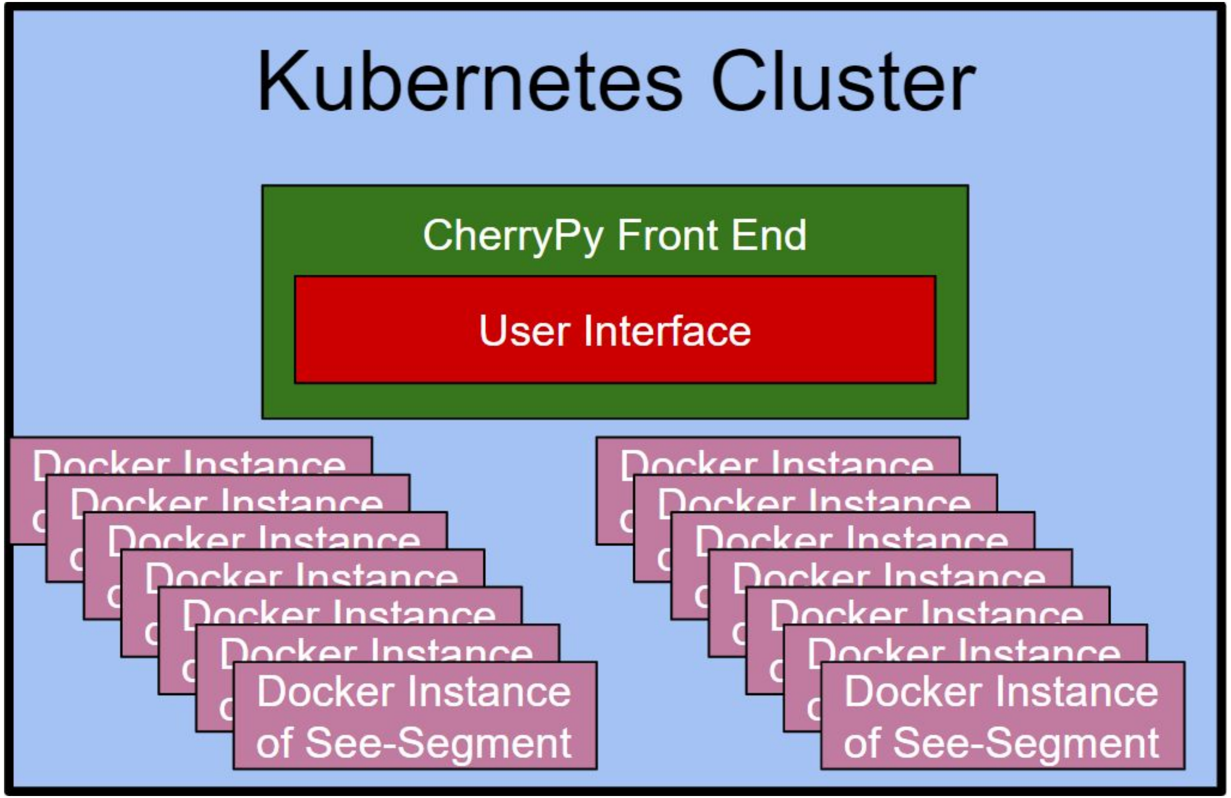
Experiments

- Preliminary Scaling results on our local HPC
- Running on multiple Kubernetes Clusters

Scaling Results

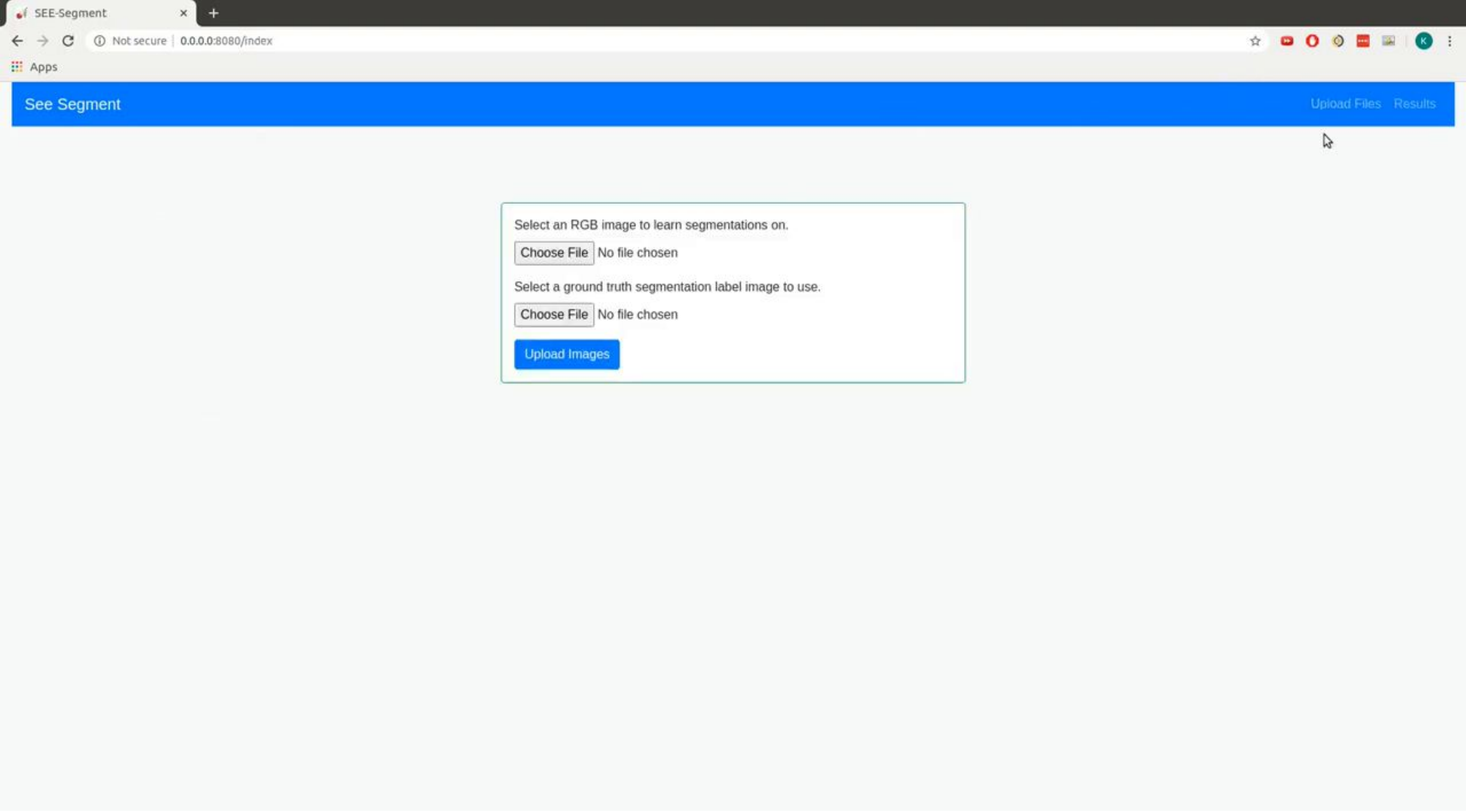


Running SEE-Segment the Cloud



Why Use Kubernetes with Docker Containers?

- The Docker Containers ***simplify*** the setup process by automatically creating the same environment for each SEE-Segment worker
- Using Kubernetes and Docker means that SEE-Segment workers and the SEE-server can be run on any ***cloud platform*** that supports Kubernetes without any code changes.
- Furthermore, Kubernetes makes ***scaling*** up the number of SEE-Segment workers as simple changing a single number.


SEE-Segment
Not secure
0.0.0.0:8080/index
Apps
See Segment
Upload Files
Results
Select an RGB image to learn segmentations on.
Choose File
No file chosen
Select a ground truth segmentation label image to use.
Choose File
No file chosen
Upload Images

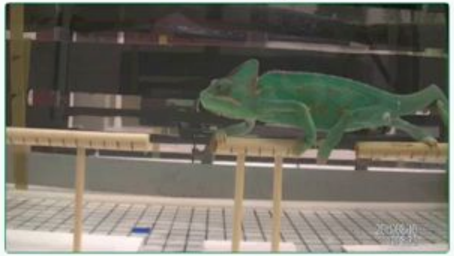
SEE-Segment

Not secure | 0.0.0.0:8080/monitor

Apps

See Segment　　Upload Files　Results

RGB Image



Current Best

Ground Truth

Segmentation Code:

```
channel_num = 3
if len(img.shape) > 2:
    num_channels = img.shape[2]
    if channel_num < num_channels:
channel = img[:, :, int(channel_num)]
    else:
hsv = skimage.color.rgb2hsv(img)
#print(f"working with hsv channel {channel_num-3}")
channel = hsv[:, :, int(channel_num)-3]
else:
    channel = img
pscale = np.max(channel)
my_mx = 0.23 * pscale
my_mn = 0.44 * pscale

output = None

if my_mn < my_mx:
```

Fitness: 0.06

Parameters:

['CT', 4594, 0.981, 7873, 0.23, 7245, 2159, 10, 0.77, 8233, 6488, 3, 8, 0.0001, 0.44, [1, 1], 3.4, 'small disk', 'checkerboard', 3, 1438, -21, 0.0, 0.0, 0.0]

0.0.0.0:8080/monitor

Questions

Nicholas Grabill - University of Michigan

- grabilln@umich.edu

Kai Pinckard - Reed College

- kaidpinck@reed.edu

Dirk Colbry - Michigan State University

- colbrydi@msu.edu

SEE-Insight Team and Alumni:

- Davin Lin (Grinnell College)
- Emma Burleson (Emory University)
- Katrina Gensterblum (Michigan State University)
- Noah Stolz (Rensselaer Polytechnic Institute)
- Nate Britton (Michigan State University)
- Emani Hunter (Michigan State University)
- Cameron Hurley (Michigan State University)
- Paul Ezimako (Texas Southern University)
- Raymond Lesiyon (Michigan State University)
- Alexander Eboru (Wayne State University)
- Amon Harris (North Carolina Central University)