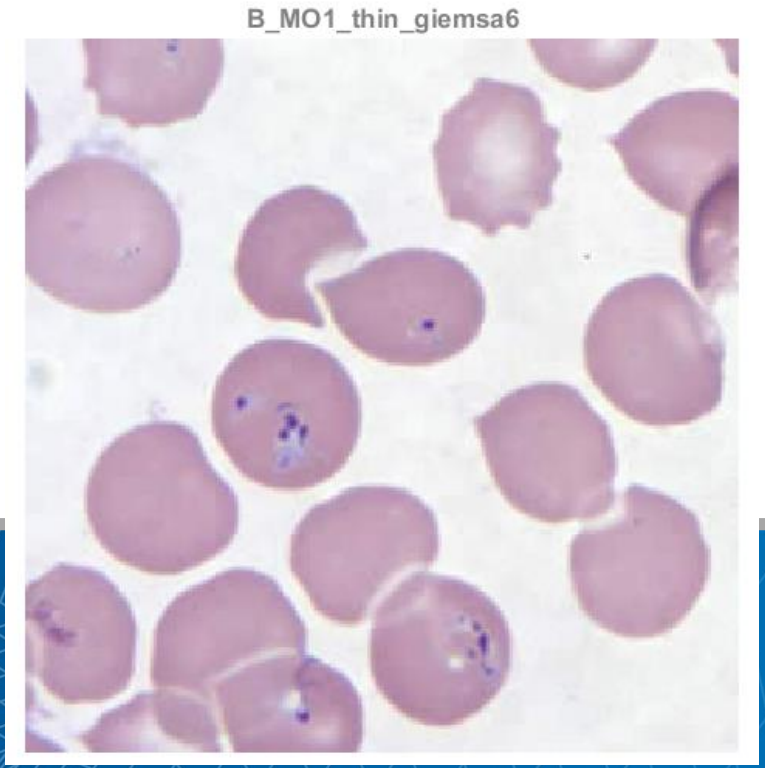
Medical Image Processing using MATLAB



Brett Shoelson, PhD — **Principal Application Engineer**
Bill Wass — **Senior Account Manager**
Adam Rogers

Session Agenda:

Medical Image Processing in MATLAB

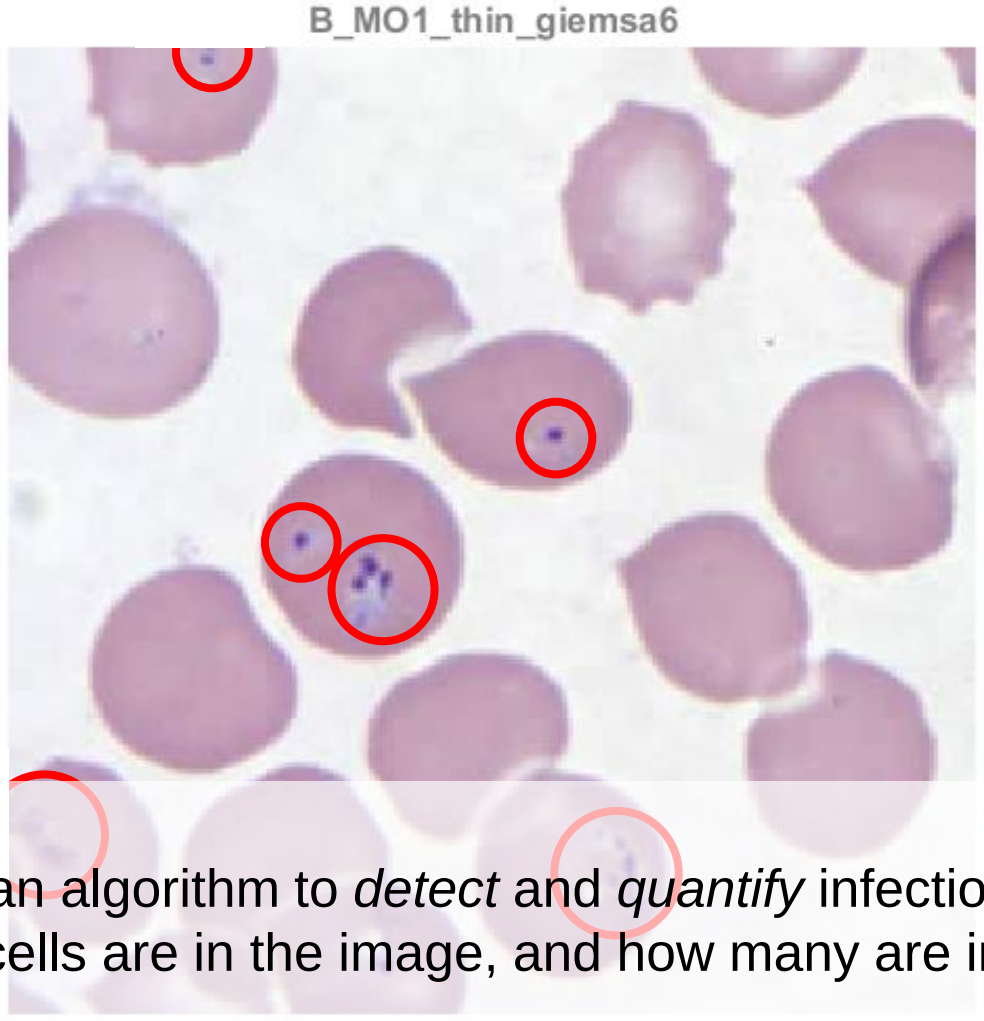
This demonstration will be particularly valuable for anyone interested in using MATLAB to process, visualize, and quantify biomedical imagery. Rather than focus on extracting information from a few homogeneous images, we will introduce a typical real-world challenge, and discuss approaches to managing and exploring collections of widely heterogeneous images. We will describe user interfaces that simplify the exploration and algorithm development processes, and demonstrate their utility in identifying and quantifying scientific or clinically relevant insights.

We will then focus on the extraction of features from images, and the use of machine learning algorithms to classify images based on their content.

In this presentation, we will:

- Explore and manage a range of real-world image sets
- Solve challenging image processing problems with user interfaces
- Develop familiarity with simple to advanced image segmentation approaches
- Classify parasitic infections using machine learning techniques

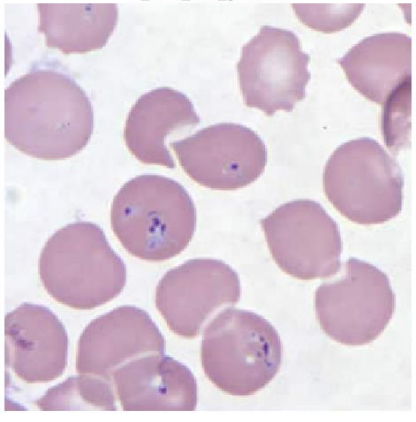
Consider this image from the Centers for Disease Control:



- Our goal:
  To develop an algorithm to *detect* and *quantify* infection.
  How many cells are in the image, and how many are infected?

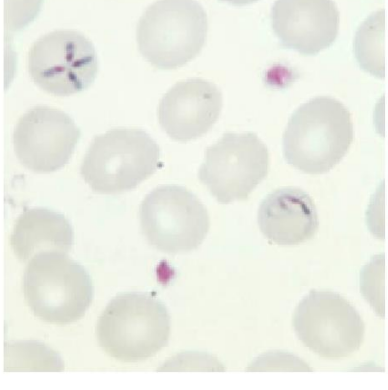
Quantifying infection across multiple images…

B_MO1_thin_giemsa6

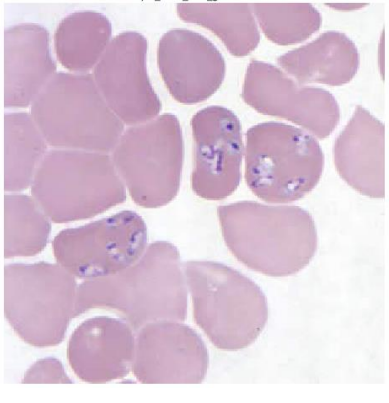


…Despite widely varying image quality

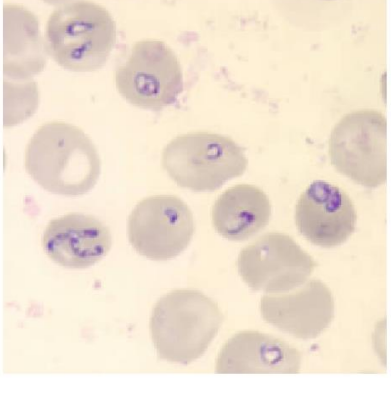
15) Babesia_thin_giemsa_tetrad3



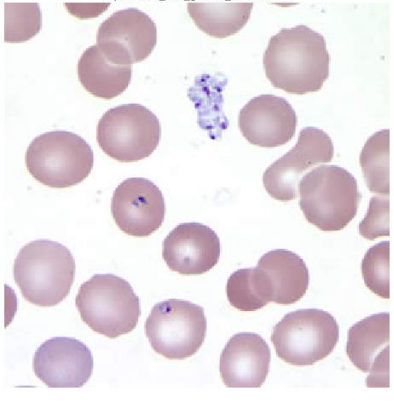
3) B_MO1_thin_giemsa5



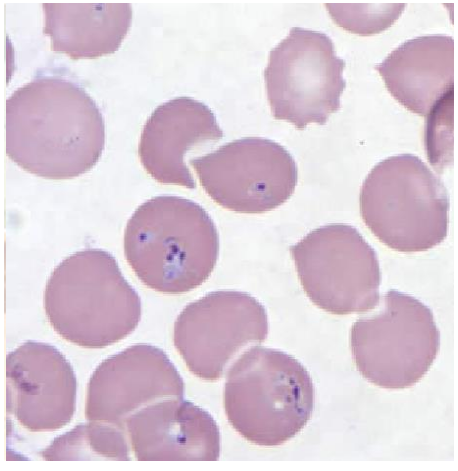
8) Babesia_thin_giemsa3



10) Babesia_thin_giemsa_extracellular



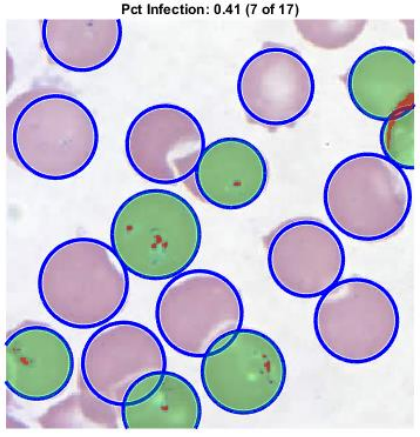
Identify key challenges, consider strategies:



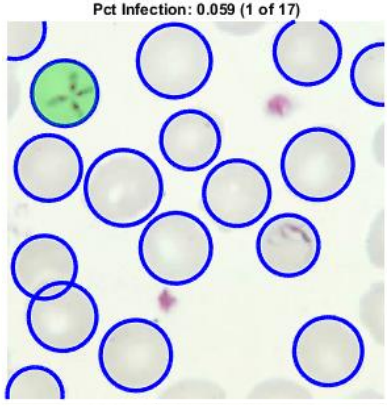
- **Challenges**:
  - Differences in color
  - Differences in illumination
  - Contiguity of cells
  - Low resolution/poor quality
- **Strategies**:
  - Using apps to explore images
  - Pre-processing
  - Watershed segmentation
  - Morphological segmentation

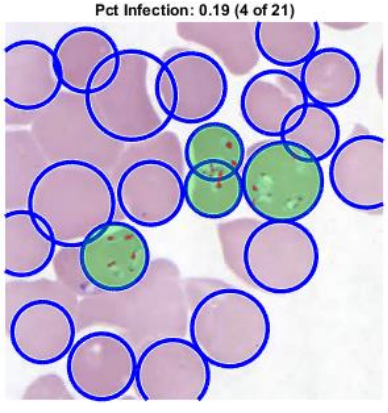
DEMO

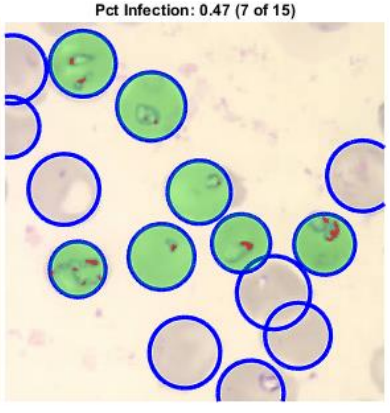
In this session…

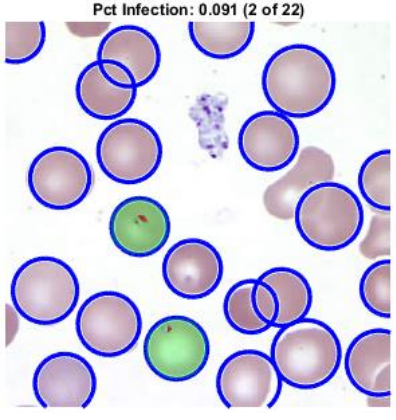


…we quantified rates of infection in heterogeneous images

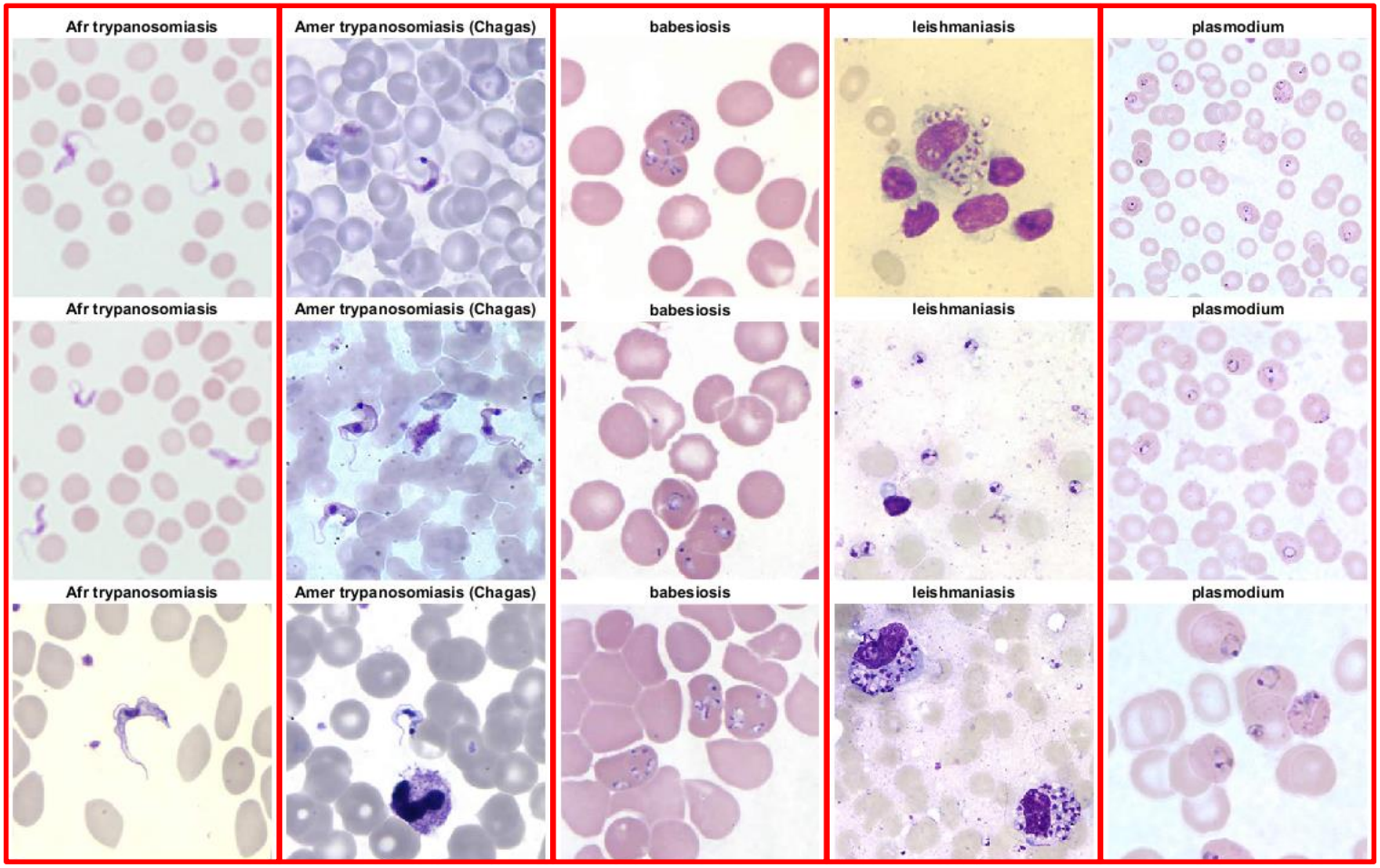






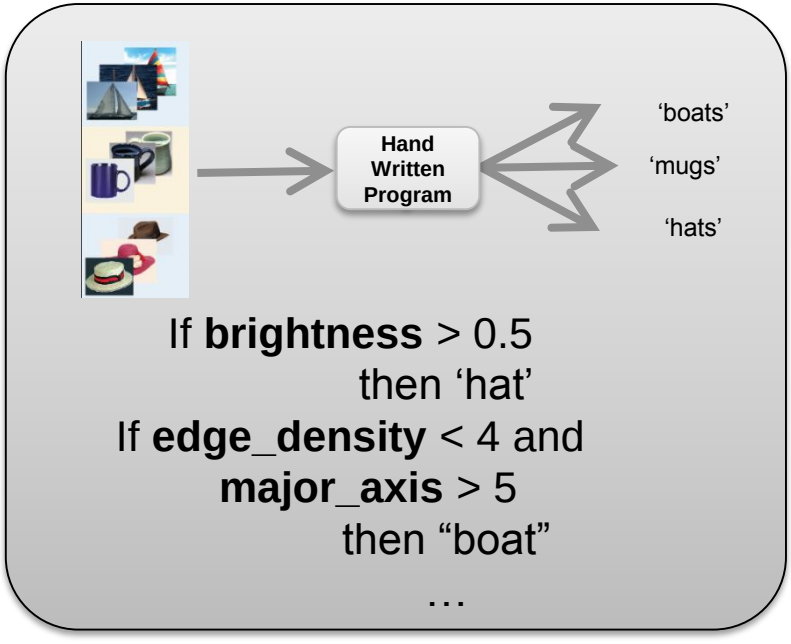


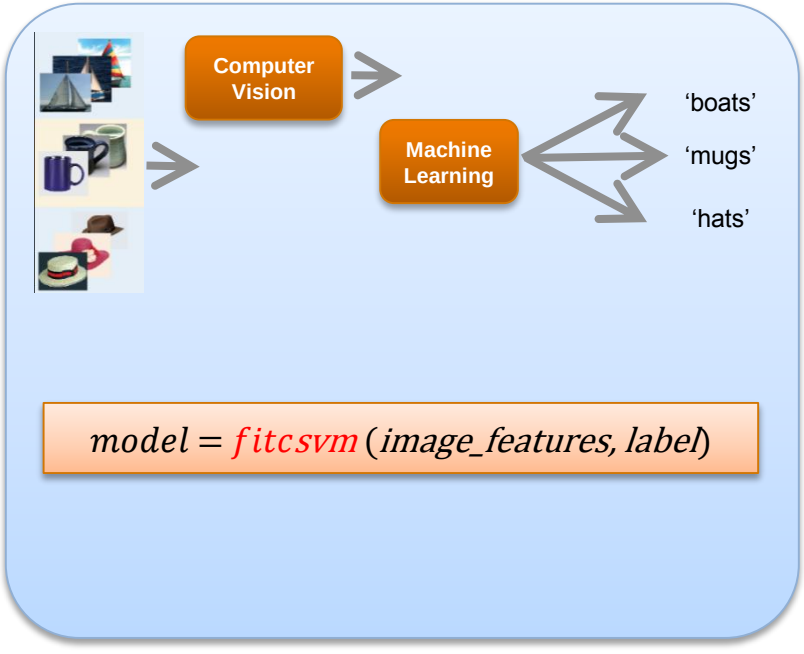
What if we wanted to classify the *type of infection*, differentiating several species of parasites?



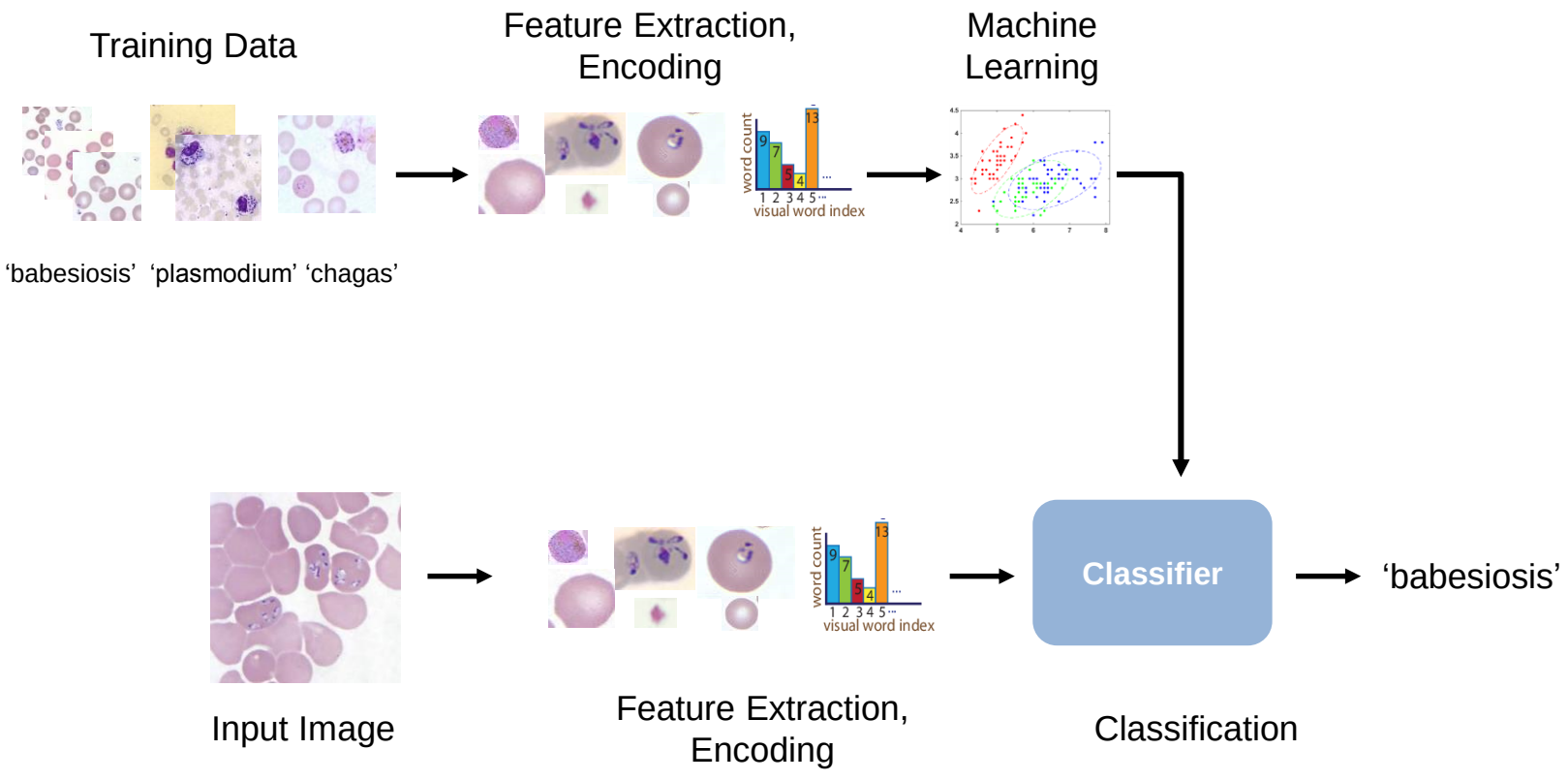
Machine Learning

A machine learning algorithm takes examples of inputs and outputs associated with a task and produces a program that can automatically differentiate them.





Machine Learning Workflow Using Images



Bag of Words

Image Processing Toolbox

Perform **image processing**, **analysis**, and algorithm development

Image Processing Toolbox™ provides a comprehensive set of reference-standard algorithms, functions, and apps for **image processing**, **analysis**, visualization, and algorithm development. You can perform **image analysis**, **image** segmentation, **image enhancement**, noise reduction, geometric transformations, and **image** registration. Many toolbox functions support multicore processors, GPUs, and C-code generation.

Image Processing Toolbox supports a diverse set of **image** types, including high dynamic range, gigapixel resolution, embedded ICC profile, and tomographic. Visualization functions and apps let you explore **images** and videos, examine a region of **pixels**, adjust color and contrast, create contours or histograms, and manipulate regions of interest (ROIs). The toolbox supports workflows for **processing**, displaying, and navigating large **images**.

Bag: image processing, analysis , image, pixels, enhancement

Class / Label

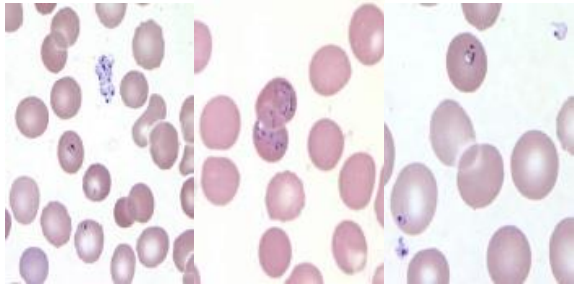
Training Data

Vocabulary / Bag of Words

Bag of “Visual Words” (features)

‘babesiosis’

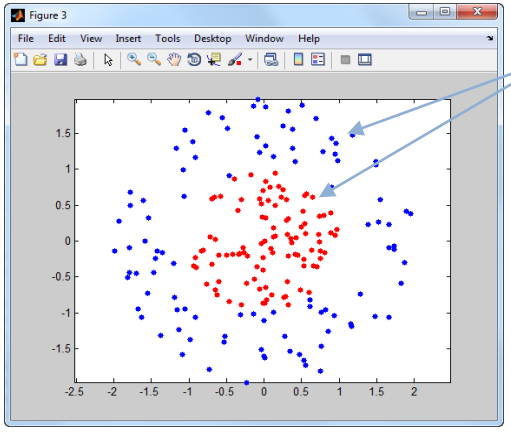
Class / Label



Training Data

Vocabulary / Bag of Features

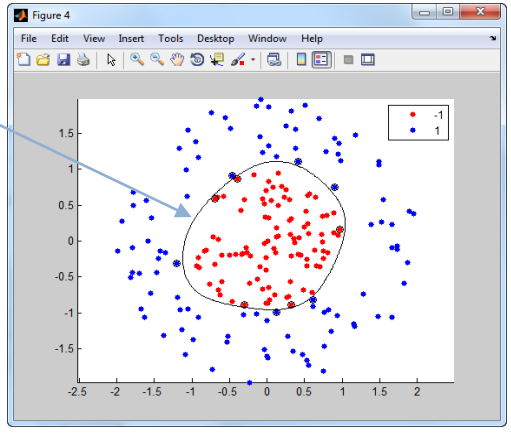
What is a Classifier ?



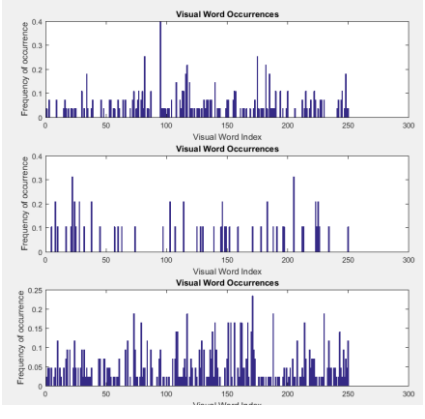
Training Data
Features

Machine Learning

Classifier



Encoded images

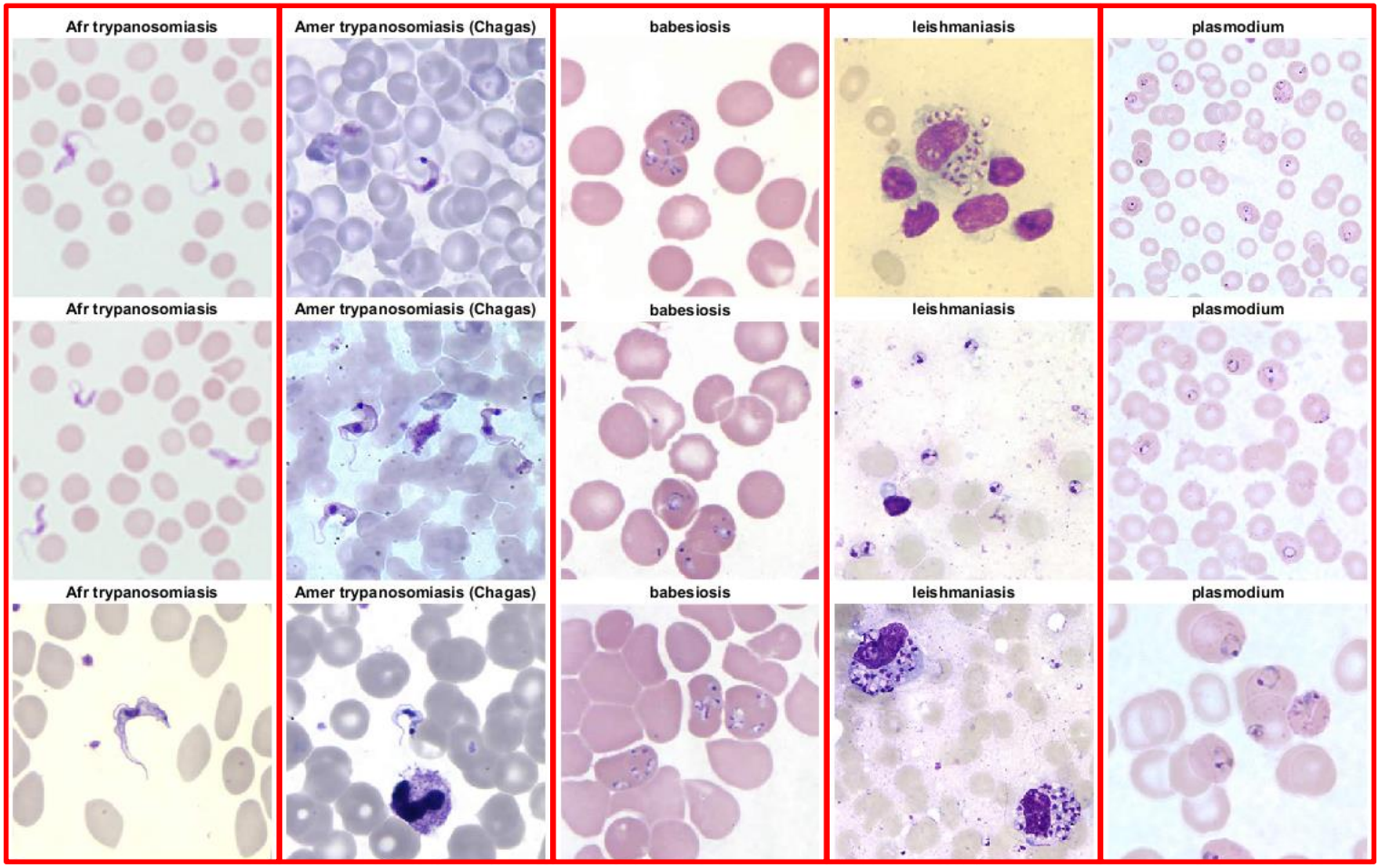


Classification

‘babesiosis’
‘plasmodium’
‘chagas’

Class
Membership

So let's give it a try…



DEMO

Using Machine Learning for Computer Vision

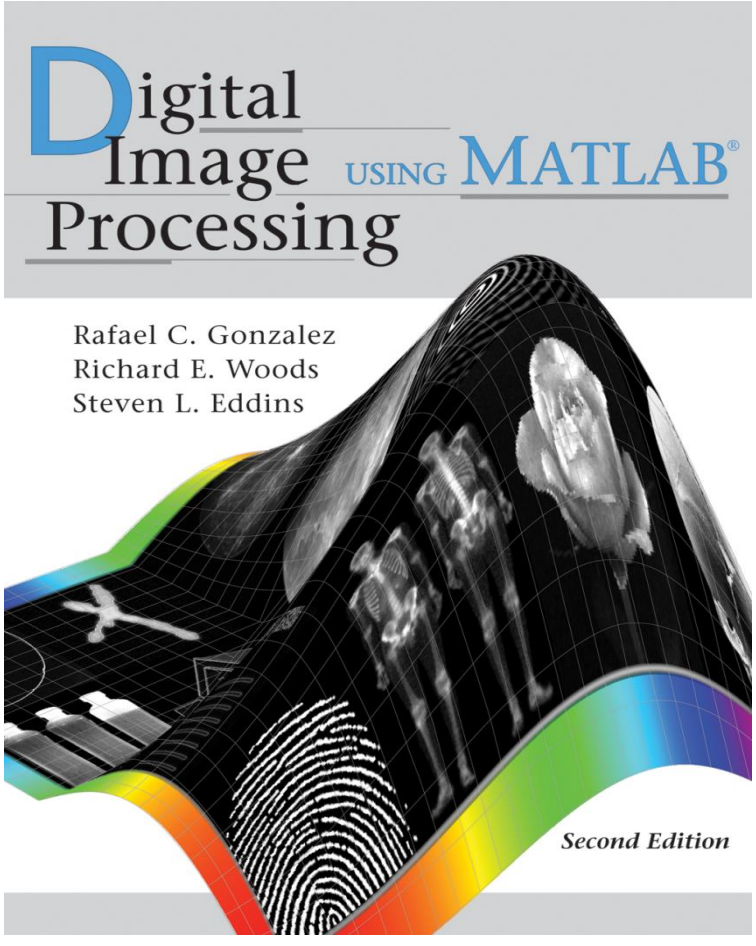
- Computer Vision System Toolbox
  - Provides tools to generate image features for training classifiers
  - See doc for full list of provided image features
- Statistics and Machine Learning Toolbox
  - Provides learning algorithms to train classifiers

Additional Resources

Digital Image Processing Using MATLAB

Gonzalez, Woods, and Eddins

Gatesmark Publishing



Additional Resources

MATLAB Central Blog: “Steve on Image Processing”

http://blogs.mathworks.com/steve/

MATLAB CENTRAL
An open exchange for the MATLAB and Simulink user community

Hosted by The MathWorks

Search: MATLAB Central Go

File Exchange | MATLAB Newsgroup | Link Exchange | Blogs | Contest | MathWorks.com

Steve on Image Processing

October 2nd, 2007

Upslope area - influence and dependence maps

In my August 7th post on upslope area, I showed how to construct and solve the flow matrix to determine the upslope area for every pixel in a digital elevation model (DEM). In addition, the flow matrix can be used to compute both the *influence map* and the *dependence map*.

The influence map shows where water starting from a particular pixel will drain. The dependence map shows which uphill pixels drain through a particular pixel. Let's look at how to compute these from the flow matrix.

Recall the form of the flow equation for a particular pixel: The upslope area of a pixel equals 1 plus a weighted fraction of the upslope area of each of its neighbors.

$$A_j = 1 + w_1 A_{n_1} + w_2 A_{n_2} + \ldots + w_8 A_{n_8}$$

The 1 term comes from assuming each pixel contributes an equal volume to the overall flow across the terrain. Think of it as rain falling equally on each pixel. To compute the influence map, then, we just have to modify our set of equations so that the 1 term appears only for the pixel or pixels of interest. Fortunately, that only affects the right-hand side of the equation, not the flow matrix itself. So solving for the influence map a particular pixel looks almost like solving for upslope area for all pixels.

Here's an example from the Milford, Massachusetts DEM that I've used before:

```
s = load('milford_ma_dem');
E = s.Zc;
```

About

Steve Eddins manages the Image & Geospatial development team at The MathWorks and coauthored Digital Image Processing Using MATLAB. He writes here about image processing concepts, algorithm implementations, and MATLAB.

Thank you!