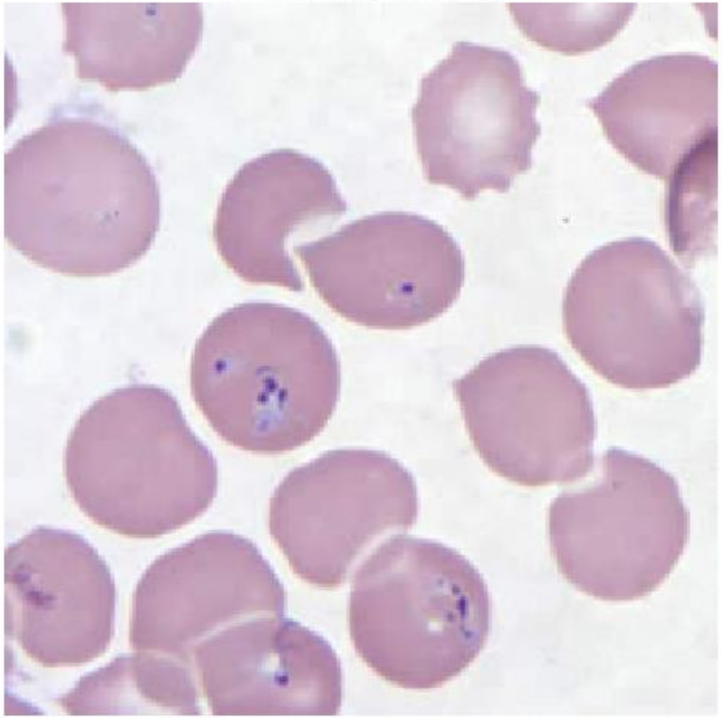
Medical Image Processing using MATLAB

B_MO1_thin_giemsa6



Louvere Walker-Hannon Application Engineer

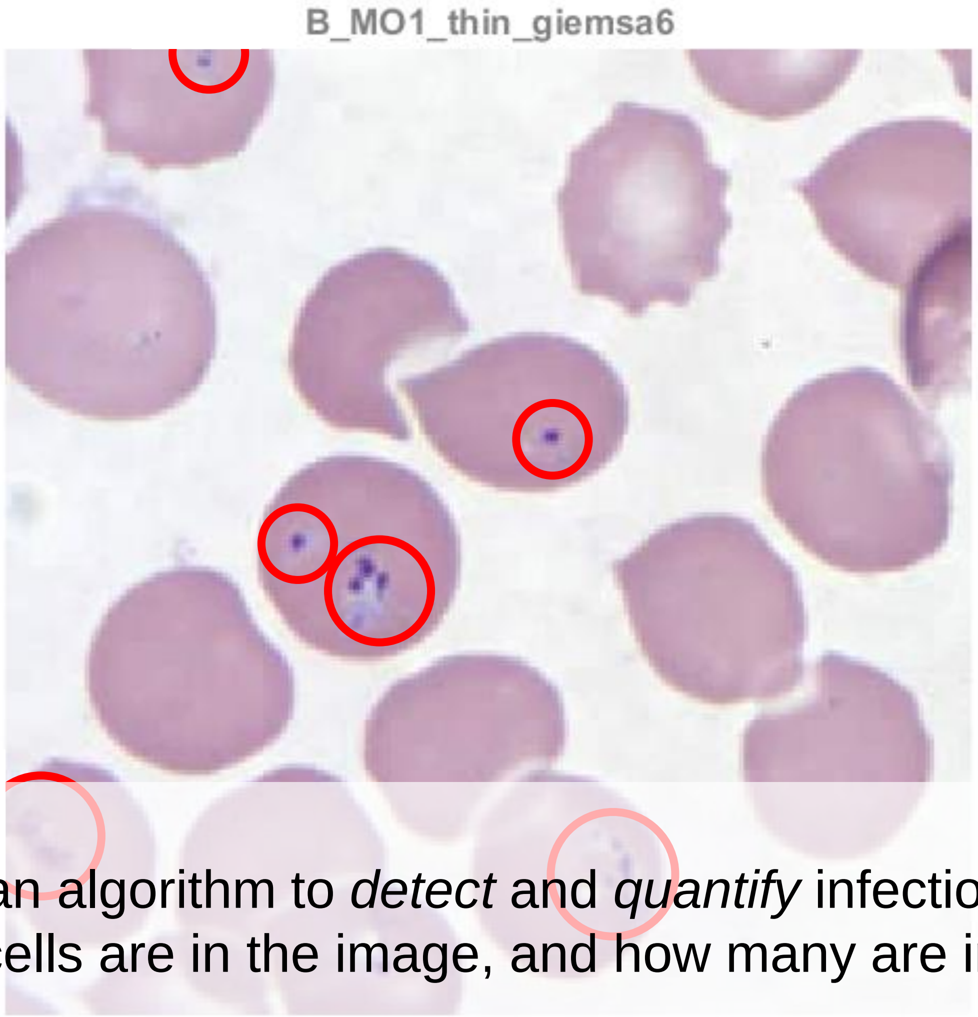
Session Agenda:

Medical Image Processing in MATLAB

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 computer vision and machine learning techniques

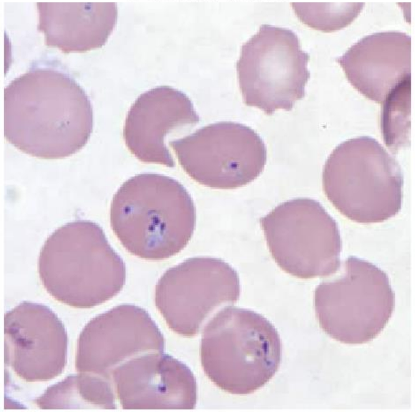
Consider this image from the Centers for Disease Control:



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

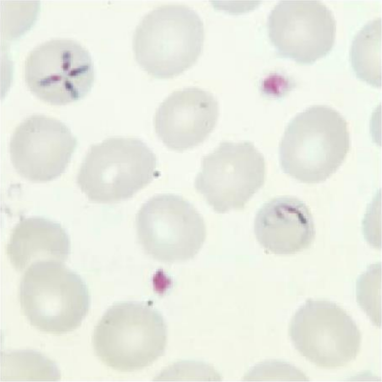
Quantifying infection across multiple images…

B_MO1_thin_giemsa6

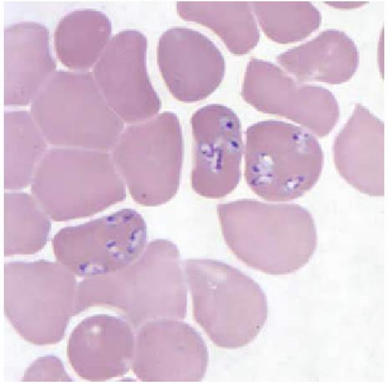


…Despite widely varying image quality

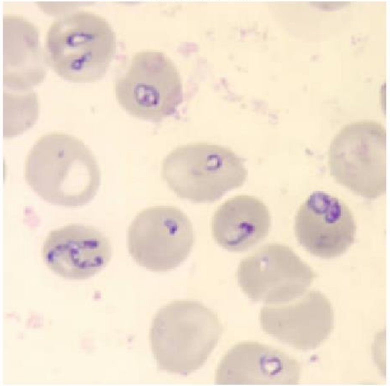
15) Babesia_thin_giemsa_tetrad3



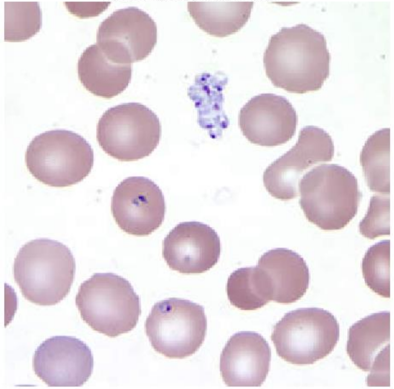
3) B_MO1_thin_giemsa5



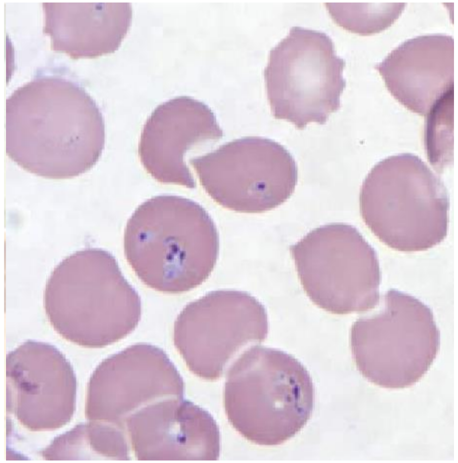
8) Babesia_thin_giemsa3

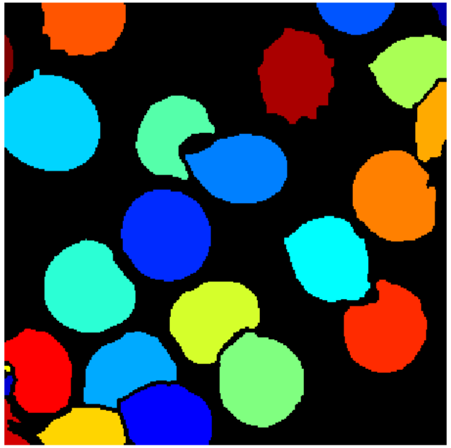


10) Babesia_thin_giemsa_extracellular



Identify key challenges, consider strategies:

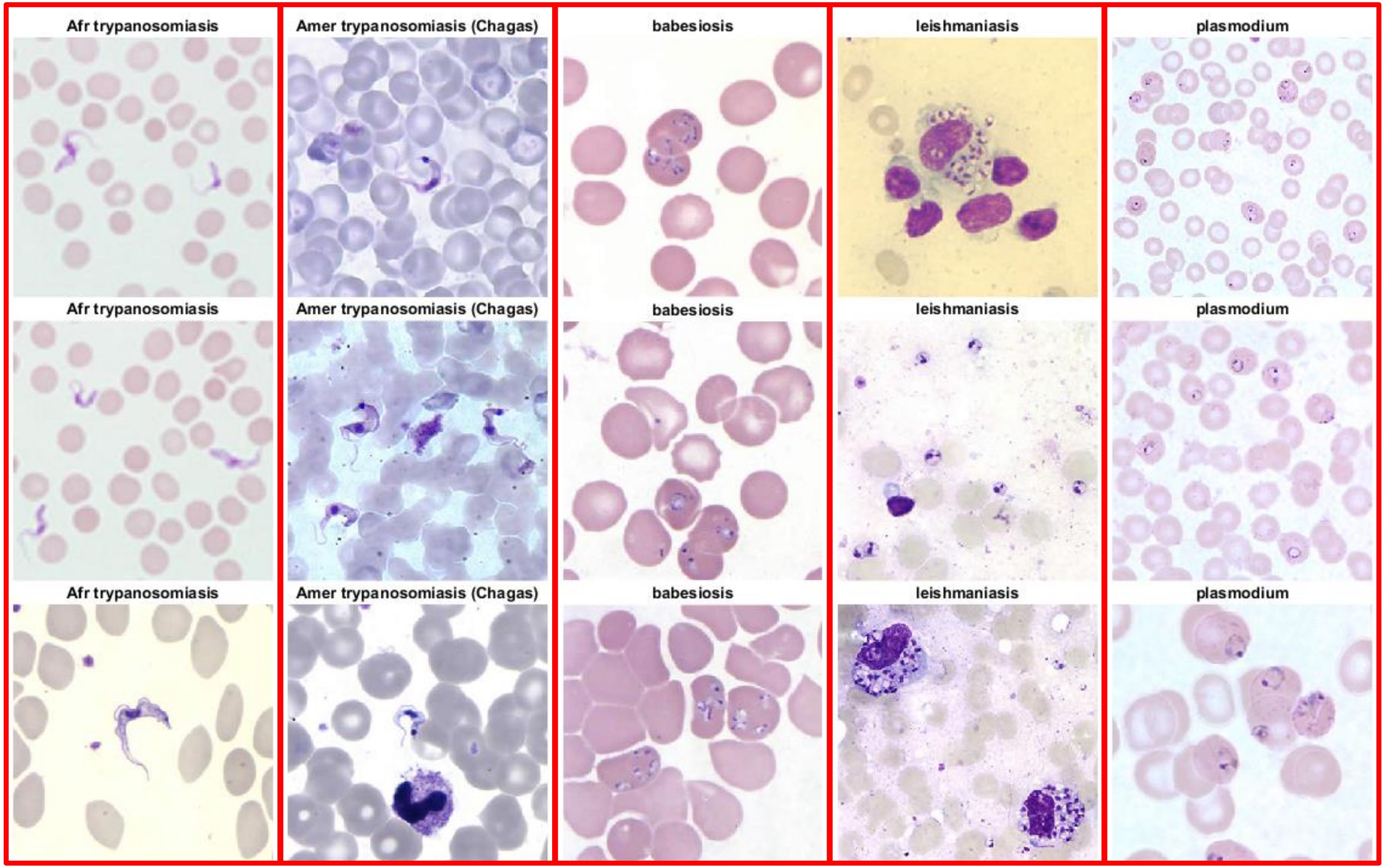




- **Challenges**:
  - Differences in color
  - Differences in illumination
  - Contiguity of cells
  - Low resolution/poor quality
- **Strategies**:
  - Using apps to explore images
  - Pre-processing
  - Watershed segmentation
  - Morphological segmentation

DEMO

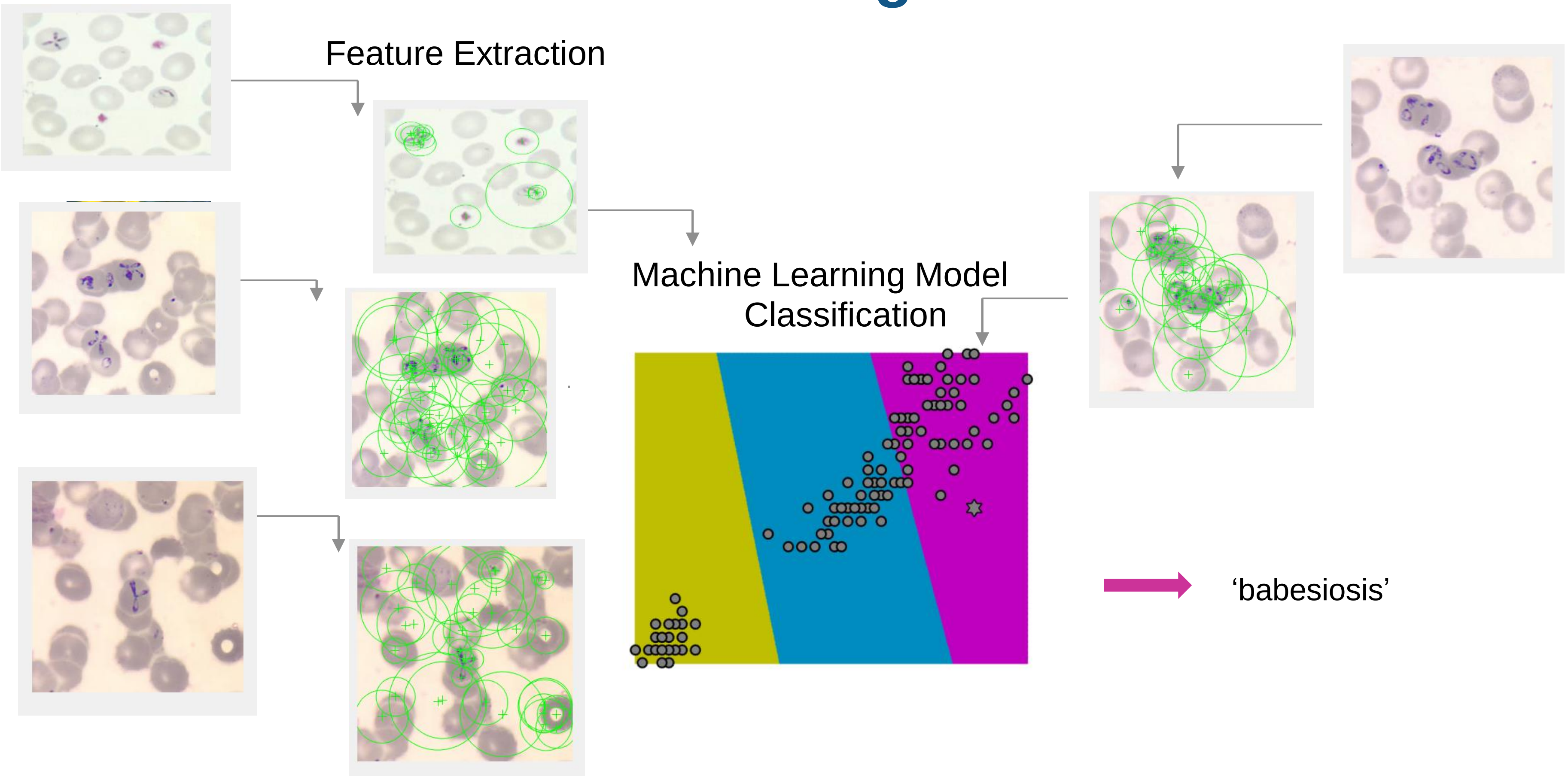
What if we wanted to classify the *type of infection*, differentiating several species of parasites?



Machine Learning Workflow

Training Data

Test Data

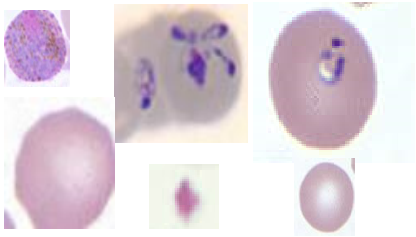


Feature Extraction

Machine Learning Model
Classification

‘babesiosis’

What is Feature Extraction ?



Feature Extraction

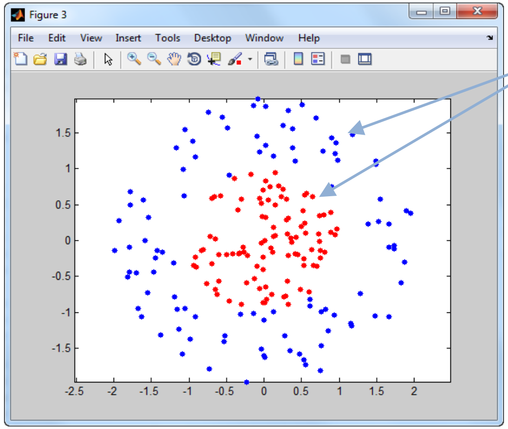
- Representations often **invariant to** changes in scale, rotation, illumination
- **More compact** than storing pixel data
- Feature selection based on nature of problem

Methods for obtaining features

- Image Pixels
- HOG
- Surf
- Bag of Words

Dense to Sparse

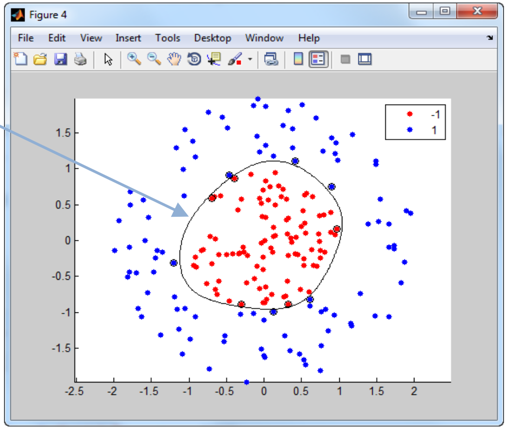
What is a Classifier ?



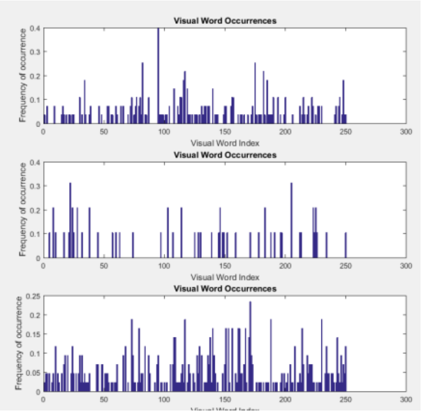
Training Data Features

Machine Learning

Classifier



Encoded images



Classification

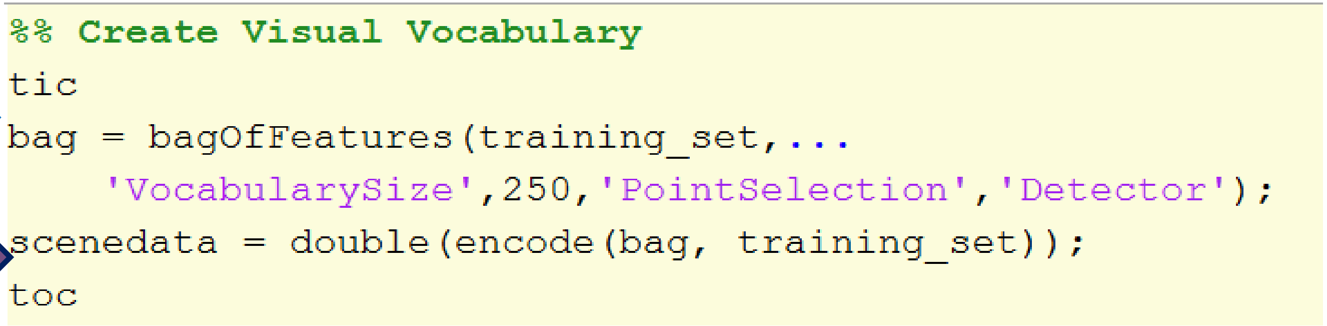
‘babesiosis’
‘plasmodium’
‘chagas’

Class Membership

Bag of Visual Words

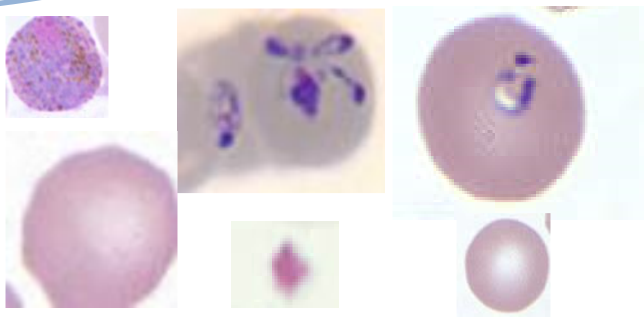
Define the words

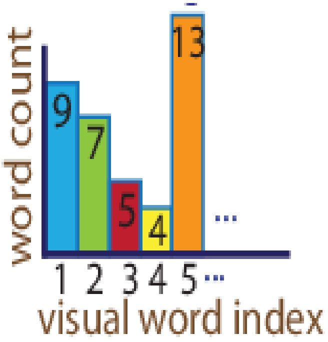
occurrence



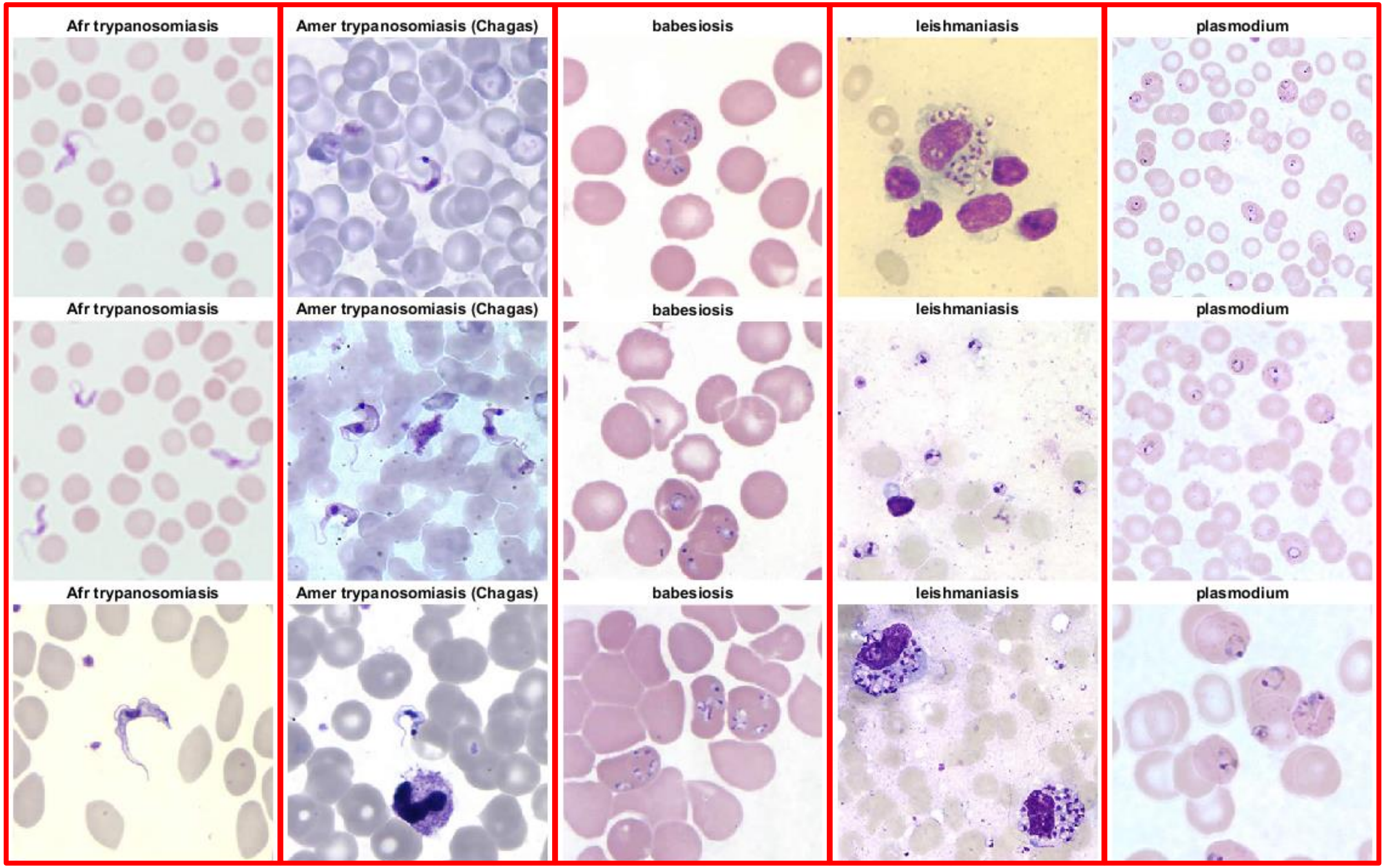
```
%% Create Visual Vocabulary
tic
bag = bagOfFeatures(training_set,...
    'VocabularySize',250,'PointSelection','Detector');
scenedata = double(encode(bag, training_set));
toc
```

Frequency of words describes the scene





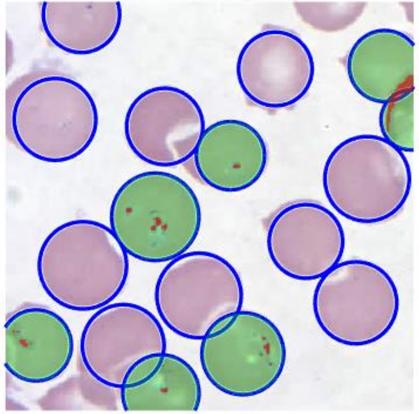
So let’s give it a try…



DEMO

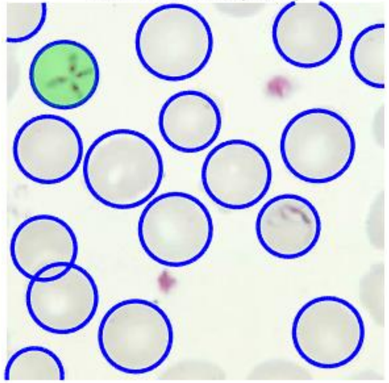
In this session…

Pct Infection: 0.41 (7 of 17)

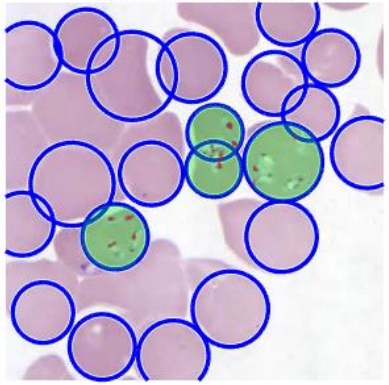


…we quantified rates of infection in heterogeneous images

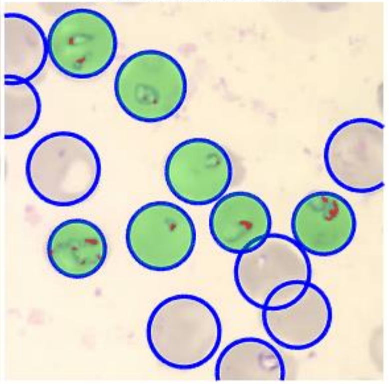
Pct Infection: 0.059 (1 of 17)



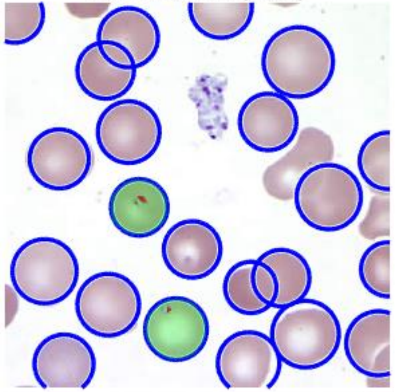
Pct Infection: 0.19 (4 of 21)



Pct Infection: 0.47 (7 of 15)



Pct Infection: 0.091 (2 of 22)



Using Machine Learning for Computer Vision

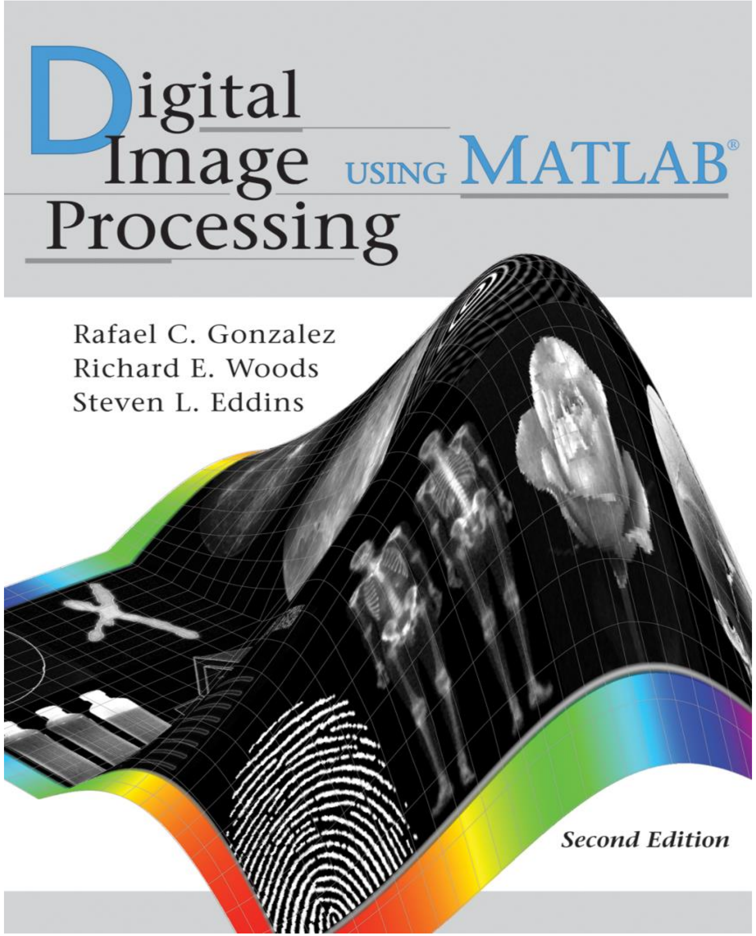
- Image Processing Toolbox
  - Provides 100s of validated functions
  - Indispensable for image processing applications
- Computer Vision System Toolbox
  - Provides tools to generate image features for training classifiers
  - See doc for full list of provided image features
- Statistics and Machine Learning Toolbox
  - Provides learning algorithms to train classifiers

Additional Resources

Digital Image Processing Using MATLAB

Gonzalez, Woods, and Eddins

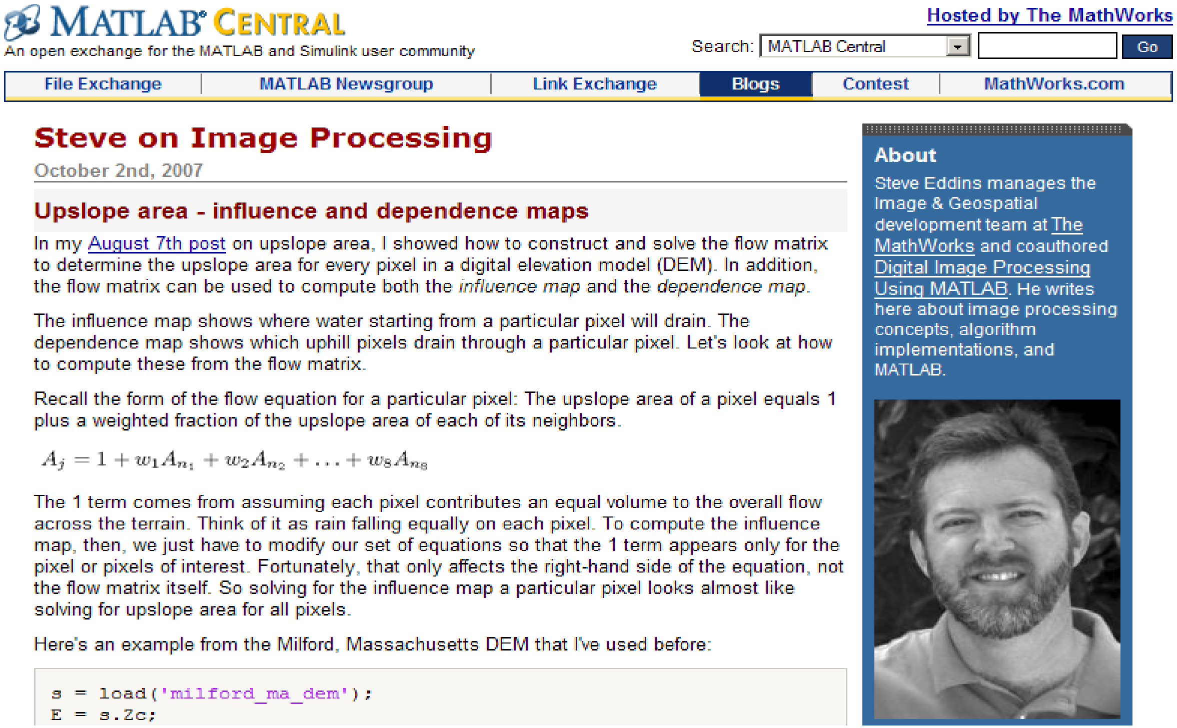
Gatesmark Publishing



Additional Resources

MATLAB Central Blog: “Steve on Image Processing”

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



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.