Automated Recognition of Wasp Species through Image Analysis

Adrian Teng-Amnuay
Taiwan Forestry Research Institute
Taipei, Taiwan

Presentation Layout

- Introduction
  - Background
  - Project description and motivation
  - Designing a solution
    - Related work and available tools
    - Describe the workflow

Presentation Layout

- System implementation
  - (Manual) Image Preprocessing
  - (Automatic) Image preprocessing
  - Feature Extraction
    - Color based features
    - Shape based features
  - Feature reduction
  - Machine learning
    - Model validation
    - Choices for statistical models
    - Model evaluation

Presentation Layout

- Conclusion
  - Results Analysis
  - Unaddressed concerns
  - Future work
  - Discussion

Background

- Adrian Teng-Amnuay
- Undergraduate student from UC San Diego
  - Pursuing B.S. in Computer Engineering (June 2012)
- Before this project
  - One course in Computer Vision
  - One course in Artificial Intelligence
  - Studied Machine Learning with Tony Fountain

Defining the Project

- Image classification of Vespidae Wasps

- Given a set of images, can we determine what species of wasp is in each image?

Motivation

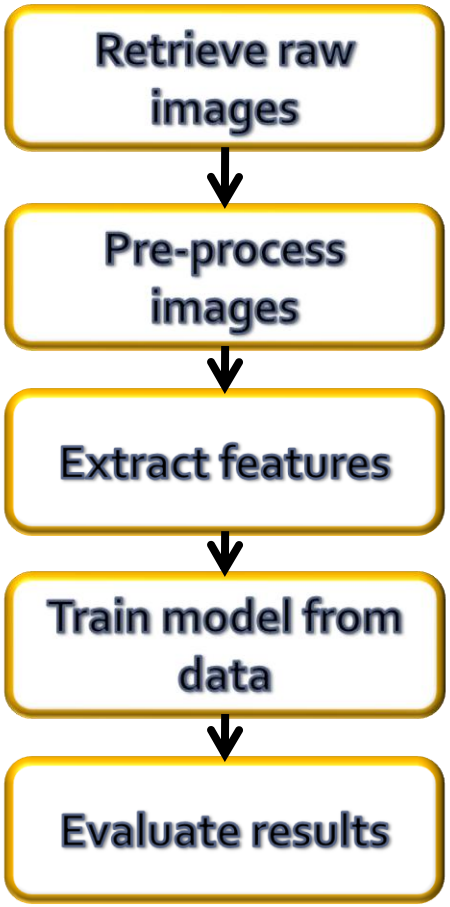
- Why do we want an automated image classification system?

How to accomplish this goal?

- Study of Computer Vision (CV) "eyes" and Machine Learning (ML) "brain"
- Can a machine learn to recognize patterns in image data?
- Classification is just one application of CV/ML
- Many CV and ML algorithms already implemented in the OpenCV library[1]

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000

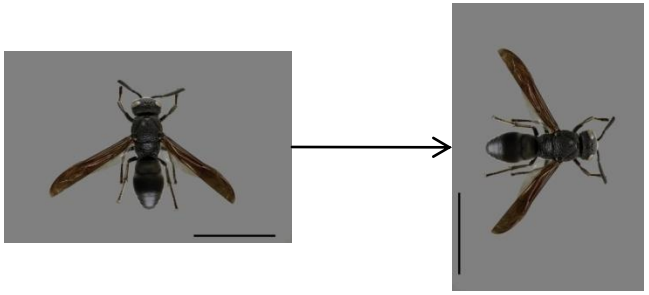
Image classification workflow



- Preprocessing
  - "Garbage in, garbage out"
  - Remove as much irrelevant data as possible
- Feature Extraction
  - How to "describe" an image?
  - Our system focuses on color and shape
- Statistical models of learning
  - How to learn and compare features?
  - How to evaluate performance?
  - Does model generalize to real-world data?

Manual image preprocessing

- Why not automate?
  - Either too hard to automate or easier to do manually
- Image compression
  - Hundreds of high-resolution images (~2 MB each)
- Rotating images
  - For consistency of images



Manual image preprocessing

- Segment out wings, legs, antennae, and knife-blade
  - Too much intraclass variance
  - Difficult to automatically detect and segment

Wings roughly 45˚

Wings closed, body obscured.

Wings wide open

Knife blade protruding

Manual image preprocessing

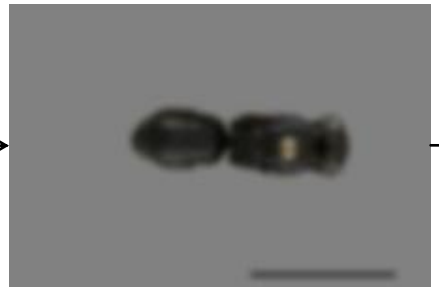
- Preserved wasp bodies



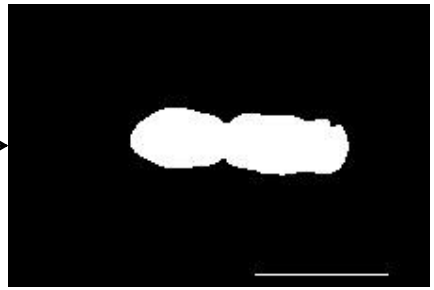
- Use Photoshop to remove unwanted parts

Automatic image preprocessing

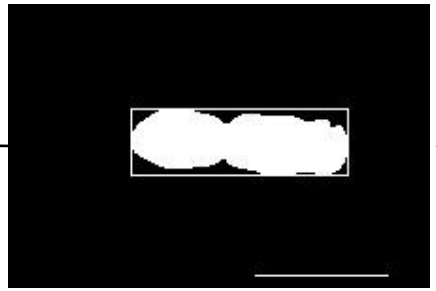
First down-sample image



Gaussian blur



Create binary image



Fit bounding rectangle

Crop region of interest
and normalize size

Feature Representation (example)

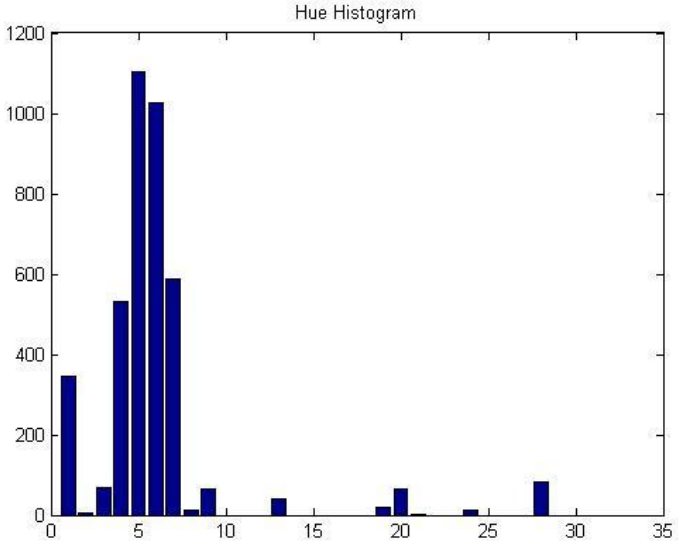
- Weight
- Length

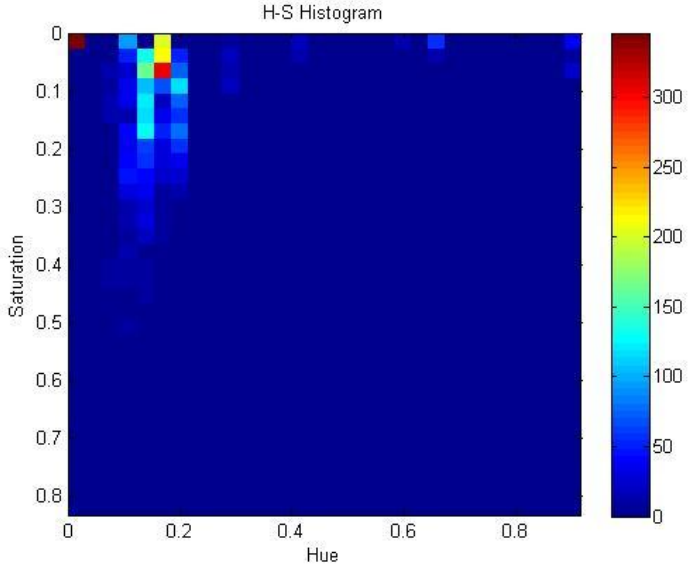
Feature Extraction

- How to describe color?
- Color Histograms
  - Compute 2D Hue-Saturation histograms
  - Disregard "Value" to remove sensitivity to lighting

Color Histograms

- Examples 1D and 2D histograms





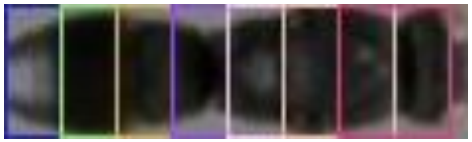
Together, hue and Saturation provide a good "description" of color

Overlapping blocks of Color Histograms

- Color histograms place pixels from the entire image into bins
  - No spatial information

- To capture spatial information about color, we take blocks from different regions of the image and calculate color histograms for each block.

- We can vary the block sizes, and allow the blocks to overlap to improve performance.

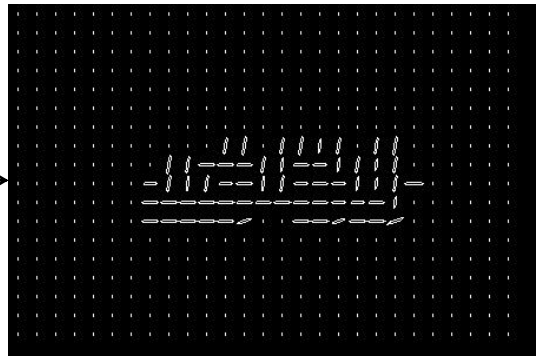


Feature Extraction

- How to describe shape?
- Histogram of Oriented Gradients (HOG)[2]
  - Robust shape descriptors created for pedestrian detection in images
  - Already implemented by OpenCV

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR, pages I: 886–893, 2005.*

Histogram of Oriented Gradients



For the sake brevity, algorithm details are not covered

HOG features are explained in detail in the paper by N. Dalal and B. Triggs [2]

Cells of grouped pixels, represented by magnitude and orientation

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR, pages I: 886–893, 2005.*

Feature Reduction

- Color and HOG feature vectors contain thousands of features (over 10,000 with the current implementation)
- Curse of dimensionality
  - Feature vectors become difficult to compare
  - Computational overhead grows exponentially with number of features
- Principle Component Analysis (PCA)
  - A way to reduce feature dimensionality
  - Principle components only need to be calculated during training (takes ~30 seconds on Intel® Core™2 Duo Processor T5550 (1.83 GHz))

Choosing a model

- Many different ML models
  - Nearest Neighbor
    - K-nearest neighbor
  - Decision Trees
  - Support Vector Machines
    - Binary classifier
    - Multiclass SVM (one-against-one, one-against-rest)
  - Many more, but only had time to test out a handful of models

Model Validation

- Need a way to evaluate model performance
- Holdout validation
- K fold cross-validation (10)
- Confusion Matrices
  - How often is a class correctly predicted?
  - How often is a class confused for another class?

Evaluating the model

- One-against-rest SVMs using overlapping blocks of color histograms
- Validation accuracy in percentage ± standard deviation across 10 folds
- (number correctly identified / total number of images)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	# of Test Images	Accuracy
0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1
1	0	12	0	1	0	0	0	0	0	0	1	0	0	0	0	0	14	0.86
2	1	0	38	0	0	0	0	1	0	0	2	0	0	1	0	0	43	0.88
3	0	0	0	47	0	0	0	0	0	0	0	0	0	0	0	1	48	0.98
4	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	40	1
5	0	0	0	0	0	18	1	0	0	0	0	0	0	0	0	0	19	0.95
6	0	0	0	0	0	1	24	0	0	0	0	0	0	0	0	0	25	0.96
7	0	0	2	1	0	0	0	16	0	0	1	0	0	1	0	0	21	0.76
8	0	0	0	1	0	1	0	0	9	0	0	0	0	0	0	0	11	0.82
9	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	21	1
10	0	0	1	0	0	0	0	0	0	0	7	0	0	1	0	1	10	0.7
11	0	0	0	0	0	0	0	0	0	0	0	33	1	0	0	0	34	0.97
12	0	0	0	0	0	0	0	0	1	0	0	3	31	0	0	0	35	0.89
13	0	0	2	0	0	0	0	0	0	0	0	0	0	44	0	0	46	0.96
14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	42	1	44	0.95
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	11	13	0.85

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	# of Test Images	Accuracy
0	10	0	0	0	0	0	0	1	0	0	0	0	0	2	0	0	13	0.77
1	0	12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	14	0.86
2	0	0	38	0	0	0	0	4	0	0	1	0	0	0	0	0	43	0.88
3	0	0	2	44	0	0	0	1	0	1	0	0	0	0	0	0	48	0.92
4	0	0	0	0	37	0	0	0	0	0	0	1	0	2	0	0	40	0.93
5	0	0	0	0	0	17	2	0	0	0	0	0	0	0	0	0	19	0.89
6	0	0	0	0	2	3	19	0	1	0	0	0	0	0	0	0	25	0.76
7	0	0	5	0	0	0	0	13	0	0	1	0	0	1	1	0	21	0.62
8	0	0	1	0	0	1	1	0	6	0	1	1	0	0	0	0	11	0.55
9	0	0	0	1	0	0	1	0	0	19	0	0	0	0	0	0	21	0.9
10	1	1	0	0	0	0	0	1	0	0	7	0	0	0	0	0	10	0.7
11	0	0	0	0	0	0	0	0	0	0	0	33	1	0	0	0	34	0.97
12	0	0	0	0	2	1	0	0	0	0	0	1	31	0	0	0	35	0.89
13	1	0	3	0	0	0	0	1	0	0	0	0	0	41	0	0	46	0.89
14	0	0	0	0	0	0	1	0	0	1	0	0	1	0	41	0	44	0.93
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	10	13	0.77

Color features:
92.906 ± 4.534 %
(406/437)

H0G features:
86.499 ± 4.843 %
(378/437)

Combining color and shape features

- Instead of only predicting the class label for a given example, return probabilistic output for each class
  - e.g. "an_image.jpg" is 98% likely to be "Allorhynchium argentatum," 0.1% likely to be "Apodynerus f.formosensis," 0.3% likely to be … and so on.
- This allows us to combine shape and color features in a naïve fashion

Evaluating the model

- One-against-rest SVMs using both color blocks and HOG features
- Validation accuracy in percentage ± standard deviation across 10 folds
- (number correctly identified / total number of images)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	# of Test Images	Accuracy
0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1
1	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	1
2	0	0	41	0	0	0	0	1	0	0	1	0	0	0	0	0	43	0.95
3	0	0	0	47	0	0	0	0	0	0	0	0	0	0	0	1	48	0.98
4	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	40	1
5	0	0	0	0	0	18	1	0	0	0	0	0	0	0	0	0	19	0.95
6	0	0	0	0	0	2	23	0	0	0	0	0	0	0	0	0	25	0.92
7	0	0	2	1	0	0	0	17	0	0	1	0	0	0	0	0	21	0.81
8	0	0	1	0	0	0	0	0	10	0	0	0	0	0	0	0	11	0.91
9	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	21	1
10	0	0	1	0	0	0	0	0	0	0	9	0	0	0	0	0	10	0.9
11	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	34	1
12	0	0	0	0	0	0	0	0	0	0	0	1	34	0	0	0	35	0.97
13	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	46	1
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	42	1	44	0.95
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	11	13	0.85

96.110 ± 1.810 %
(420/437)

Results Analysis

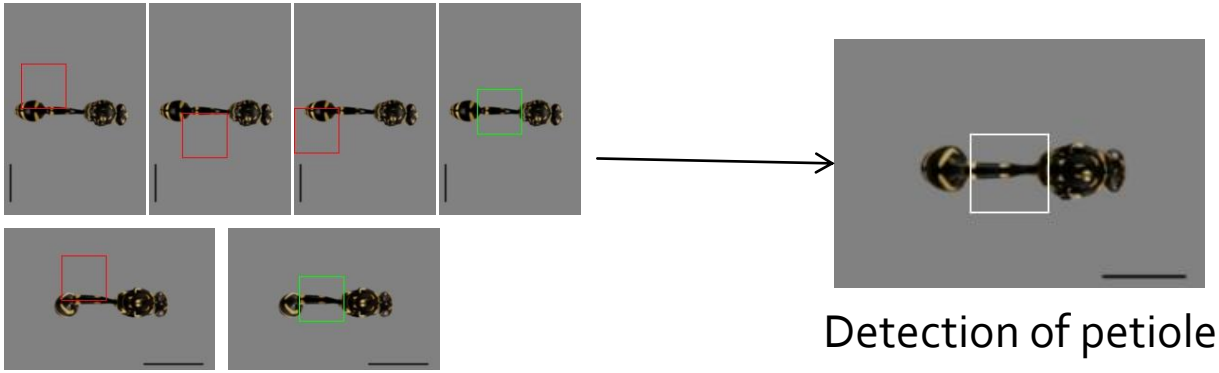
- We have presented a workflow for automated image classification, and achieved an overall performance of ~96%
- Using local color information and HOG descriptors for shape captures much of the interclass variance
- Still room for improvement

Unaddressed concerns

- Tuning parameters
  - Large number of parameters tested and chosen manually, not optimally
    - Number of bins for color histograms
    - HOG parameters
    - Cropped image resizing dimensions
    - SVM parameters
    - Probability output function parameters
- Number of features
  - Shouldn't have 10,000 features
- Scale bar too close to wasp body
  - Leads to bad binary image

Future work

- Automating the entire workflow
- An important step that was simplified in the problem was manual segmentation of wings, legs, etc.
  - To automate, look into part-based detection as described in Week 6 Report



Detection of petiole

  - But train on entire wasp body, focusing on wings as negatives
  - Feed detected "wasp body" into current system

Future work

- Modify current implementation to work with wings
  - Might not be necessary to remove wings
  - Other creative ideas? Tweak system implementation?
- Expand system to include full set of species
  - Did not include several species due to small number of training samples
- Add graphical user interface
  - Include visual display of results (confusion matrices, species labels, etc.)

Discussion

- Any questions?

References

- [1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000
- [2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR, pages I: 886–893, 2005.*

Acknowledgements

- *Mentors*
- Tony Fountain (CALIT2)
- Chau Chin Lin (TFRI)
- Sheng-Shan Lu (TFRI)
- Yu-Huang Wang (TFRI)

- *PRIME coordinators*
- Gabriele Wienhausen
- Teri Simas
- Peter Arzberger
- Tricia Taylor

- *Advice on Computer Vision*
- Serge Belongie (UCSD faculty)