Vision and Machine Learning for Automated Arthropod Biodiversity Studies

Students: N. Larios, H. Deng, W. Zhang, N. Payet, M. Sarpola, C. Fagan, J. Yuen, S. Ruiz Correa
Postdocs: G. Martinez
Faculty: R. Paasch, A. Moldenke, D. A. Lytle, E. Mortensen, L. G. Shapiro, S. Todorovic, T. G. Dietterich

Oregon State University
University of Washington

Automated Rapid-Throughput Arthropod Population Counting

- Population counts of small arthropods are an important source of data for
  - community ecology
  - biodiversity studies
  - biomonitoring of soils, lakes, streams, and oceans
- Manual identification and counting of specimens
  - very time-consuming
  - requires high degree of expertise
  - very few experts in the world
- Goal:
  - technician collects specimens in the field by various means
  - robotic device automatically manipulates, photographs, classifies, and sorts the specimens
- Three applications:
  - stoneflies in freshwater streams
  - soil mesofauna
  - freshwater zooplankton

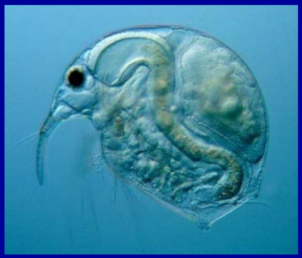
Application 1: Stonefly populations in freshwater streams

- differentially sensitive to many pollutants
- live in rivers; reliable indicator of stream health
- difficult and expensive for people to classify (particularly to genus or species levels)

Application 2: Small arthropods in soil: “soil mesofauna”

Application 3: Freshwater Zooplankton

Daphnia



Bosmina

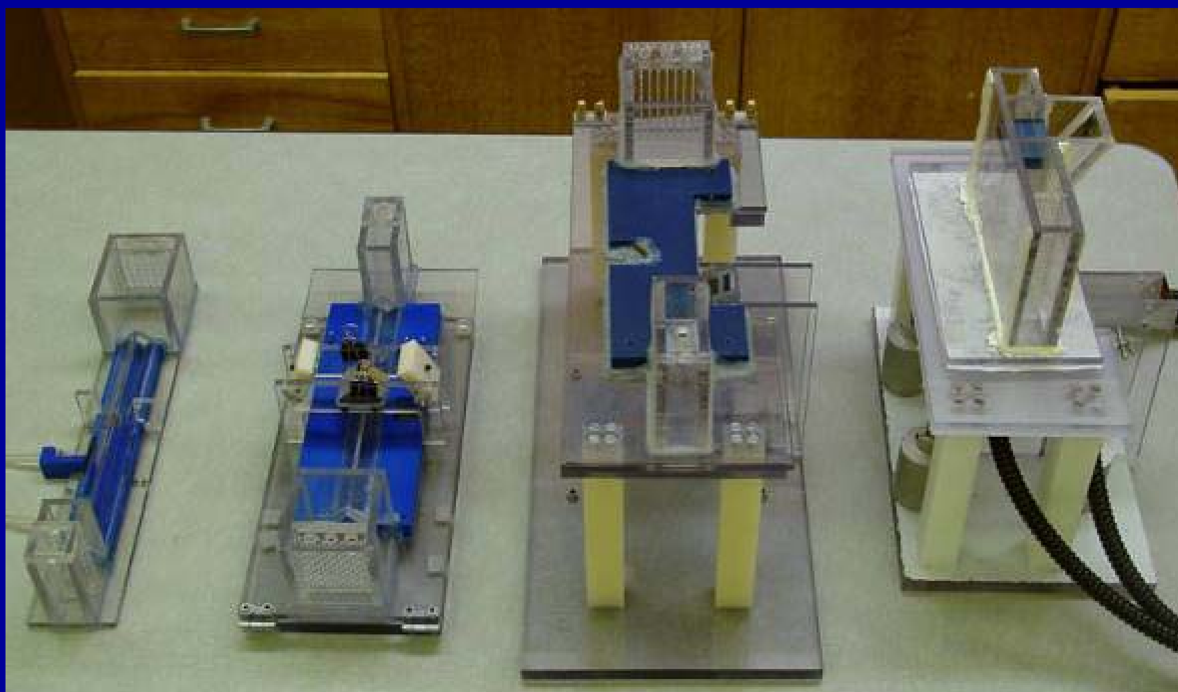
Polyphemus
(cladocerans)

Cyclops
(copepod)

- Measure biodiversity in freshwater lakes
- 70 species
  - 100-1000 specimens per sample

Images from Microscopy-UK.

Image Capture Apparatus



Stonefly Imaging

Soil Mesofauna / Zooplankton Imaging

Robotic Extraction of Specimens

Computer Vision Challenges(1)

- Highly-articulated objects with deformation

Computer Vision Challenges(2)

- Huge intra-class changes of appearances due to development and maturation

tergites become wings

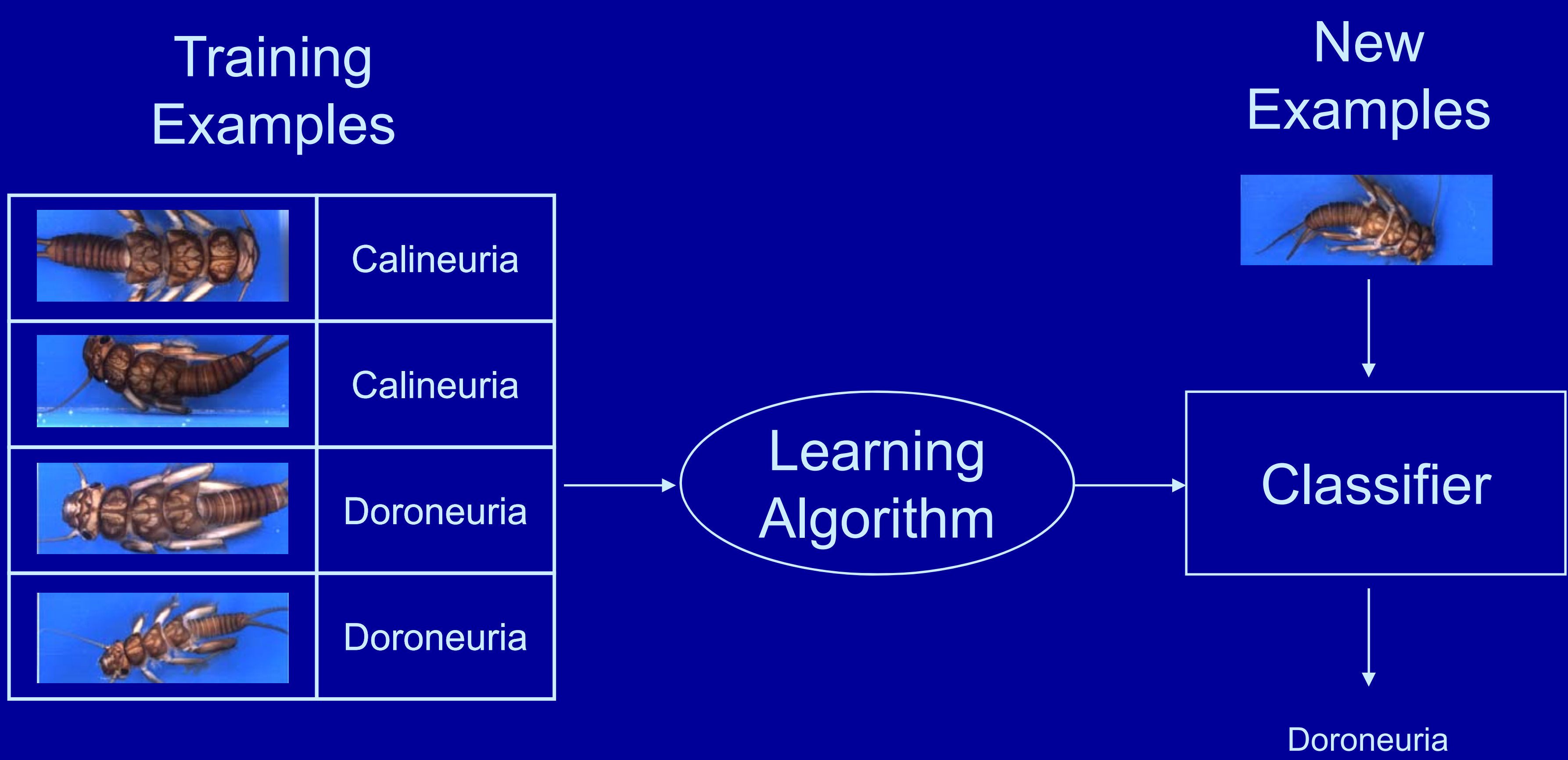
Computer Vision Challenges(3)

- Small between-class differences

Calinueria

Doronueria

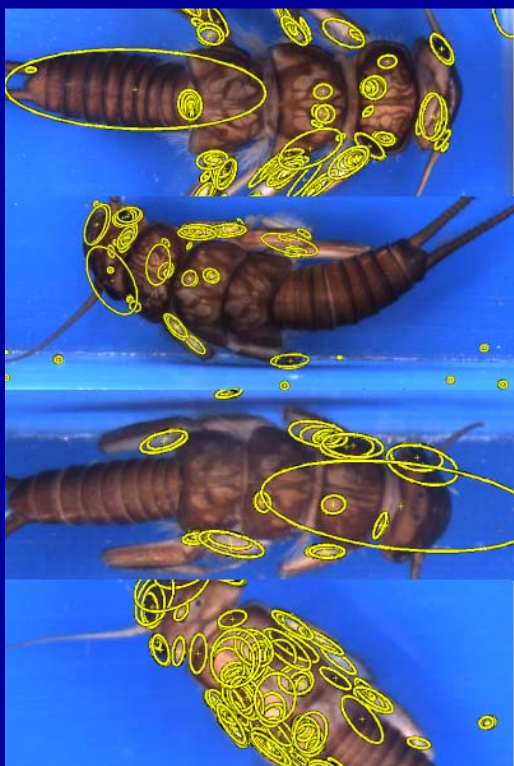
Machine Learning



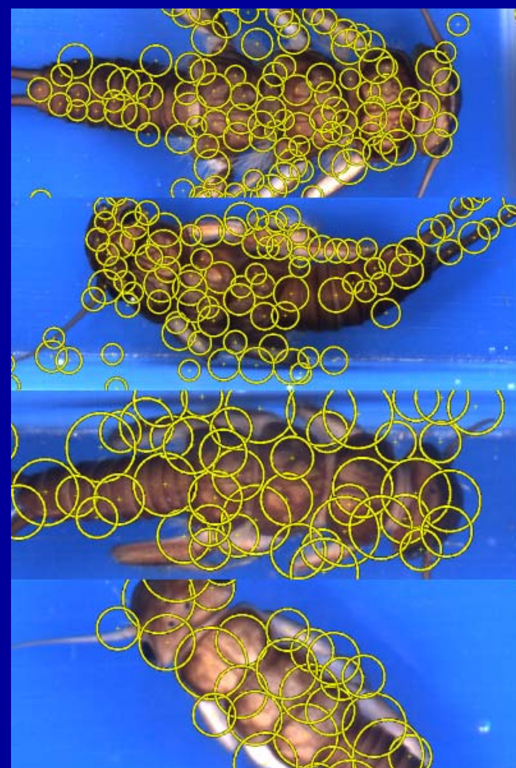
Stonefly Identification System

- Semi-automated specimen handling and photography
- Computer Steps:
  1. Dorsal view detection
  2. Region detection
  3. Region description
  4. Classification

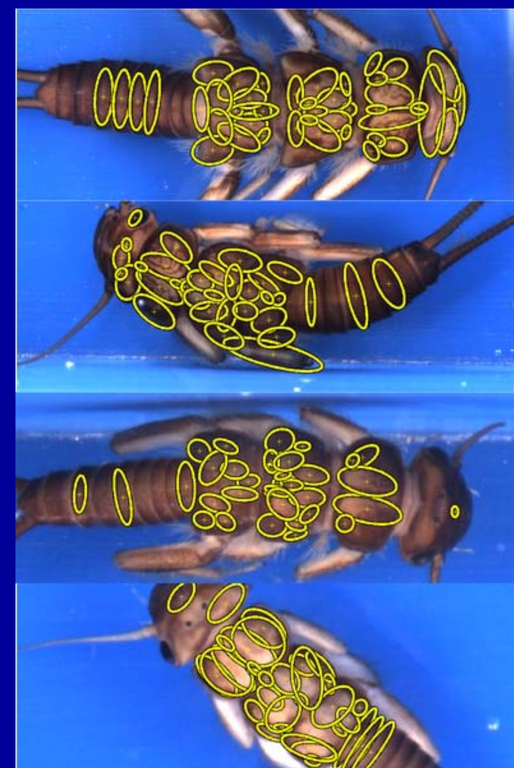
Region Detectors



Hessian-Affine Detector

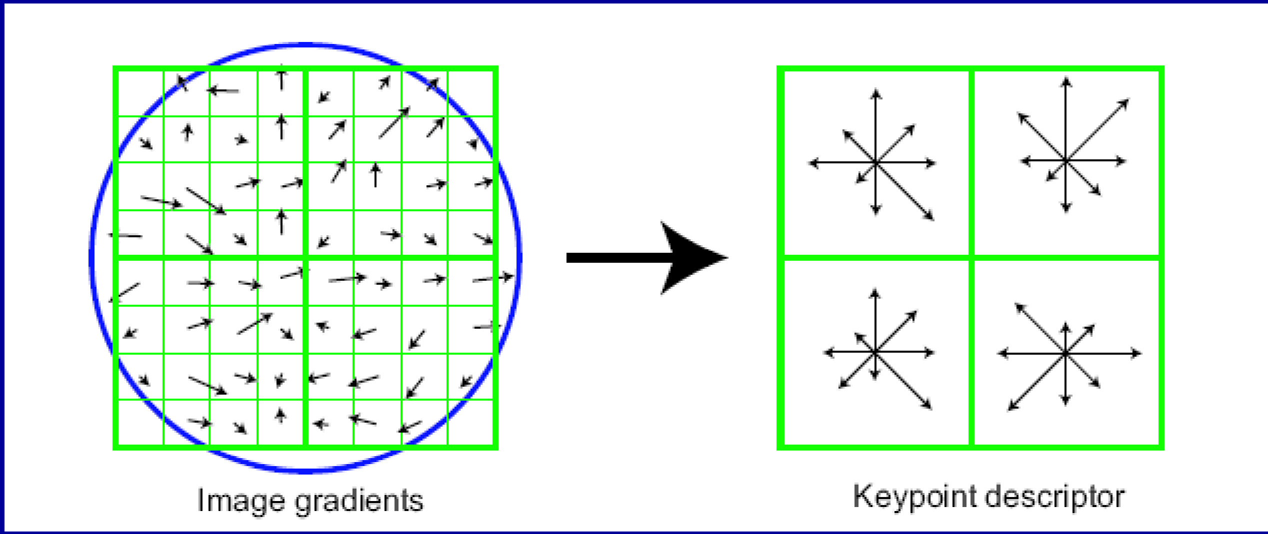


Kadir Entropy Detector



PCBR Detector

Scale-Independent Feature Transform
SIFT (Lowe, 1999)



(Lowe, 1999)

- Compute intensity gradient at each pixel in 16x16 region
- Weight them by a Gaussian distribution (indicated by circle)
- Collect into histograms within each 4x4 region (gives 16 histograms)
- Result: 128-element vector normalized to have Euclidean norm 1

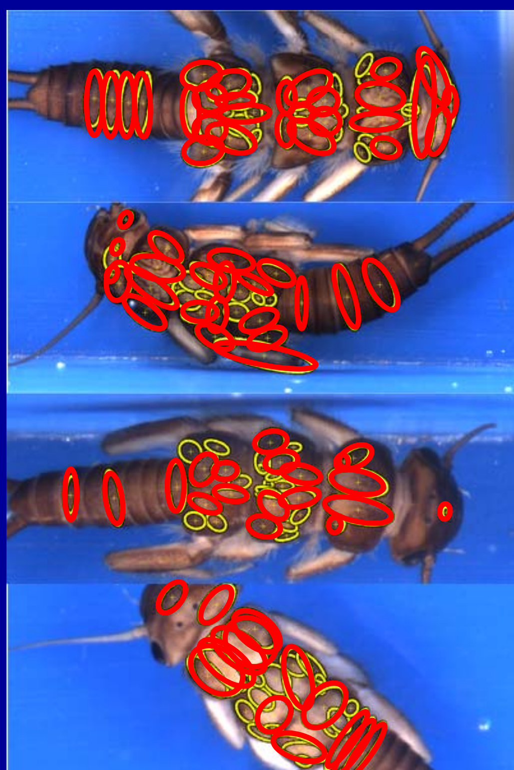
Classification

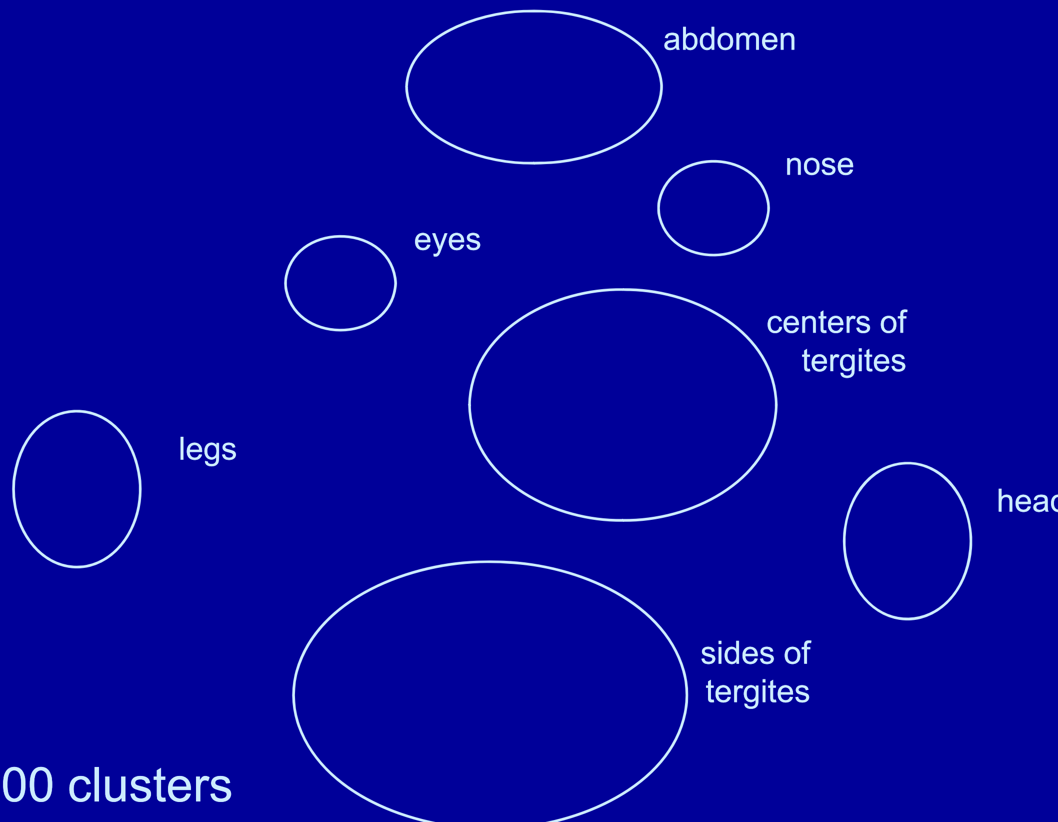
- Dominant Approach:
  - Learn visual dictionary
  - Map bag of SIFTs to keyword histogram
  - Bag-of-word classifier

- A New Approach:
  - Direct multiple instance classification
  - Stacked “evidence trees”

Learn Visual Dictionary by Clustering

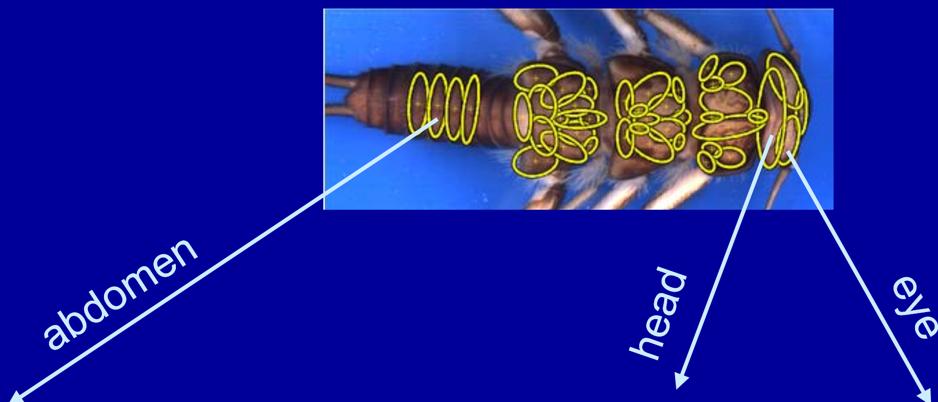
- Gaussian Mixture Model (k=100) with diagonal covariance matrices (EM, initialized with K-means)





100 clusters

Count each detected keyword into a “feature vector”



4	2	0	4	2	5	.	.	.	.	.	1	.	.	.	2	1	2		

Classification

- Boosted Decision Trees

- Other Approaches in the Literature
  - Boosted Logistic Model Trees
  - Support Vector Machines
    - Earthmover Distance kernel
    - Pyramid Match kernel

Issues with Visual Dictionaries

- Unsupervised
  - Several efforts to construct discriminative dictionaries (Moosman et al., 2006)
- Do not scale to many classes
  - 3 detectors × 9 classes × 100 keywords = 2700 features
  - Some efforts to learn shared / universal dictionaries (Winn, et al., 2005; Perronnin, et al., 2007)

Multiple-Instance Learning

- Given:
  - Labeled *bags* of feature vectors:

    (B_i, y_i)

    where each $B_i = \{x_{i,1}, \ldots, x_{i,N_i}\}$

    and each $x_{i,j}$ is a 128-element SIFT vector

    $y_i \in$ {Cal, Dor, Hes, Iso, Mos, Pte, Swe, Yor, Zap}
- Find
  - A scoring function $f(B,y)$ such that

    $y_i = \text{argmax}_y \, f(B_i, y)$

⇩

134 SIFT vectors

⇩

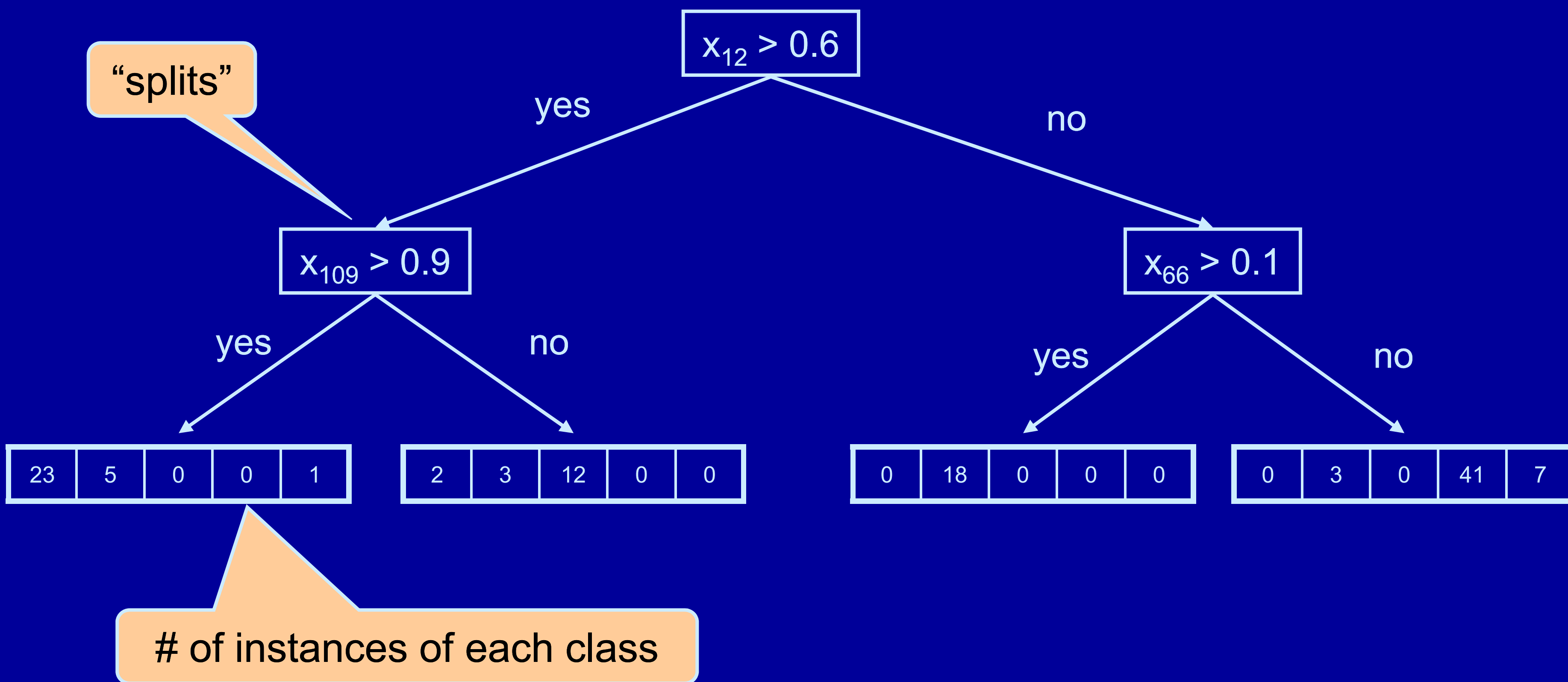
Calineuria

Multiple-Instance Learning is Hard

- Not all of the SIFTs are relevant
  - some are measuring parts shared across all species
  - some are measuring the same region at multiple scales
- A good predictor needs to combine evidence from multiple SIFTs
  - each individual SIFT just captures a small patch of the image

Evidence Trees

- Standard decision trees, but output the evidence rather than a decision



Our Method: Random Forest Evidence Trees + Stacking

1. Convert bags to individual labeled SIFTs
2. Train an *ensemble* of evidence trees using random forests & bootstrapping
3. Re-represent each Bag by the total predicted counts from these trees
4. Train a stacked classifier (boosted decision trees) to make the final decision using these counts

Step 1: Convert Labeled Bags to Labeled SIFTs

- Input labeled bags:

  (B_i, y_i)

- Create a new training set consisting of labeled SIFTs:

  $(x_{i,j}, y_i)$ for each $x_{i,j} \in B_i$

Step 2: Bootstrap Ensemble of Random Forest Evidence Trees

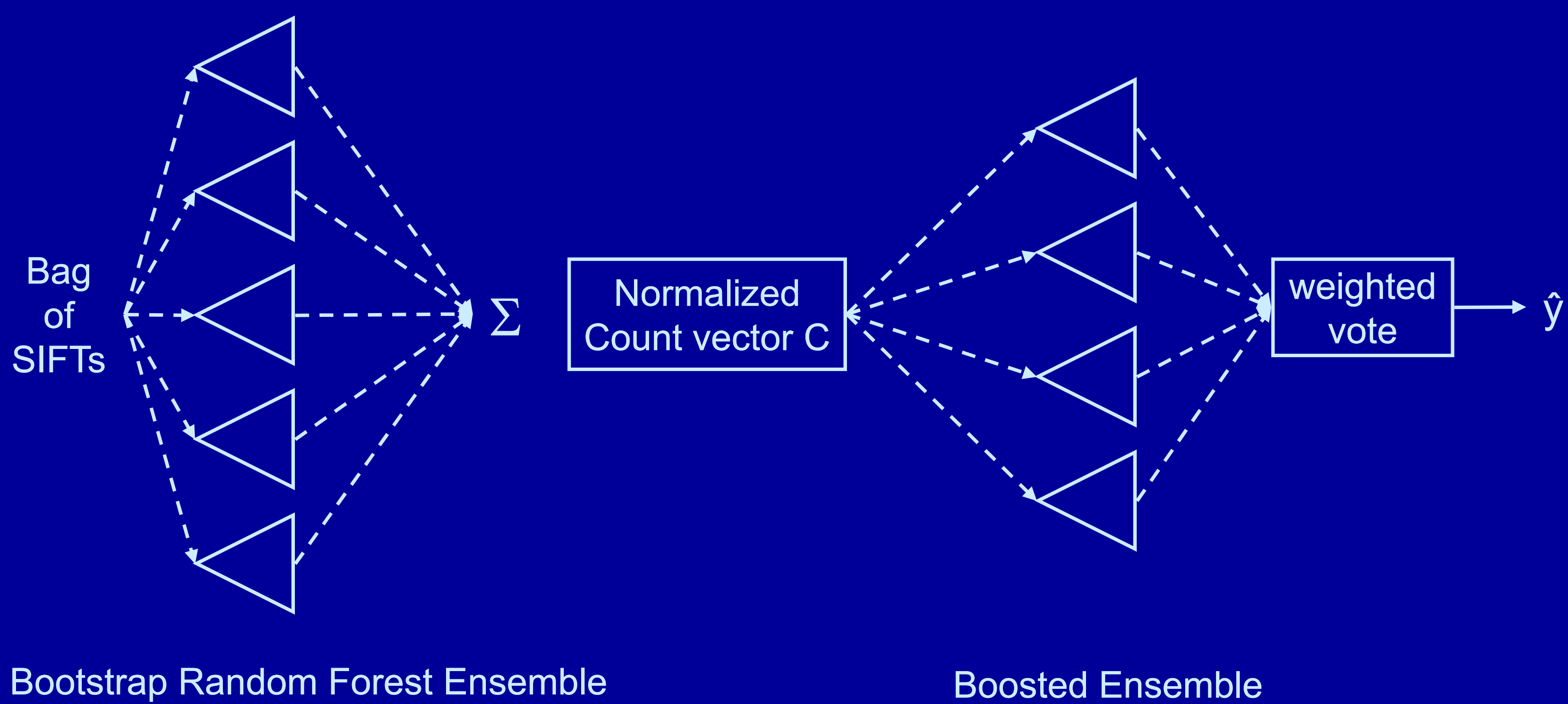
- For L = 1, …, 60
  - Draw a bootstrap replicate training set S'_L by sampling with replacement from entire Bags (B_i, y_i)
  - Convert S'_L to labeled SIFTs
  - Train a random forest evidence tree T_L
    - at each node, choose floor(1 + $\log_2$ n) attributes at random
    - choose the best attribute to split on from these
    - each leaf constrained to contain ≥ 20 points

Step 3: Create a Stacking Training Set

- For each original labeled bag (B_i, y_i)
  - Take each SIFT $x_{ij} \in B_i$, process it through each tree T_L in which it was *not used in training* (B_i not $\in S'_L$)
  - Let C_{ij} be count vector at the leaf of the tree
- Compute the vector sum C_i of these and normalize it to sum to 1.0
- Form a new training example (C_i, y_i)

Step 4: Train a Boosted Ensemble on the Stacking Data Set

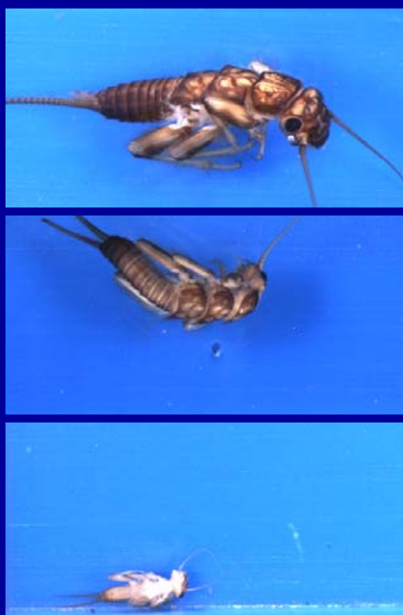
Final Classifier



Additional Details

- Train a separate bootstrapped random forest for each of three detectors
  - Harris-Affine
  - Kadir
  - PCBR
- Concatenate the resulting feature vectors prior to stacking
- Adaboost: 100 C4.5 decision trees
- Can also grow random forests based on other features (e.g., shape)

Experimental Study
9 Taxa of Stoneflies



Calineuria

Doroneuria

Hesperoperla

Yoroperla

Stonefly9 Dataset

- 3826 images
- 773 specimens
- 9 classes
- Error estimation by 3-fold cross-validation
  - all images of a specimen belong to the same fold

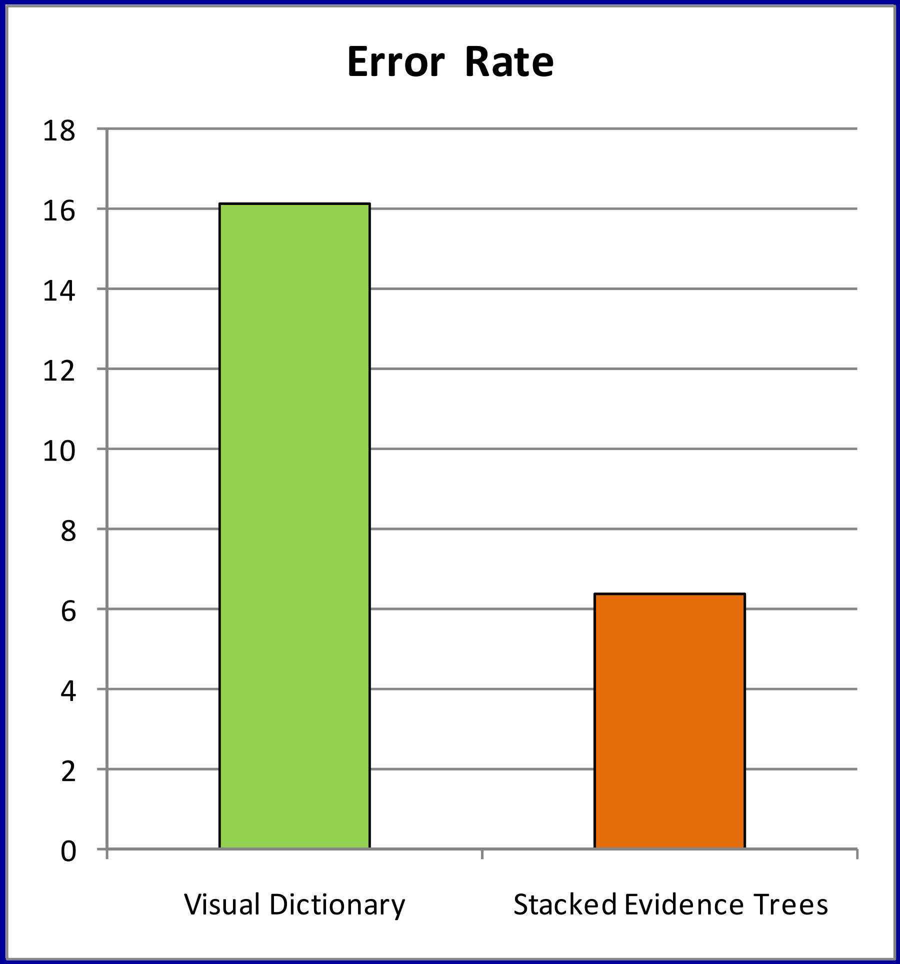
Results: 94.6% Correct

Predicted Species

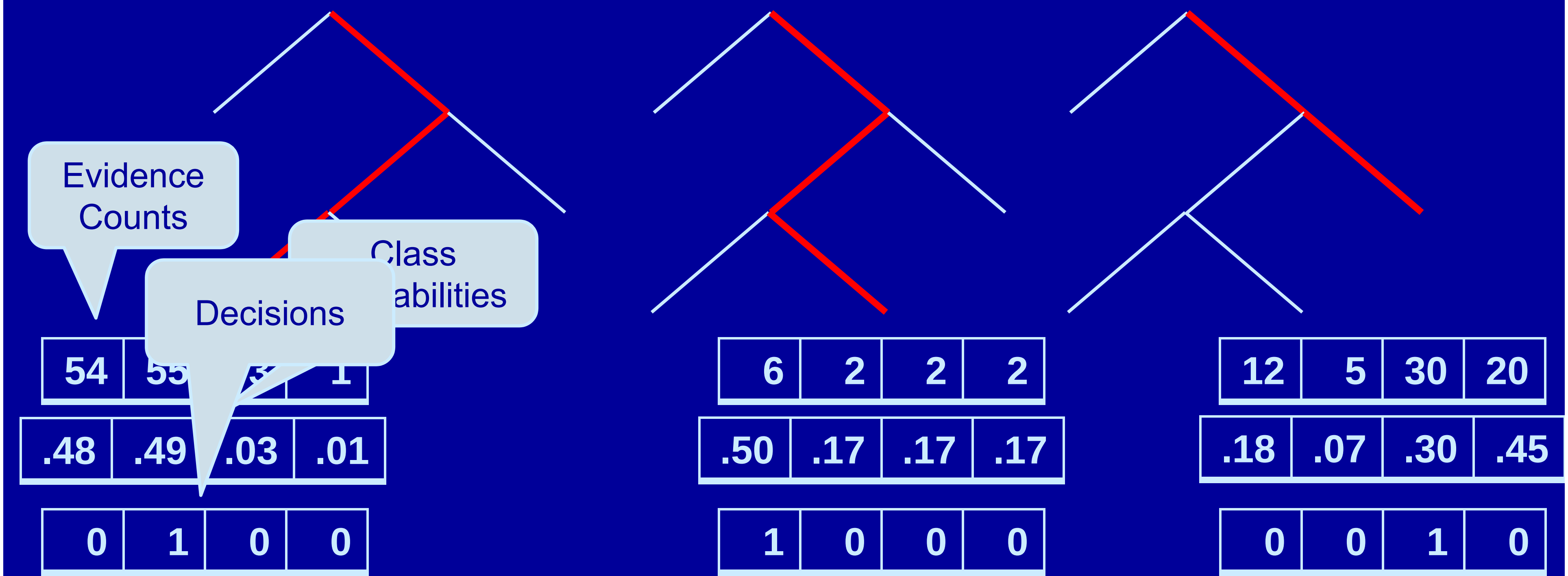
True Species

	Cal	Dor	Hes	Iso	Mos	Pte	Swe	Yor	Zap
Cal	443	17	3	4	0	0	20	0	5
Dor	19	489	1	10	1	0	7	0	5
Hes	6	5	460	5	0	1	12	0	2
Iso	3	6	3	456	0	2	27	0	3
Mos	0	0	0	1	107	0	3	0	8
Pte	0	3	0	0	0	203	6	5	6
Swe	4	10	2	23	0	1	433	1	5
Yor	1	1	1	1	1	3	0	481	3
Zap	0	0	2	8	4	9	3	4	468

Comparison of Methods



Combining Evidence is better than Voting Decisions or Probabilities



Evidence Counts

72	62	35	23
.38	.32	.18	.12

Class Probabilities

1.16	0.73	0.50	0.63
.38	.24	.17	.21

Decisions

1	1	1	0
.33	.33	.33	.00

Mathematical Model

- Parameters:
  - C training examples in each leaf
  - L trees in the ensemble
  - D regions detected in the test image
  - γ: probabilistic margin of each leaf
    - one class has probability $1/2 + \gamma$
    - one class has probability $1/2 - \gamma$

Applying Chernoff Bounds

- Voting classifications
  - Each leaf has a probability ε of being mislabeled during training

    $\varepsilon \approx \mathrm{Exp}[-2C\gamma^2]$
  - The vote has a probability of error

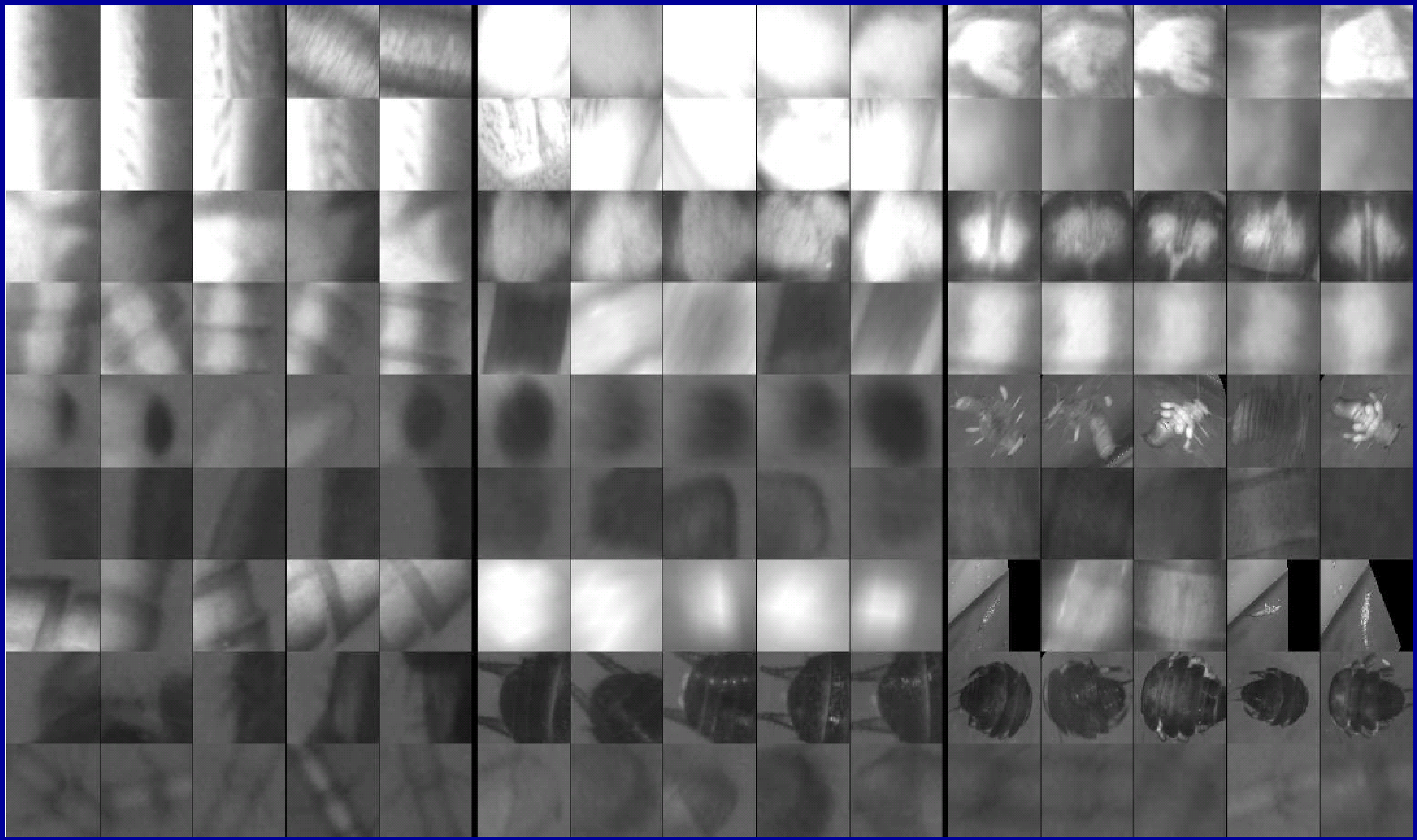
    $\varepsilon_{vy} \approx \mathrm{Exp}[-2DL\gamma^2(1 - 2\varepsilon)^2]$
- Voting evidence
  - Vote has probability of error

    $\varepsilon_{v\#} \approx \mathrm{Exp}[-8CDL\gamma^4]$

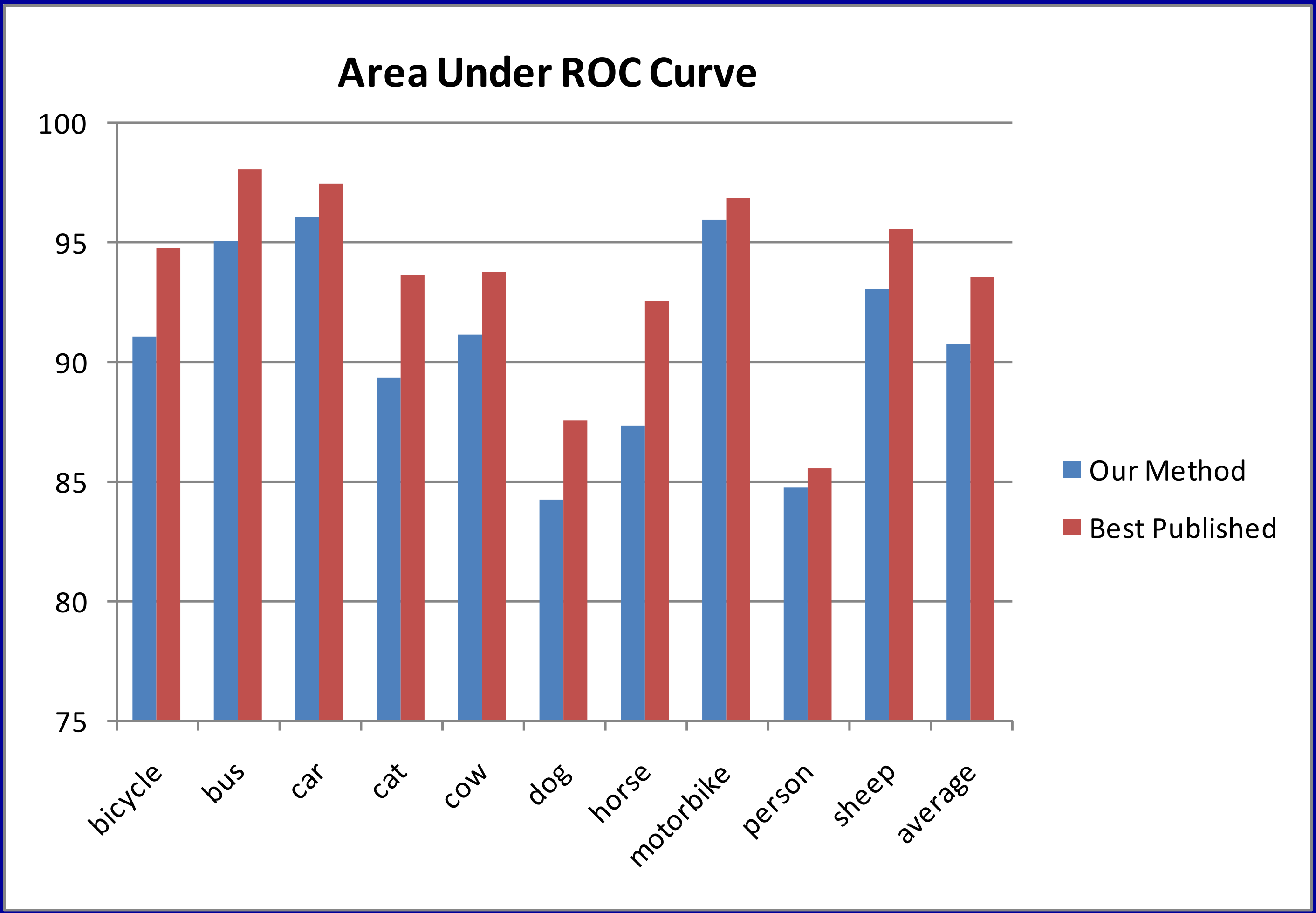
Result

- If $C > 1/(4\gamma^2)$ then voting evidence is better than voting decisions: $\varepsilon_{v\#} < \varepsilon_{vy}$
- A good tree has $\gamma > 0.25$, so voting evidence will be better than voting decisions if $C > 4$

Most Discriminative Regions



Generic Object Recognition: PASCAL 2006 VOC



Rank:
5th out of 21

Next Steps

- Stoneflies
  - Detecting and Rejecting “Distractors”
  - Extending coverage to Ephemeroptera (mayflies) and Trichoptera (caddis flies)
  - EPA field study
- Soil Mesofauna
- Freshwater Zooplankton
- Moths
- Shellfish Larvae

- PASCAL 2007; PASCAL 2008 Challenges
- Night Calls of Migrating Birds?

Conclusions

- Computer vision and machine learning methods can achieve high accuracy classification of stoneflies
- For computer vision problems involving multiple detections per image, voting the evidence is more accurate than voting class probabilities or voting decisions
- Our methods are competitive on generic object recognition problems

Acknowledgements

- Grant Support: US National Science Foundation