
Boosted Evidence Trees for Object Recognition with Applications to Arthropod Biodiversity Studies

Students: N. Larios, H. Deng, W. Zhang, N. Payet,
M. Sarpola, C. Fagan, C. Baumberger, J. Lin, J.
Yuen, S. Ruiz Correa

Postdoc: 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

Arthropod Population Counts: An Important Form of Ecological Data

- ◆ Arthropods are a powerful data source
 - Found in virtually all environments
 - streams, lakes, oceans, soils, birds, mammals
 - Easy to collect
 - Provide valuable information on ecosystem function
 - Consume the primary producers: bacteria, fungi, plants
 - Are consumed by more charismatic organisms: birds, mammals, fish
- ◆ Problem: Identification is time-consuming and requires scarce expertise
- ◆ Solution: Combine robotics, computer vision, and machine learning to automate classification and population counting



Automated Rapid-Throughput Arthropod Population Counting

◆ **Goal:**

- technician collects specimens in the field by various means
- robotic device automatically manipulates, photographs, classifies, and sorts the specimens

◆ **Two applications:**

- EPTs in freshwater streams
- Soil mesofauna

Application 1: EPT Larvae

- ◆ EPTs: Mayflies, Stoneflies, Caddis flies (Ephemeroptera, Plecoptera, Trichoptera)
- ◆ Live in freshwater streams
- ◆ Population surveys are used for
 - assessing stream health
 - measuring success of stream restoration
 - understanding basic stream ecology



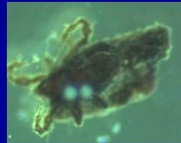
Application 2: Small arthropods in soil: “soil mesofauna”



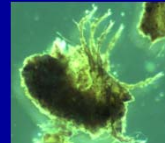
AchipteriaA



BdellozoniumI



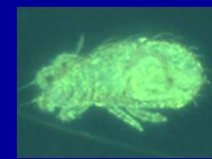
BelbaA



Belbal



CatoposurusA



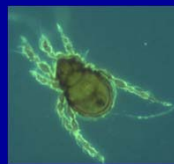
EniochthoniusA



PtenothrixV



EntomobryaTM



EpidamaeusA



EpilohmanniaA



EpilohmanniaD



EpilohmanniaT



HypochthoniusLA



PtiliidA



HypogastruraA



IsotomaA



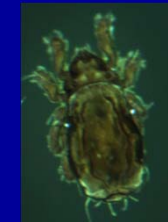
IsotomaVI



LiacarusRA



MetrioppiaA



NothrusF



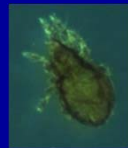
QuadropiaA



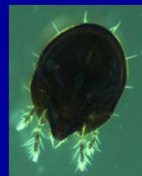
TomocerusA



onychiurusA



OppiellaA



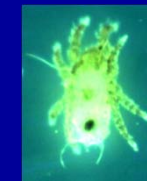
PeltenuiaA



Phthiracarusa



PlatynothrusF



PlatynothrusI



SiroVI

1/25/2011

Caltech

5

Previous Results: 9 Taxa of Stoneflies

Cal



Iso



Dor



Mos



Hes



Pte



Swe



Yor



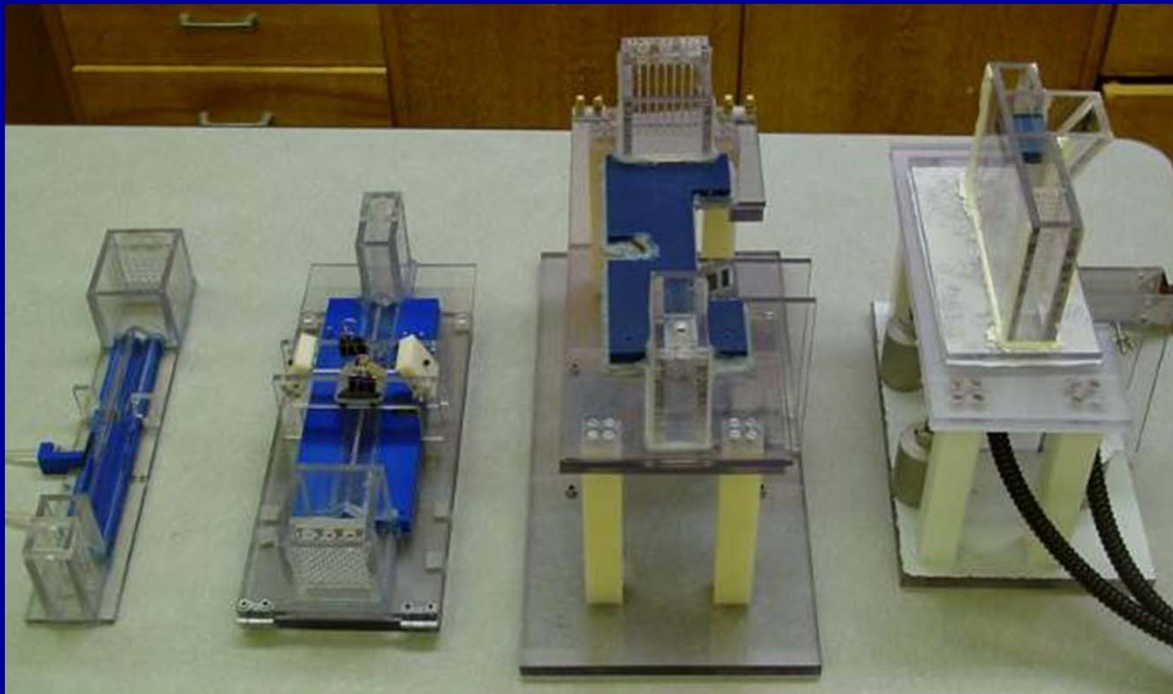
Zap



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

Image Capture Apparatus



Stonefly Imaging



Soil Mesofauna
Imaging

Computer Vision Challenges(1)

- ◆ Highly-articulated objects with deformation



Computer Vision Challenges(2)

- ◆ Huge intra-class changes of appearance due to development and maturation



tergites

become



wings

Computer Vision Challenges(3)

- ◆ Small between-class differences







Calinueria



Doronueria

Machine Learning

Training
Examples

| | |
|---|------------|
|  | Calineuria |
|  | Calineuria |
|  | Doroneuria |
|  | Doroneuria |

Learning
Algorithm

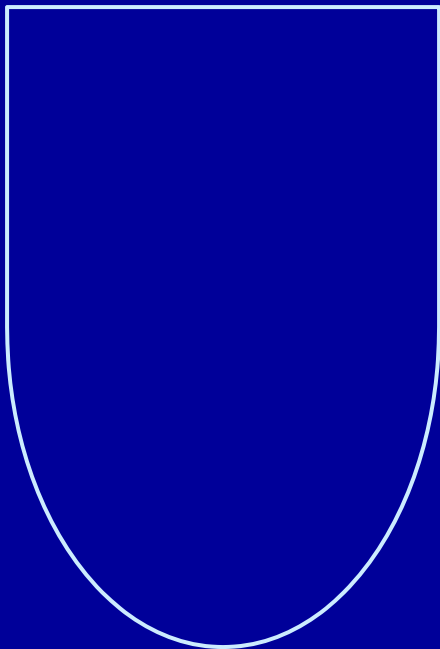
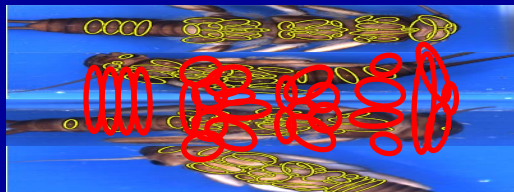
New
Examples



Classifier

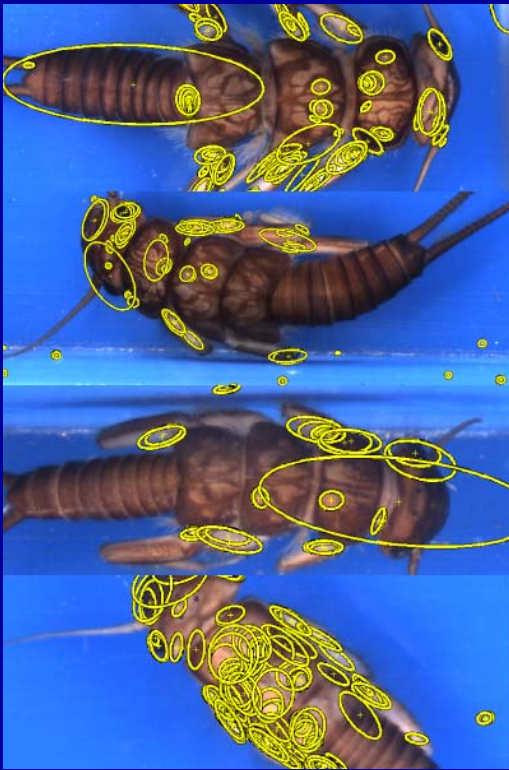
Doroneuria

Region-Based Approaches: Convert Image to Bag of Patches

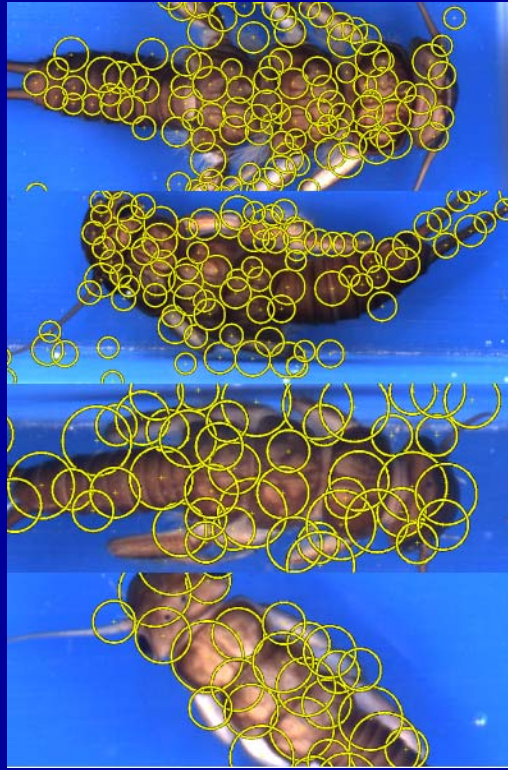


- ◆ Handles
 - Occlusion
 - Rotation, translation
 - Scale (with scale-independent patch representation)
 - Partial out-of-plane orientation
 - Articulation / Pose
- ◆ Problem:
 - How to define the patches?
 - How to represent each patch?
 - How to classify a BAG of patches?

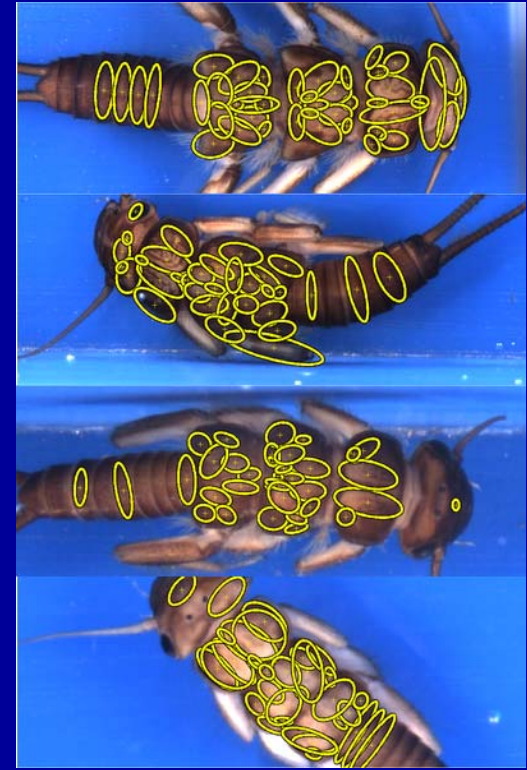
Defining the Patches: Interest Region Detectors



Hessian-Affine Detector

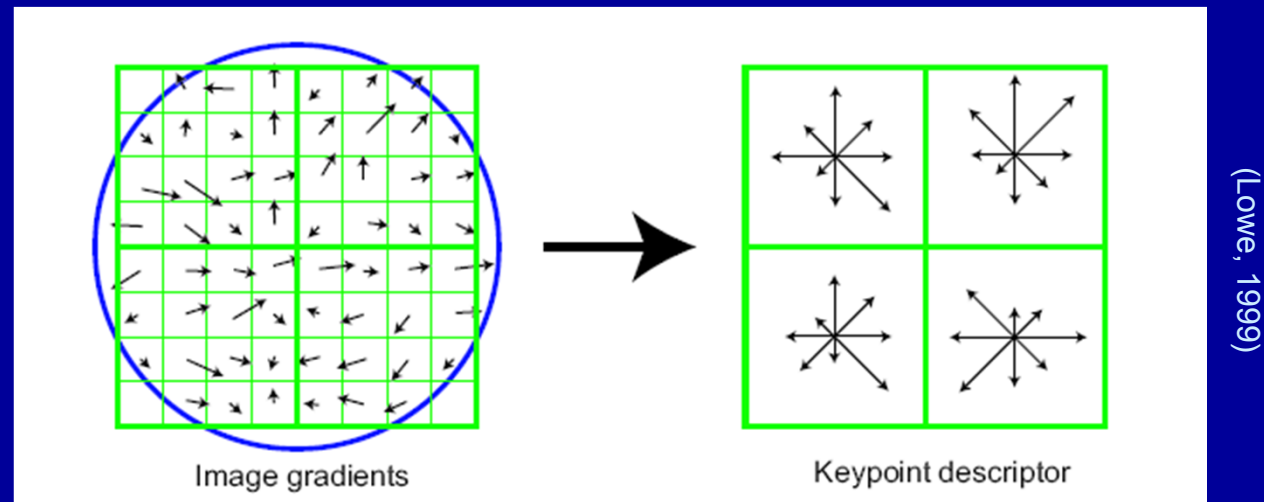


Kadir Entropy Detector



PCBR Detector

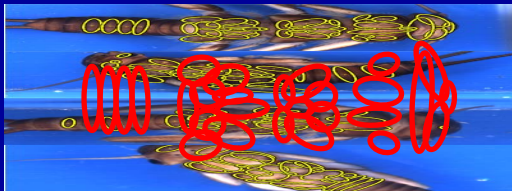
Representing the Patches: SIFT (Lowe, 1999)



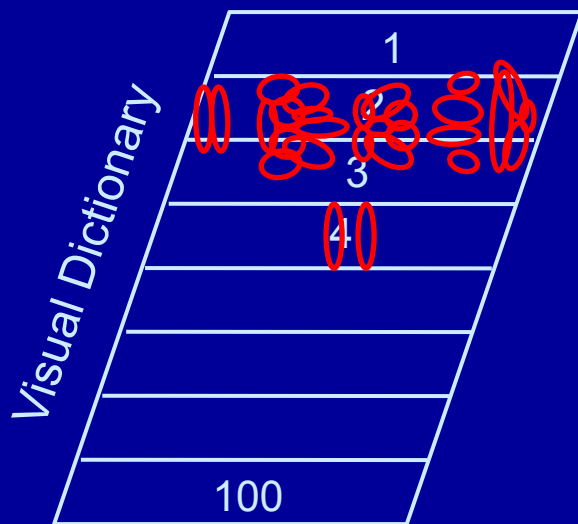
- Morph ellipse into a circle
- Compute intensity gradient at each pixel in 16x16 region
- Rotate whole circle according to dominant intensity gradient
- Weight gradients by a gaussian distribution (indicated by circle)
- Collect into histograms within each 4x4 region (gives 16 histograms)
- Result: 128-element vector normalized to have Euclidean norm 1

Classify Bag of Patches

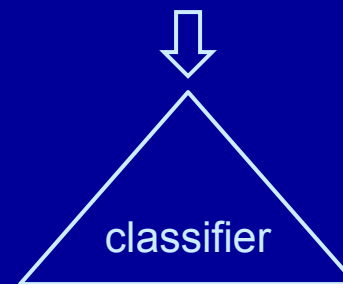
Method 1: Visual Dictionaries



- ♦ “look up” each patch in dictionary and count into a feature vector
- ♦ feature vector is then given to the classifier



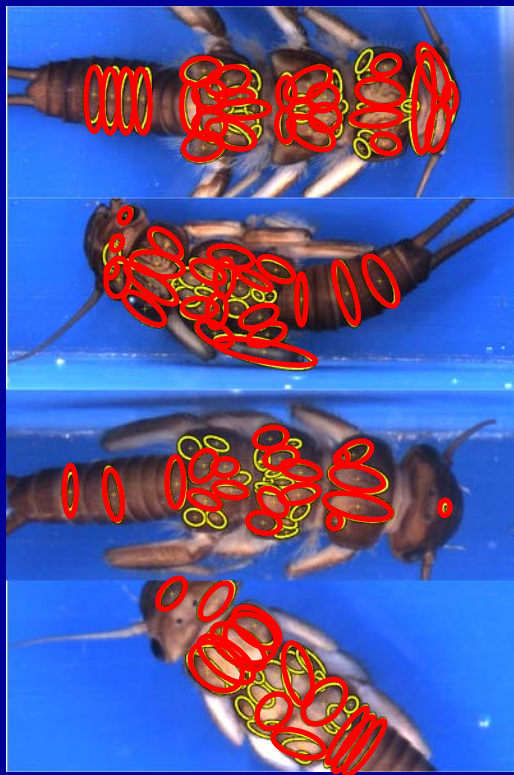
| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 2 | 6 | 4 | 9 | 0 | . | . | . | . | . | 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|



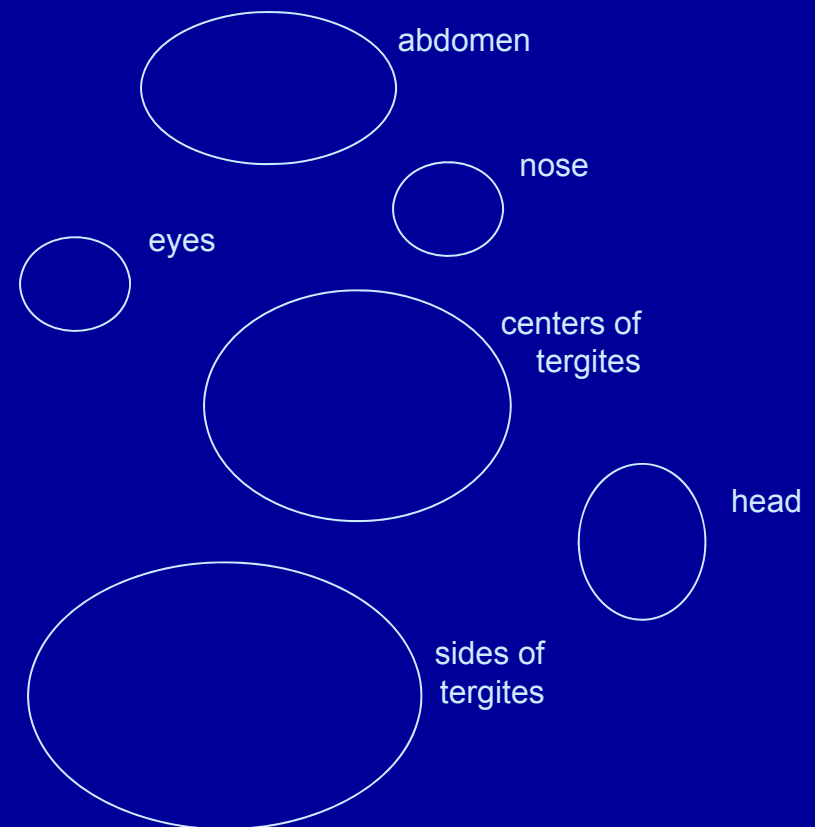
$\hat{y}=2$

Learn Visual Dictionary by Clustering

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



100 clusters



Issues with Visual Dictionaries

- ◆ Information is lost
- ◆ Unsupervised
 - Several efforts to construct discriminative dictionaries (Moosman et al., 2006)
- ◆ Do not scale to many classes
 - $3 \text{ detectors} \times 9 \text{ classes} \times 100 \text{ keywords} = 2700$ features
 - Some efforts to learn shared / universal dictionaries (Winn, et al., 2005; Perronnin, et al., 2007)

Boosting Visual Dictionaries

For each image i , assign weight $w_i = 1$

For $t = 1, \dots, T$

For each SIFT s_{ij} , assign it weight w_i

Apply weighted k-means clustering to construct a dictionary D_t

Train classifier F_t on the training images encoded using D_t

Update the image weights according to the Adaboost formula

Final classifier is weighted vote of the F_t

Why is this a good idea?

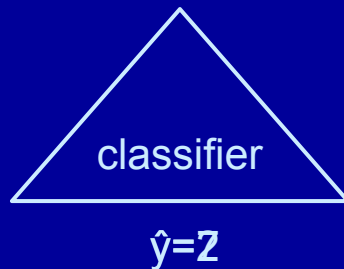
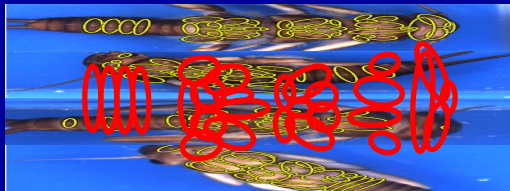
- ◆ If D_t is not adequate for correctly classifying some images, then the next dictionary D_{t+1} will allocate more representational resources to those images
- ◆ This will lead to reduced quantization error for the SIFTs in those images
- ◆ This will allow the next classifier F_{t+1} to do a better job

Additional Details

- ◆ Feature vectors are reweighted using TF-IDF weights
- ◆ Classifier in each iteration: 50-fold bagged C4.5 decision trees (no pruning)
- ◆ 30 boosting iterations
- ◆ Each iteration learns 100 codewords per detector (300 codewords total)
- ◆ Final classifier is using a dictionary of 9000 codewords (but partitioned into 300-word parts)

Classify Bag of Patches

Method 2: Multiple-Instance Classifier



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 8 | 1 | 3 | 0 | 0 | 6 | 4 | 2 |
|---|---|---|---|---|---|---|---|---|

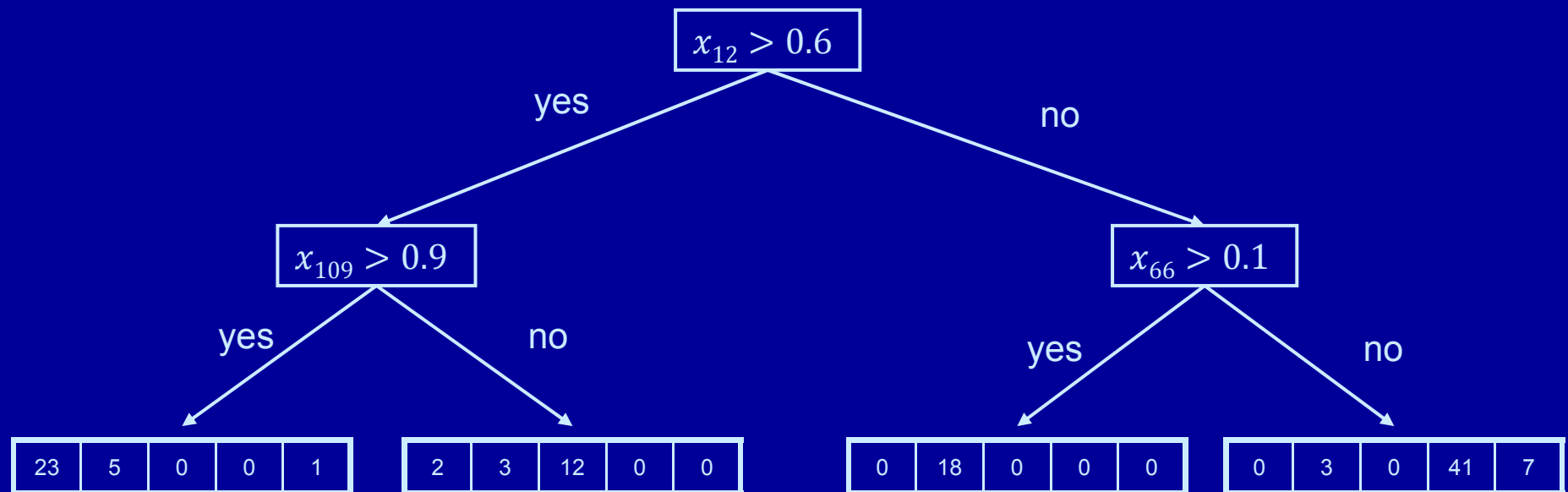
votes

Final prediction: $\hat{y} = 2$

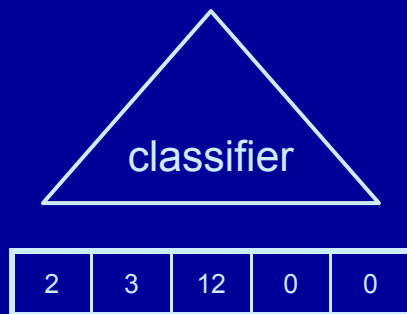
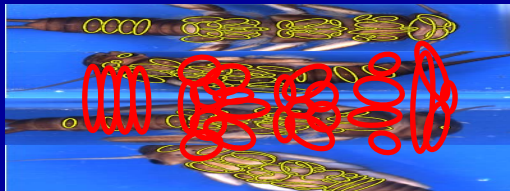
- ♦ The classifier predicts the class of the image separately using each patch
- ♦ These vote to make the final decision

Improved Multiple-Instance Classification

- ◆ Evidence Trees: Like decision trees, but store the “evidence” in each leaf
- ◆ Given an input, output the evidence



Classify Bag of Patches Voted Evidence Trees



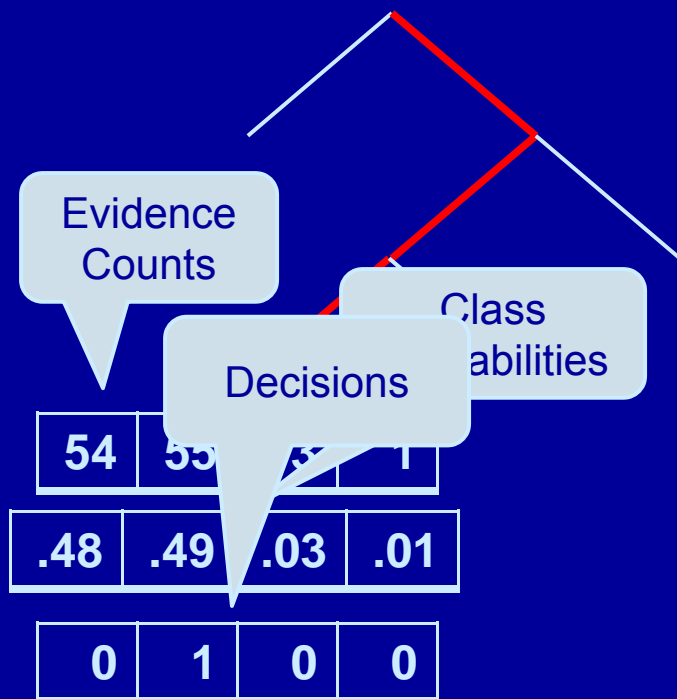
| | | | | |
|----|----|----|---|----|
| 87 | 14 | 34 | 6 | 61 |
|----|----|----|---|----|

votes

- ◆ The classifier predicts the class of the image separately from each patch
- ◆ These vote to make the final decision

Final prediction: $\hat{y} = 1$

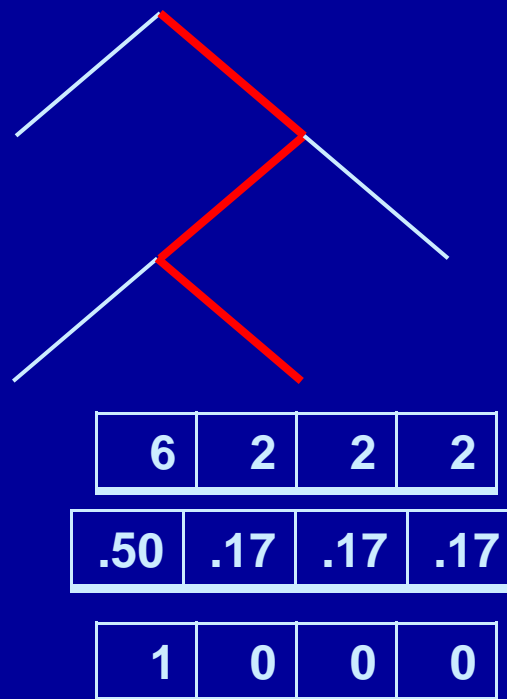
Claim: Combining Evidence is better than Voting Decisions or Probabilities



Evidence Counts

| | | | |
|----|----|----|----|
| 72 | 62 | 35 | 23 |
|----|----|----|----|

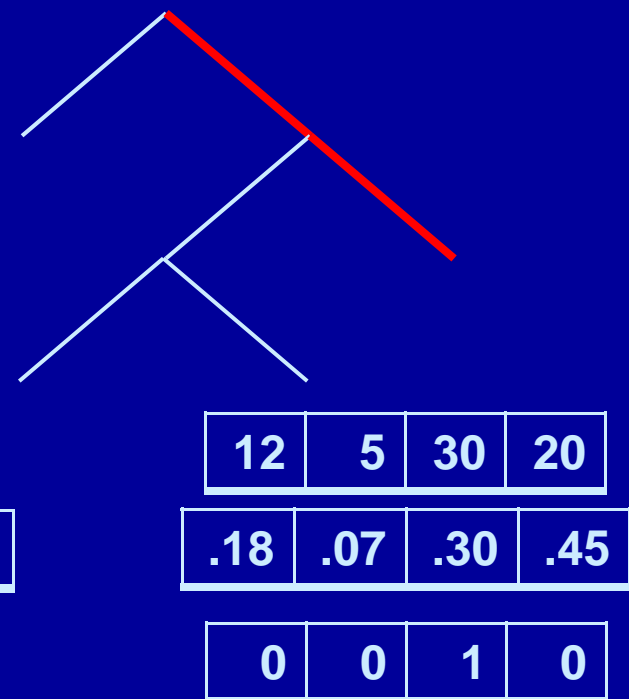
| | | | |
|-----|-----|-----|-----|
| .38 | .32 | .18 | .12 |
|-----|-----|-----|-----|



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 $1/2 - \gamma$

Proof

- ◆ Let $\beta = 2 \gamma^2 \pi^2$
- ◆ Voting decisions. Lower-bound binomial tail by largest term:

$$\epsilon_{vd} \geq \left(\frac{1}{2} - \beta\right)^{\frac{D}{2}}$$

- ◆ Voting evidence. Upper-bound binomial tail via Chernoff bound:

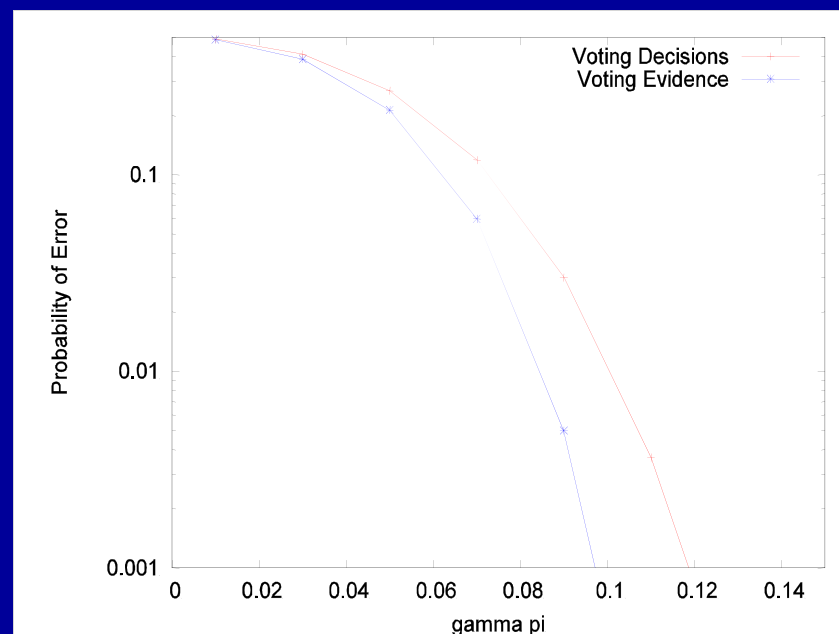
$$\epsilon_{v\#} \leq \exp[-8CDL\gamma^4]$$

Result

- ♦ If $C > -\log(\frac{1}{2} - \beta)/4\beta^2$ then voting evidence is better than voting decisions: $\epsilon_{v\#} < \epsilon_{vd}$
- ♦ Exact computation for reasonable values (e.g., $C=21$, $D=301$) verifies this

Theorem: Voting Evidence is Better than Voting Decisions

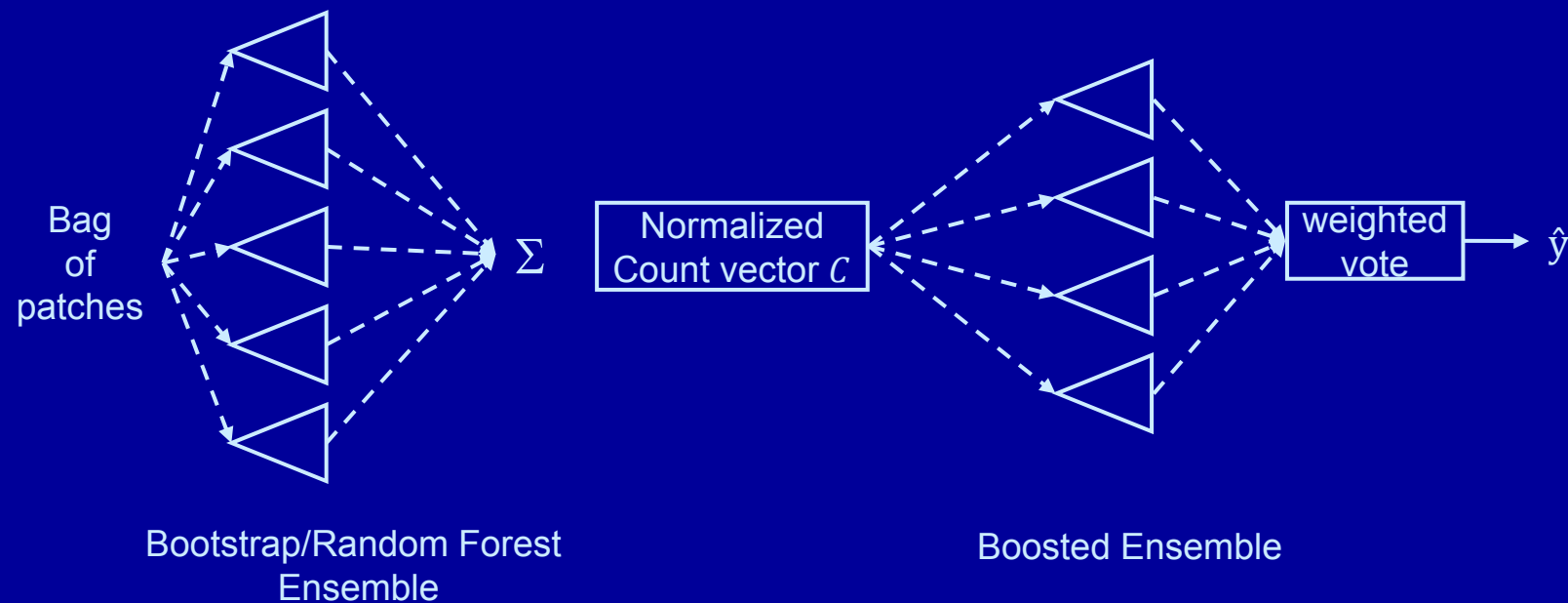
- ♦ Intuition: When voting decisions, there are two opportunities to make a mistake:
 1. Making the wrong decision at each leaf
 2. Making the wrong decision when combining the votes
- ♦ With evidence trees, the first opportunity is avoided



γ = margin of decision tree nodes
 π = fraction of non-noise patches

Final Classifier: Stacked Evidence Tree Random Forest

1. Each patch is processed by a **random forest** of evidence trees
2. Evidence is summed and normalized to produce \mathcal{C}
3. \mathcal{C} is classified by a second-level **boosted decision tree ensemble**



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

Cal



Dor



Hes



Iso



Mos



Pte



Swe



Yor



Zap



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

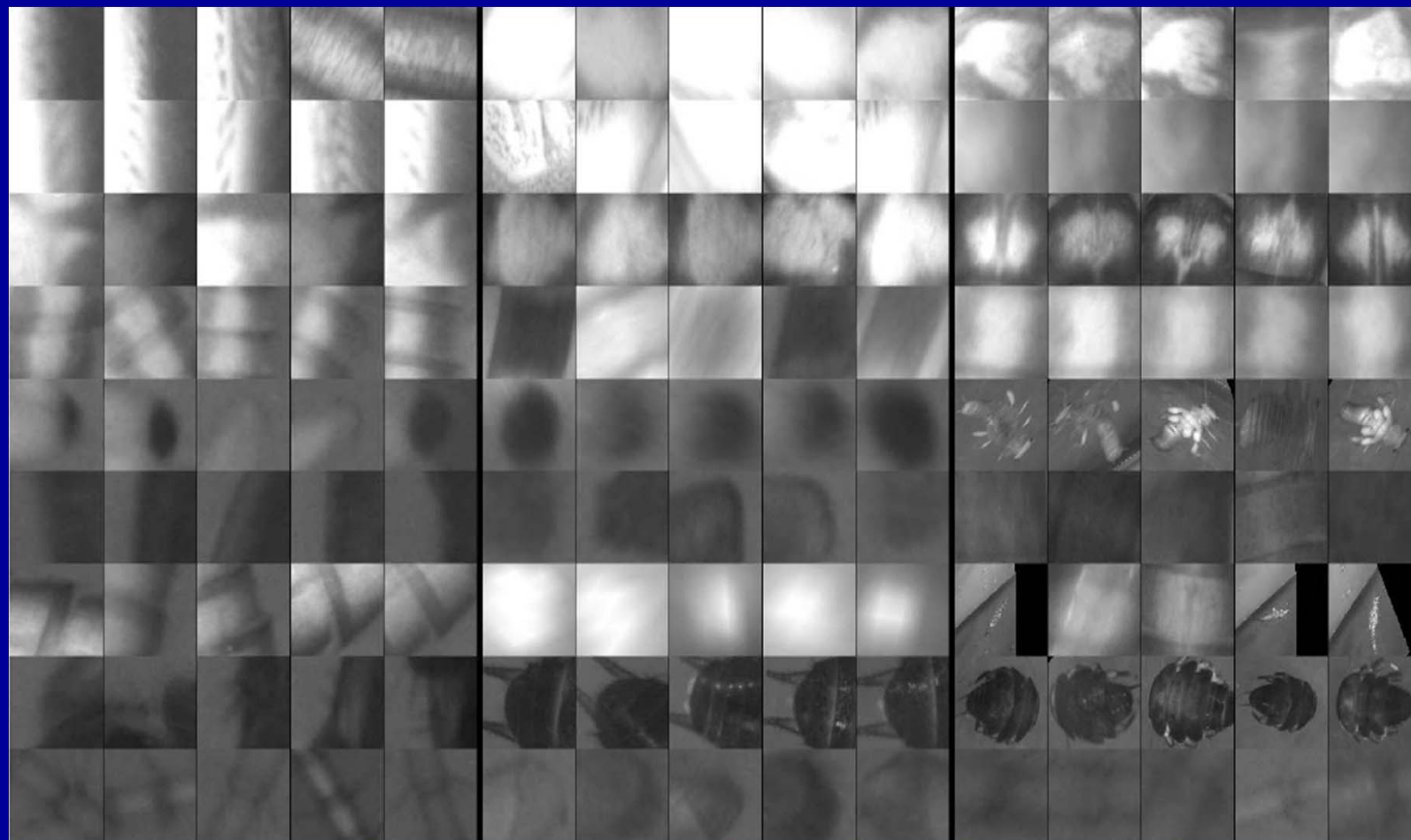
Results

| Configuration | Error Rate |
|--|------------|
| Single GMM Dictionary + Boosted Decision Trees | 16.1% |
| 30-fold Boosted Dictionaries | 4.9% |
| Stacked Evidence Trees | 5.6% |

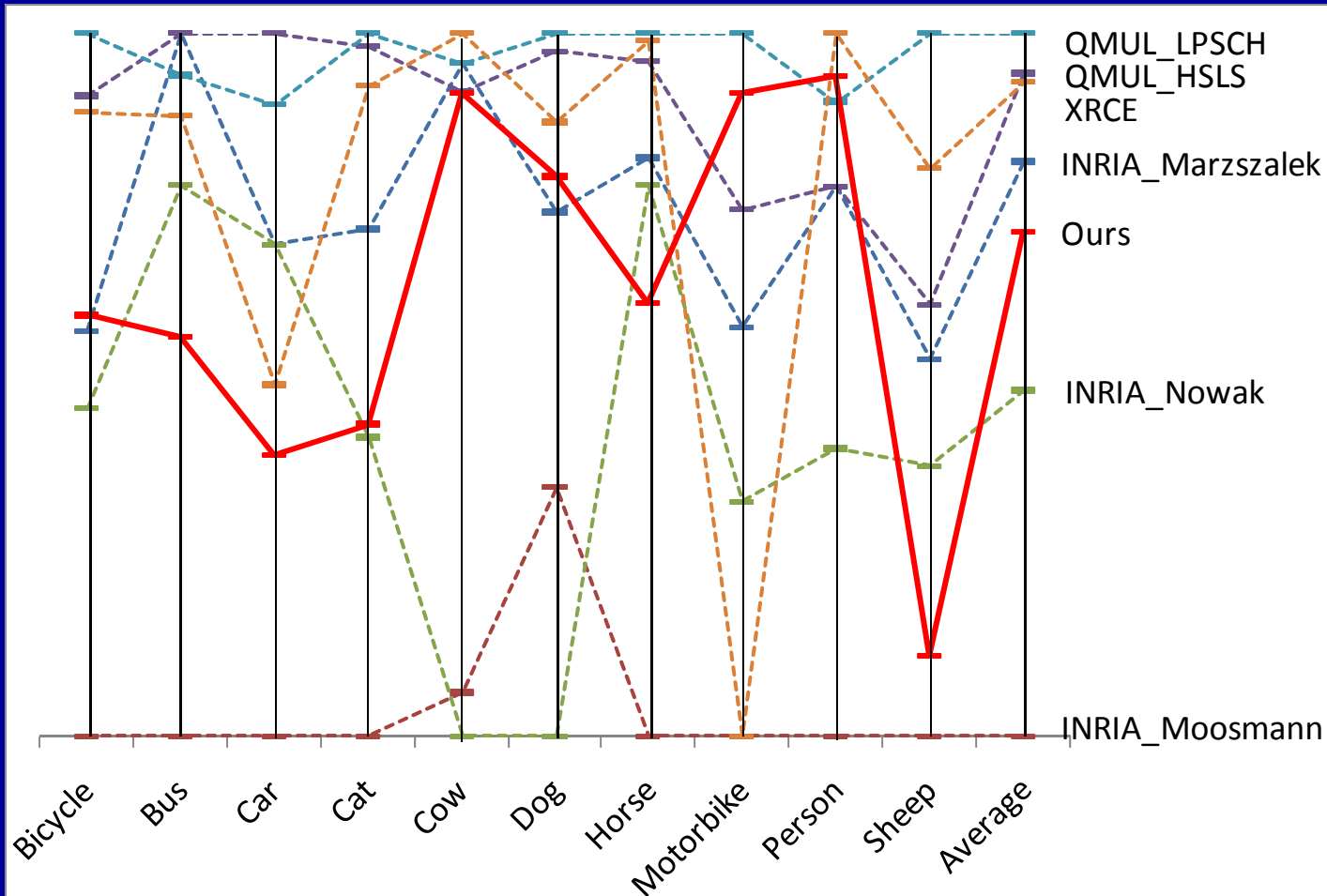
Evidence Tree Confusion Matrix

| True Species | Predicted 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 |

Most Discriminative Regions

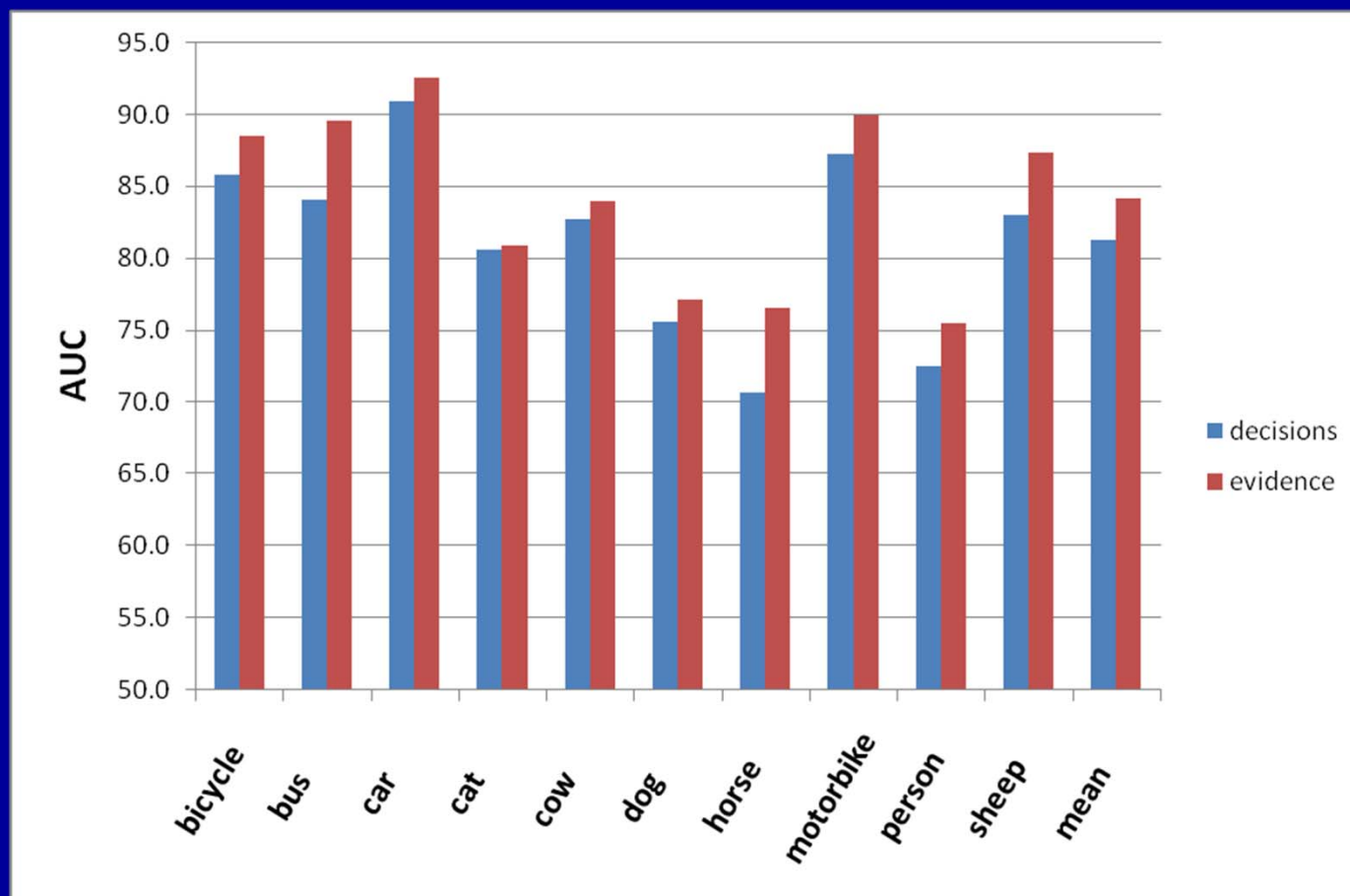


Generic Object Recognition: PASCAL 2006 VOC



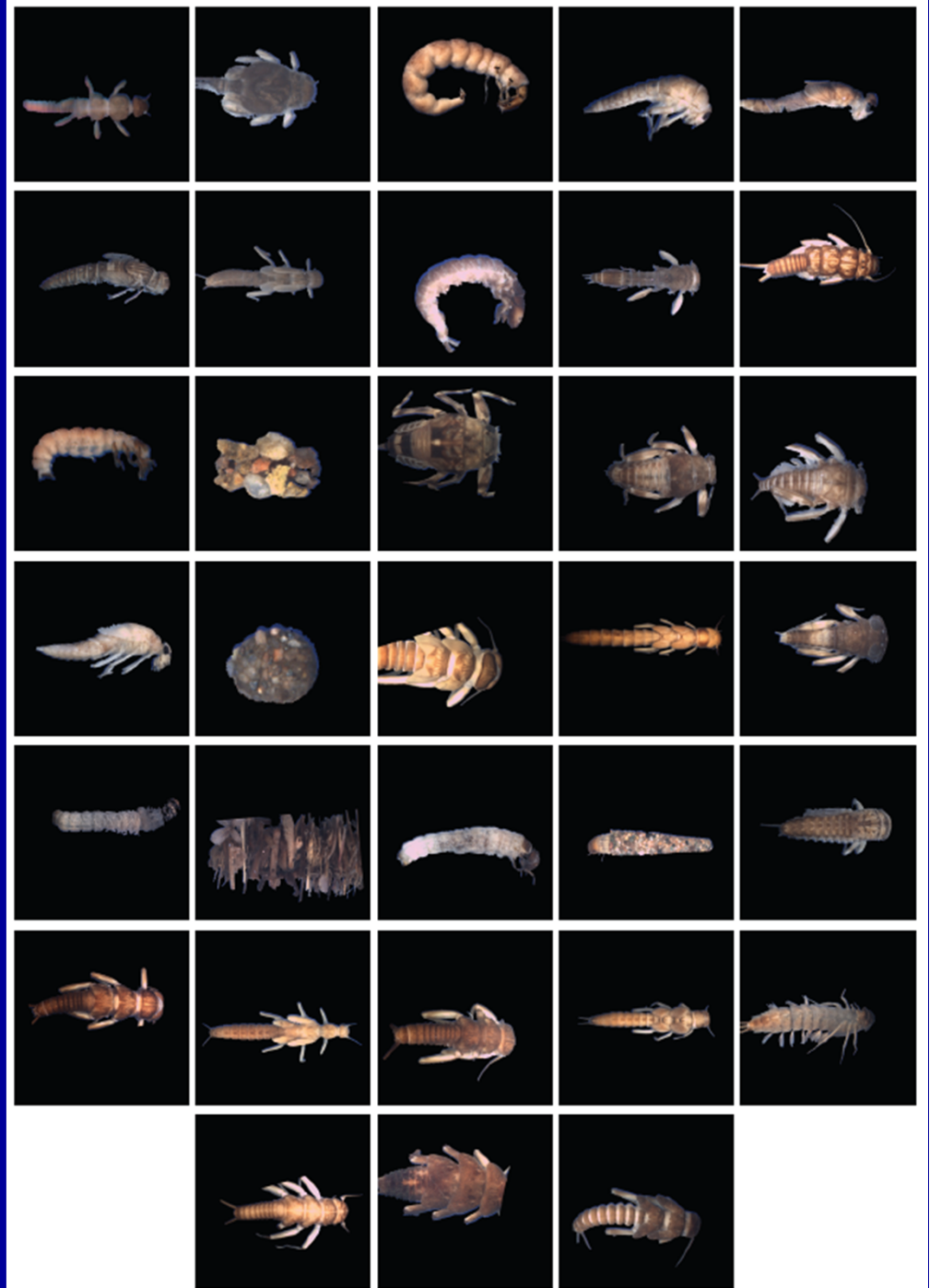
AUC Rank:
5th out of 21

Comparison: Voting Evidence vs. Voting Decisions



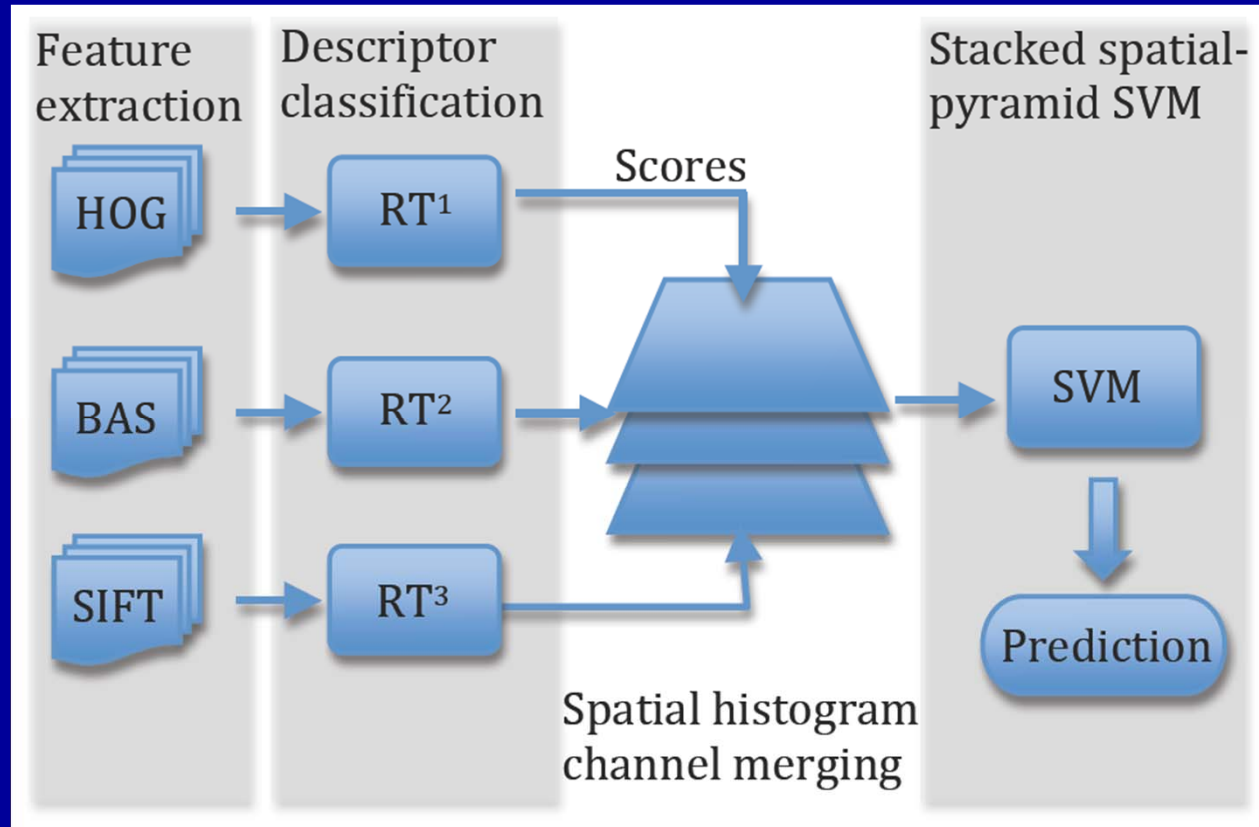
EPT 29 Data Set

- ◆ 29 taxa of stoneflies (Plecoptera), caddisflies (Trichoptera), and mayflies (Ephemeroptera)
- ◆ 4722 images
- ◆ 1-4 images per specimen
- ◆ automatically segmented, rotated, and aligned to face left
- ◆ 3 folds (all images per specimen in same fold)



Method 3: Stacked Spatial Pyramid

Natalia Larios

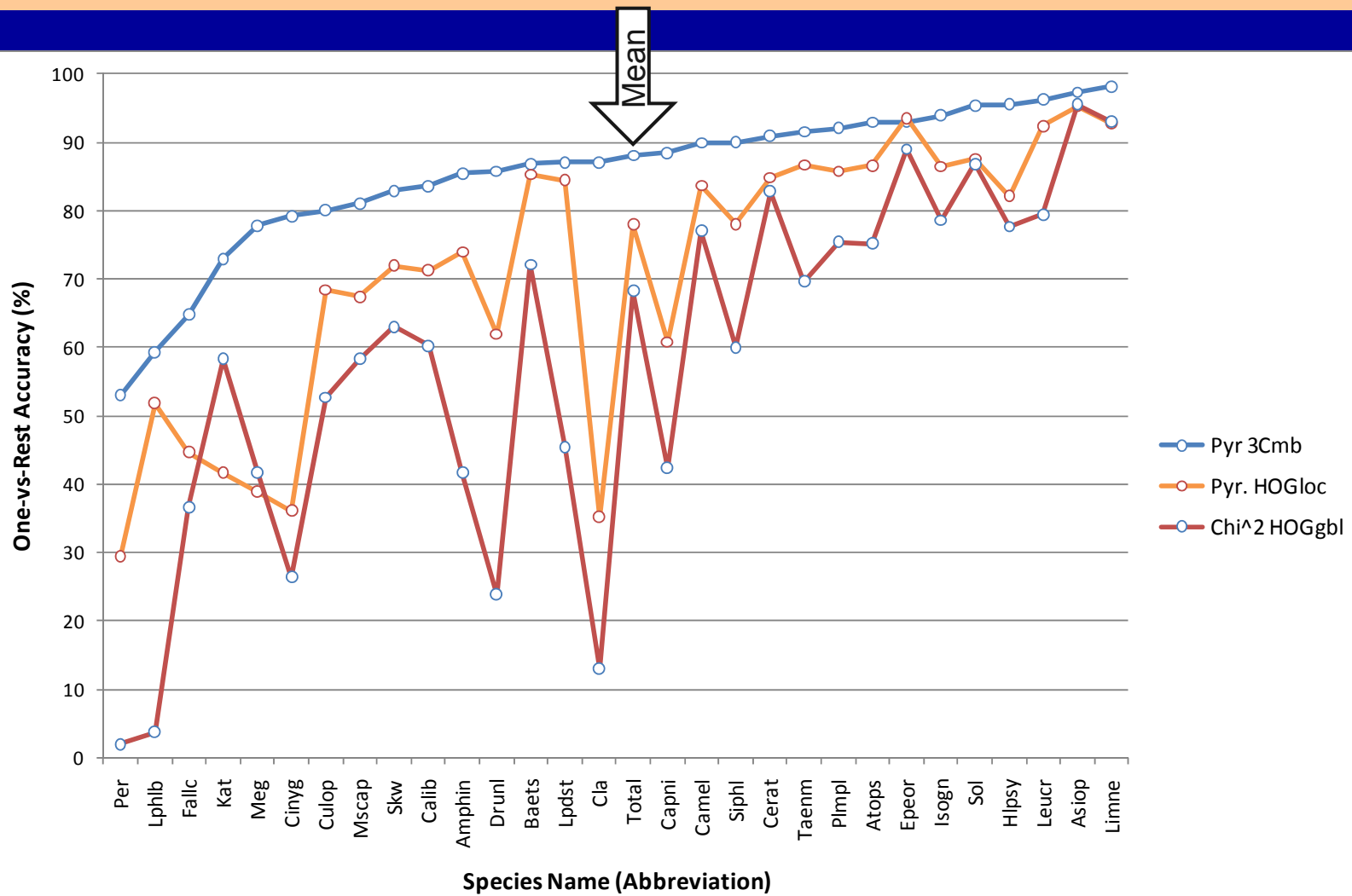


Larios, N., Lin, J., Zhang, M., Lytle, D., Moldenke, A., Shapiro, L., Dietterich, T. (2011). Stacked Spatial-Pyramid Kernel: An Object-Class Recognition Method to Combine Scores from Random Trees. WACV 2011.

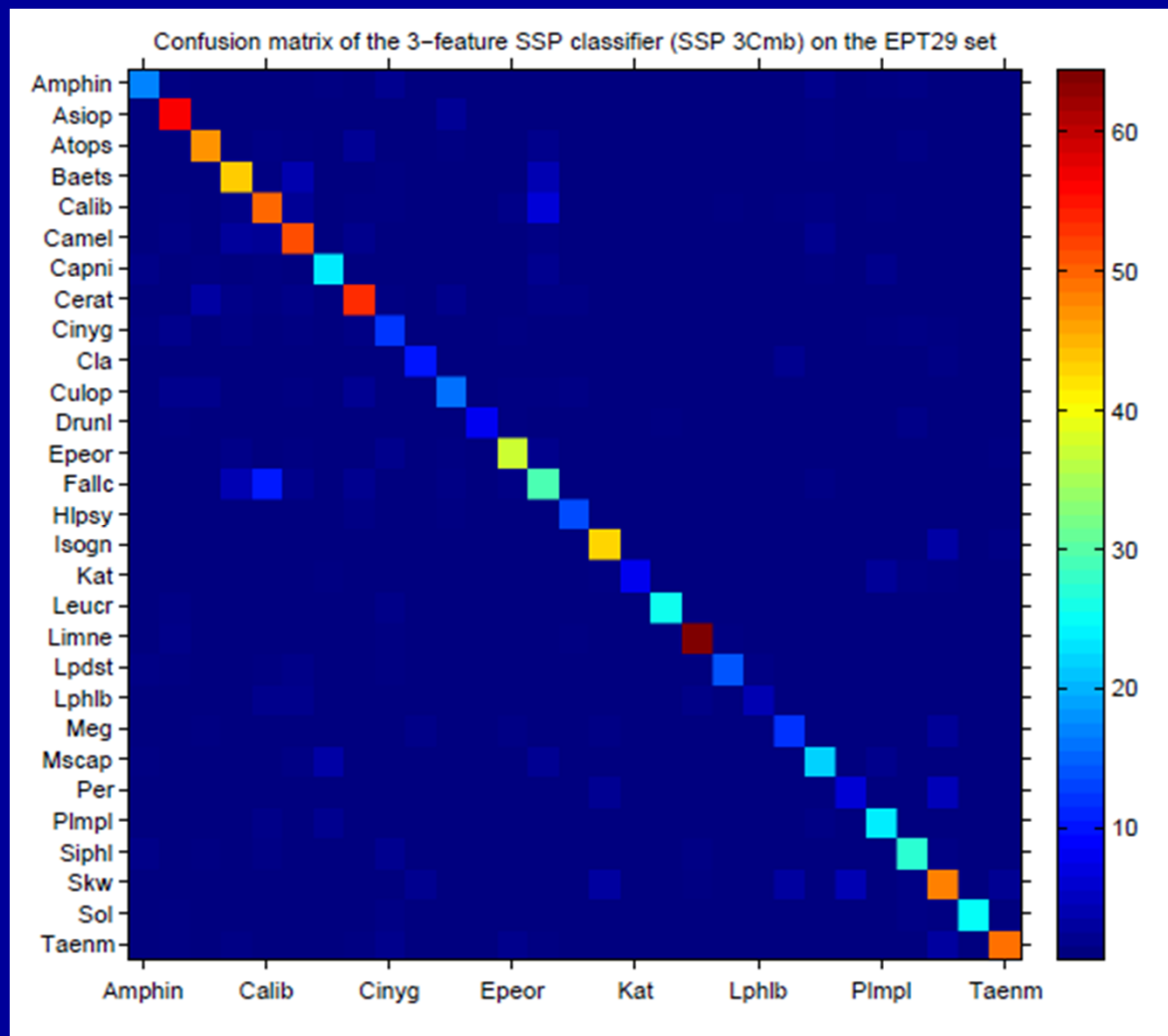
Experiment Details

- ♦ Detectors/Descriptors
 - HOC: Dense 16x16 pixels with 8 pixel overlap
 - BAS: salient points on perimeter, beam angle statistics + SIFT at each salient point
 - SIFT: DoG detector + SIFT descriptor
- ♦ Random Forest classifiers (RT)
 - 150 trees with max depth 25
 - trained to predict class of image from single patch descriptor (HOG, BAS, or SIFT)
 - score every patch, sum and normalize to obtain class probabilities
 - based on Evidence Trees but with normalization
- ♦ Stacked classifier
 - 3-level pyramid (16, 4, 1)
 - intersection kernel
 - trained via “out of bag” instances

Results

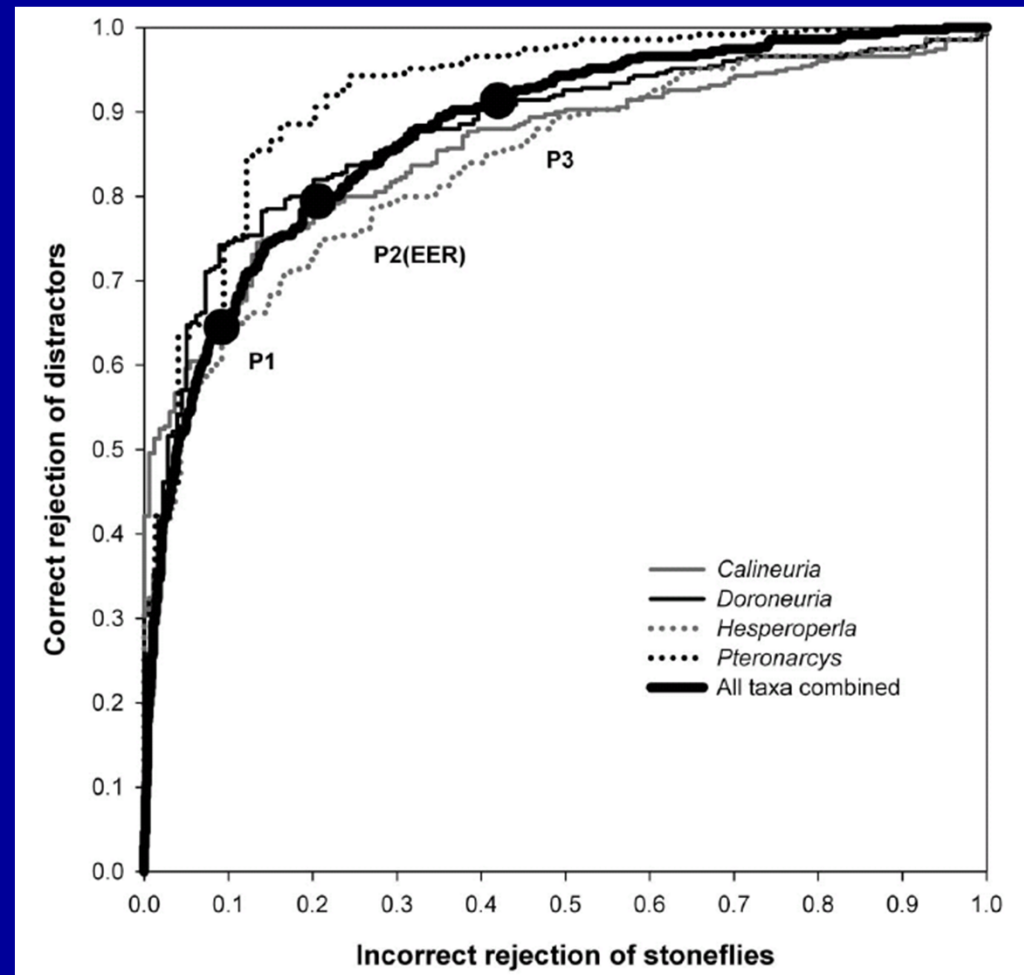


Confusion Matrix



Challenge Problem: Detecting and Rejecting “Novel” Species

- ◆ Can the system detect that a specimen does not belong to any of the training classes?
- ◆ Stonefly 9 with 10 “Distractor Classes”
- ◆ P2: Equal-Error Rate 21.3%



Novelty Detection Methods

- ◆ Density estimation (applied to BoW histograms)
 - Projection Pursuit Density Estimation (Friedman, Stuetzle & Schroeder, 1984)
 - Boosted Density Estimation (Rosset & Segal, 2002)
 - PCA + GMM
 - Manifold Embedding + GMM
 - Mixtures of Factor Analyzers
- ◆ Density ratio estimation
 - uLSIF (Hido et al, 2010)
- ◆ Reconstruction error methods
 - PCA + reconstruction
 - Sparse coding + reconstruction error
- ◆ One-class SVM

Preliminary Results

| Method | Equal Error Rate (accept/reject) |
|---------------------------------------|----------------------------------|
| Supervised classification lower bound | ~3.5% |
| PCA + GMM | 16.3% |
| Gaussian Naïve Bayes + tricks | 21.3% |
| Boosted GMMs | Numerical problems |
| PCA + reconstruction error | 29.2% |
| Sparse Coding + reconstruction error | 40.0% |
| uLSIF | >38.0% |
| One-class SVM | >34.6% |

Next Steps

◆ EPTs

- EPT52 data set
- Field studies using EPA data
- Comprehensive rejection experiments

◆ Soil Mesofauna

- Samples collected; awaiting photography

◆ Other Applications

- Freshwater Zooplankton
- Flies
- Moths
- Mosquitoes
- Soil Mesofauna

Evidence Trees: A New Machine Learning Paradigm

- ◆ General Principle:
 - Store evidence in the leaves of random forest trees
 - Combine evidence via non-parametric method to make final decision
- ◆ The purpose of the tree is NOT to make a decision but to identify the evidence relevant to making the decision

Another Example: Hough Forests

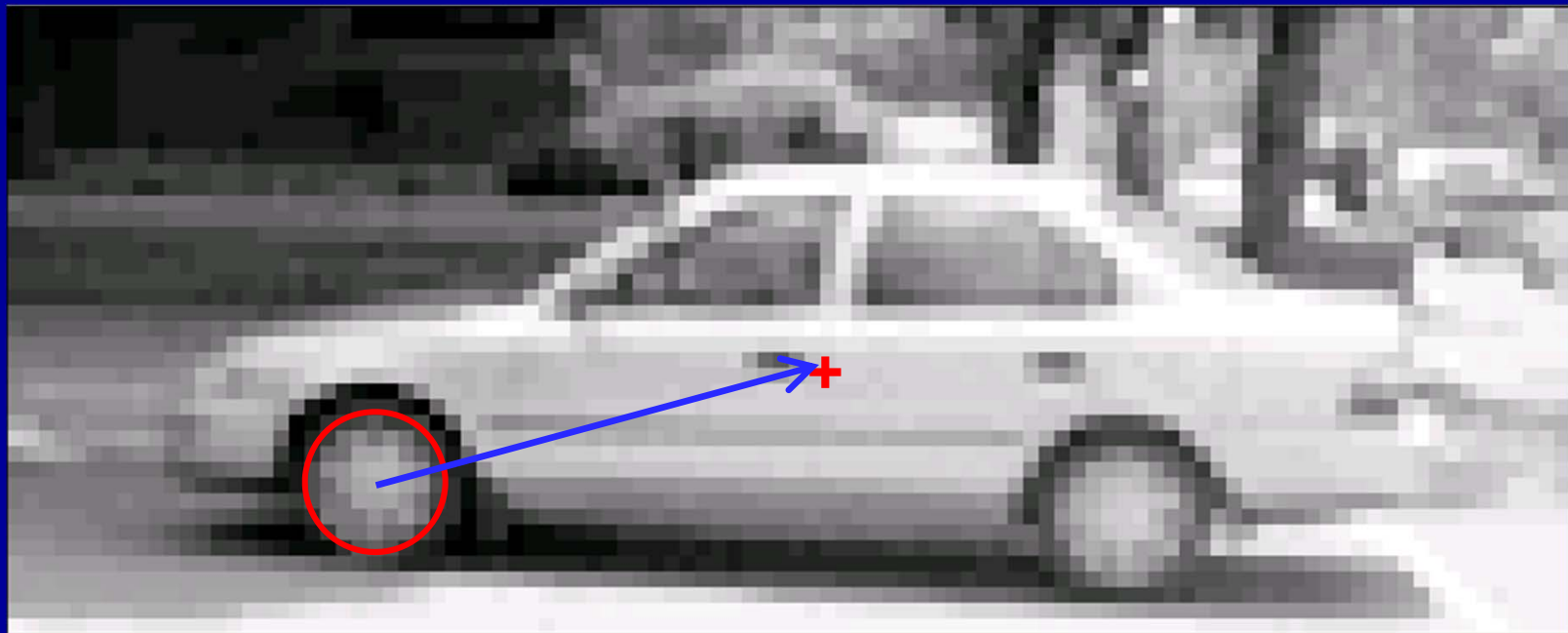
[Gall & Lempitsky, CVPR 2009]

- ◆ Task: Object Detection (aka Localization)
 - Find all instances of object class in image



Training Examples

- ◆ At each interest point, compute (dx, dy, class)



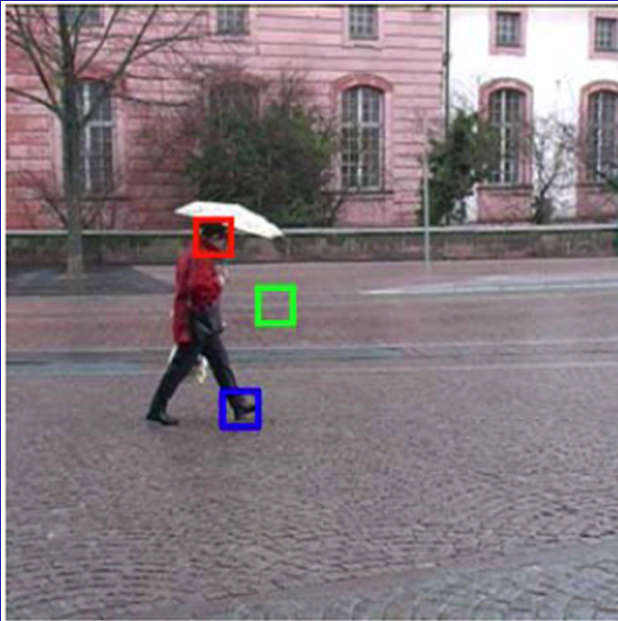
Evidence Trees

- ◆ Training criterion
 - all examples in a leaf should
 - belong to the same class
 - have similar (dx,dy) offsets (2-D variance)
- ◆ Note: All training images are scaled to a fixed scale based on the size of the car

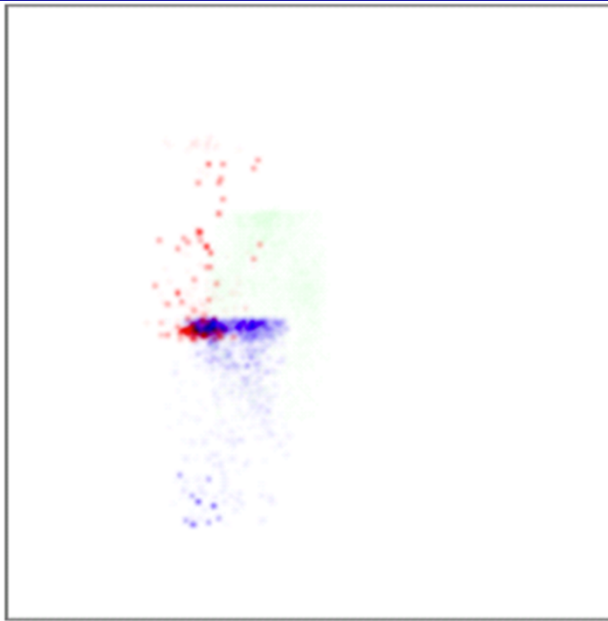
Predicting New Images

- ◆ For each interest region (x, y) in test image
 - Drop SIFT vector through each tree
 - For each (dx, dy, k) stored in leaf
 - Predict that an object belonging to class k is located at $(x + dx, y + dy)$
- ◆ Apply mode-finding algorithm (e.g., mean shift) to find peaks in the distribution of predictions
- ◆ Repeat at multiple scales; choose best scale; predict a car at the top N peaks

Example for Pedestrian Detection



(a) – Original image with three sample patches emphasized



(b) – Votes assigned to these patches by the Hough forest



(c) – Hough image aggregating votes from all patches

Gall & Lempitski, CVPR 2009

Tree Splitting

- ◆ Gall & Lempitski:
 - alternate between splitting on class information gain and splitting on variance of (dx,dy)
- ◆ Our work (Martinez & Dietterich)
 - split to maximize information gain:
 $I(\text{split} ; dx, dy, \text{class})$

Results: UIUC Cars (multiple)

| Method | Equal Error Rate |
|--------------------------------------|------------------|
| Mutch & Lowe (CVPR 06) | 90.6% |
| Lampert, et al. (CVPR 08) | 98.6% |
| Gall & Lempitsky (CVPR 09) | 98.6% |
| Stacked Evidence Trees (unpublished) | 98.5% |
| Stacked Decision Trees (unpublished) | 89.5% |

We can probably improve the results by using the re-centering technique employed by Gall & Lempitsky

Conclusions

- ◆ Computer vision and machine learning methods can achieve high accuracy classification of stoneflies
 - two methods scoring ~5% error on 9 classes
- ◆ Similar techniques achieve ~12% error on 29 classes of EPTs
- ◆ 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
- ◆ Major challenge: novel class detection / rejection

Acknowledgements

- ◆ Grant Support: US National Science Foundation