

A Class of One's Own

Image Classification Using Forward Recursion

Christopher Tsai
June Zhang

June 4, 2008

EE 378 – Statistical Signal Processing

Attending Class without a Teacher

- Unsupervised Learning
 - Prescribe only the *number* of classes
 - Classes will emerge themselves
- Motivation
 - Scientific curiosity
 - Classes not known in complicated images
 - Interesting subclasses may emerge
- No manual assignment
 - Number of classes can be suggestive
 - Definite structure emerges

Problem 1 - Three-Way Linear Classification

no. 2

Spatial Dependence

- Features localized to a block
 - (4 × 4) or (8 × 8) block
- First-order Markov model assumed
 - Dependence only on *one* neighbor
- Hidden Markov State X = Class
- Markov Observation Y = Features
- Scan methods

\bar{Y}_1 = Features Observed in Block (1, 1)
 X_1 = State of Block (1, 1)

\bar{Y}_2 = Features Observed in Block (1, 2)
 X_2 = State of Block (1, 2)

\bar{Y}_3 = Features Observed in Block (2, 1)
 X_3 = State of Block (2, 1)

\bar{Y}_4 = Features Observed in Block (3, 1)
 X_4 = State of Block (3, 1)

no. 3

Classification System Diagram

Training Images

[kClasses / blockSize / nFeatures / scanMode]

Feature Extraction

```
@getRGB
@getHSV
@getLumin
@getDCT
@strelConvolve
```

Parameter Estimation

Repeat until convergence

Bayes-Gauss Likelihoods

Vector Quantization

Lloyd-Max Binning

Bin Statistics

Test Image

Classification

Forward Recursion

$$\alpha_i(x_i) = \frac{\beta_i(x_i)p(y_i|x_i)}{\sum_{j=1}^k \beta_j(x_i)p(y_j|x_i)}$$

$$\beta_{i+1}(x_{i+1}) = \sum_{x_i} \alpha_i(x_i)p(x_{i+1}|x_i)$$

$p(x_{i+1}|x_i)$

$p(y_i|x_i)$

no. 4

Classification System Diagram

Training Images

[kClasses / blockSize / nFeatures / scanMode]

Feature Extraction

```
@getRGB
@getHSV
@getLumin
@getDCT
@strelConvolve
```

Parameter Estimation

Repeat until convergence

Bayes-Gauss Likelihoods

Vector Quantization

Lloyd-Max Binning

Bin Statistics

Test Image

Classification

Forward Recursion

$$\alpha_i(x_i) = \frac{\beta_i(x_i)p(y_i|x_i)}{\sum_{j=1}^k \beta_j(x_i)p(y_j|x_i)}$$

$$\beta_{i+1}(x_{i+1}) = \sum_{x_i} \alpha_i(x_i)p(x_{i+1}|x_i)$$

$p(x_{i+1}|x_i)$

$p(y_i|x_i)$

no. 5

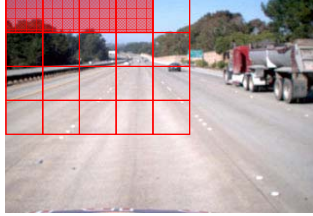
Turning the Knob – Choice of Parameters

- blockSize – How finely should we sample our image?
 - Smaller regions → better point accuracy (like with color)
 - Larger regions → larger-scale features (like counting edges)
- kClasses – How many categories should our algorithm seek?
- nFeatures – Which quantities best separate the objects?
- nTrainingImages – How much variation does the data contain?
- scanMode – Horizontal, Vertical, or Diagonal?

no. 6

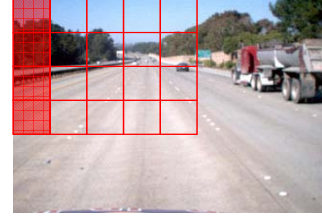
Parameter Selections

- Number of classes
- Image block size → nBlocks
- Type of feature(s)
- Number of training image(s)
- 'horizontal, vertical, diagonal' scan mode



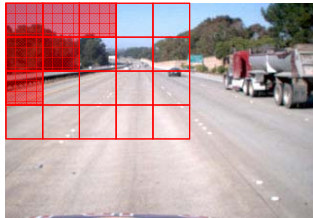
Parameter Selections

- Number of classes
- Image block size → nBlocks
- Type of feature(s)
- Number of training image(s)
- 'horizontal, vertical, diagonal' scan mode

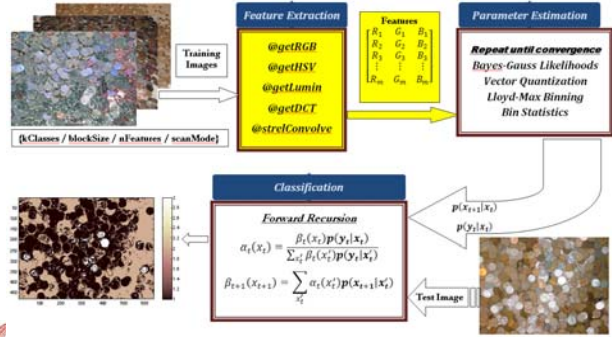


Parameter Selections

- Number of classes
- Image block size → nBlocks
- Type of feature(s)
- Number of training image(s)
- 'horizontal, vertical, diagonal' scan mode



Classification System Diagram



Feature Extraction – Function Pointers

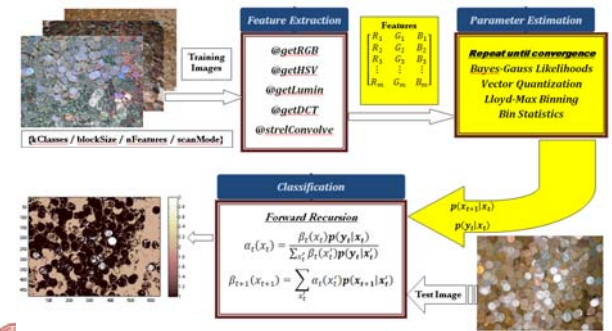
```
featureVector = feval(featureFunc, currentBlock);
y = feval(@getRGB, currentBlock);
```

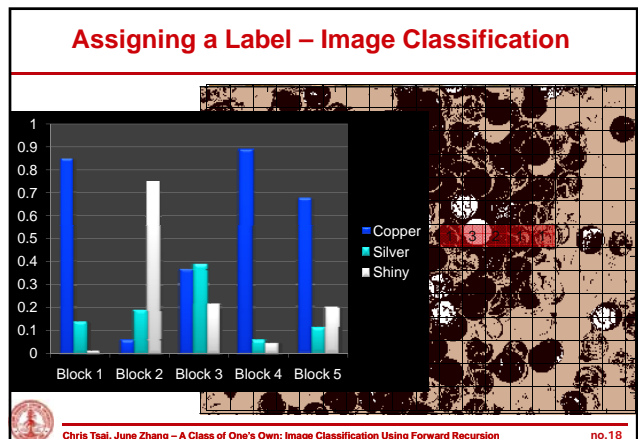
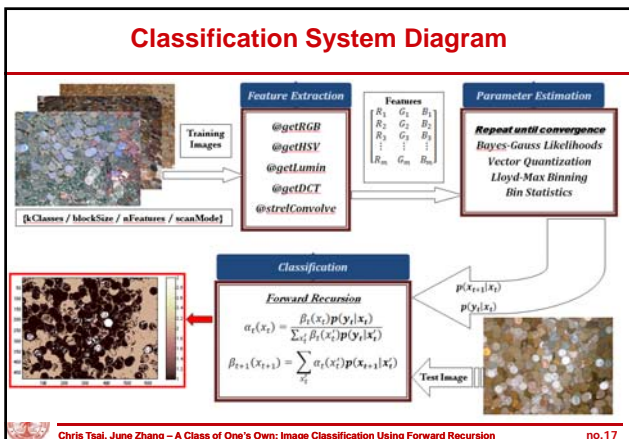
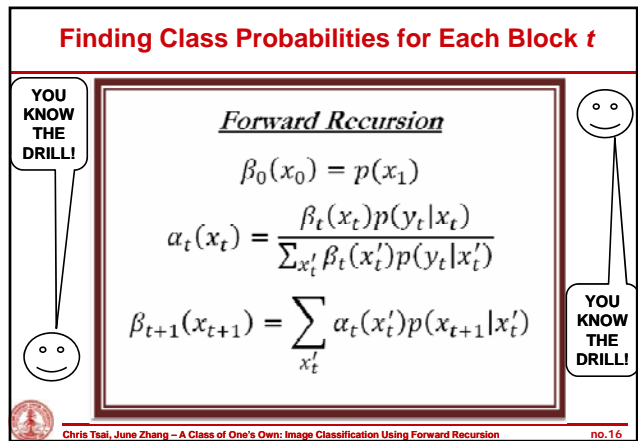
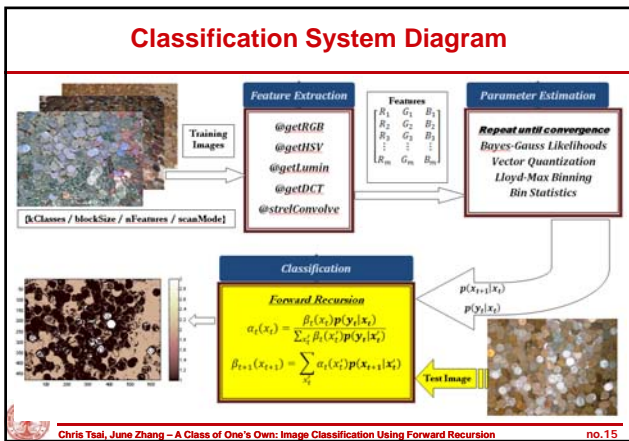
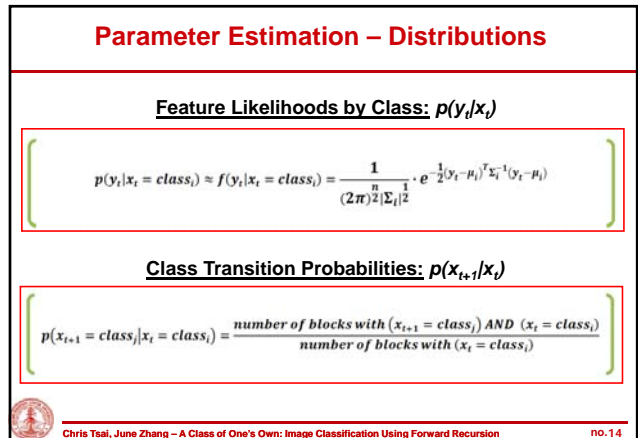
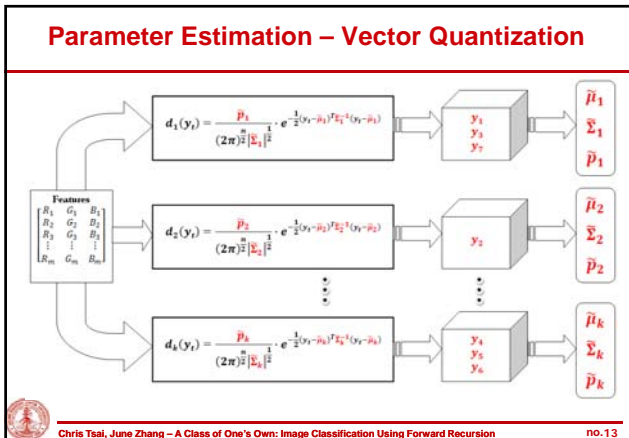


- Application-dependent**
- Block-Mean Red/Green/Blue
- Block-Mean Luminance
- Block-Mean Hue
- Block DCT (1, 1) Coefficient
- Block DCT Directional Coefficients
- 2-D Haar Wavelet Coefficients
- Texture: Smoothness/Roughness
- Response to Structuring Element



Classification System Diagram

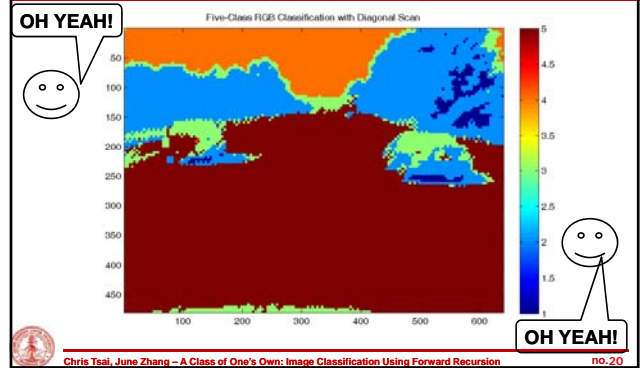




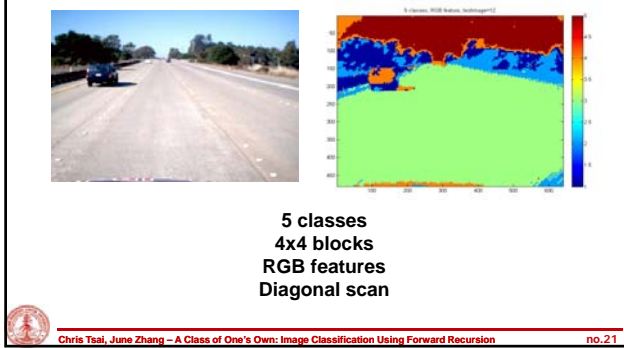
Taking the Test – Experimental Data



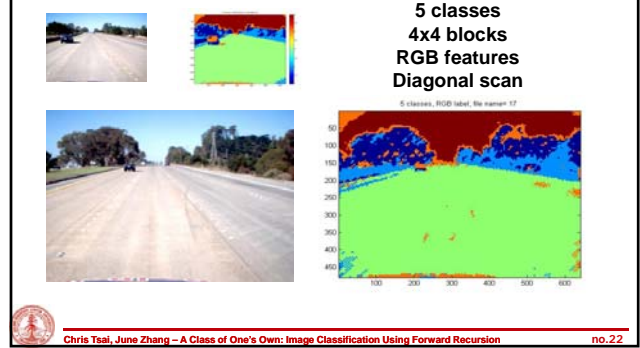
Taking the Test – Experimental Data



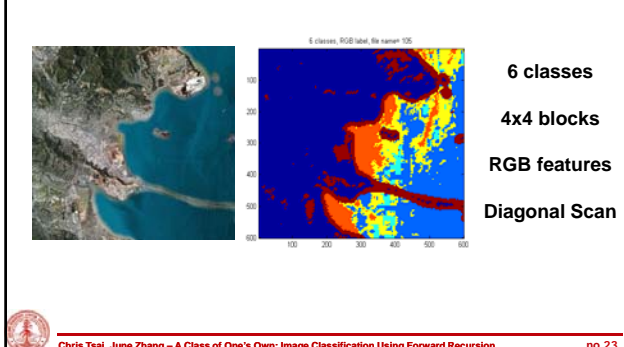
Car Images: Training (RGB)



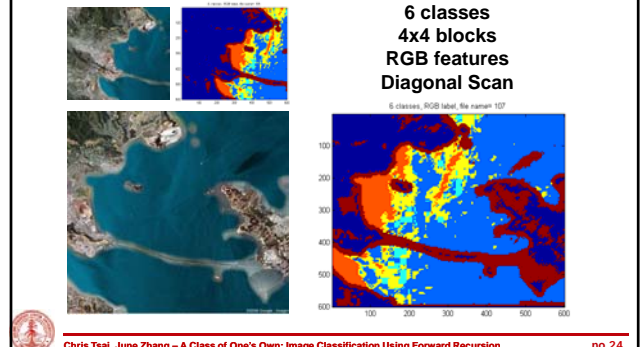
Car Images: Testing (RGB)



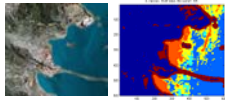
Aerial Images: Training (RGB)



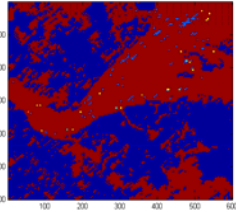
Aerial Images: Testing (RGB)



Aerial Images: Testing (RGB)



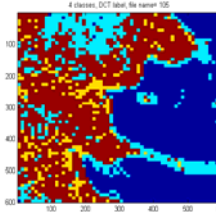
6 classes
4x4 blocks
RGB features
Diagonal Scan



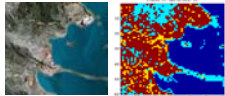
Aerial Images: Training (DCT)



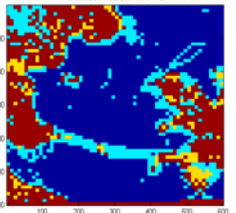
4 classes
10x10 blocks
DCT features
Diagonal Scan



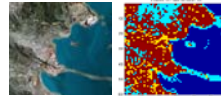
Aerial Images: Testing (DCT)



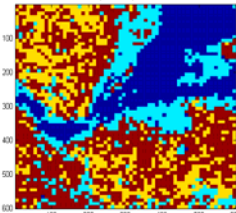
4 classes
10x10 blocks
DCT features
Diagonal Scan



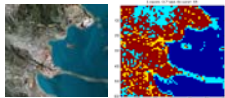
Aerial Images: Testing (DCT)



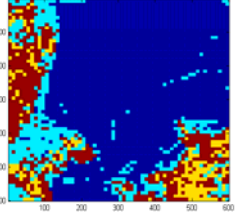
4 classes
10x10 blocks
DCT features
Diagonal Scan



Aerial Images: Testing (DCT)



4 classes
10x10 blocks
DCT features
Diagonal Scan



The End

- Thanks for listening. Do you have any questions? Raise your hand!



- Special thanks to Kivanc Ozonat for his assistance.

