Automatic Photo Quality Assessment

Presenter: Yupu Zhang, Guoliang Jin, Tuo Wang

Computer Vision – 2008 Fall

Estimating the photorealism of images:

Distinguishing paintings from photographs

Florin Cutzu, Riad Hammoud, Alex Leykin
Department of Computer Science
Indiana University
Bloomington, IN 47408

Presenter: Yupu Zhang
yupu@cs.wisc.edu

Outline

- Introduction
- Distinguishing Features
- Proposed Classifiers
- Classifier Performance
- Psychophysical Experiments

Introduction

- Photorealism
  - a style of painting in which an image is created in such exact detail that it looks like a photograph

Introduction

- Class Definition
  - Photograph (degree of photorealism is high)
    - color photographs of 3D real-world scenes
  - Painting (degree of photorealism is low)
    - conventional canvas paintings, frescoes and murals
- Goal
  - Automatically differentiate photographs from paintings without constraints on the image content

Distinguishing Features

- Three sets of features for discriminating paintings from photographs:
  1. Visual features
  2. Spatial-color feature
  3. Texture-based feature

Visual features (1)

- If we convert a color picture to gray-scale
  - edges in photos are still clear
  - most of the edges in paintings are eliminated
- Color edges vs intensity edges
  - Paintings: color edges, due to color changes
  - Photos: intensity edges, resulting from intensity changes
- Quantify : **Eg**

$$E_g = \frac{\#\text{ pixels: intensity, not color edge}}{\text{total number of edge pixels}}$$

Paintings will have smaller Eg

Visual features (2)

- Color changes to a larger extent from pixel to pixel in paintings than in photos
- Spatial variation of color : **R**
  - For each pixel and its 5x5 neighborhood, calculate the local variation of color around the pixel
  - R is the average of the variations taken over all pixels
    - Paintings have larger R

Visual features (3)(4)

- Number of unique colors : **U**
  - Paintings have a larger color palette than photos
  - U = # of unique colors / # of pixels
- Pixel saturation : **S**
  - Paintings contain a larger percentage of highly saturated pixels
  - RGB => HSV
  - Create a saturation histogram, using 20 bins
  - **S** is the ratio between the highest bin and the lowest bin

Spatial-color feature

- RGB (3D) => RGBXY (5D)
  - X and Y are the two spatial coordinates
- Calculate a 5x5 covariance matrix of the RGBXY space
- The singular value could represent the variability of the image pixels in both color space and "location space"
  - paintings should have a larger singular value

Texture-based feature

- Observation
  - In photos texture elements tend to be repetitive
  - In paintings it's difficult to maintain texture uniformly
- Method
  - Gabor filter: extract textures from images
  - Calculate the mean and the standard deviation of the result of the filter over all pixels
    - Photos tend to have larger values at horizontal and vertical orientations
    - Paintings should have larger values at diagonal orientations

Proposed Classifiers

- Individual classifier:
  - {Eg, U, R, S} space
    - a combination of four thresholds
  - RGBXY space
    - singular value
  - Gabor space
    - means and standard deviations
- Implementation
  - Neural network
  - Training

Proposed Classifiers

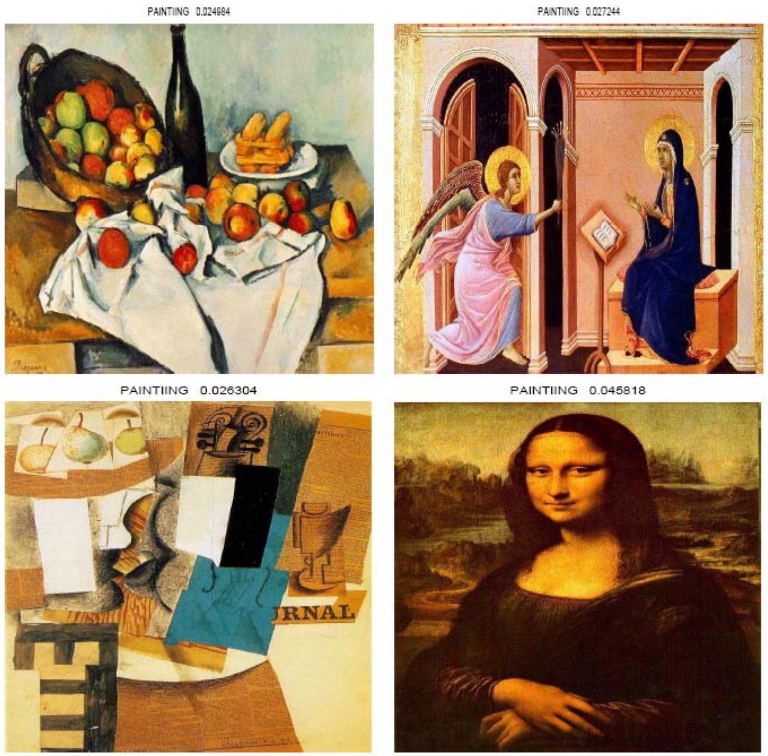
- Combine all three classifiers
  - the "committees" of neural networks
- How?
  - individual classifier gives a score between 0 and 1
    - 0 perfect painting, 1 perfect photo
  - take the average of the three scores
    - <= 0.5 => painting
    - > 0.5 => photo

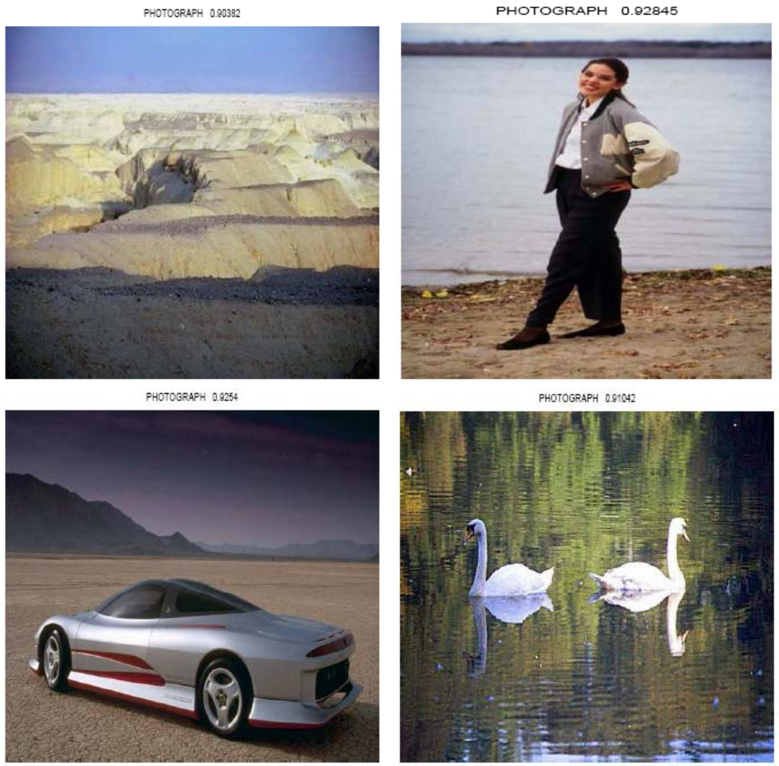
Classifier Performance

- C_1 {Eg, U, R, S}
- C_2 RGBXY
- C_3 Gabor
- C committee

Classifier	P hit rate ($\mu \pm \sigma$)	Ph: hit rate ($\mu \pm \sigma$)
C_1	72±5%	71±4%
C_2	81±3%	81±3%
C_3	79±5%	78±4%
C	94±3%	92±2%

Classifier Performance

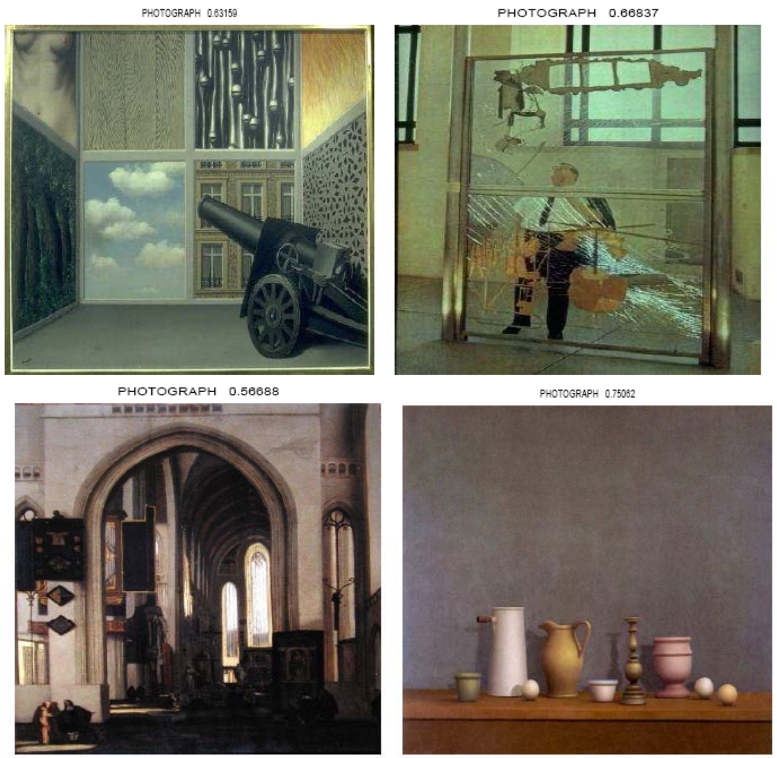




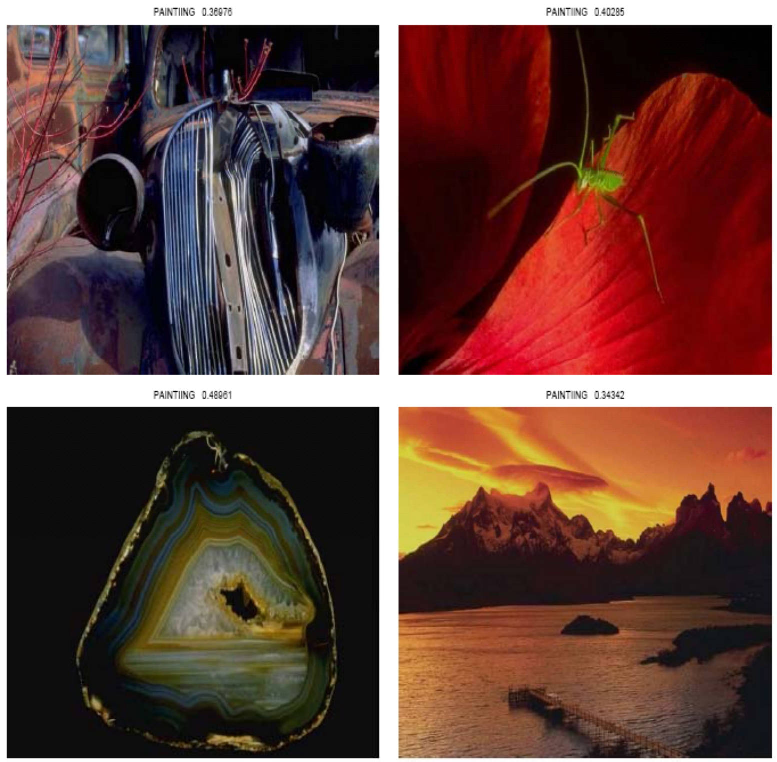
paintings (score < 0.1)　　photos (score > 0.9)

0 perfect painting, 1 perfect photo

Classifier Performance



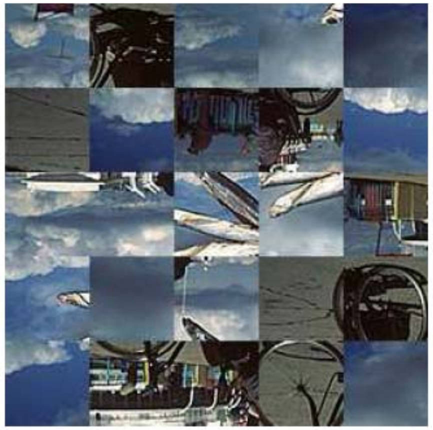
paintings classified as photos



photos classified as paintings

Psychophysical Experiments

- The mistakes made by the classifiers seems to reflect the degree of perceptual photorealism of the image
- How to verify this?
  - psychophysical experiment
  - human testers read
    - scrambled image patches
    - content independent
  - give scores [0, 10]
    - 0 perfect painting
    - 10 perfect photo
  - calculate the correlation coefficient between human ratings and classifier outputs
  - result: 0.865



Studying Aesthetics in Photographic Images Using a Computational Approach

Presenter: Guoliang Jin

What they did

Established significant correlation between various visual properties of photographic images and their aesthetics ratings.

- using a community-based database and ratings
- extracting certain visual properties
- build a classifier that can qualitatively distinguish between pictures of *high* and *low* aesthetic value

Community-based Photo Ratings

Data Source: *Photo.net*

- A large online photo sharing community
- Primarily intended for photography enthusiasts
- More than 400, 000 registered members
- Photographers share photos, and rate and comment on photos taken by peers

The rating system in *Photo.net*

Photo peer-rated in terms of two qualities, namely *aesthetics* and *originality*

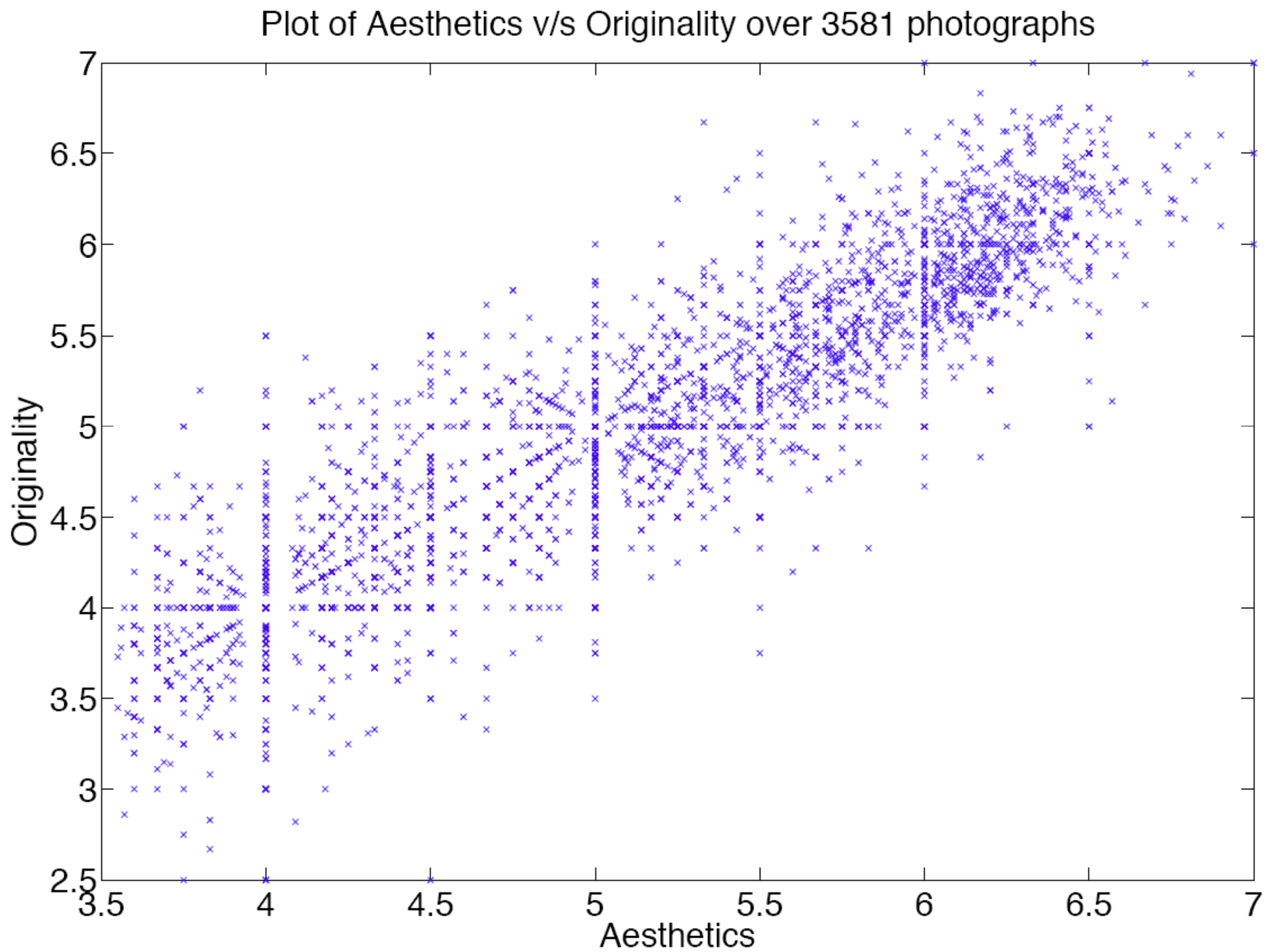
- In the range of one to seven, with a higher number indicating better rating
- Pro: photos are rated by a relatively diverse group which ensures generality in the ratings
- Con: the acquired data was fairly noisy

How they use *Photo.net*

Download pictures and associated metadata

- average aesthetics score between 1.0 and 7.0
- average originality score between 1.0 and 7.0
- number of times viewed by members
- number of peer ratings

Aesthetics and Originality



Correlation between the aesthetics and originality ratings for 3581 photographs.

Visual Feature Extraction

- Choice of features was principled, based on
  1. rules of thumb in photography
  2. common intuition
  3. observed trends in ratings
- They extracted 56 visual features for each image

  refer them as candidate features

  denote them as $F = \{f_i \mid 1 \leq i \leq 56\}$

Visual Feature Extraction (Cont.)

- The RGB data of each image is converted to HSV color space, producing two-dimensional matrices IH, IS, and IV, each of dimension X × Y

  color tones and saturation play important roles, and hence working in the HSV color space makes computation more convenient
- The image is also transformed into the LUV space, since in this space locally Euclidean distances model the perceived color change well, so it will be easy to use a fast segmentation method based on clustering

Visual Feature Extraction (Cont.)

- Exposure of Light and Colorfulness $\{f_1, f_2\}$
- Saturation and Hue $\{f_3, f_4\}$
- The Rule of Thirds $\{f_5 \sim f_7\}$
- Familiarity Measure $\{f_8 \sim f_9\}$
- Wavelet-based Texture $\{f_{10} \sim f_{21}\}$
- Size and Aspect Ratio $\{f_{22}, f_{23}\}$
- Region Composition $\{f_{24} \sim f_{52}\}$
- Low Depth of Field Indicators $\{f_{53} \sim f_{55}\}$
- Shape Convexity $\{f_{56}\}$

Feature Selection

To discover features that show correlation with community-based aesthetics scores

- Use a one-dimensional support vector machine (SVM)
- SVMs are essentially powerful binary classifiers that project the data space into higher dimensions where the two classes of points are linearly separable
- Two classes: *high containing samples with aesthetics scores greater than* 5*.8, and low with scores less than 4.2*
- The top 15 among the 56 features in terms of model accuracy are obtained

Feature Selection, Classification, and Regression

- A classifier that can separate *low from high*
- Use SVM as well as the classification and regression trees (CART) algorithm
- *Filter-based methods* and *wrapper-based methods* are two broad techniques for feature selection
- Stop at 15 iterations (i.e. 15 features) and use this set to build the SVM-based classifier
- Use the recursive partitioning (RPART) implementation to build a two-class classification tree model for the same set of 1664 data samples
- Perform linear regression on polynomial terms of the features values to see if it is possible to directly predict the aesthetics scores in the 1 to 7 range from the feature vector

Measure the quality of regression

- *residual sum-of-squares error*

$$R_{res}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

- worst case $\overline{Y}$ *is chosen every time without using the* regression model, yielding $R_{res}^2 = \sigma^2$ *(variance of Y).*
- if the independent variables explain something about *Y , it must be that* $R_{res}^2 \leq \sigma^2$

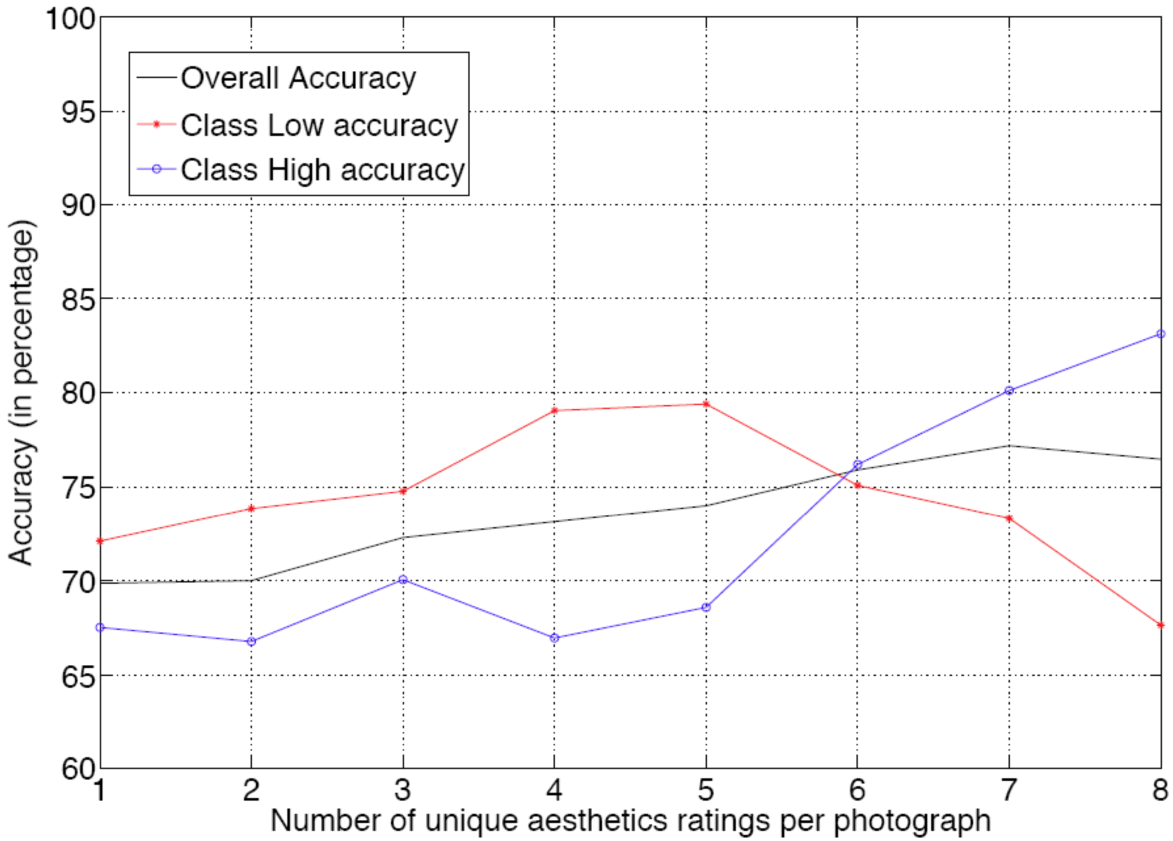
Experimental Results

- The top 15 classification rates achieved by *{*f_{31}, f_1, f_6, f_{15}, f_9, f_8, f_{32}, f_{10}, f_{55}, f_3, f_{36}, f_{16}, f_{54}, f_{48}, f_{22}*}*, with accuracy over 54%.
- The maximum classification rate achieved by any single feature was f_{31} *with 59.3%.*
- They act as weak classifiers and hence show some correlation with the aesthetics.

Experimental Results (Cont.)

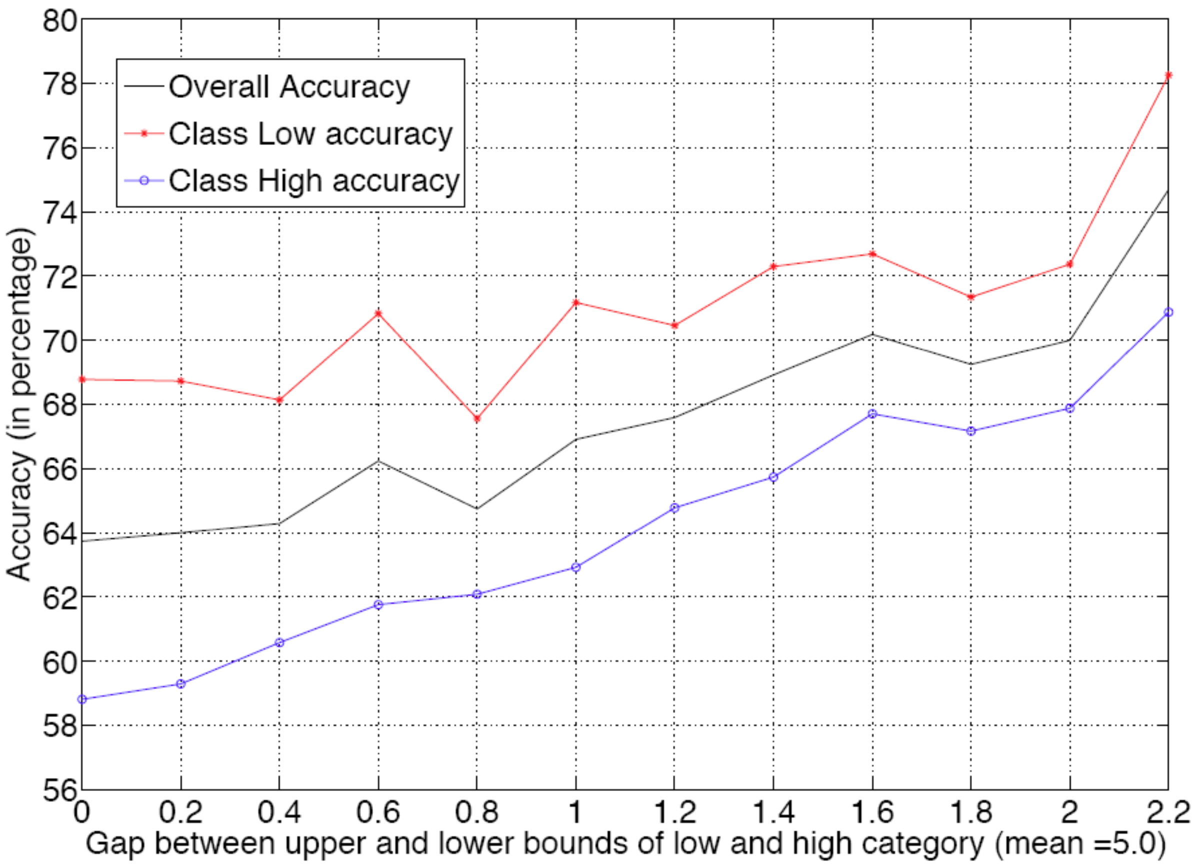
- The combined filter and wrapper method for feature selection yielded the following set of 15 features:$\{f_{31}, f_{1}, f_{54}, f_{28}, f_{43}, f_{25}, f_{22}, f_{17}, f_{15}, f_{20}, f_{2}, f_{9}, f_{21}, f_{23}, f_{6}\}$. The accuracy achieved with 15 features is 70.12%, with precision of detecting *high* class being 68.08%, and *low* class being 72.31%.

Experimental Results (Cont.)



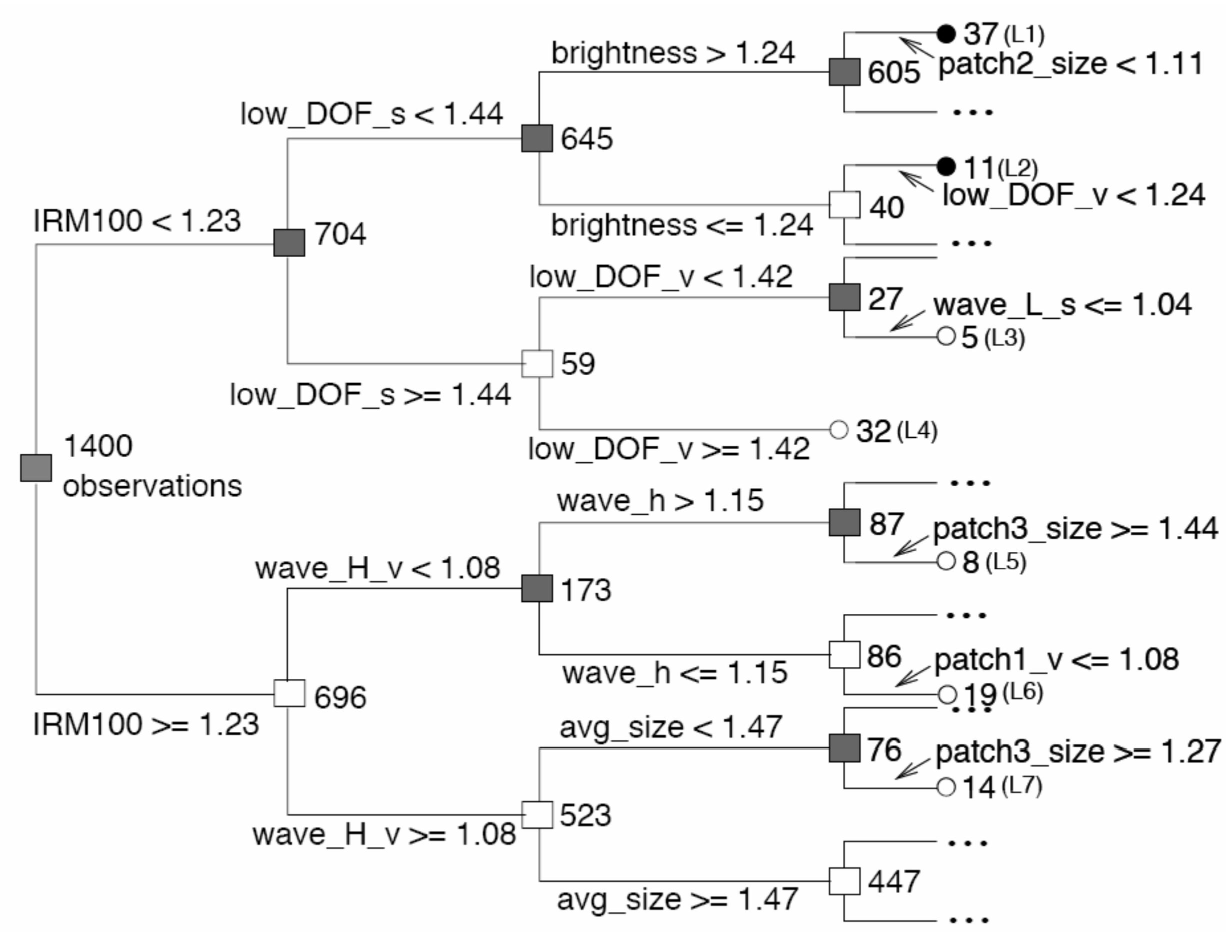
Variation of *accuracy with the minimum number of unique* ratings per picture

Experimental Results (Cont.)



Variation of *SVM accuracy with inter-class gap δ.*

Experimental Results (Cont.)



Decision tree obtained using CART and the 56 visual features (partial view)

Experimental Results (Cont.)

- The variance σ^2 *of the aesthetics* score over the 3581 samples is 0*.69.*
- With 5 polynomial terms for each of the 56, achieves a residual sum-of-squares $R_{res}^2 = 0.5020$
- Randomly permuted the aesthetics scores (breaking the correspondence with the features) and performed the same regression. This time, R_{res}^2 *is 0.65, clearly showing that the reduction in expected error was not merely* by the over-fitting of a complex model.

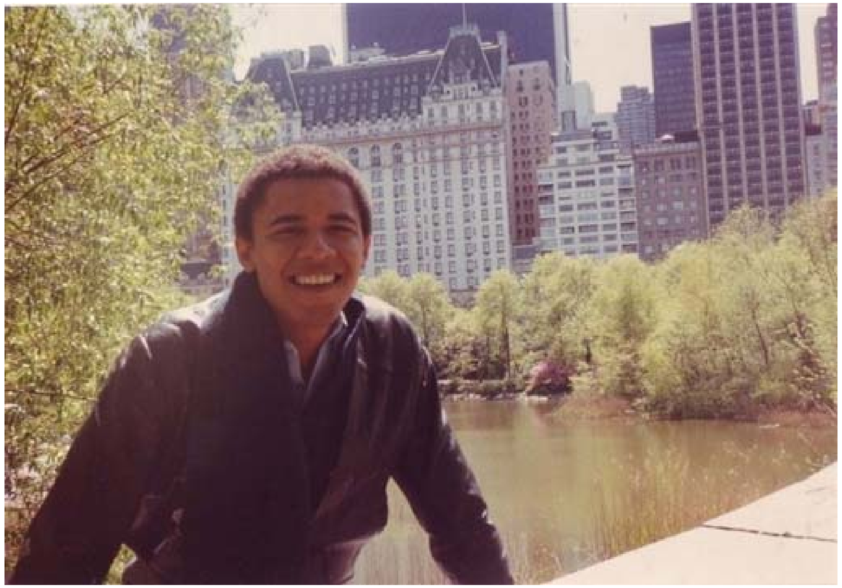
Conclusion

- Despite the inherent noise in data, our SVM-based classifier is robust enough to produce good accuracy using only 15 visual features in separating *high* and *low* rated photographs.

The Design of High-Level Features for Photo Quality Assessment

Yan Ke, Xiaoou Tang, Feng Jing

Presented by Tuo Wang
Computer Vision – Fall 2008




AC CA 645

Anyone can take great photos

… if you can recognize the good ones.

Outline

- High and low quality photo
- Criteria between high and low quality photo
- Proposed feature
- Classifier
- Experiment dataset
- Results

High and low quality photo

- Images classification:
  - ➢ photos or graphics
  - ➢ taken indoors or outdoors
  - ➢ city or landscape
  - ➢ photos or paintings
  - ➢ real or rendered photo
  - ➢ …

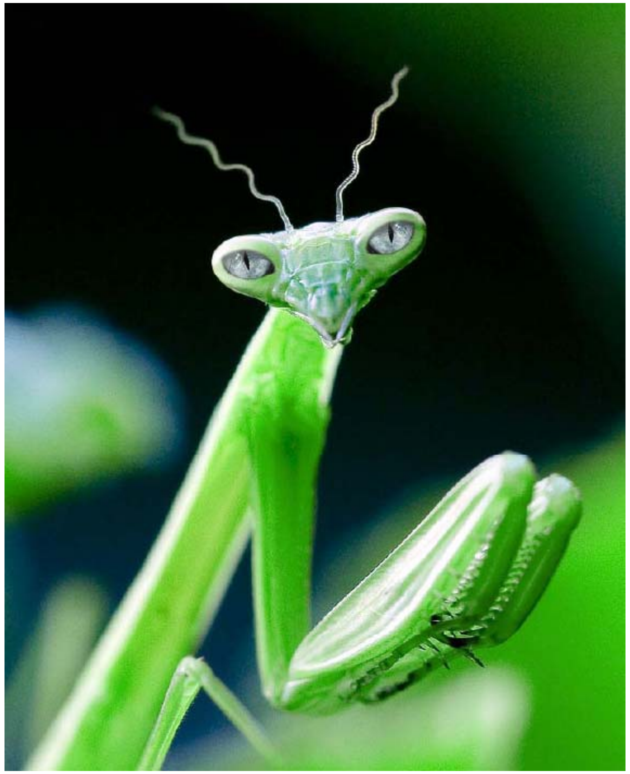
- What makes a high quality photo?

  Before design features to assess a photo’s quality, we must determine the perceptual criteria that people use to judge photos.

What makes a high quality photo

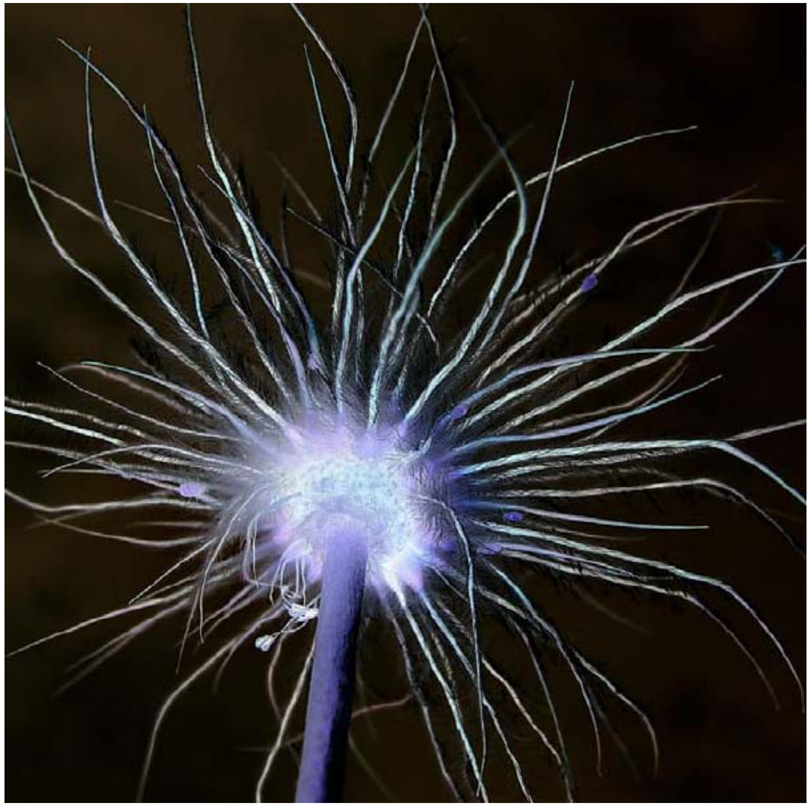
- Three distinguishing factors between the high quality of photos and low quality of photos
  - ➢ Simplicity
  - ➢ Realism
  - ➢ Basic photographic technique

Simplicity
- Background out of focus



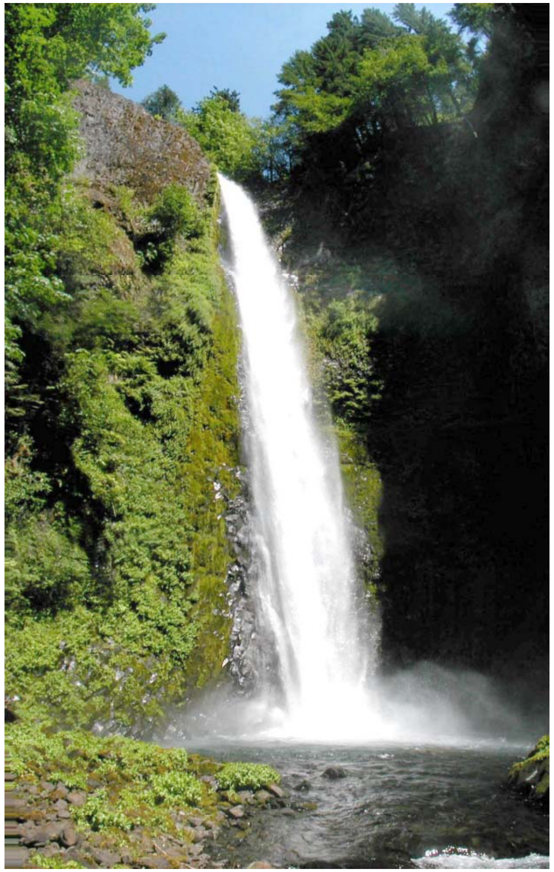
Simplicity
- Color contrast

Simplicity
- Lighting contrast

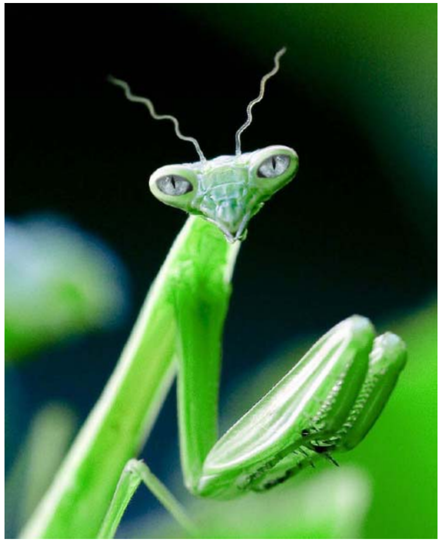
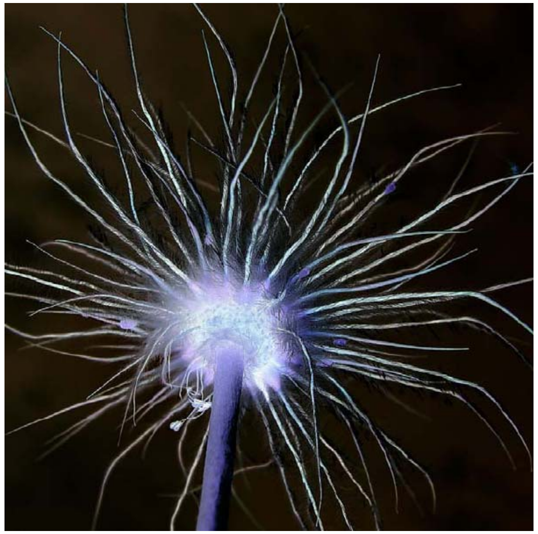



Realism
- Color palette

Realism
- Camera settings



Realism
- Subject matter


Basic photographic technique -- Blur

Basic photographic technique
-- Contrast and Brightness

What features can we extract

- Spatial Distribution of Edges
- Color Distribution
- Hue Count
- Blur
- Low Level Features

Features – Spatial Distribution of Edges

- Edge spatial distribution feature extractor:
  - 3*3 Laplacian filter
  - Resize Laplacian image size to 100*100
  - Normalize the image sum to 1
  - The quality of probe image: $q_l = d_s - d_p$, where:

$$d_s = \sum_{x,y} |L(x,y) - M_s(x,y)| \quad d_p = \sum_{x,y} |L(x,y) - M_p(x,y)|$$

M_p and M_s are the mean Laplacian image of the professional photos and snapshots

Features – Spatial Distribution of Edges

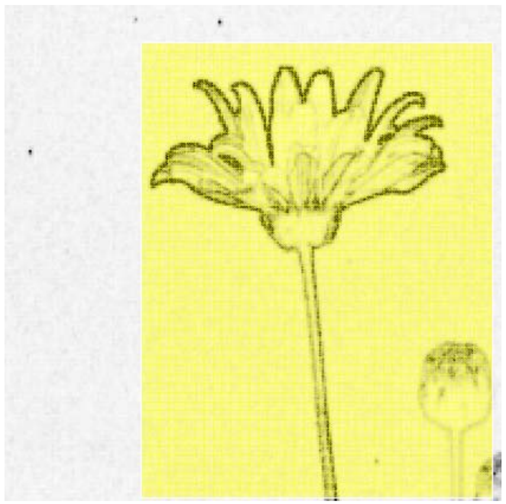
Features – Spatial Distribution of Edges

- The amount of area that the edges occupy

$$P_x(i) = \sum_y L(i, y) \qquad P_y(j) = \sum_x L(x, j)$$

w_x and w_y be the width of 98% mass of the projections P_x and P_y respectively. So the bounding box: $w_x w_y$, the quality of the image q_a: 1- $w_x w_y$.

Features – Spatial Distribution of Edges

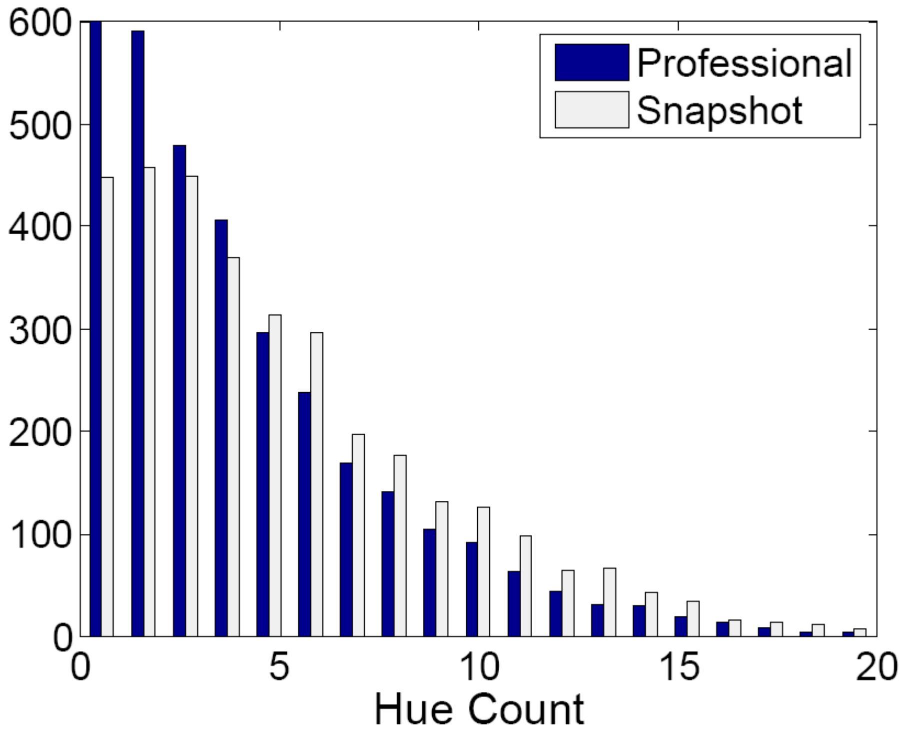


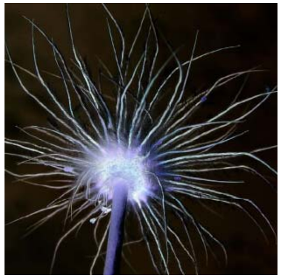
Features – Color Distribution

- Quantize the red, green, and blue channels into 16 values
- A 4096 = 16^3 bin histogram is created
- Normalize the histogram to unit length
- Use the L1 metric to calculate the distance between histograms
- *k*NN on color histogram
- $q_{cd} = n_p - n_s$

  = #professional_neighbors - #snap_neighbors

Features – Hue Count





||N||= (# *hues* > *threshold*)

q_h = 20 – ||N||

Features – Blur

- Model a blurred image I_b as the result of a Gaussian smoothing filter G_σ applied to an otherwise sharp image I_s,

$$I_b = G_\sigma * I_s$$

$$F = FFT(I_b)$$

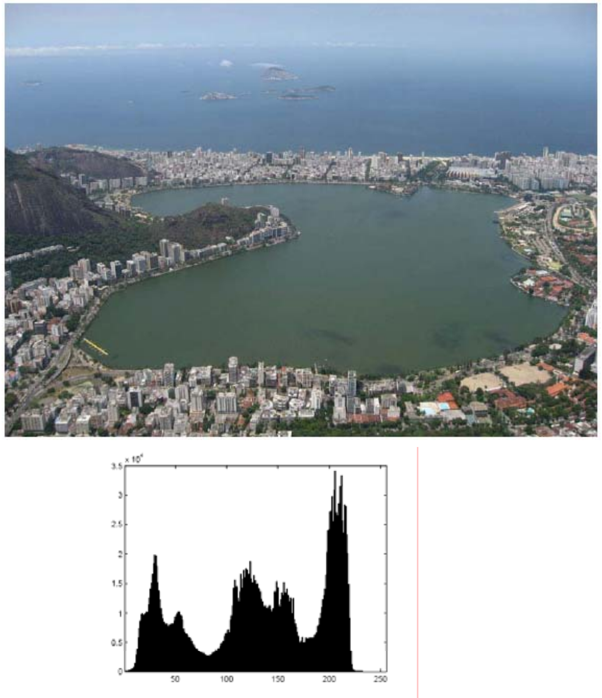
$$C = \{(u, v) \mid |F(u, v)| > \theta\}$$

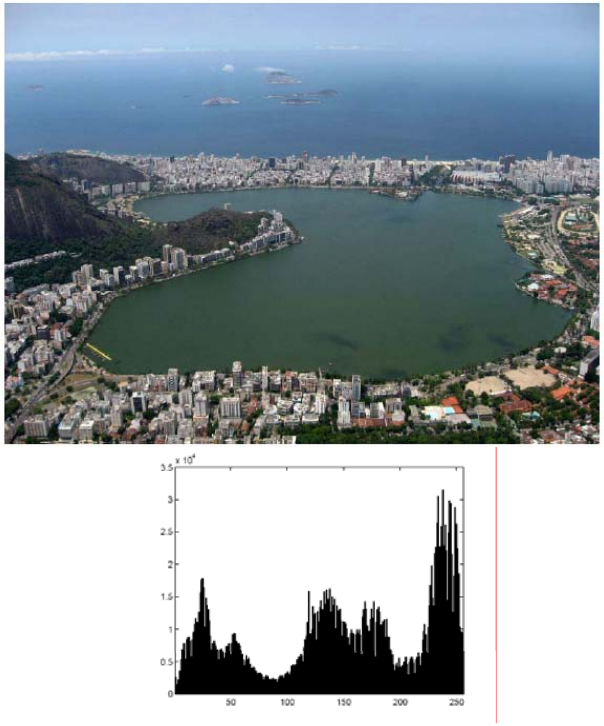
$$q_f = \frac{||C||}{||I_b||} \sim \frac{1}{\sigma}$$

Low level features - Contrast

$$H(i) = H_r(i) + H_b(i) + H_g(i)$$

- q_{ct}: equal to the width of middle 98% mass of the histogram





Low level features – Average Brightness

- Average brightness: *b*

Classifier

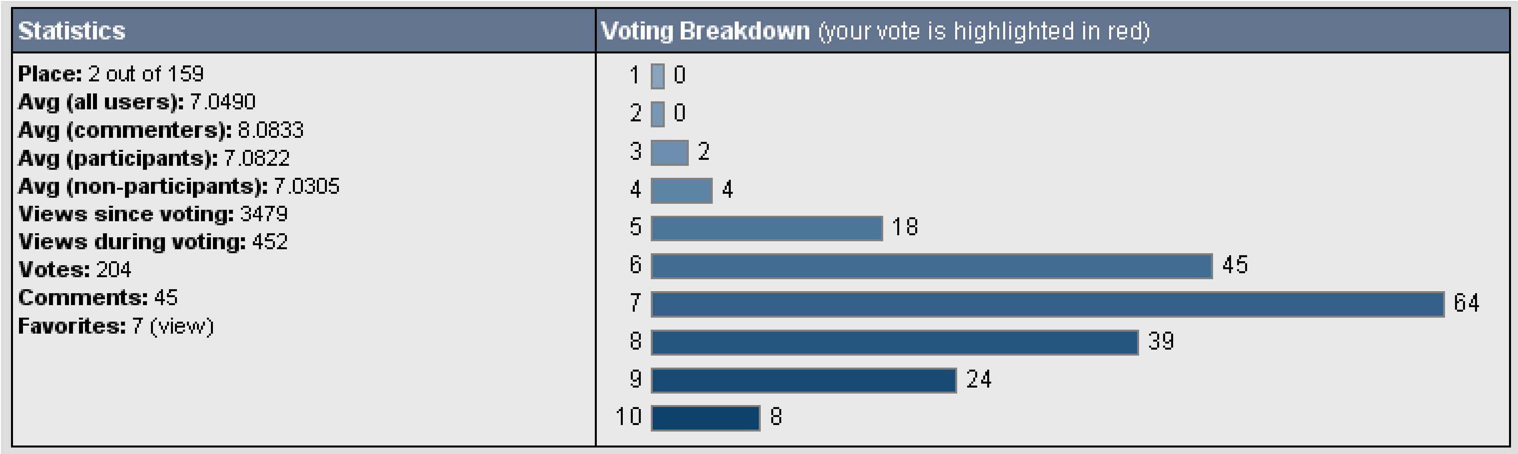
- Naives Bayes
- Assume independence of the features

$$
\begin{aligned}
q_{all} &= \frac{P(Prof \mid q_1 \ldots q_n)}{P(Snap \mid q_1 \ldots q_n)} \\
&= \frac{P(q_1 \ldots q_n \mid Prof)P(Prof)}{P(q_1 \ldots q_n \mid Snap)P(Snap)}
\end{aligned}
$$

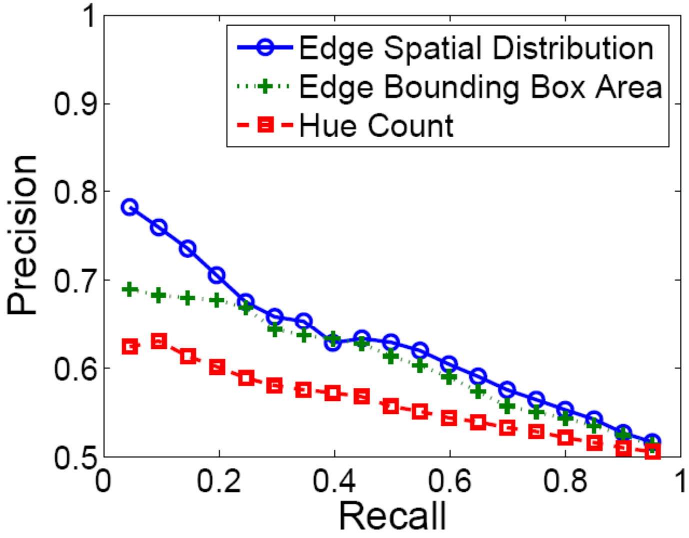
$$
q_{all} = \frac{P(q_1 \mid Prof) \ldots P(q_n \mid Prof)P(Prof)}{P(q_1 \mid Snap) \ldots P(q_n \mid Snap)P(Snap)}
$$

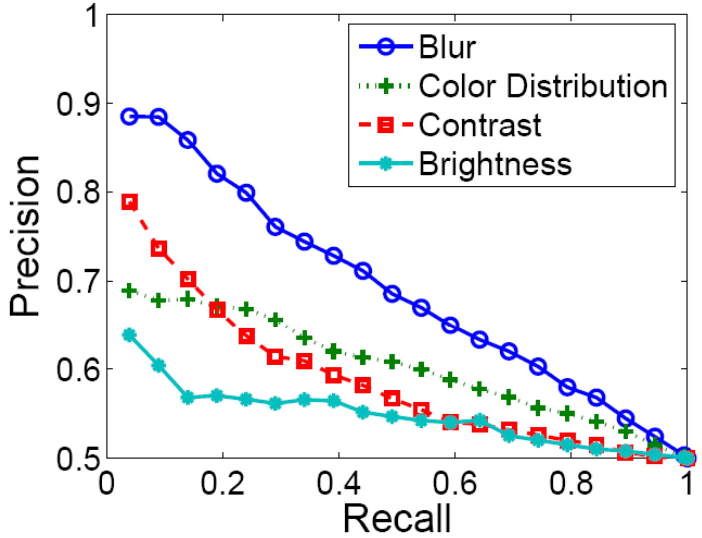
Dataset – DPChallenge.com

- 60K photos
- 40K photographers
- 10/90 percentile



Results

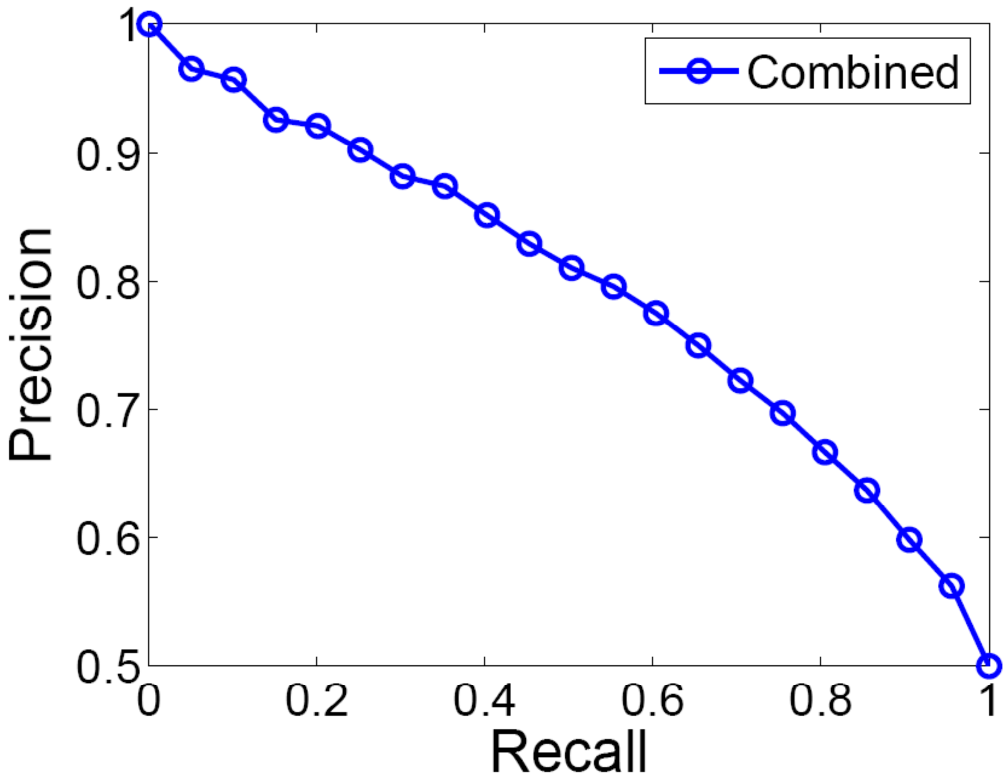




$$recall = \frac{\#\ professional\ photos\ above\ threshold}{total\ \#\ professional\ photos}$$

$$precision = \frac{\#\ professional\ photos\ above\ threshold}{\#\ photos\ above\ threshold}$$

Results



Results

- The difference between high and low quality photos are exaggerated when using a smaller test set. The error rate decreases as well, which suggests the quality metrics match the perceptual criteria in judging photos

	Testing on top and bottom n%				
	10%	8%	6%	4%	2%
Error rate	28%	26%	24%	23%	19%

Web Retrieval Results






Web Retrieval Results






Reference

- Y. Ke, X. Tang, and F. Jing. *The Design of High-Level Features for Photo Quality Assessment*. Computer Vision and Pattern Recognition, 2006

- Yan Ke, Taking Great Pictures, http://www.cs.cmu.edu/~yke/photoqual/20071127_GreatPictures.pdf

The end

Thank you!