A Closer Look at Boosted Image Retrieval

Nicholas R. Howe

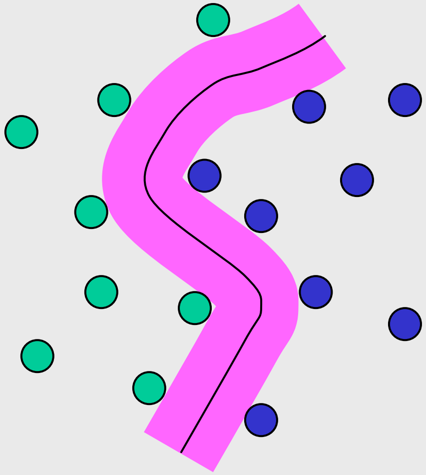
nhowe@cs.smith.edu

SMITH COLLEGE

Never the Twain Shall Meet?

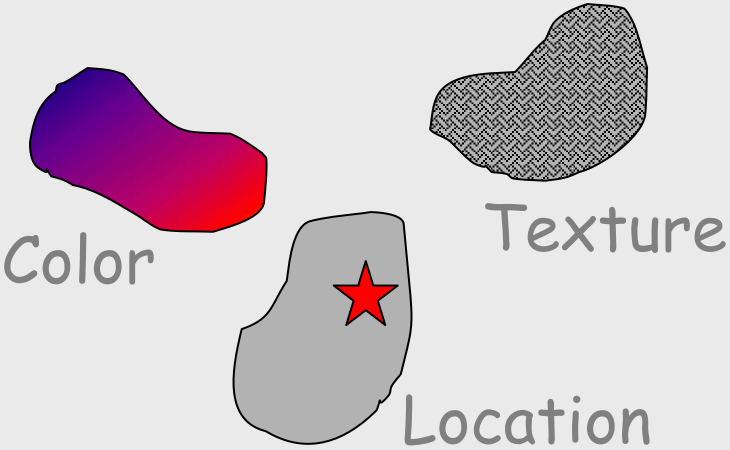
<u>Machine Learning</u>

☞ *Improved classification through "boosting" & other large-margin techniques.*



<u>Image Retrieval</u>

☞ *Improved performance through better, more comprehensive image representations.*



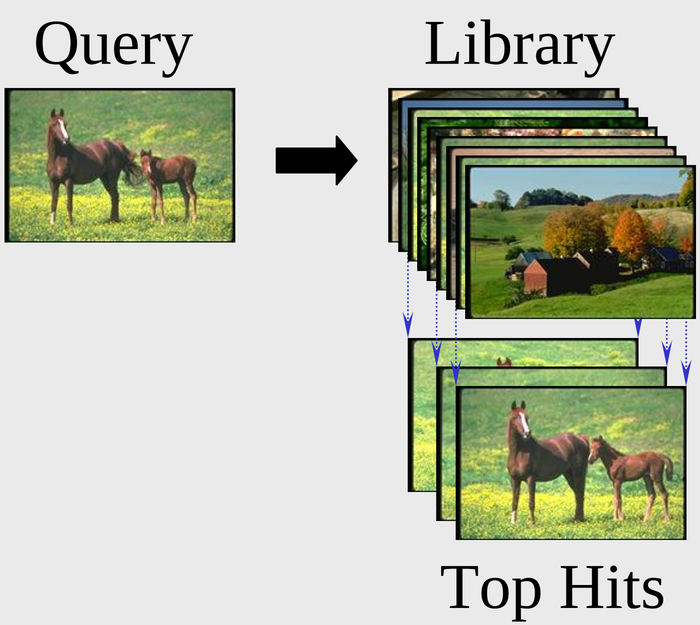
Previous Work

- Tieu and Viola (2000) – a good start…
  - Looks at just one candidate image representation
  - Simple, feature-based boosting
    (i.e., decision stumps)
- Can we apply boosting more effectively?

Boosting + Image Representations = ?

Retrieval vs. Classification

Retrieval paradigm:



Classification paradigm:

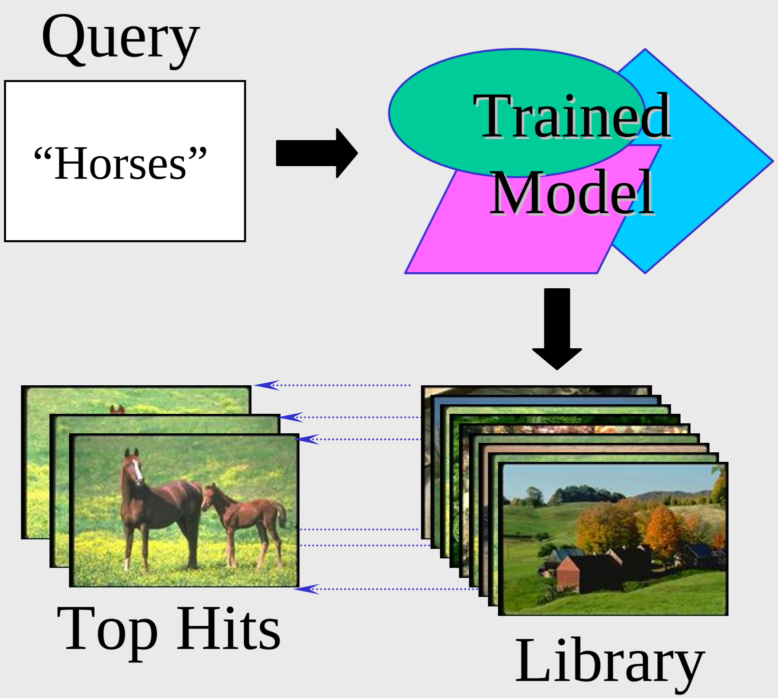
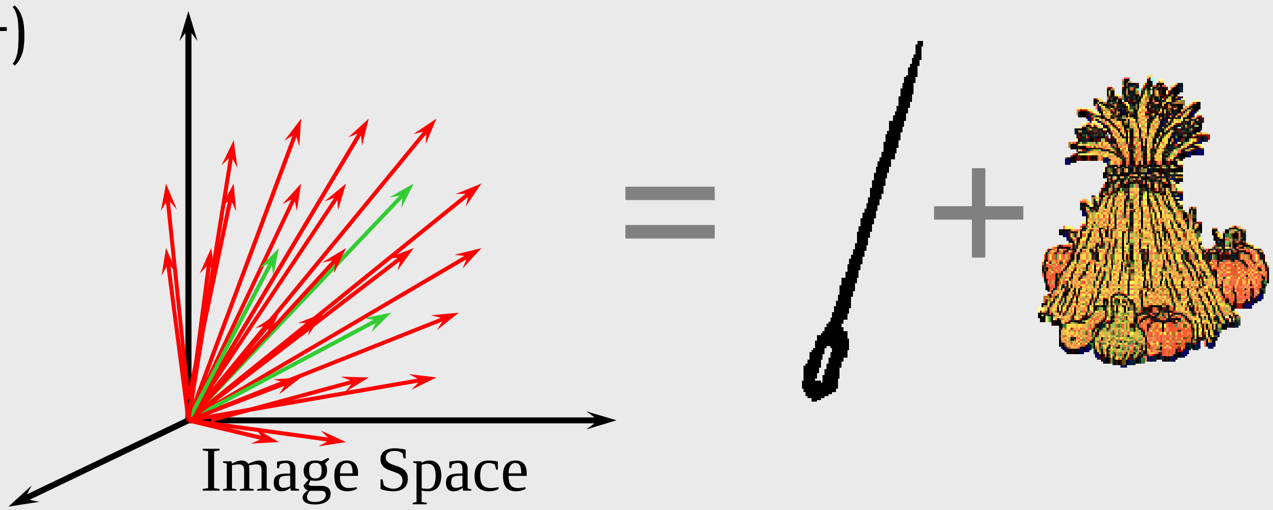


Image Classification is Hard

- Classes are diffuse.
- Features correlate weakly with class.
- High dimensionality (10K+)



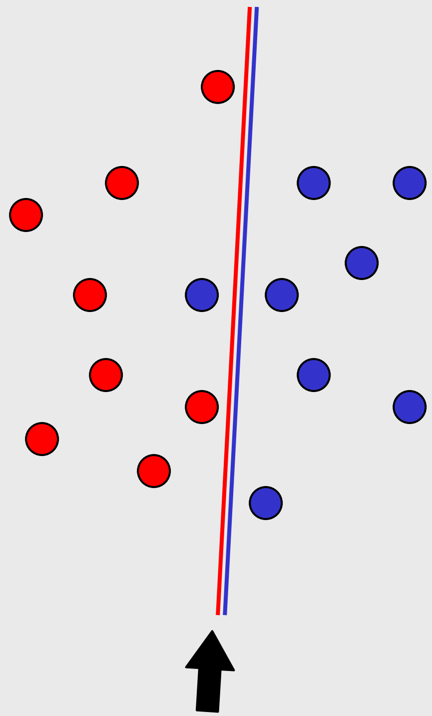
Boosting Can Help

- Designed for complicated problems
  - Irregular & complex decision boundaries
  - Mislabeled training data*
- Known to help in wide range of machine learning problems.
- Tieu & Viola provide example.

*Some forms of boosting, anyway

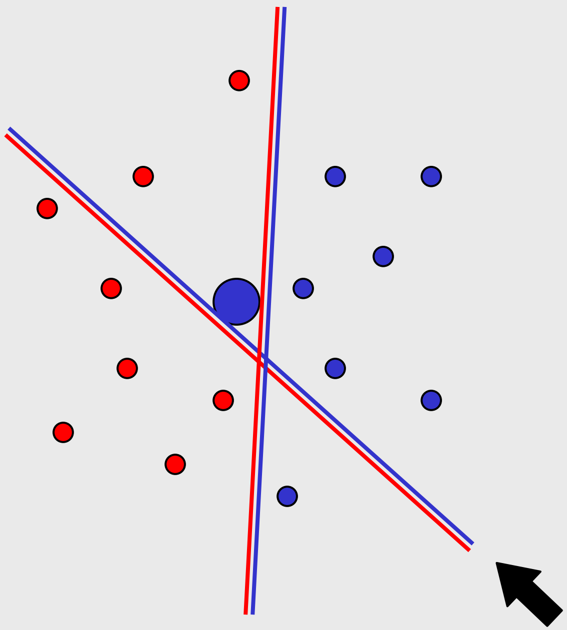
Review of Boosting

- Base classifier must score **>50%** on arbitrarily weighted training set.



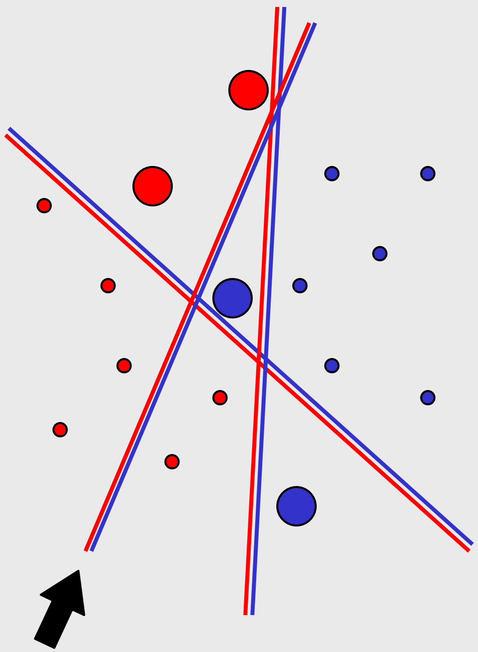
Review of Boosting

- Base classifier must score **>50%** on arbitrarily weighted training set.
- Repeatedly train base classifier using multiple weightings of training data.



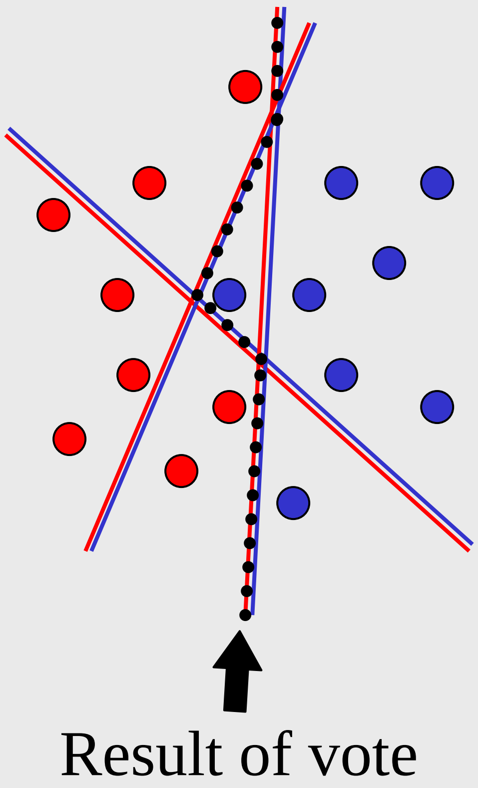
Review of Boosting

- Base classifier must score **>50%** on arbitrarily weighted training set.
- Repeatedly train base classifier using multiple weightings of training data.



Review of Boosting

- Base classifier must score **>50%** on arbitrarily weighted training set.
- Repeatedly train base classifier using multiple weightings of training data.
- Combined predictions better than single classifier alone.
  - Weighted majority vote



Open Questions

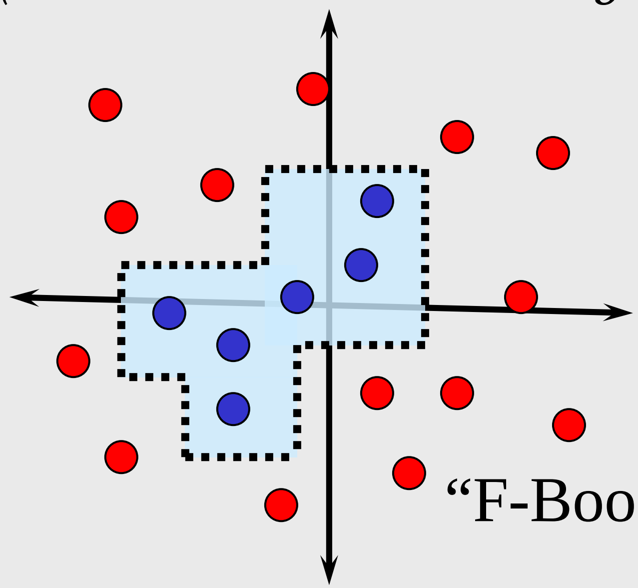
1. How do we apply boosting with standard image representations?
   - Larger than most used in machine learning.
2. Are some representations better for boosting?
3. Does boosting work better with some classes of images?

Image-Friendly Base Classifiers

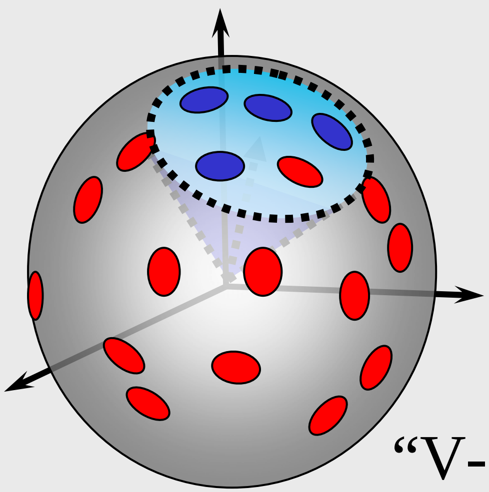
- Many standard classifiers are “feature-based”.

  (Decision boundaries orthogonal to feature axes.)
- “Vector-based” classifier may suit images better.

  (Decision boundaries = angular neighborhood around a vector.)



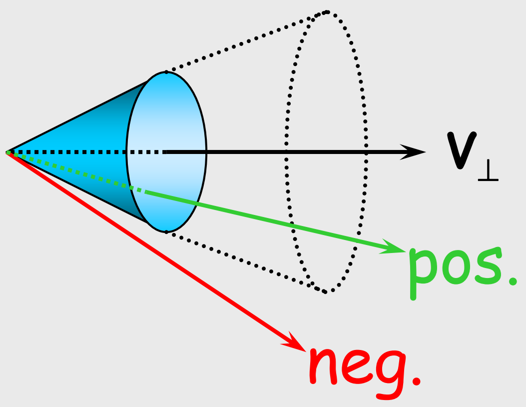
“F-Boost”



“V-Boost”

Vector-Based Classifier

- Identify a central vector $V_{\perp}$ within a concentration of positive instances.
- Classify instances within some angular radius of $V_{\perp}$ as positive examples (Salton's cosine metric).
- Question: How to find $V_{\perp}$?

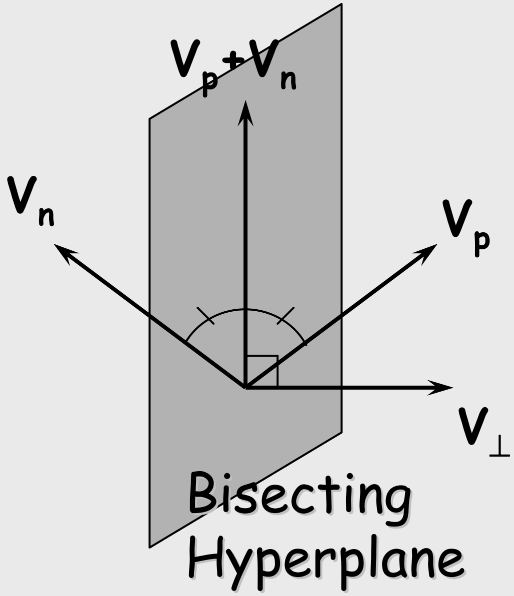


Vector-Based Classifier

$\mathbf{V_p} = \sum$ weighted positives

$\mathbf{V_n} = \sum$ weighted negatives

$$\mathbf{V}_{\perp} = \mathbf{V}_p - \frac{\mathbf{V}_n \bullet (\mathbf{V}_p + \mathbf{V}_n)}{\left\| \mathbf{V}_p + \mathbf{V}_n \right\|}$$

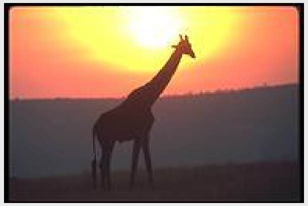


☞ Consistently generates good classifiers (empirical observation).

Experimental Design

- 3 algorithms
  (F-Boost, V-Boost, control)
- 4 image reps.
- 5 classes + chaff
  - 20K images (Corel)
- 5x2 cross validation
  - Data split: training/test
  - 5 repetitions

Churches



Sunsets

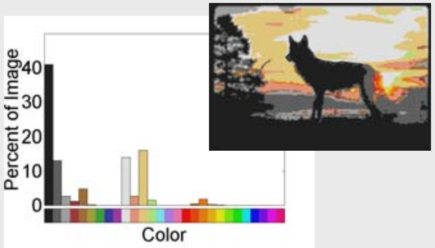
Race Cars

Tigers

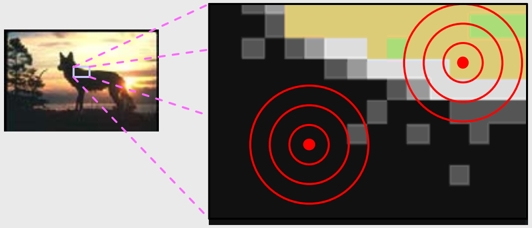
Wolves

Image Representations

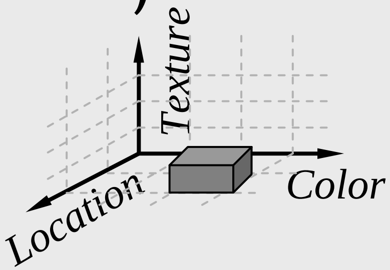
- Histogram (Swain & Ballard)



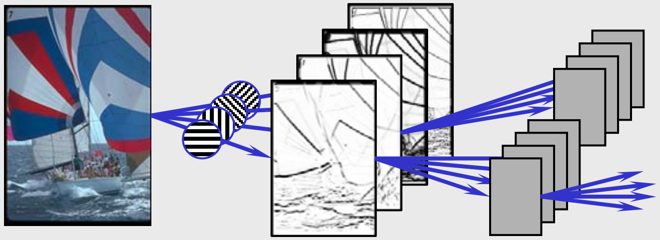
- Correlogram (Huang et. al.)



- Stairs (Howe & Huttenlocher)



- Tieu-Viola



Choosing a Control

- Poor control: Single Base Classifier
  - Does only slightly better than chance.
- Also poor: Nearest Neighbor using entire training set

- <u>Good</u> control: Nearest Neighbor using greedy selection of exemplars
  - Select one training example with best 1-NN accuracy
  - Add additional exemplars greedily as long as they increase accuracy on training set.

Result Preview

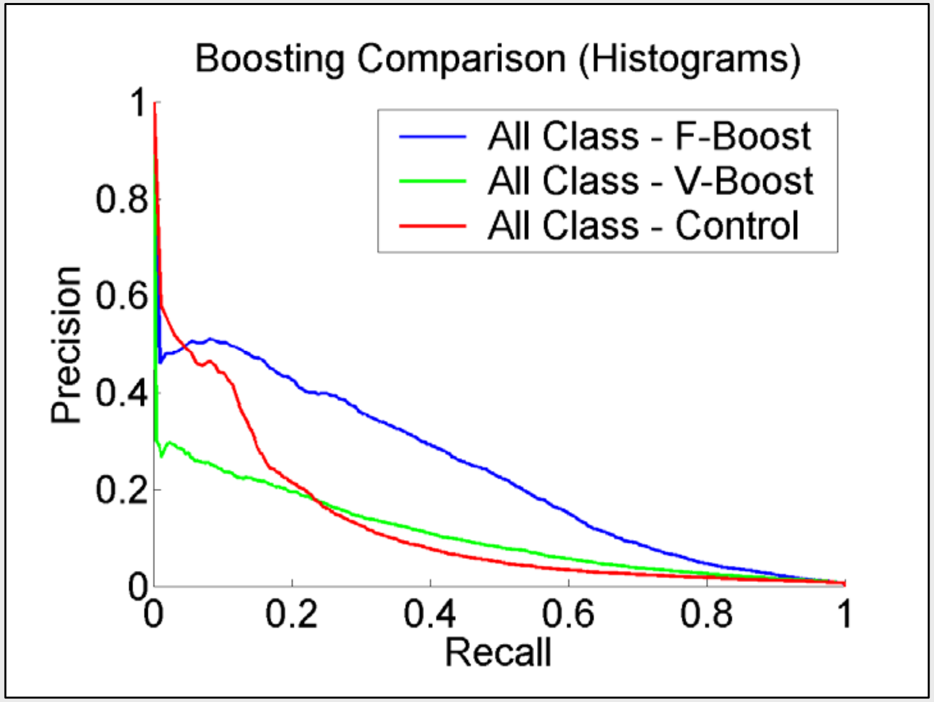
1. Comparison of different base classifiers with each other and control
2. Comparison of different image representations under boosting
3. Contrasting results for different classes

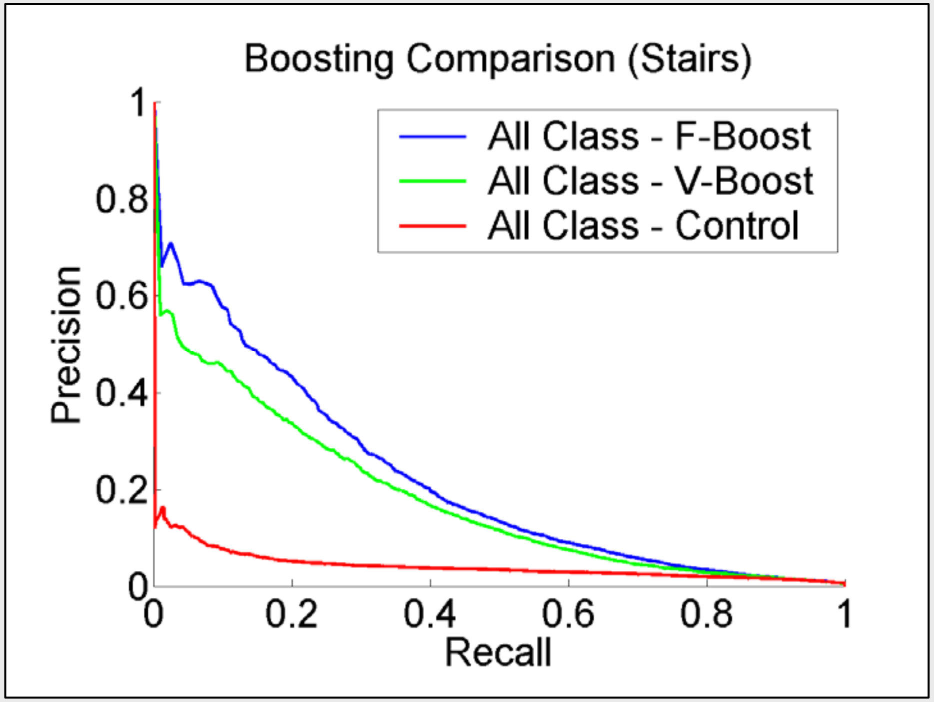
Recall = % of target class that is retrieved

Precision = % of retrieved images that are correct

Comparison: Boosting Type

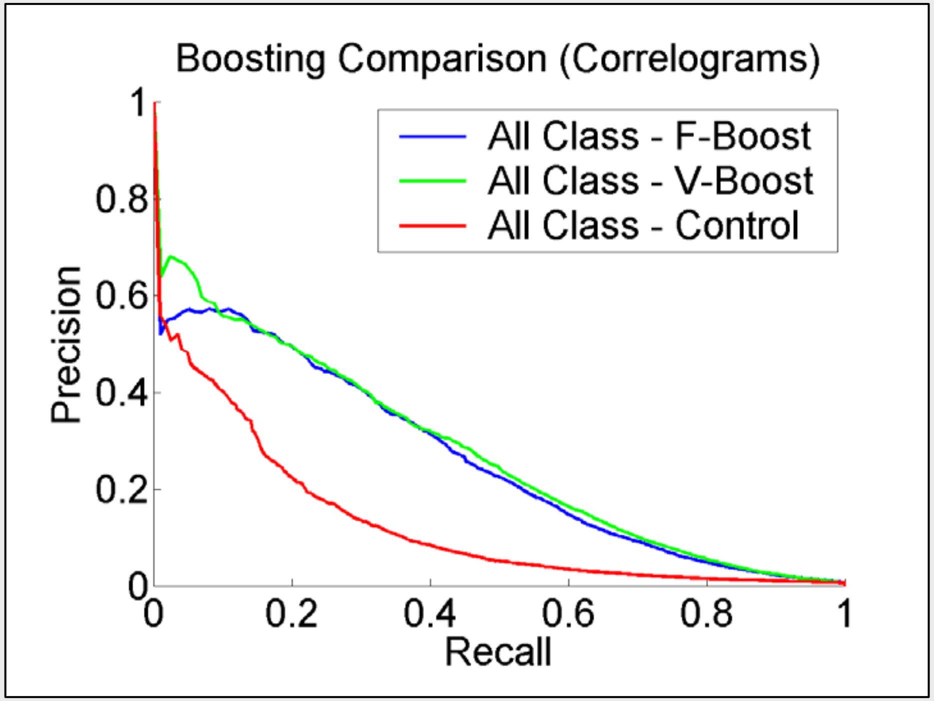
- Boosting beats controls nearly everywhere.
- F-Boost does best with Histograms, Stairs.

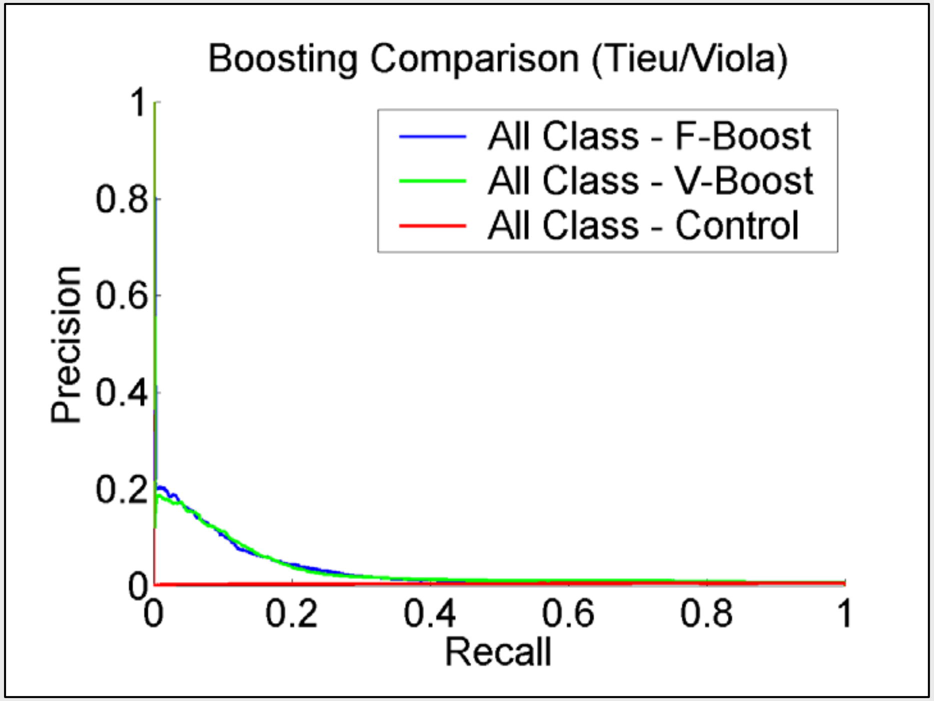




Comparison: Boosting Type

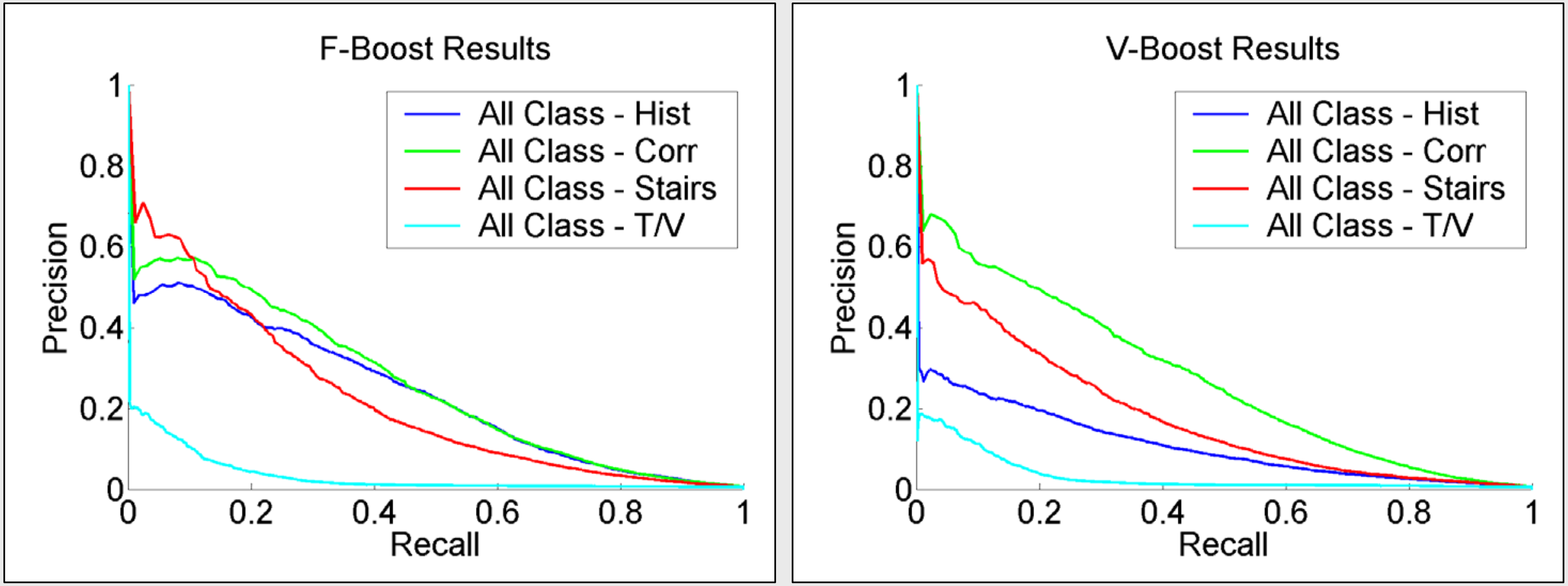
- Vboost ties Fboost on Correlograms, T/V.



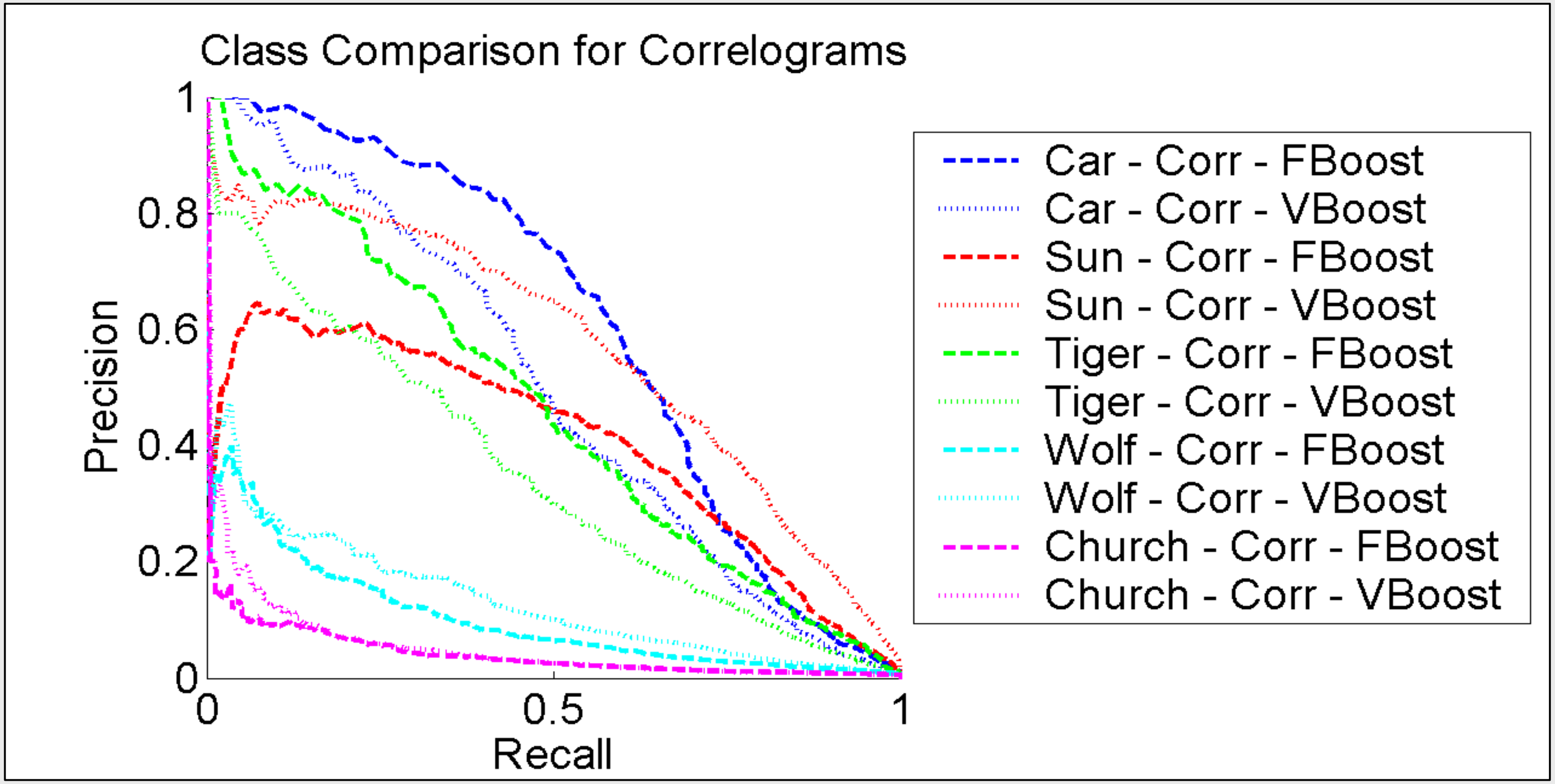


Comparison: Image Reps

- Correlograms do best, Tieu-Viola worst.



Results By Image Classes



- V-Boost, F-Boost better for different classes.

Conclusions

- Boosting improves precision & recall with a range of image representations.
  - No surprise!
  - But: better than Tieu & Viola indicate.
- Boosted correlogram is most successful representation.
  - Boosted effectiveness mirrors unboosted.
- Best base classifier may vary.
  - V-Boost faster, but sometimes worse.

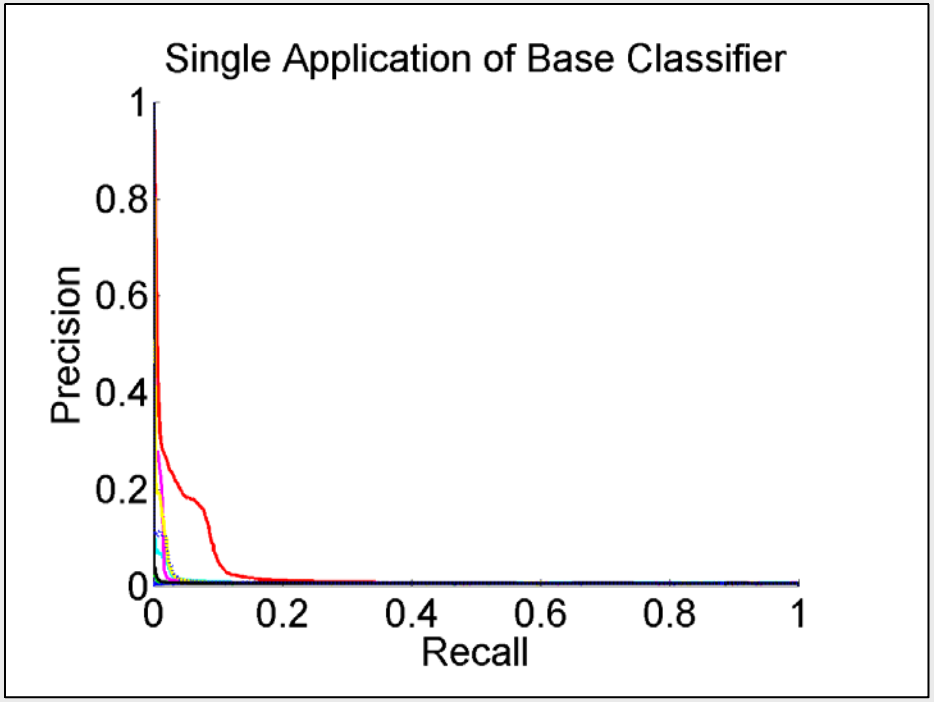
Questions?

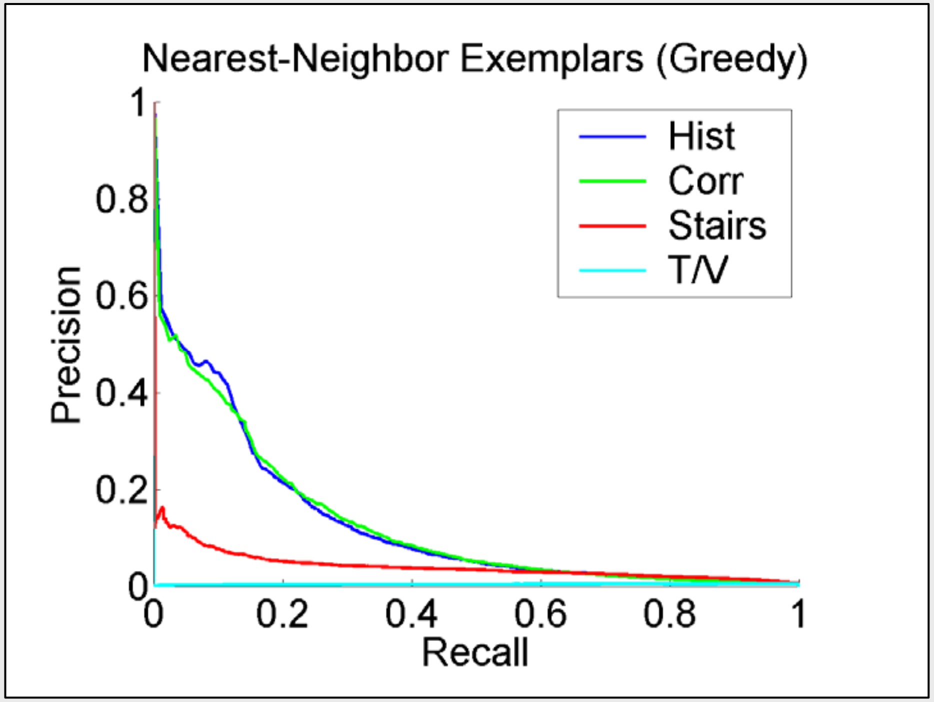
Summary of Work

- Investigated boosting using two types of base classifier plus control.
- Compared effectiveness of different image representations with boosting.
- Looked at image classes with range of difficulty.

Choosing a Control

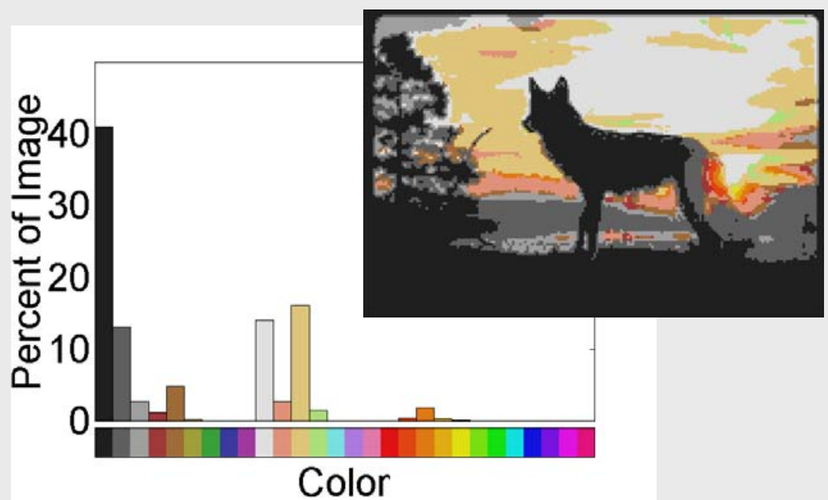
- Base Classifier alone does poorly vs. Nearest Neighbors with greedy exemplar selection.





Color Histograms (Swain & Ballard)

- Map image to limited set of colors.
- Count fraction of pixels in each color.



Color Correlograms (Huang et. al.)

- Map image to limited set of colors.
- Count co-occurrence probability of same colors at fixed distances.

Color Correlograms

- Correlograms consist of a table of probabilities.

$$C(x,y) = P\big(color(b) = x \,\big|\, (color(a) = x) \wedge (\|a-b\| = y)\big)$$

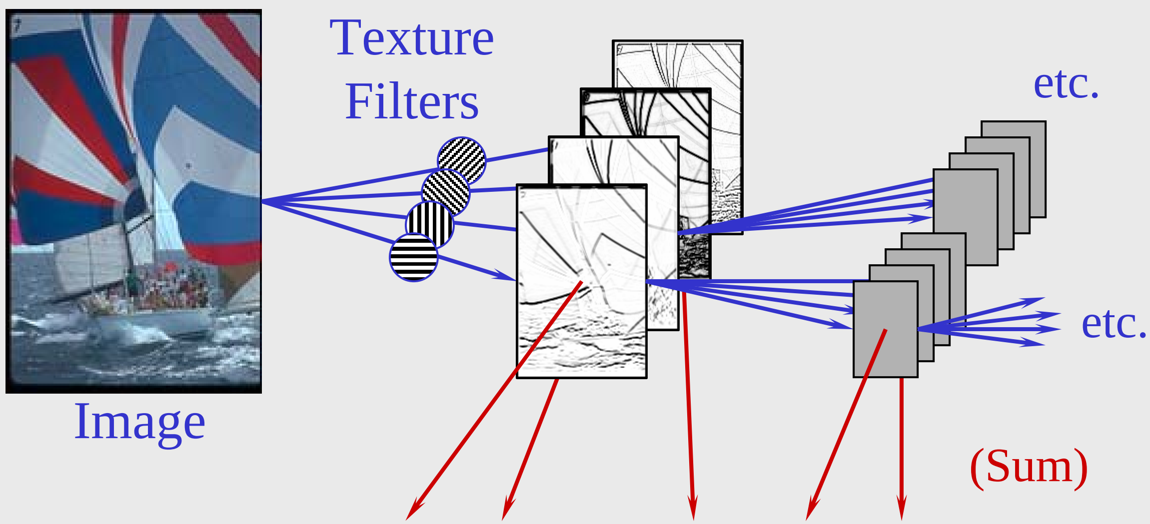
	Red	Orange	Yellow	etc...
1 pixel	0.32	0.0	0.06	0.14
3 pixels	0.16	0.0	0.04	0.0
5 pixels	0.08	0.0	0.03	0.0

"Given a pixel of color *x*, the probability that a pixel chosen distance *y* away is also color *x*"

- Correlograms can be compared like vectors.

DeBonet & Viola

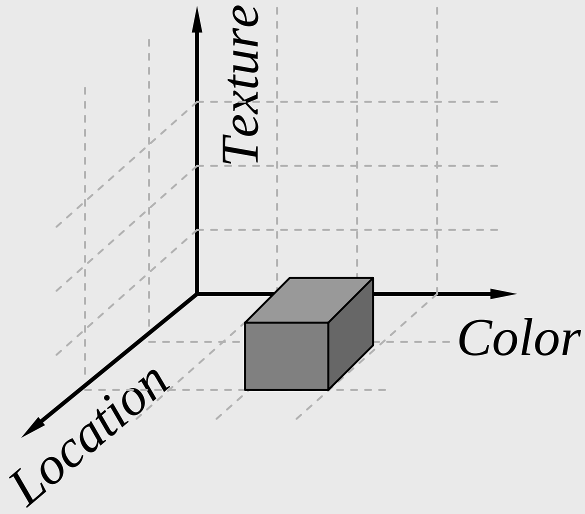
- Pass set of simple filters over image; sum.
- Repeat on filtered images 4 levels deep.



Description = (T_0, T_1, ..., T_n, T_{00}, T_{01}, ..., T_{nn}, ...)

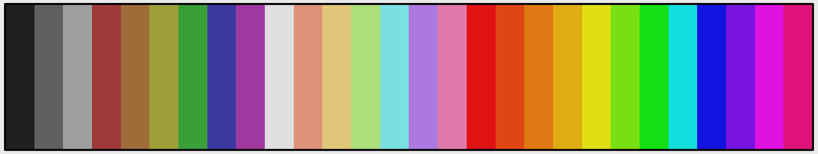
Stairs (Howe & Huttenlocher)

- Discretize the range of each feature.
  (Color, texture, and location)
- Count area in image described by each combination of features.
  - Blue-Smooth-TopLeft: 5,
    Blue-Smooth-TopMiddle: 1,
    …
    Green-Smooth-TopLeft: 0, etc.

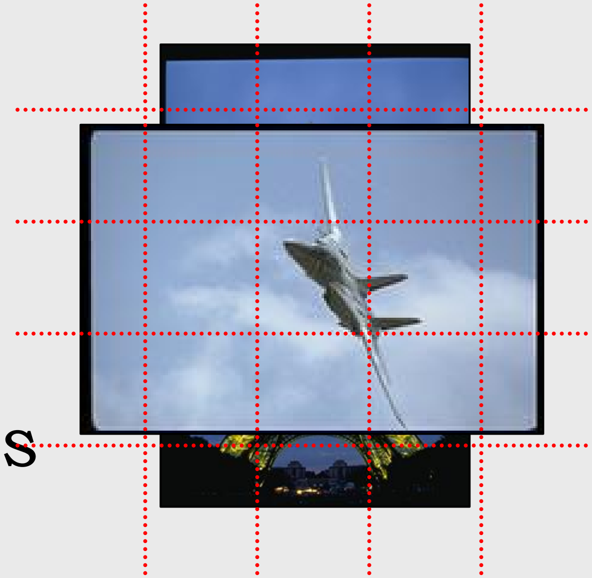


Discretization

- Color: 28 bins



- Texture: 3 bins
  (smooth, textured, rough)
- Location: 25 bins



- Total: 28×3×25
  = 2100 combinations

Vector Representation

- Final representation of image is a vector with 19200 dimensions.

$$\mathbf{v} = \left\langle v_{c_1 t_1 l_1}, v_{c_1 t_1 l_2}, \ldots, v_{c_1 t_1 l_{25}}, v_{c_1 t_2 l_1}, \ldots, v_{c_{128} t_6 l_{25}} \right\rangle$$

- Each dimension records how much of a particular type of material is present.
  - e.g., how much smooth blue in the top left corner?