Predicting Visual Saliency of Building using Top down Approach

Sugam Anand ,CSE
Sampath Kumar,CSE
Mentor : Dr. Amitabha Mukerjee
Indian Institute of Technology, Kanpur

Outline

- Motivation
- Previous Work
- Our Approach
- Saliency Computation
    - Itti and Koch - A saliency-based search mechanism for overt and covert shifts of visual attention, 2000
- Object Detection
    - A simple object detector with boosting- by Antonio Torralba
    - Haartraining: Detect objects using Haar-like features
- Problems Faced
- Work Done
- References

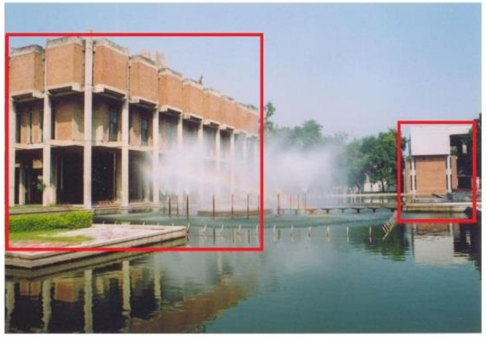
Motivation

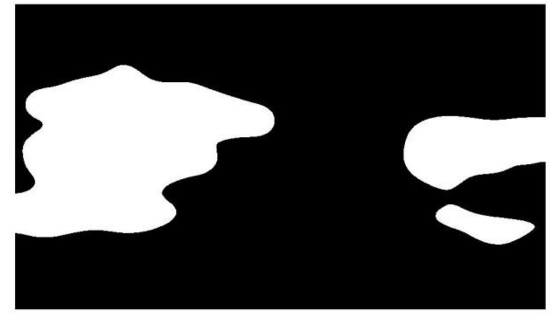
- What landmarks (buildings) does human choose for describing a route.
- Applications in robotics.
- Less work done in top down approach of visual saliency

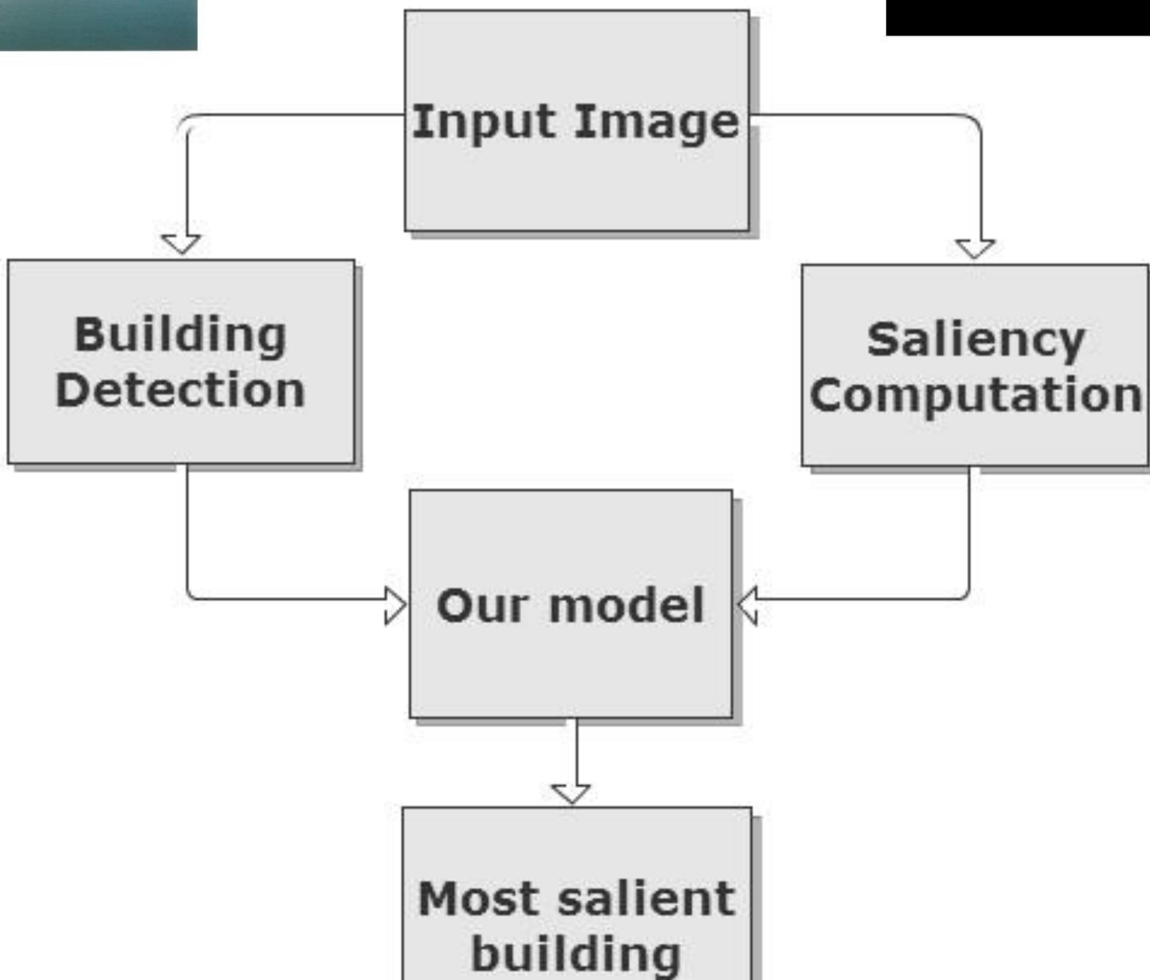
Previous Work

- L. Itti, C. Koch, & E. Niebur (1998)- A Model of Saliency-Based Visual Attention for Rapid Scene Analysis
  - Uses low level features
  - Not able to predict correctly where humans actually look ,upto 28.4 % [3]
- Tilke judd, Krista Ehinger , Fredo Durand, Antonia torralba(2009)-Learning to Predict where humans look
  - A learning based model
  - Uses high level features also
  - State of the art in visual saliency prediction

Our Approach







Saliency Models

- Based on neuro biologically linear filters
- Take into account low level features like intensity, contrast , illumination and color.
- Apart from these low level ,Some mid and high level features .
- All use bottom approach

Itti and Koch Model,[1998]

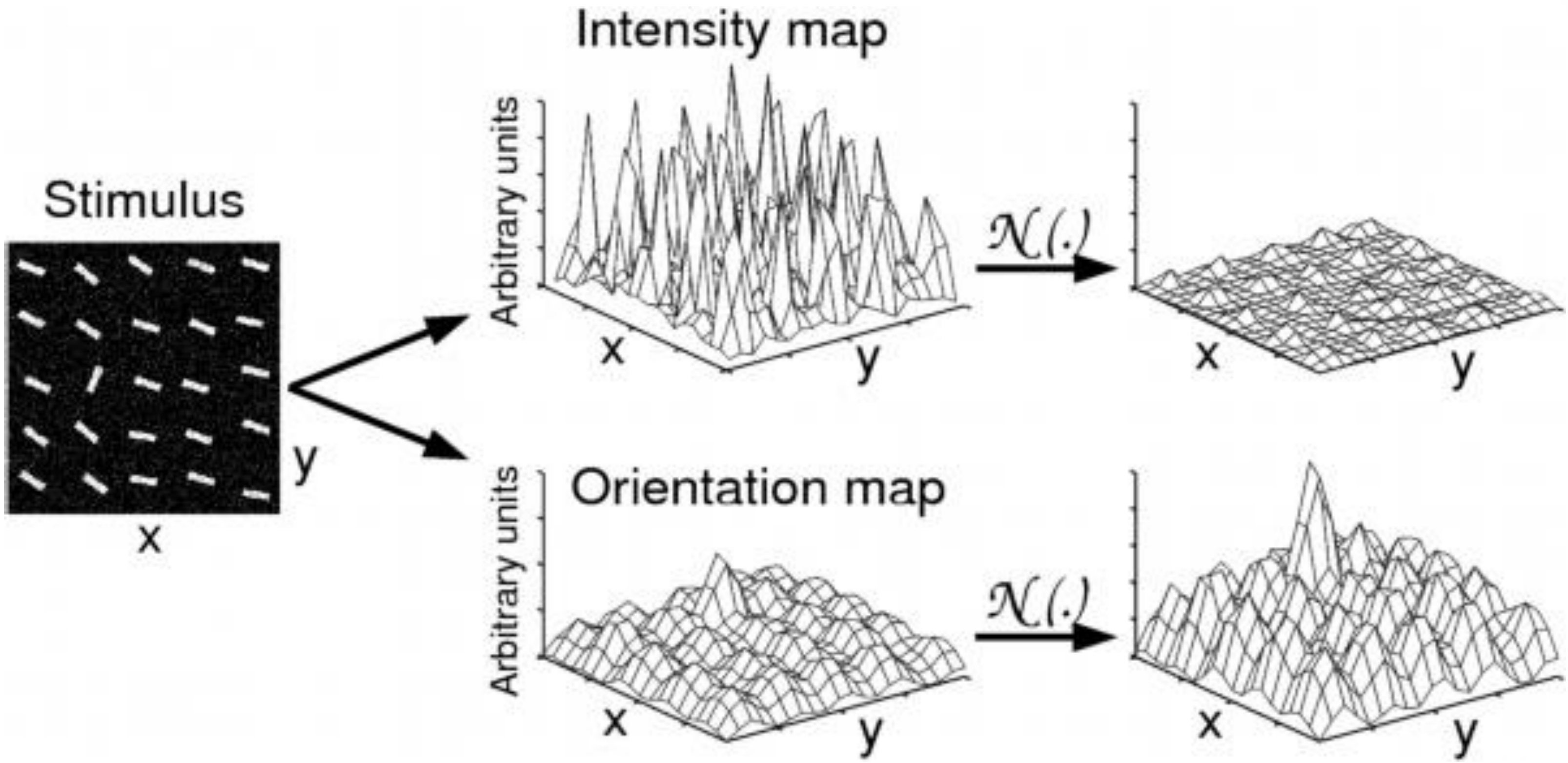
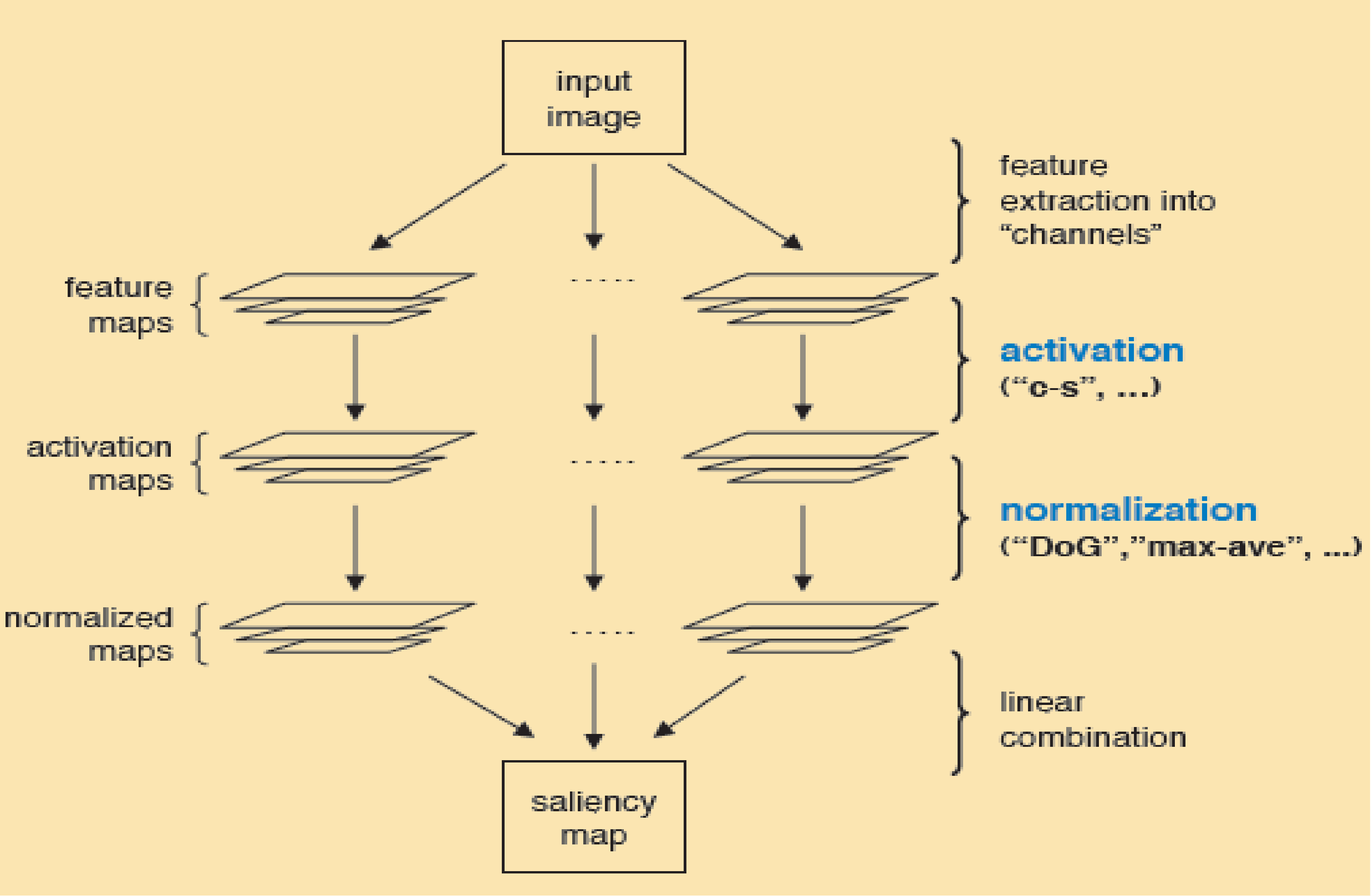
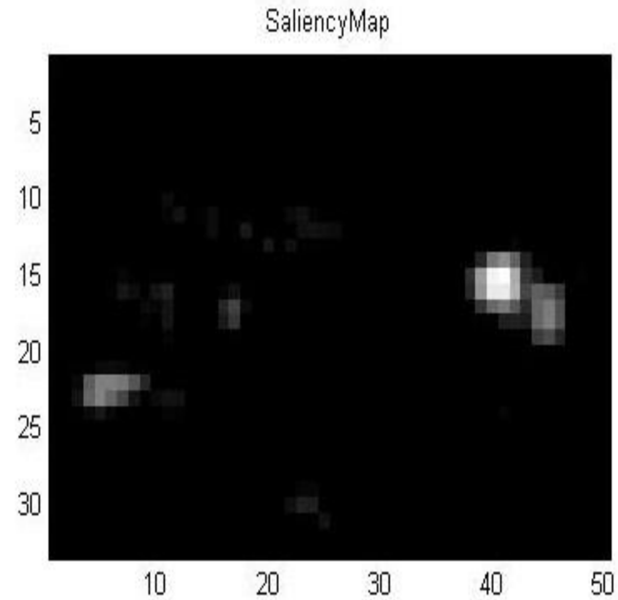


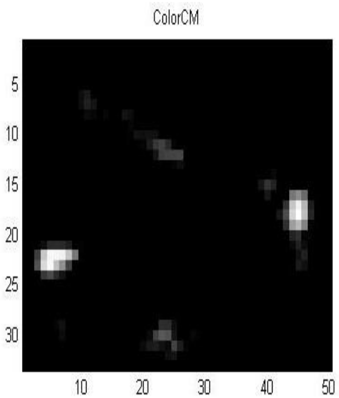
Figure taken from [1]

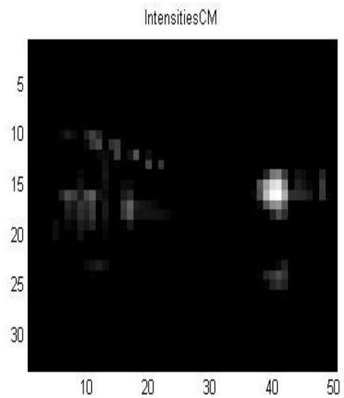
Algorithm

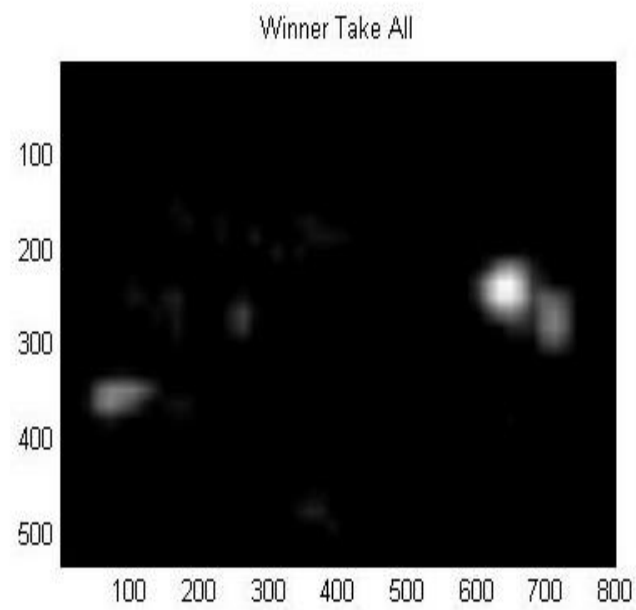


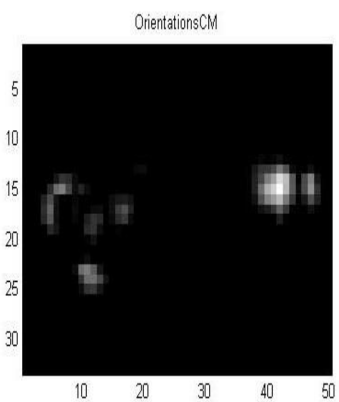
Taken from [6]


SaliencyMap


ColorCM


IntensitiesCM

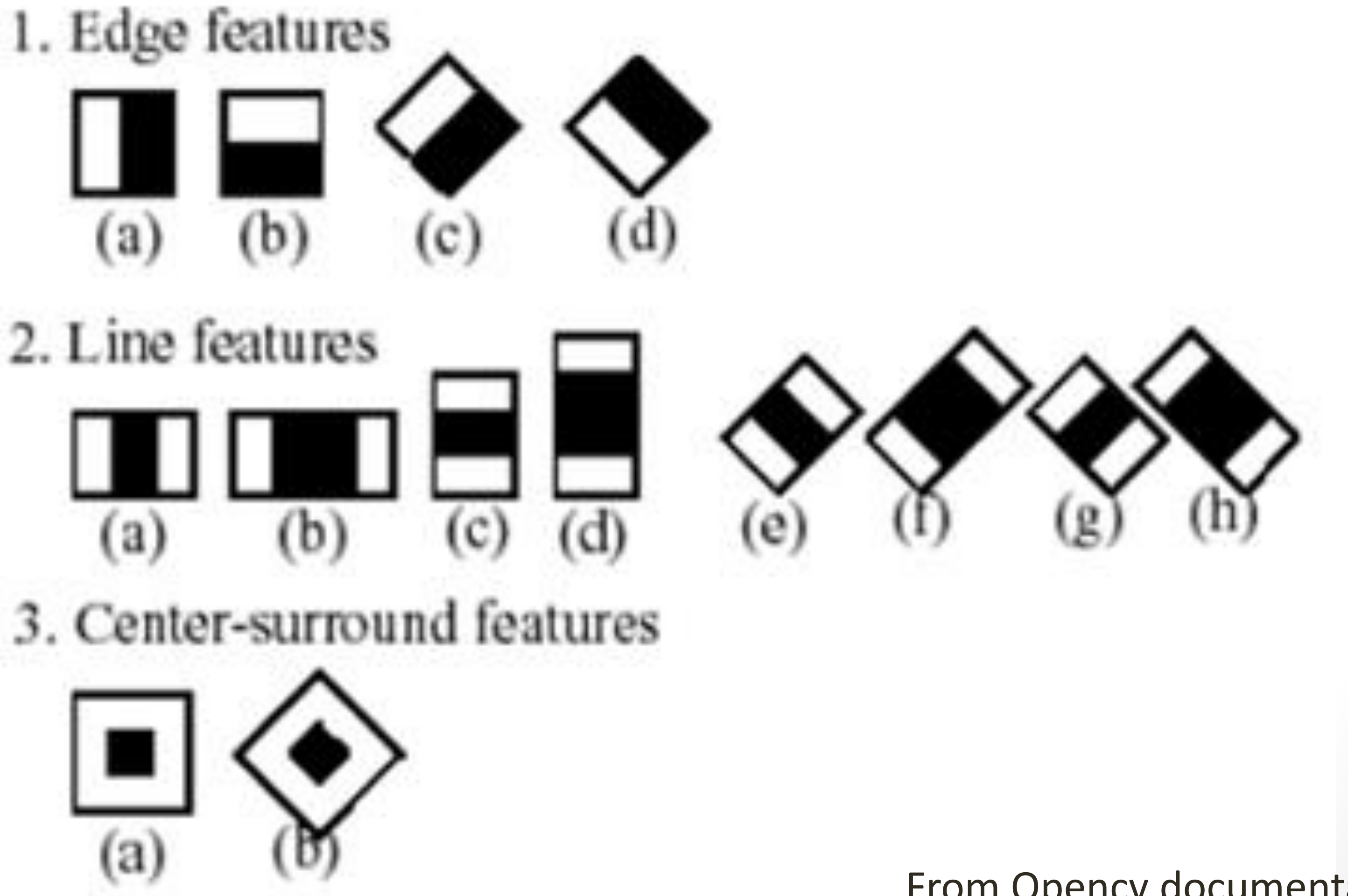

Winner Take All


OrientationsCM

Object Detection

- OpenCV Haartraining: Detect objects using Haar-like features
- Take multiple “positive” samples, i.e., objects of interest, and “negative” samples, i.e., images that do not contain objects.
- Different features are extracted from samples and distinctive features are “compressed” into the statistical model parameters.
- A classifier after training period is obtained for object detection of that class.

Haar-like Features



From Opencv documentation

- Haar like feature’s value is computed as the difference between the sum of the pixels within white and black rectangular regions for that feature.

$$f_i = \mathrm{Sum}(\mathrm{r}_{\mathrm{i,\ white}}) - \mathrm{Sum}(\mathrm{r}_{\mathrm{i,\ black}})$$

$$h_i(x) = \begin{cases} 1 & if\ f_i > threshold \\ -1 & if\ f_i < threshold \end{cases}$$

Adaboost Learning

$$F = sign(w_1 h_1 + w_2 h_2 + \ldots + w_n h_n)$$

$$\text{where, } h_i(x) = \begin{cases} 1 & if \;\; f_i > \theta_i \\ -1 & if \;\; f_i < \theta_i \end{cases}$$

Weak classfiers (h_i (x)) with less error rate ,gets larger weight . Hence ,contributes in strong classifier.

Object Detection in OpenCV

1. Generating the database of positive and negative samples.
2. Make the bounding box for the object by objectmarker.exe
3. Generate the vec file out of positive samples using createsamples.exe
4. For generating classifier run the haartraining.exe
5. Run haarconv.exe to convert classifier to .xml file

Where Do People Look

- Faces
- Text
- People
- Body parts
- animals

Problem faced

Unconventional buildings attract attention against low level features used by us

Contd…

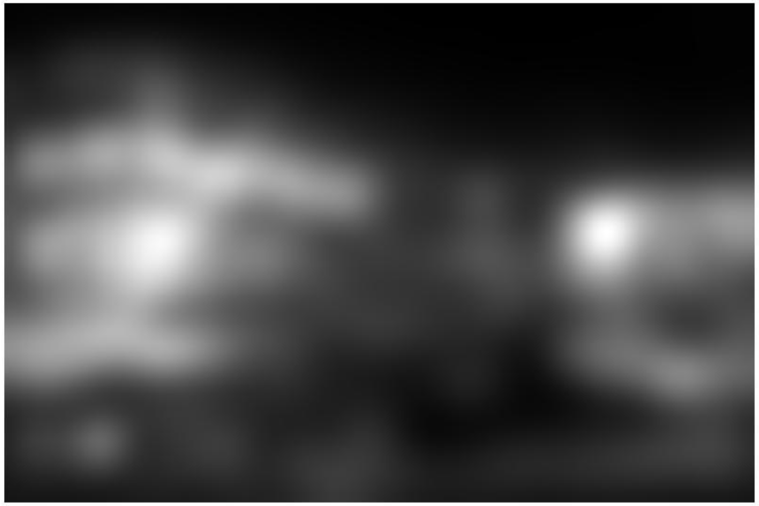
- Text ,faces etc on buildings attract more attention.

Work done

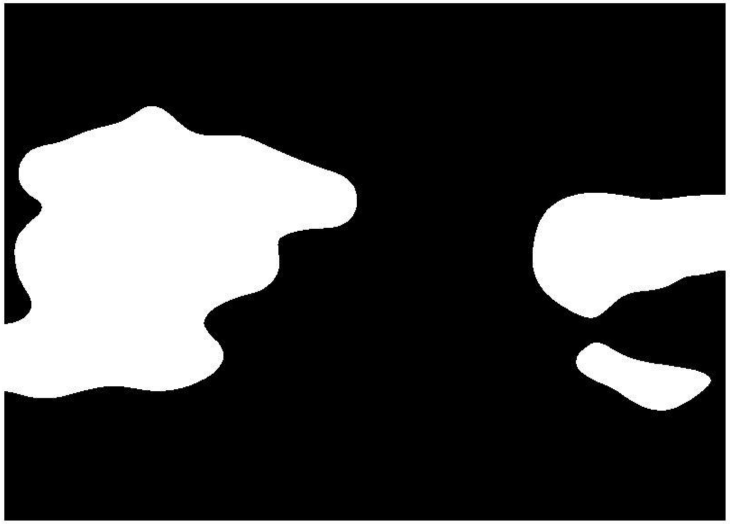
Input image

- Saliency Detection completed

After applying itti koch algo

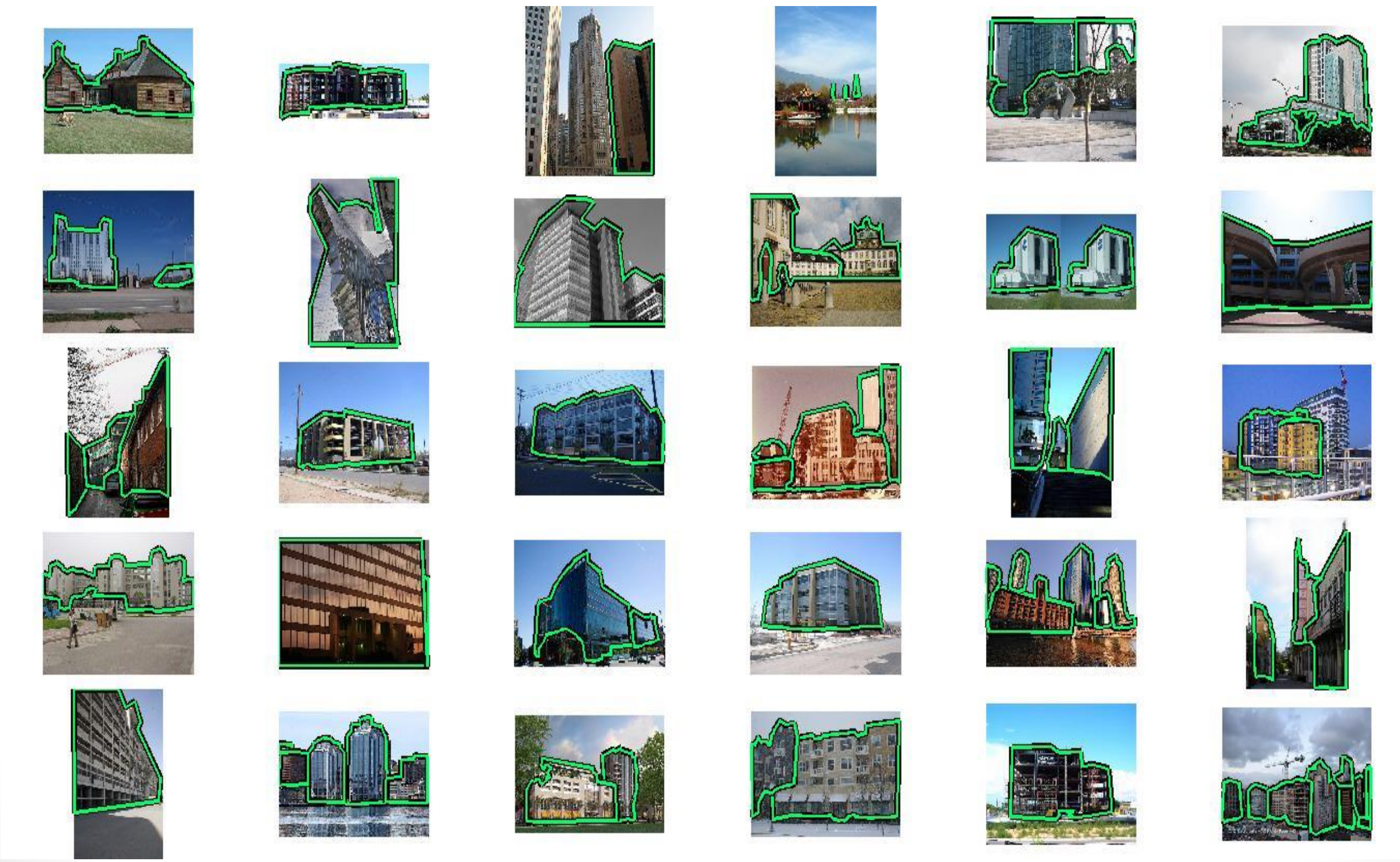


thresholding



Work done

- Our Label me[4] database consisting 150 annotated images



Resources

- Saliency Tool box
  - Contains functions for implementing visual saliency based on itti and koch model
- Cascade Classifier Training in opencv
- J. Harel, A Saliency Implementation in MATLAB: http://www.klab.caltech.edu/~harel/share/gbvs.php
- Training images from Imagenet

References

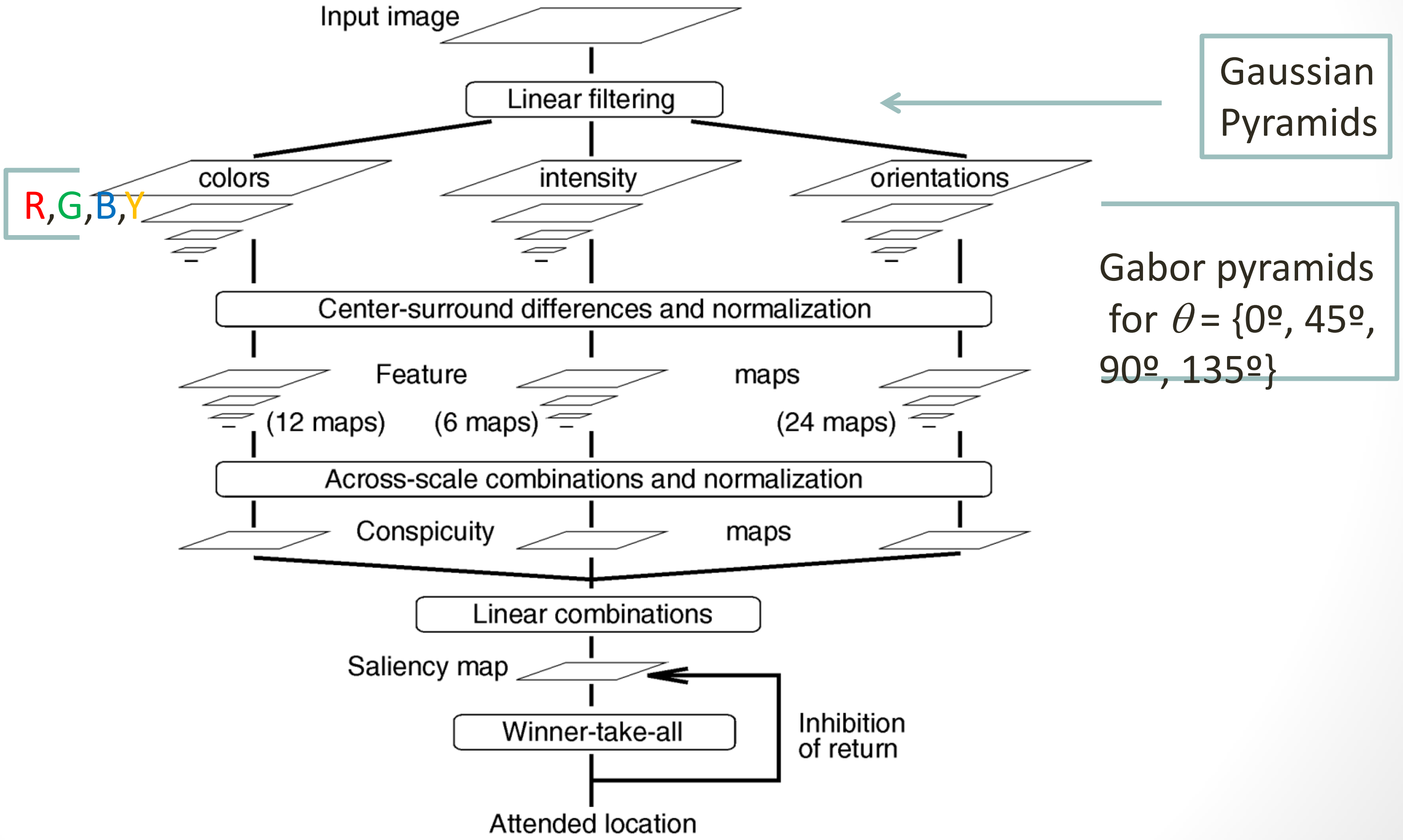
- [1]Itti and Koch - A saliency-based search mechanism for overt and covert shifts of visual attention, 2000
- [2] Tilke judd, Krista Ehinger , Fredo Durand, Antonia torralba(2009)- Learning to Predict where humans look
- [3]A Benchmark of Computational Models of Saliency to Predict Human Fixations by Tilke Judd, Fredo Durand and Antonio Torralba.[2012] .
- [4] LabelMe: online image annotation and applications
  A. Torralba, B. C. Russell, J. Yuen
- [5] Paul Viola, Michael Jones[2001]. Rapid Object Detection using a Boosted Cascade of Simple Features. Conference on Computer Vision and Pattern Recognition
- [6] http://www.klab.caltech.edu/~harel/pubs/gbvs_nips_poster.pdf

Questions

???

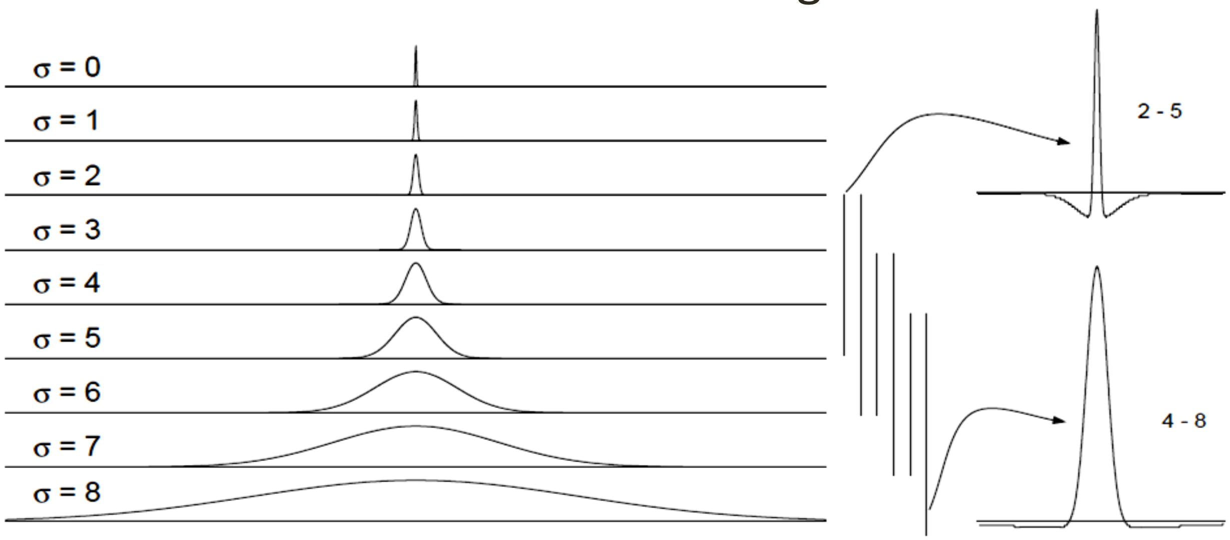
L. Itti's approach

- Architecture:



L. Itti's approach

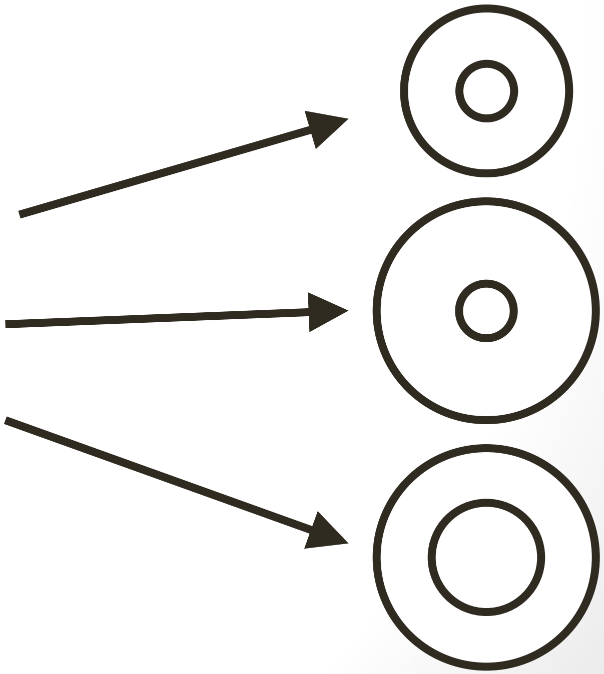
- **Center-surround Difference**
- Achieve center-surround difference through across-scale difference



- Operated denoted by Θ: Interpolation to finer scale and point-to-point subtraction
- One pyramid for each channel: $I(\sigma)$, $R(\sigma)$, $G(\sigma)$, $B(\sigma)$, $Y(\sigma)$ where $\sigma \in [0..8]$ is the scale

L. Itti's approach

- **Center-surround Difference**
  - **Intensity Feature Maps**
- $I(c, s) = |\ \mathrm{I}(c)\ \Theta\ \mathrm{I}(s)|$
- $c \in \{2, 3, 4\}$
- $s = c + \delta$ where $\delta \in \{3, 4\}$
- So $I(2, 5) = |\ \mathrm{I}(2)\ \Theta\ \mathrm{I}(5)|$
  $I(2, 6) = |\ \mathrm{I}(2)\ \Theta\ \mathrm{I}(6)|$
  $I(3, 6) = |\ \mathrm{I}(3)\ \Theta\ \mathrm{I}(6)|$
  …
- → 6 Feature Maps



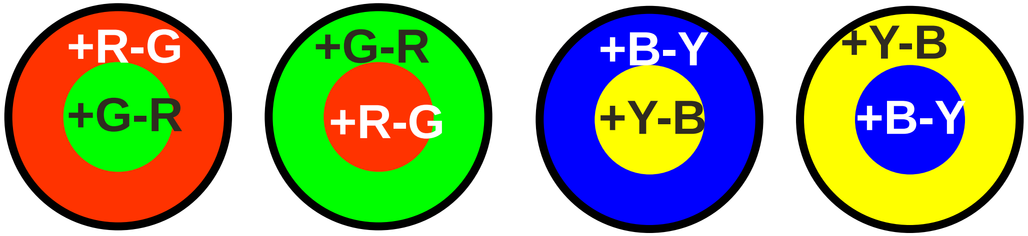
L. Itti's approach

- **Center-surround Difference**
  - **Color Feature Maps**

Red-Green and Yellow-Blue

- **Center-surround Difference**
  - **Orientation Feature Maps**
- $O(c,s,\theta) = |O(c,\theta) - O(s,\theta)|$

Same *c* and *s* as with intensity



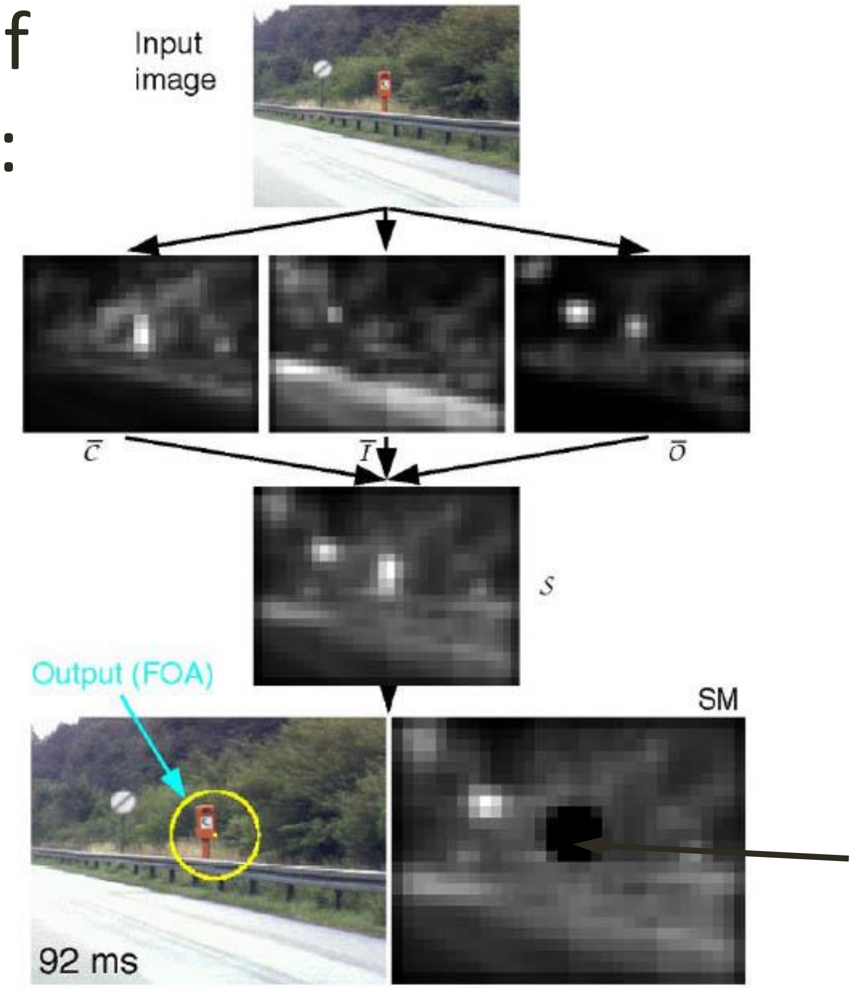
$RG(c, s) = |\,(R(c) - G(c)) \ominus (G(s) - R(s))\,|$
$BY(c, s) = |\,(B(c) - Y(c)) \ominus (Y(s) - B(s))\,|$

L. Itti's approach

- **Normalization Operator**
- Promotes maps with few strong peaks
- Surpresses maps with many comparable peaks
  1. Normalization of map to range $[0 \ldots M]$
  2. Compute average m of all local maxima
  3. Find the global maximum M
  4. Multiply the map by $(M - m)^2$

L. Itti's approach

Example of Operation:

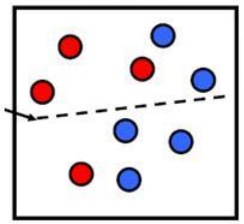


Inhibition of return

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

AdaBoost Algorithm

Start with uniform weights on training examples



$\{x_1, \ldots x_n\}$

For T rounds

Evaluate *weighted* error for each feature, pick best.

Re-weight the examples:
Incorrectly classified -> more weight
Correctly classified -> less weight

Final classifier is combination of the weak ones, weighted according to error they had.

Slide from K.Grauman

Acknowledgement

- The slides 22-28 are based on the tutorial from http://disp.ee.ntu.edu.tw/class/saliencymap.