A Comparative study of Unsupervised Texture Segmentation Algorithms

Presented by:

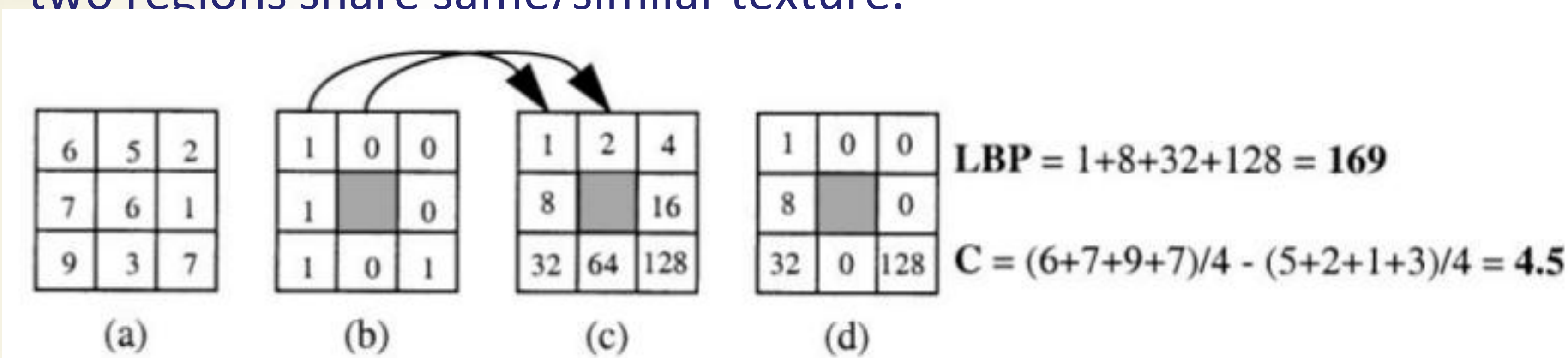
Rohit Girdhar (201001047)
Nishit Soni (201002026)
Divyansh Agarwal (201001027)
Team #4

Problem Statement

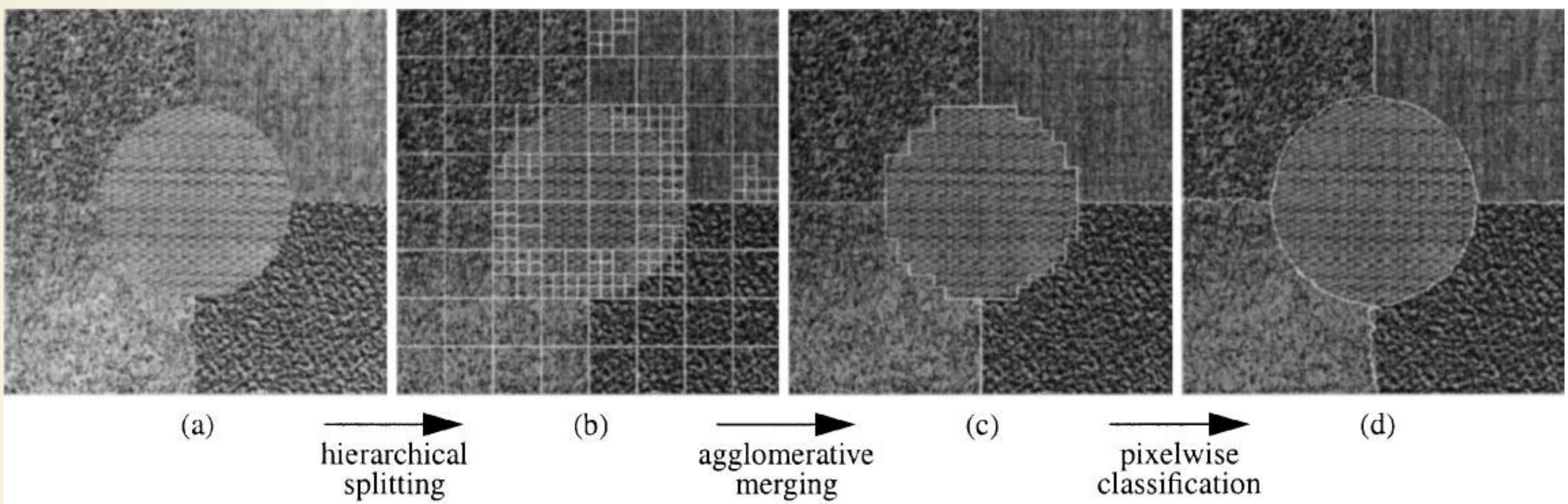
- Problem: To compare unsupervised texture segmentation algorithms based on objective and subjective parameters
- Importance: Texture segmentation has various applications in PR, vision like CBIR, recognition etc
- References:
  - Algorithm 1
    - A. K. Jain, F. Farrokhnia, “Unsupervised texture segmentation using Gabor filters,” *Pattern Recognition*, vol. 24, no. 12, pp.1167-1186, 1991
  - Algorithm 2
    - Timo Ojala, Matti Pietikainen, “Unsupervised Texture Segmentation Using Feature Distributions”, ICIAP '97 Proceedings of the 9th International Conference on Image Analysis and Processing-Volume I , pp 311-318
  - Algorithm 3
    - Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59:2004, 2004.

LBP Based Texture Segmentation

- Applicable for grayscale images
- Textures characterized by joint distribution of local binary pattern (LBP) and contrast (C) features. LBP stores the spatial information and C stores the contrast
- LBP/C histogram is calculated as a texture measure of a texture region
- The LBP/C distribution is approximated by a discrete two-dimensional histogram of size 256xb, where b is the number of bins for C (normally 8).
- Comparing two texture regions – G statistic (log likelihood ratio) – a similarity measure. Lesser the value, higher is the probability that the two regions share same/similar texture.



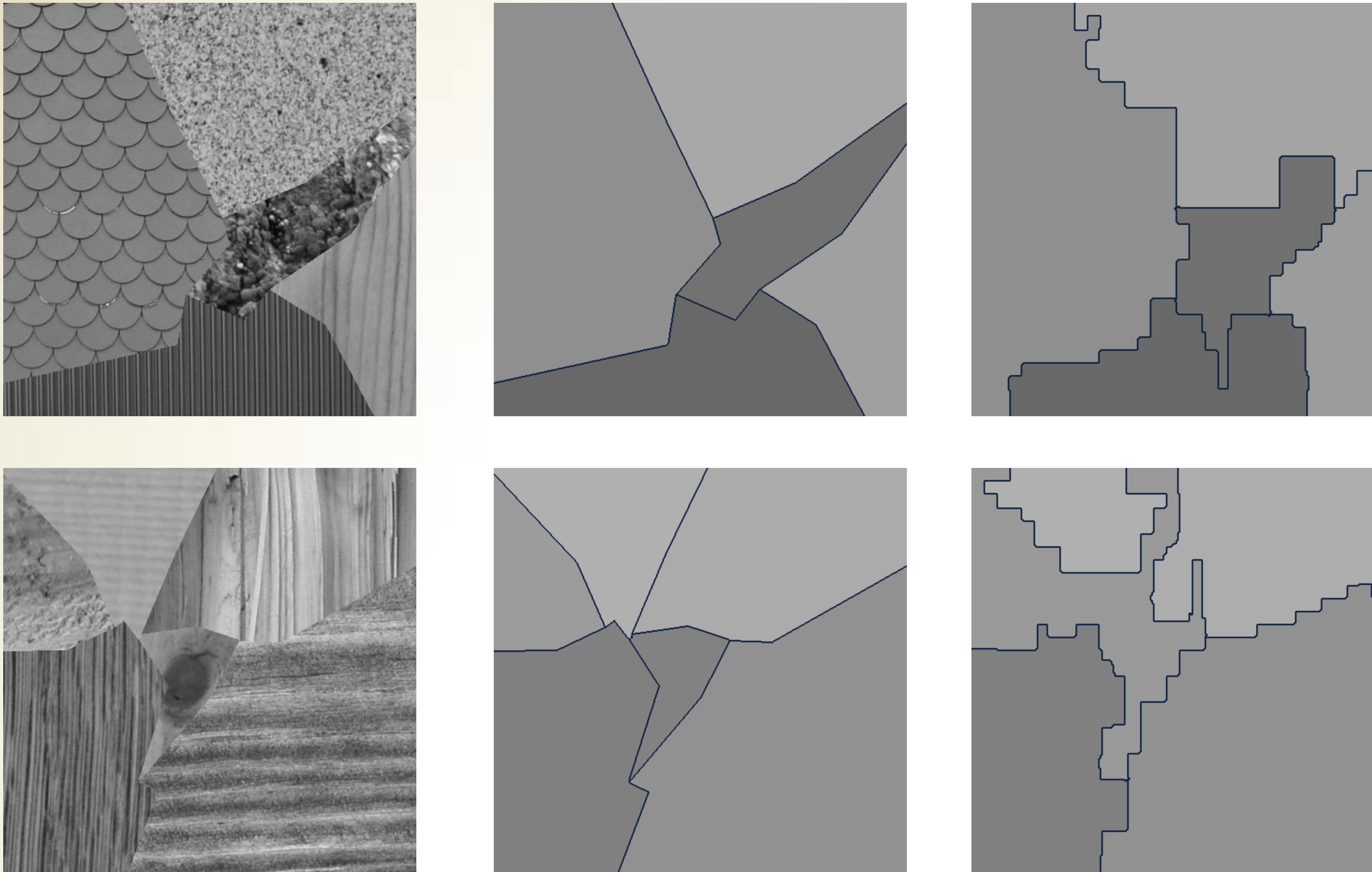
- Hierarchical splitting:
  - Recursively splits the original image into square blocks of varying size. Decision that a block gets divided into 4 sub blocks is based on a uniformity test. For a block, 6 G values are computed (4C2 pairs) and if the ratio = Gmax/Gmin exceeds a threshold, then division is done.
- Agglomerative merging
  - Merges similar adjacent regions until a stopping criterion is satisfied. At a particular stage of the merging, we merge that pair of adjacent segments, which has the smallest merger importance (MI) value = pxG (p is the number of pixels in the smaller of the two regions and G is the distance measure.)
  - Stopping rule: MIR = MI(cur)/MI(max) > Y

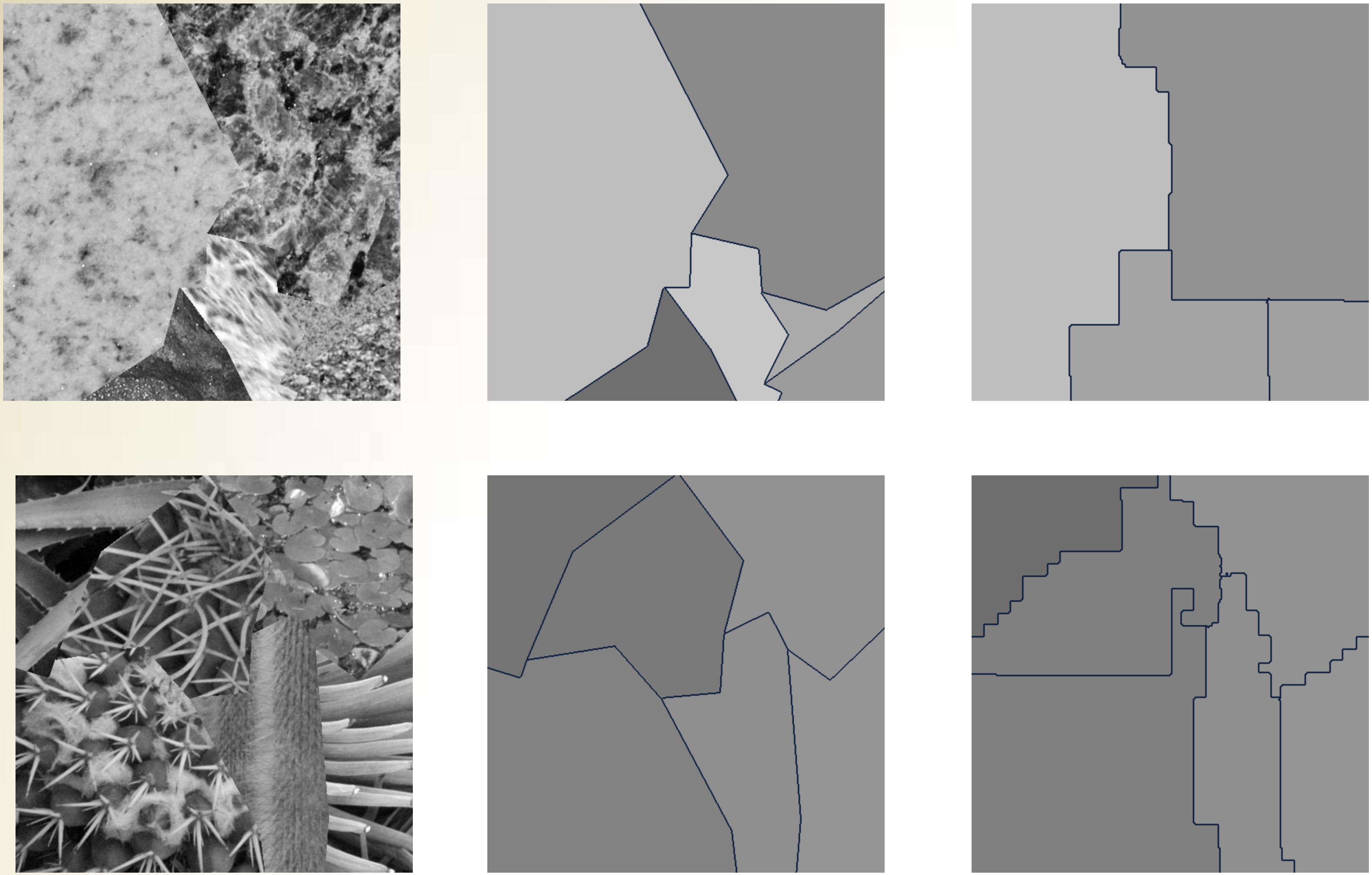


- Pixel-wise classification
  - To improve the localization of the boundaries
  - If a pixel has at least one pixel of different label in the 4-neighbourhood group, then it is/is not re-labelled
  - The process of pixel-wise classification continues until no new pixels are to be relabelled or maximum number of sweeps is reached.
  - For relabeling the pixels, the paper proposed a circular disk of radius r ($r = 2*S_{min}$), however, for simplicity and computational efficiency, we use a square of side 4*r+1.

- Our Contribution
  - We propose (and implemented) the use of disjoint set data structure for computationally efficient representation of regions.

Results (LBP based Segmentation)





Gabor Filter based Segmentation

- Decomposition of the input image using a Gabor Filter bank
  - The filter set forms an approximate basis for a wavelet transform, with the Gabor function as the wavelet
  - We use four values of orientation θ: 0, 45, 90, and 135. The restriction to four orientations is made for computational efficiency.
  - For an image array with a width of N, pixels, where N , is a power of 2, the following values of radial frequency u0 are used:

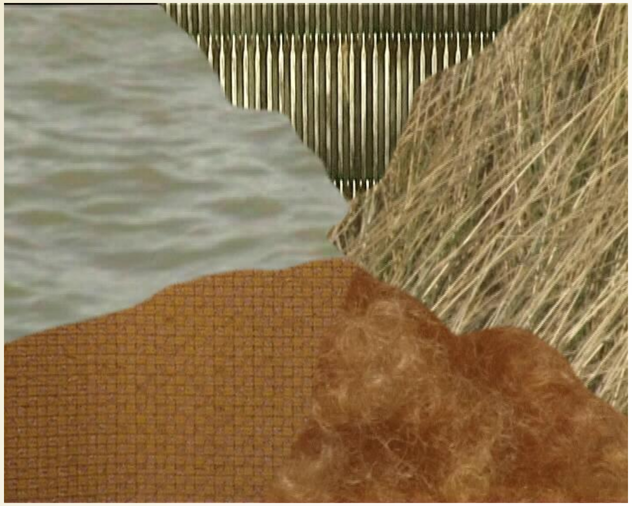
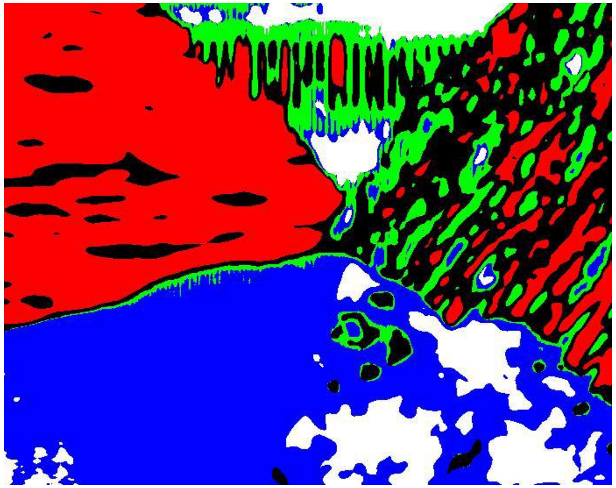
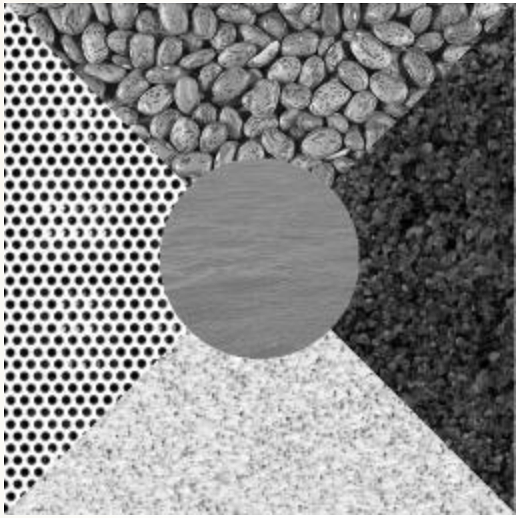
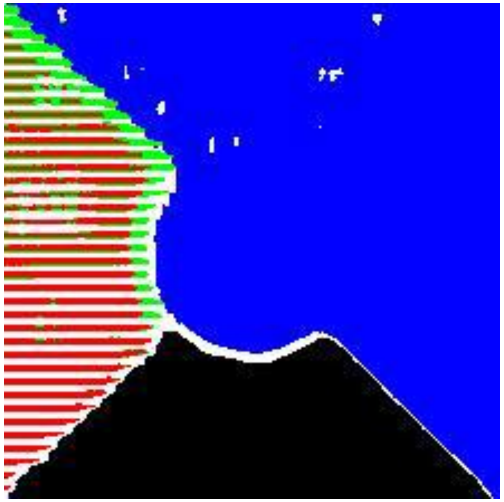
$$1\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, \ldots, \text{ and } \frac{N_c}{4}\sqrt{2} \text{ cycles/image-width}$$

- Feature extraction
  - Filtered images are subjected to a non-linear transformation.
  - This transforms the sinusoidal modulations in the filtered images to square modulations and acts as a blob detector
  - Accurate localization of texture boundaries is done through Gaussian weighted windows
- Clustering
  - Clustering is done through k-means algorithm which expects prior knowledge of number of texture regions K centroids are assigned randomly
  - the spatial coordinates of the pixels as additional features to take into account the spatial adjacency information in the clustering process

Drawbacks

- The Gabor filters are better in supervised texture segmentation where filter locations can be chosen based on empirical information of power spectrum characteristics of different textures.
- Algorithm very complex and required extensive fine-tuning of parameters
- K-means clustering makes algorithm supervised. Unpredictable even for same input, due to convergence to local minima in k-means.

Results (Gabor)

Graph based Texture Segmentation

- Image represented as a graph
  - All pixels as vertices
  - Adjacent pixels connected by edges
  - Weights as the Euclidian distance between intensity values at each pixel
- Split the graph into connected regions
- Comparing similarity in regions
  - Internal difference of a component

$$Int(C) = \max_{e \in MST(C,E)} w(e)$$

  - Difference between 2 components

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j))$$

- Evidence for boundary between a pair of components

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } \frac{Dif(C_1,C_2)}{MInt(C_1,C_2)} > 1 \\ \text{false} & \text{otherwise} \end{cases}$$

Where

$$MInt(C_1, C_2) = min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$$

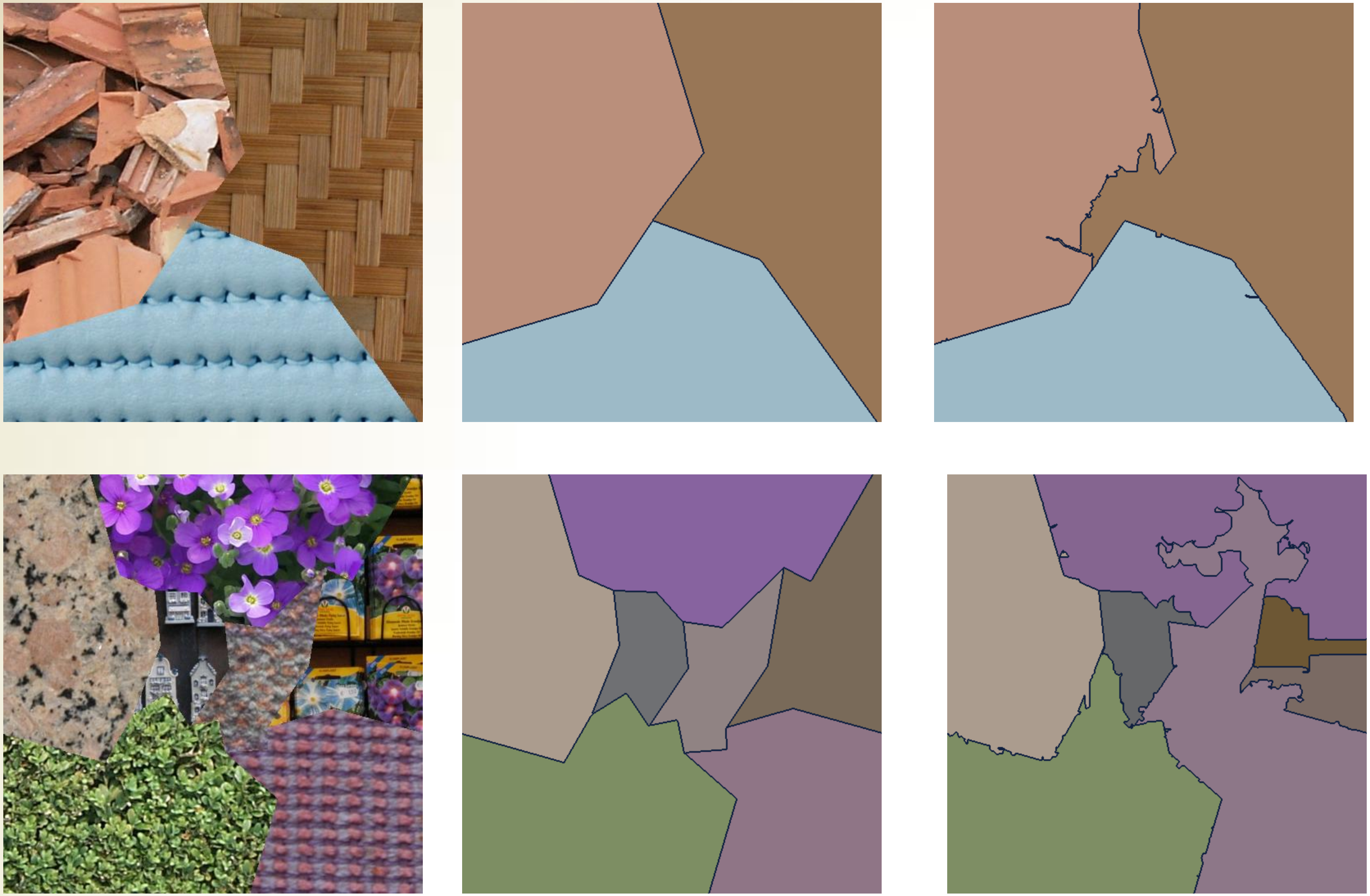
Here tau controls the degree to which difference between 2 components should be greater than internal difference for boundary to exist.

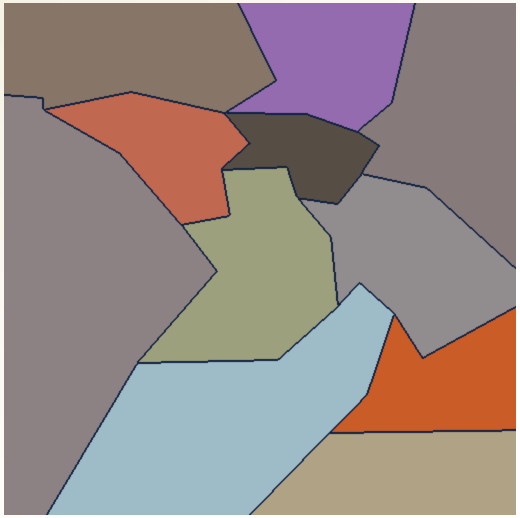
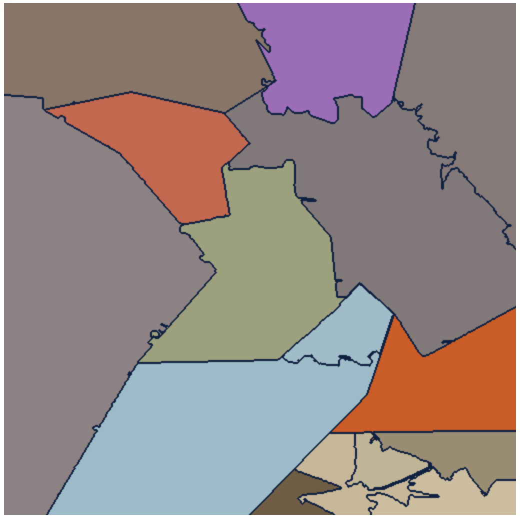
$$\tau(C) = \frac{k}{|C|}$$

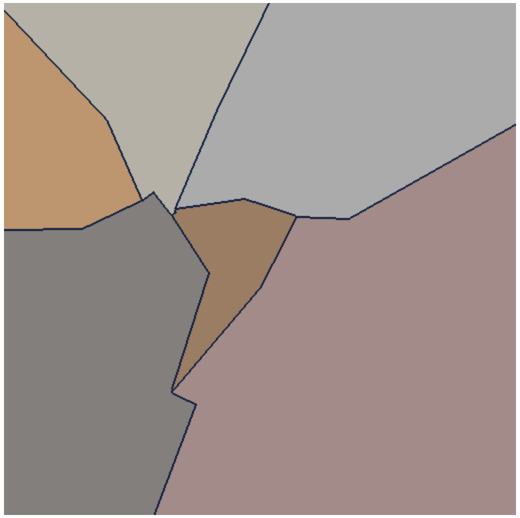
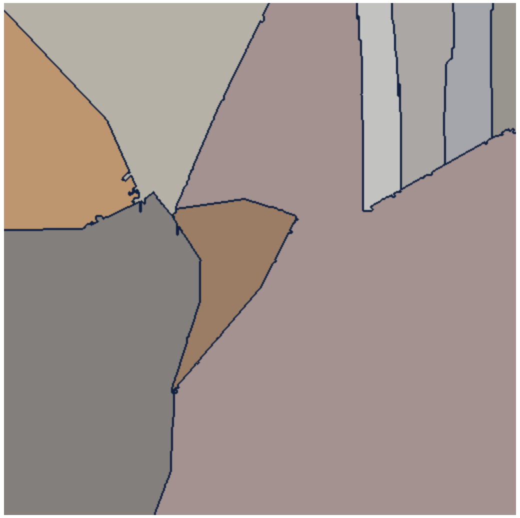
Algorithm

- Sort edges by increasing order of weights
- Initialize each vertex in its own component
- Consider each edge in above sorted list
  - Let e=(vi, vj)
  - If vi and vj in different components, and weight of edge less than internal difference of two components
    - Merge the 2 components
- Return final segmented graph

Results



IIIT Hyderabad

Evaluation and Benchmarking

- Criteria:
  - Region based: Compare the machine segmented regions with the correct ground truth regions. The overlap acceptance is controlled by a parameter k, as, correct detection occurs when, (where R_i (ground truth) and R_j are the regions to be compared)

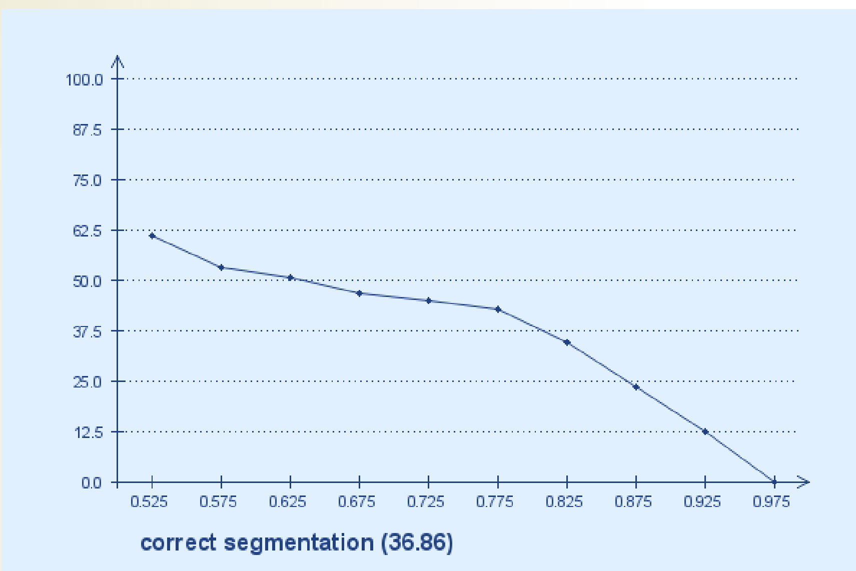
$$| R_i \cap R_j | \geq k \, | R_i |$$

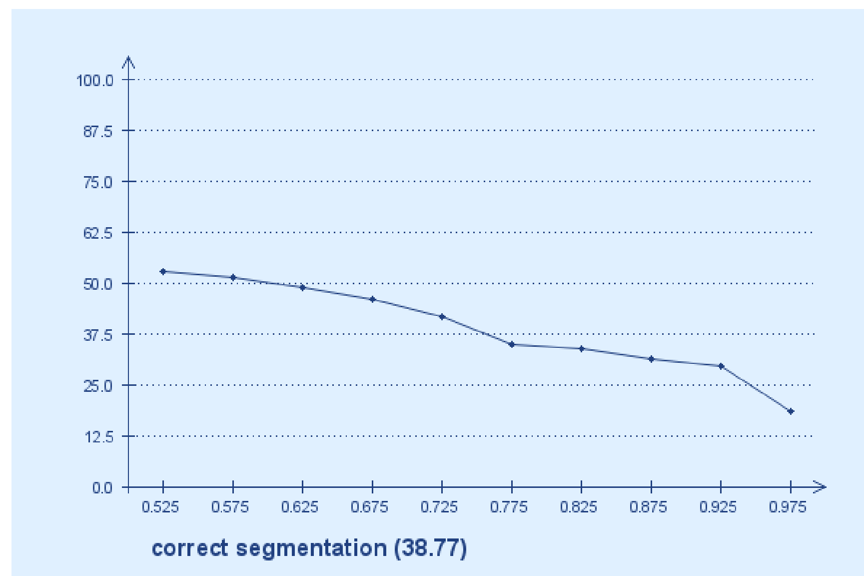
- Automatic evaluator: http://mosaic.utia.cas.cz
- Generates random texture mosaics using large collection of high resolution texture images

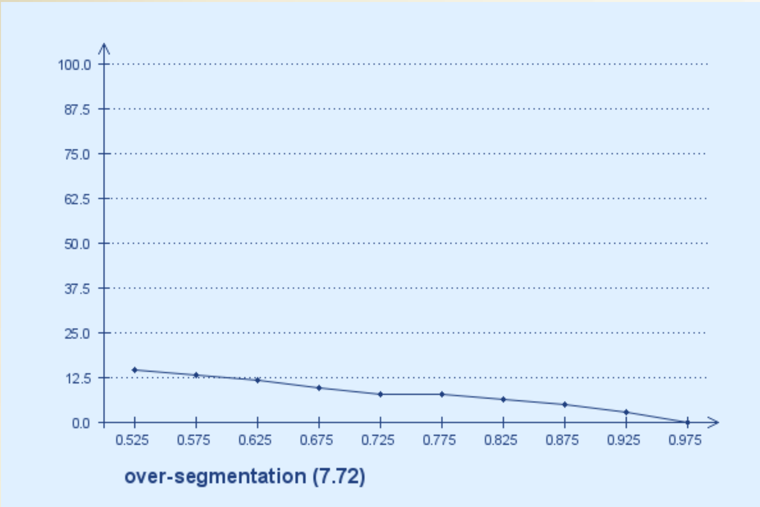
Michal Haindl and Stanislav Mikes. Texture segmentation benchmark In ICPR, pages 1-4. IEEE, 2008.

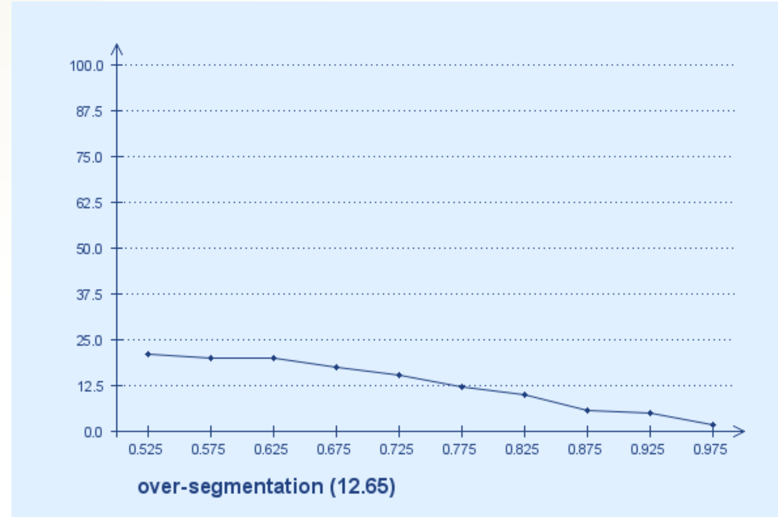
Evaluation Results

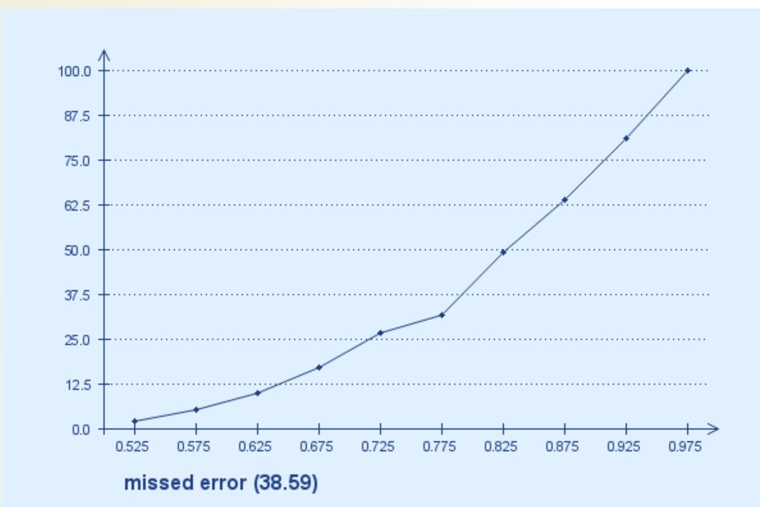
- LBP based Segmentation
  - Average 42% Correct segmentation
  - Ranked first in all grayscale segmentation algorithms implementations
- Graph based Segmentation
  - Average 40% Correct segmentation
  - Ranked 23rd in all color segmentation algorithms implementations

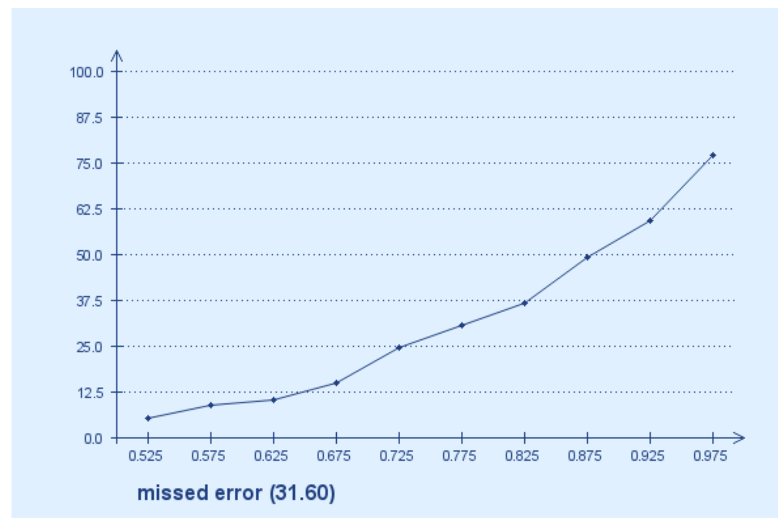








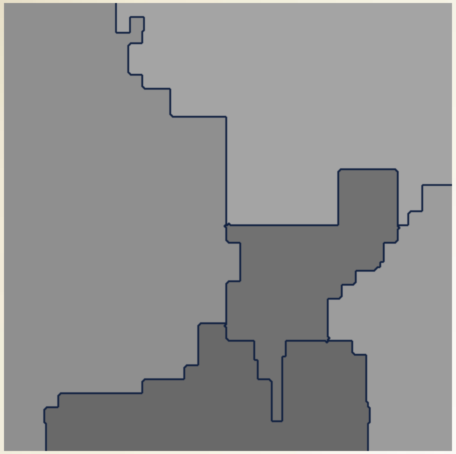


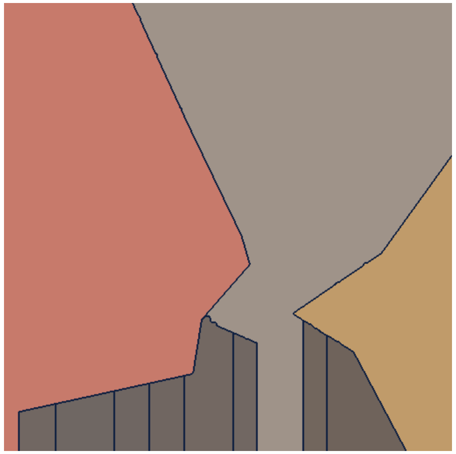


LBP Algorithm

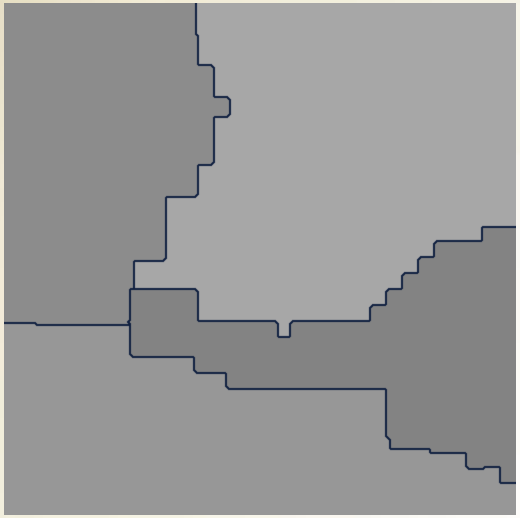
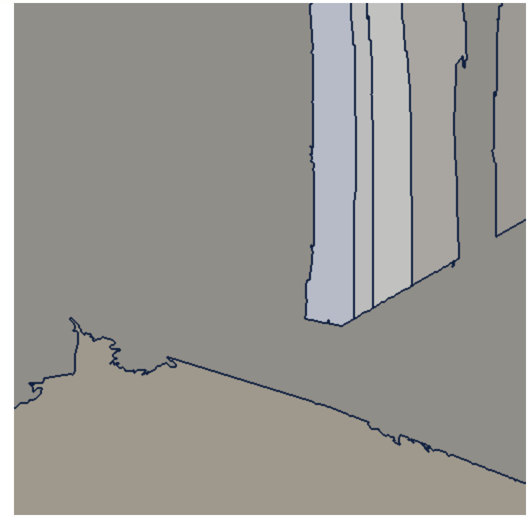
Graph Algorithm

Observations/Comparison



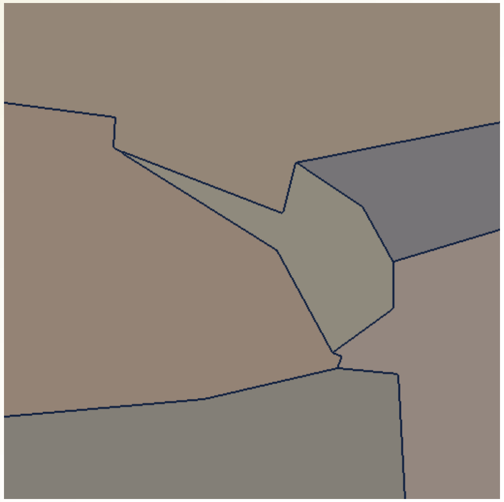


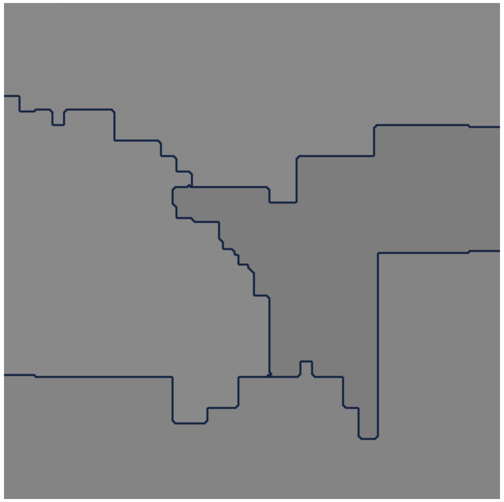
- Accuracy
  - LBP gives better accuracy
  - LBP/C histogram encodes texture information well
  - Pixel level precision better for Graph based, uses color information
  - LBP results mostly blocky, due to split and merge

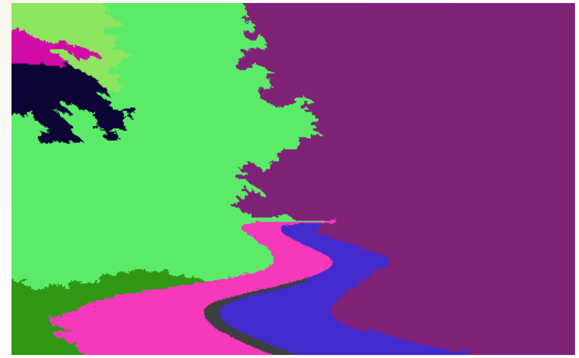
- Sensitivity
  - Graph more sensitive to in homogeneity in textures
  - LBP more stable to variations within texture pattern
  - LBP takes local characteristics into consideration as LBP/C histogram, while graph based is limited to pixel level analysis.
  - Graph based required fine-tuning of parameters, LBP more autonomous.

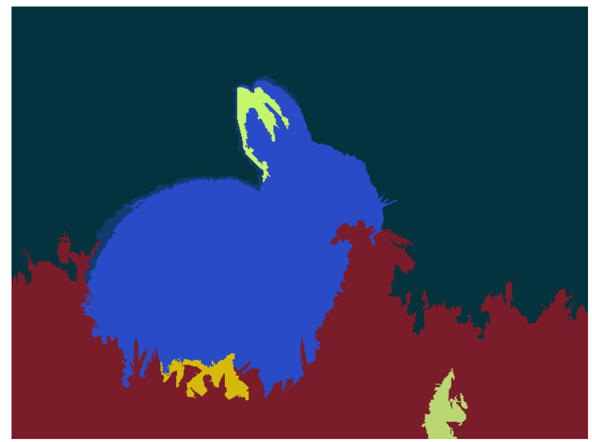
Real-life Applications





- LBP Based:
  - Very stable in encoding texture
  - Can be used for segmentation in complicated and very similar textures
  - Comparatively slow to Graph based





- Graph Based
  - Uses color information
  - Excellent segmentation in natural scenes
  - Very Fast, can be used in real-time videos

Thank you