GNR602
Advanced Methods in Satellite Image Processing

Instructor: Prof. B. Krishna Mohan

CSRE, IIT Bombay

bkmohan@csre.iitb.ac.in

Slot 13

Lecture 07-08 Contextual Classification by Relaxation Labeling Process

Contents of the Lecture

- Role of Context
- Initial likelihoods of pixels
- Modeling context
- Updation of pixel likelihoods
- Applications

Role of Context

- Same words can result in different meanings in different contexts.

The summer heat can *kill* you in India.

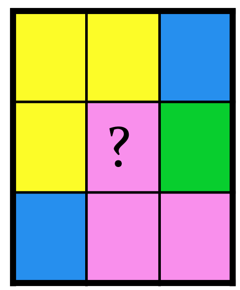
He made a *kill* during the stock market boom.

Some of the political parties are trying to *kill* the nuclear deal

The forest guards caught a person who was trying to *kill* a tiger.

Role of Context

“ Tell me about your friends and I will tell you who you are”—Albert Einstein

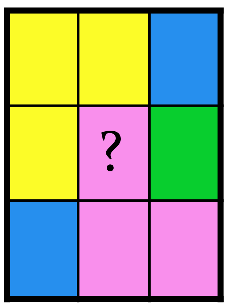


? **pixel** under consideration for classification
Rest of the pixels :: **neighboring** information

Context Based Image Analysis

- Context in image analysis is very important for classification, as well as for post-classification refinement.
- The presence or absence of a class at a pixel can have a bearing on the class to be assigned to a neighboring pixel.

Context Based Image Analysis

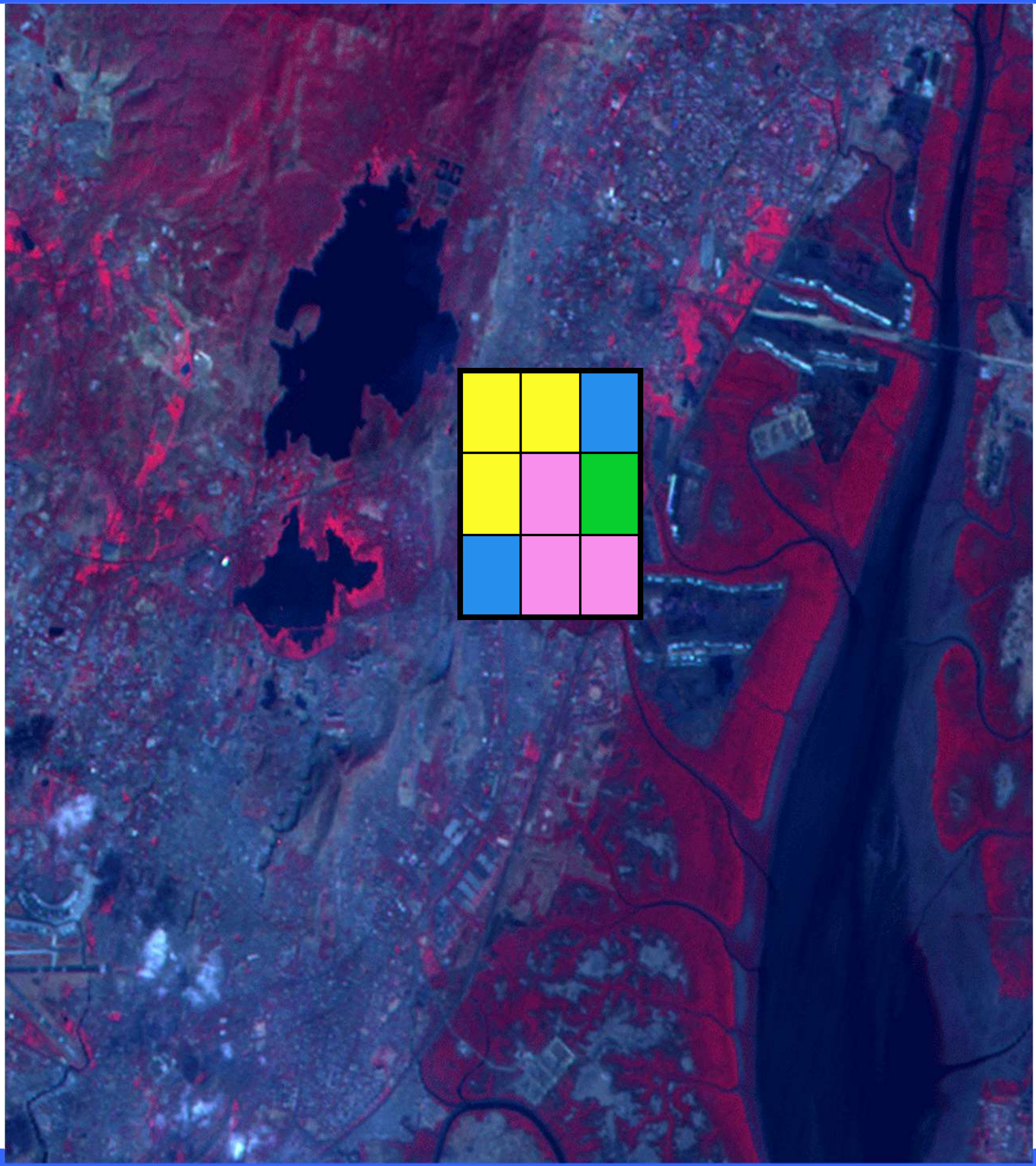


- Contextual information is derived from neighboring pixels

- It is expressed in terms of **inter-class compatibility at adjacent positions**, **neighbor support** and class **label probabilities**.

- Since the classification is context based hence the name contextual classification.

- It is based on the fact that pixels of the same class/label/ feature tend to be near each other.



Some Examples

- A water pixel may have pixels of same class in the neighborhood
- A sea water pixel may not have a paddy field pixel in its neighborhood
- A building should have a road nearby

Neighborhood Class Constraints

The presence of a class at a pixel can affect the likelihood of another class to be in the vicinity.

- A tall tree will have a shadow in the neighborhood
- A building will have a road in the neighborhood

These relationships need not be symmetric

- A shadow may not imply a building nearby
- A road may not have any particular object nearby

Typical Classification Problem

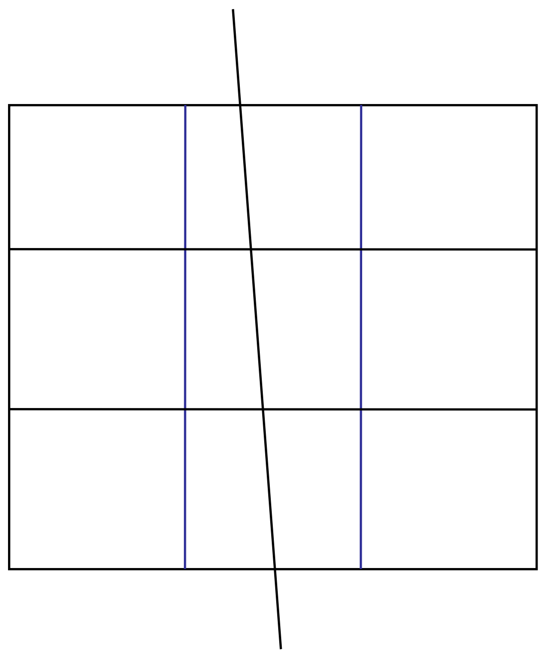
- For a landuse classification problem, we have the sample labels such as:
- Ponds, Lakes, Sea, Forest, Open land, Road, Buildings, Fields, Hills
- We extract a set of features (attributes) for each object (region, tile) in the image based on:
  - Spectral features
  - NDVI
  - Texture features
  - Shape features
  - Contextual features …

Initial Classification by ML

- **Hard Classification Rule**
  - Assign pixel with feature vector X to class λ if $p(\lambda|x)$ is highest over all classes
  - Information present in the likelihoods of X to belong to other classes is discarded
  - Even if another class has likelihood very close to that of λ, that information is not made use of

Mixed Pixel

- Class 1 Class 2



Class 1 pixels will have signatures different from signatures of class 2

Mixed pixels will have influence of signatures of both classes 1 and 2

Mixed Pixel

- Classification of mixed pixels
  - Mixed pixels can get classified into class 1 or class 2 depending on which class influences the signature
  - However, the difference in likelihoods will not be substantial as in case of pure pixels

$$|P(\lambda_1| x) - P(\lambda_2|x)| < T$$

  - If likelihoods of two classes are comparable, we are dealing with mixed pixels

Mixed Pixel

The spectral information associated with mixed pixels will be inadequate for unique classification of such pixels

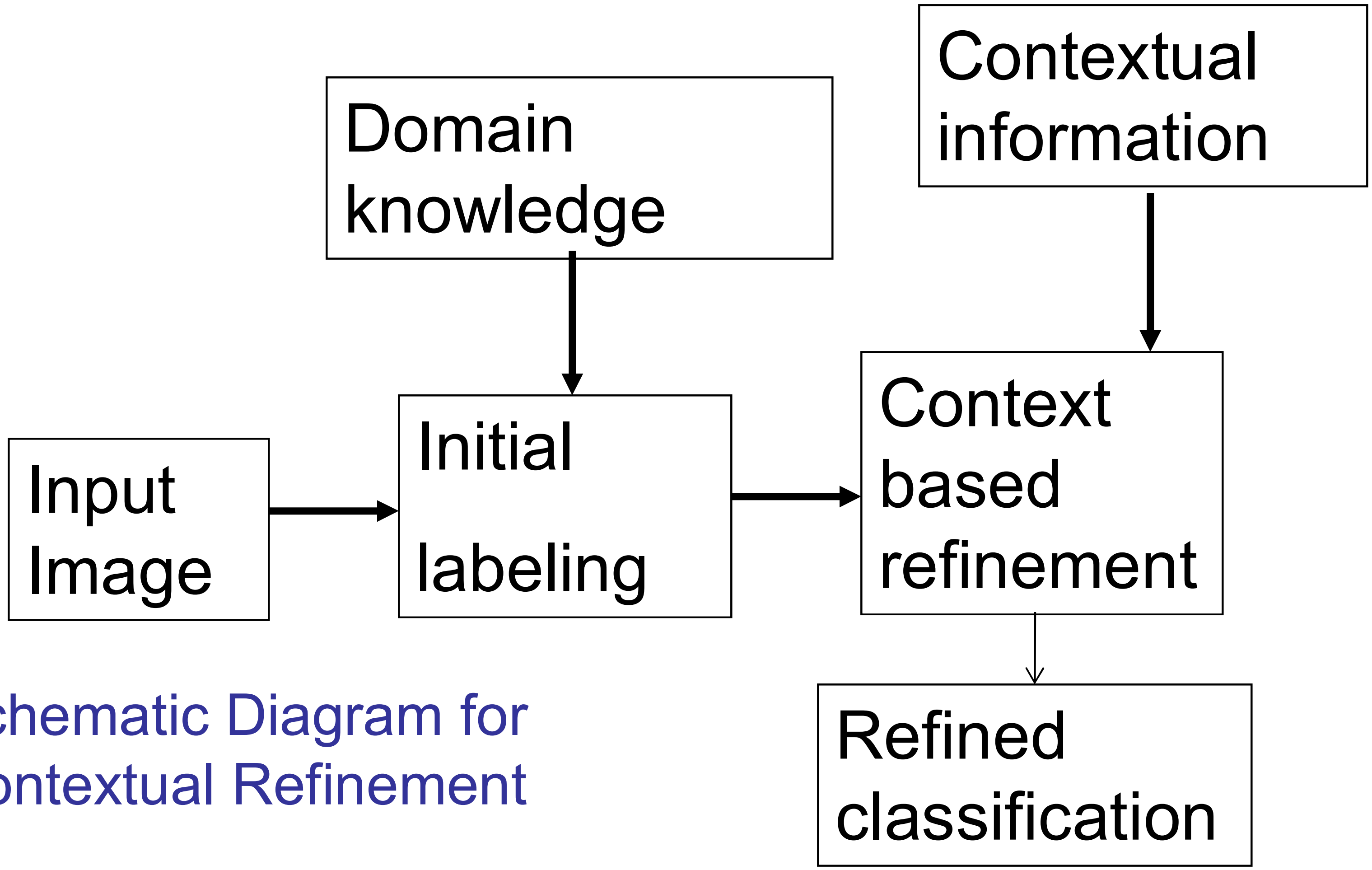
The spatial information in the form of local neighborhood can resolve the ambiguity in their classes

Noisy Pixels

- In case of noisy images, certain pixels may be assigned to classes inconsistent with the classes assigned to their neighbors
- For example, a water pixel within settlements, building pixels within desert areas, …
- Spatial information can reduce the misclassification caused by noise

Approach in Context Based Image Analysis

- Image analysis using a small neighborhood at a time (e.g. 3x3)
- OR
- Analyze the image on per-pixel basis and refine the analysis based on context
- The latter part is the focus of the discussion here.



Schematic Diagram for Contextual Refinement

Initial Classification by ML

- **Soft Classification Approach**
  - Likelihood of a pixel to belong to all the possible classes retained
  - If the likelihoods of two classes λ_1, λ_2 are close, i.e.,

$$| p(\lambda_1|X) - p(\lambda_2|X) | \leq \varepsilon$$

the location of the pixel can be noted to examine its neighborhood.

Initial labeling

- When we compute the class likelihoods, we can retain the individual class conditional probabilities $p(\lambda|x)$ as the initial labeling
- These probabilities are computed per pixel (or per object), and most of the conventional methods have no way of knowing if the class present at the neighbors is inconsistent with the class being assigned to a pixel

Classification by ML

- To compute the conditional probability that a region takes label λ ∈L, given its feature vector X, we use Bayes' Rule:

$$p(\lambda \mid \mathbf{X}) = \frac{p(\lambda)\, p(\mathbf{X} \mid \lambda)}{\sum_{\lambda' \in \Lambda} p(\lambda')\, p(\mathbf{X} \mid \lambda')}$$

$$= \frac{p(\lambda)\, p(\mathbf{X} \mid \lambda)}{p(\mathbf{X})}$$

Initial probabilities for segmentation by thresholding

- gray level of pixel $f_i < f_{mean}$
- $p_i(black) = 0.5 + 0.5 [(f_{mean} - f_i) / (f_{mean} – B)]$
- gray level of pixel $f_i > f_{mean}$
- $p_i(white) = 0.5 + 0.5 [(f_i - f_{mean}) / (W - f_{mean})]$

Contextual Information

- The inter-class dependence can be directional (if the labels are line/no-line, edge/no-edge, road, canal etc.) or omni-directional (water, forest, agricultural fields, open lands etc.)
- In general, the class-class compatibility is denoted by $r_{ij}(\lambda,\lambda')$
- The notation stands for
- **Compatibility of class λ at pixel i with class λ' at pixel j, where pixel i and pixel j are neighbors**
- If directionality is not important, $r_{ij}(\lambda,\lambda') = r(\lambda,\lambda')$
- The position of occurrence of classes is of no consequence.

Nature of compatibility coefficients

- Compatibility coefficients are modeled on several intuitive statistical relationships:
  - Correlation Coefficient
  - Conditional Probability
  - Mutual Information
  - Specially designed functions in case of line/edge labeling
- They can be numerically in the range [-1 1] or [0 1].

Philosophy of relaxation labeling

- Initial labeling is assigned by any standard classification process – ML, Bayes, ANN, K-means …
- Class compatibilities are defined based on the actual classes and the application
- The class label probability at a pixel is updated based on the ***support*** received from the neighbors
- Support from any neighbor is a function of its own class likelihoods and the inter-class compatibilities

Philosophy of relaxation labeling

- Support from the entire neighborhood is the weighted average of supports from all neighbors
- Finally, the likelihood of a class for the pixel under consideration is updated based on the support received from the entire neighborhood

Calculation of Neighbor Support

- Support from neighbor j to label λ at pixel i is defined as:
- $q_{ij}(\lambda) = \frac{1}{L}\sum_{\lambda'=1}^{L} r_{ij}(\lambda,\lambda')p_j(\lambda')$
- In case $-1 \le r_{ij}(\lambda,\lambda') \le +1$, then
- $q_{ij}(\lambda) = 0.5(1 + q_{ij}(\lambda))$ (shift supports from [-1 1] to [0 1]
- (Various models for compatibility coefficients are discussed later)

Calculation of Neighbor Support

- Overall support received for label λ at pixel i from the entire neighborhood is given by
- $$q_i(\lambda) = \frac{1}{|\{N_i\}|}\sum_{j=1}^{N_i} q_{ij}(\lambda)$$

  $$OR$$

  $$q_i(\lambda) = \sum_{j=1}^{N_i} c_j q_{ij}(\lambda), \text{ where } 0 \le c_j \le 1, \text{ and } \sum_j c_j = 1$$
- The overall support is the simple or weighted average of the individual neighbor supports. $q_i(.)$ are computed for every class label λ.

Label Probability Updating

- There are several models for updating label probabilities. One of the commonly used models is given by

$$p_i^{n+1}(\lambda) = f(p_i^n(\lambda), q_i(\lambda))$$

- If a label λ receives strong support from the neighborhood compared to other labels, then its likelihood gets stronger, while the likelihoods of the labels get weaker with iterations. Eventually one label probability tends to 1 while rest to 0.

Label Probability Updating

- Method 1:

- $p_i^{n+1}(\lambda) = \dfrac{p_i^n(\lambda)q_i(\lambda)}{\sum_{\lambda'=1}^{L} p_i^n(\lambda')q_i(\lambda')}$

- The denominator acts as a normalizing factor to ensure that

$$\sum_{\lambda=1}^{L} p_i^{n+1}(\lambda) = 1 \text{ given that } \sum_{\lambda=1}^{L} p_i^{n}(\lambda) = 1$$

Standard Label Updating Rule

- Depends on the neighbor support for each class
- If the overall support for a label is high, the label probability increases
- The neighbor support is dependent on the strength of the class probabilities of the neighbors and inter-class compatibilities

Product Rule

- In this case, the neighbor support is taken as the product of individual neighbor supports

$$q_i(\lambda) = \prod_j q_{ij}(\lambda)$$
$$= \prod_j r_{ij}(\lambda, \lambda') p_j(\lambda')$$

- The update rule is given by

$$p_i^{n+1}(\lambda) = \frac{p_i^n(\lambda) \prod_j q_{ij}(\lambda)}{\sum_{\lambda'=1}^{L} p_i^n(\lambda') \prod_j q_{ij}(\lambda')}$$

Product Rule

- The supports being multiplied, if a single neighbor offers strong negative support the overall support can sharply reduce
- This is a conservative method of using neighborhood support

Arithmetic Rule

- One approach for label probability updating is based on an *online* approach, where for each neighbor the label probabilities are updated
- $p_{ij}^{n+1}(\lambda) = \dfrac{p_i^n(\lambda) q_{ij}(\lambda)}{\sum_{\lambda'=1}^{L} p_i^n(\lambda') q_{ij}(\lambda')}$
- $p_i^{n+1}(\lambda) = \sum_{j=1}^{|N_i|} c_j p_{ij}^{n+1}(\lambda)$

c_j are weights associated with contributions of different neighbors; c_j can be a constant for all neighbors, or can vary.

Arithmetic Rule

- Since label probabilities are updated *online*, one can include/exclude some neighbor supports
- One should be able to incorporate weightages to neighbor supports based on the relative assessment of the neighbor supports

Initial probabilities for edge detection

- Use an edge detector that generates an edge magnitude and edge orientation at each pixel
- Initial probabilities are computed from edge magnitude, updation employs both magnitude and orientation
- $p_i(\text{edge}) = \text{edgemag}_i$ / max(edgemag in image)
- $p_i(\text{noedge}) = 1.0 - p_i(\text{edge})$

Initial probabilities for segmentation by thresholding

- gray level of pixel $f_i < f_{mean}$
- $p_i(black) = 0.5 + 0.5 [(f_{mean} - f_i) / (f_{mean} – B)]$
- gray level of pixel $f_i > f_{mean}$
- $p_i(white) = 0.5 + 0.5 [(f_i - f_{mean}) / (W - f_{mean})]$

Initial probabilities from K-means/MDM

- Given class mean vectors μ_i and pixel feature vector **x,** initial class likelihoods may be computed in terms of inverse of distances to respective class mean vectors

$$p(C_i \mid X) = \frac{\| X - \mu_i \|^{-1}}{\sum_{j=1}^{K} \| X - \mu_j \|^{-1}}$$

- K is the total number of classes. The distance be Euclidean / Mahalanobis …

Initial probabilities from ANN

- Given output layer responses o_i and pixel feature vector **x,** initial class likelihoods may be computed in terms of inverse of distances to respective class mean vectors

$$p(C_i \mid X) = \frac{o_i}{\sum_{j=1}^{K} o_j}$$

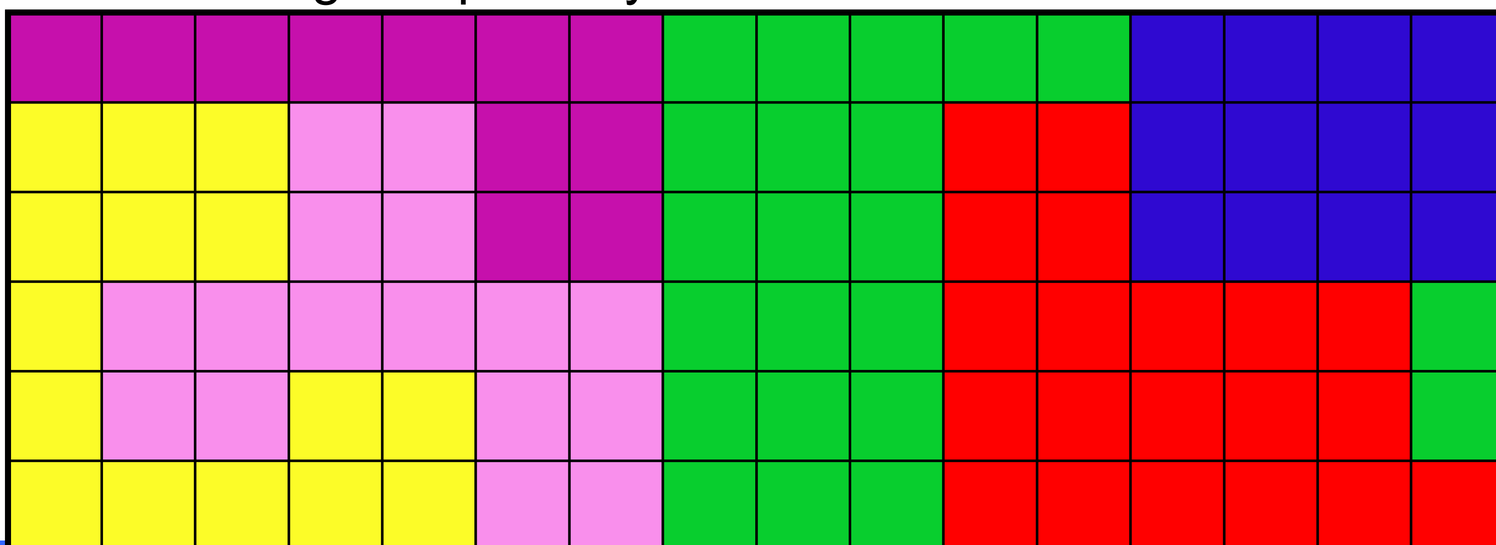
- K is the total number of classes; It is assumed that the output layer responses are non-negative. $P(C_i \mid X)$ is the same as $p_i^0(\lambda)$ used in the previous equations.

Compatibility coefficients

Compatibility coefficients

• task of relaxation labeling is to assign a consistent and unambiguous label to each point/pixel based on contextual information.

• contextual information concerning label consistency is usually represented using compatibility coefficients.



Compatibility coefficients

- The probabilities are updated using a set of given compatibility coefficients, represented by $r_{ij}\ (\lambda,\lambda')$
- The notation stands for the compatibility of label λ at pixel *i* to co-occur with label *λ'* at neighboring pixel *j*.
- These coefficients range between [-1 1] or [0 1]

- Properties :
  - If two labels are compatible then high positive
  - If two labels are incompatible then high negative
  - If neither labeling is constrained by the other, then 0 or 0.5 depending on numerical range chosen
  - The magnitude represents the strength of the compatibility.

j1	j	j2
j7	i	j3
j6	j5	j4

Compatibility coefficients

- Mutual information model

$$r_{ij}(\lambda,\lambda') = \ln \frac{p_{i,j}(\lambda,\lambda')}{\bar{p}(\lambda)\,\bar{p}(\lambda')}$$

- Principle
- I(A;B) = I(A) – I(A|B) = -ln(P(A) – (-lnP(A|B))

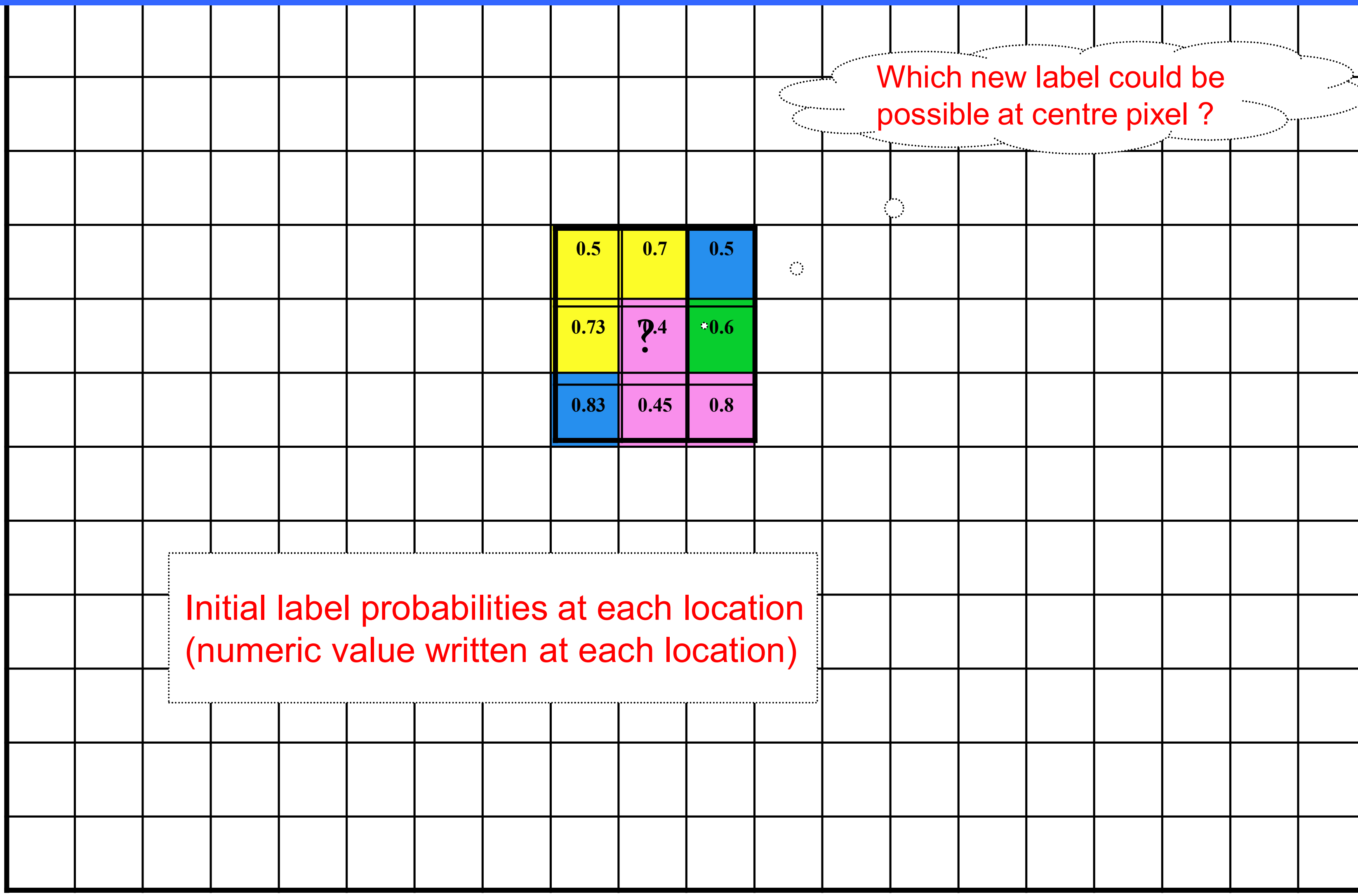
  = ln(P(A|B)/P(A)] = ln[P(A,B)/P(A)P(B)]
- Mutual information can take values in the range [-1 1]. If labels λ and λ' support each other, then mutual information tends to +1, and if they oppose each other, it tends towards -1.

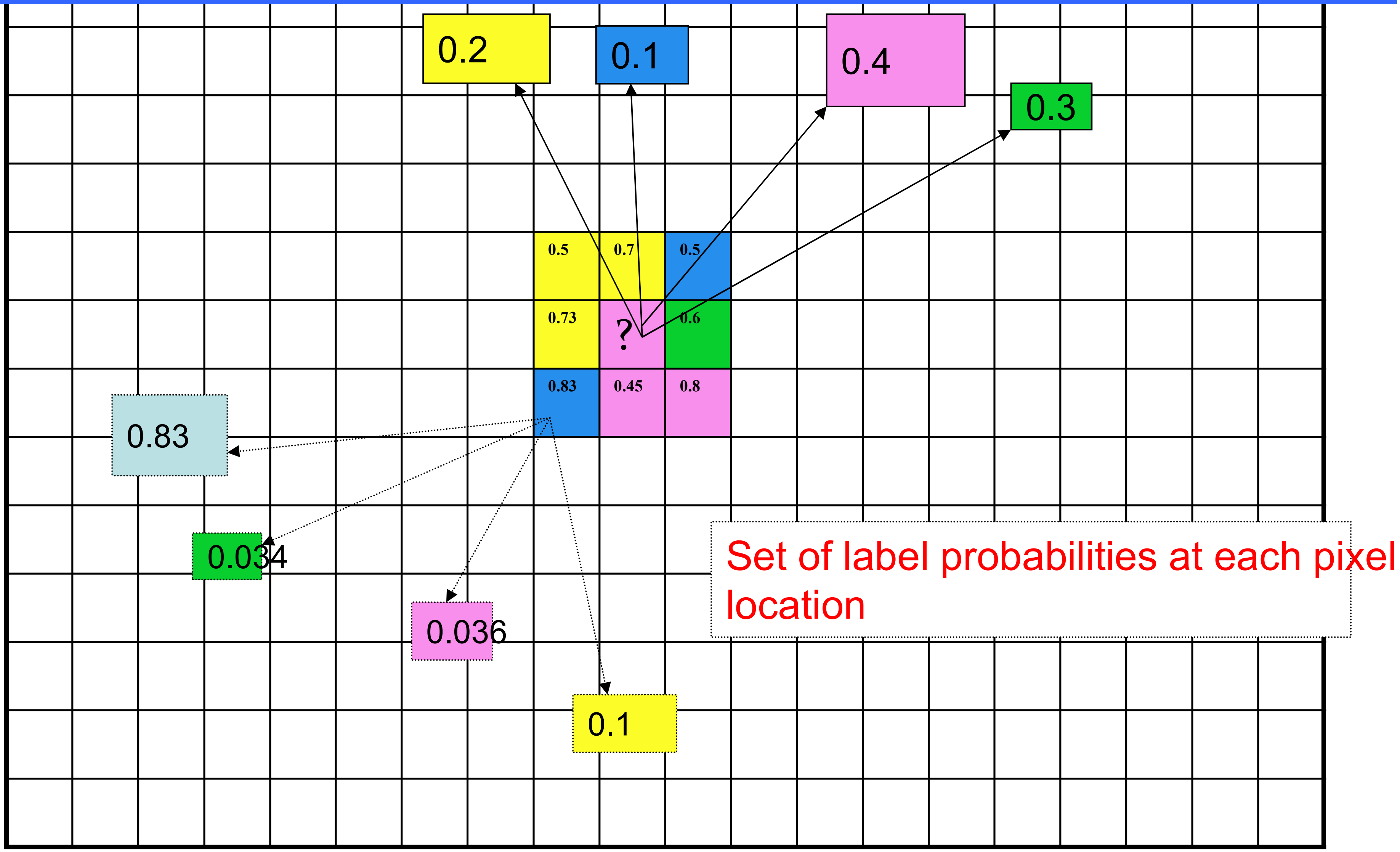
Compatibility coefficients

- Correlation Coefficient
- $r_{ij}(\lambda,\lambda') = \dfrac{\sum_i \sum_j \left[p_i(\lambda) - \bar{p}(\lambda)\right]\left[p_j(\lambda') - \bar{p}(\lambda')\right]}{MN.\sigma(\lambda)\sigma(\lambda')}$
- Correlation coefficient can take values in the range [-1 1]. If labels λ and λ' support each other, then correlation coefficient tends to +1, and if they oppose each other, it tends towards -1.

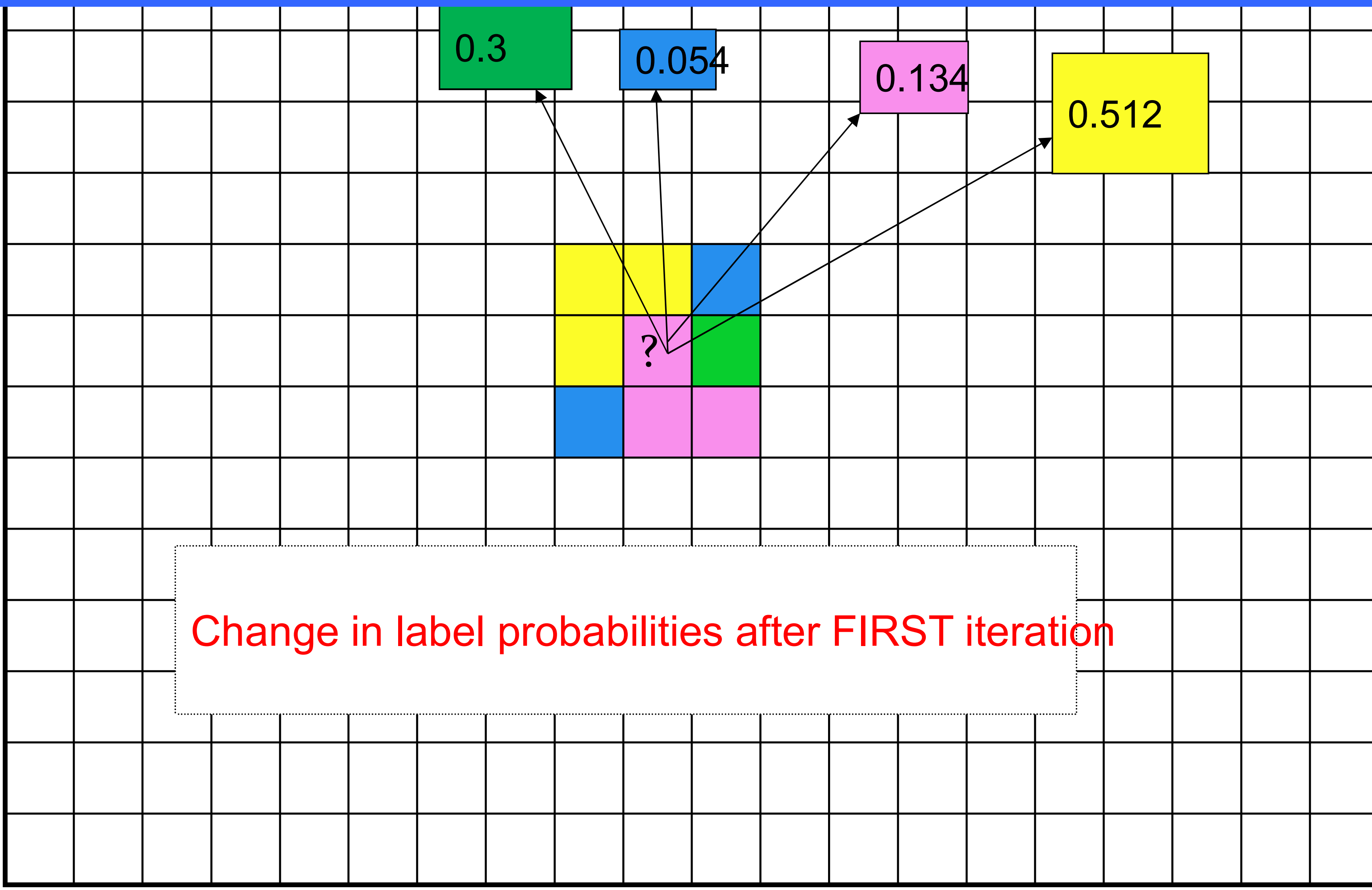
Conditional Probability

- Conditional probability model of compatibility coefficients is based on the relation
- $P(A|B) = P(A,B) / P(B)$
- Marginal probabilities and joint probabilities can be estimated from the co-occurrence matrix computed on the classification output.
- $r_{ij}(\lambda,\lambda') = \dfrac{\bar{P}_{ij}(\lambda,\lambda')}{\bar{P}(\lambda')}$

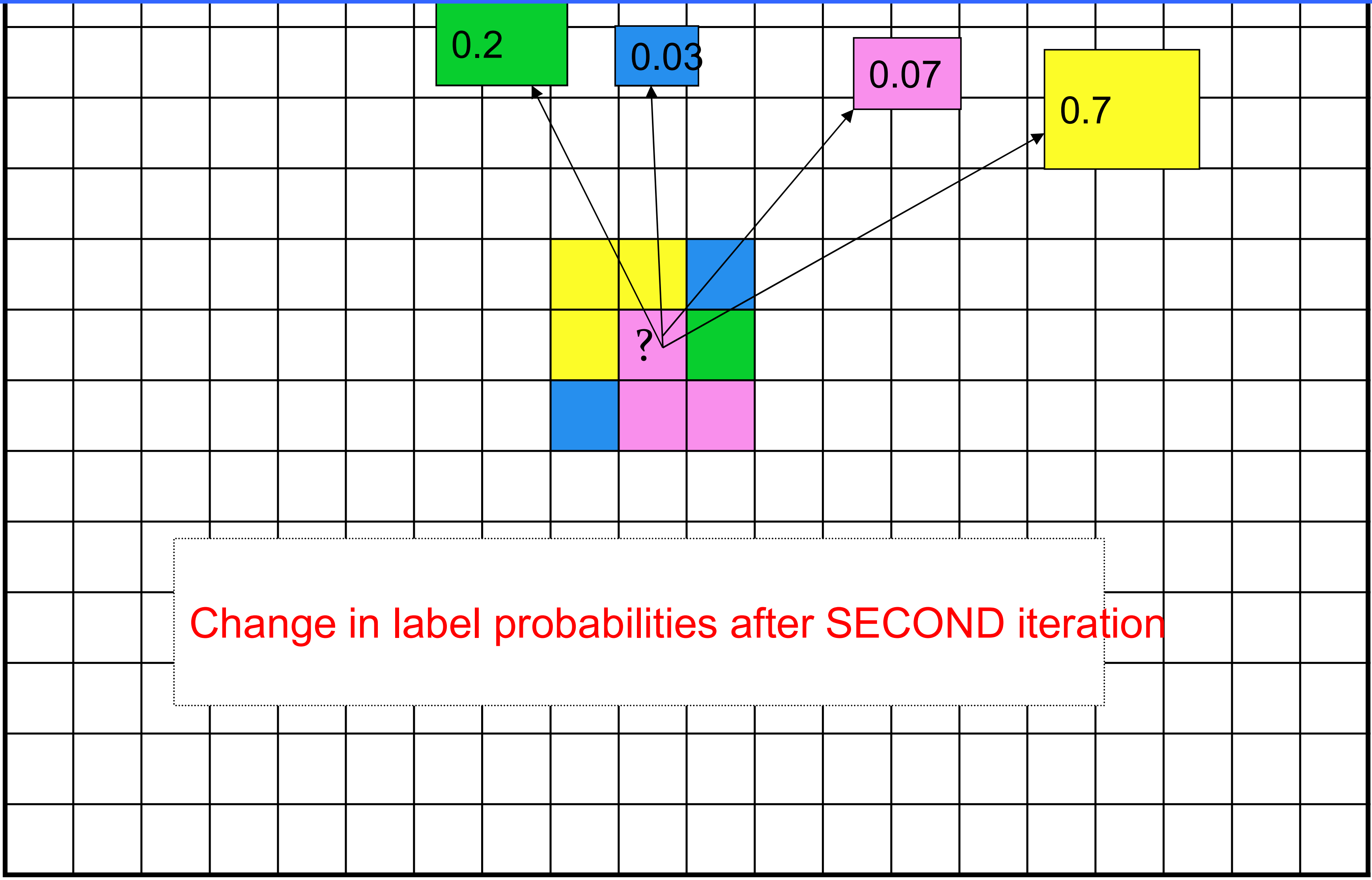

Which new label could be possible at centre pixel ?
0.5
0.7
0.5
0.73
?
0.4
0.6
0.83
0.45
0.8
Initial label probabilities at each location (numeric value written at each location)



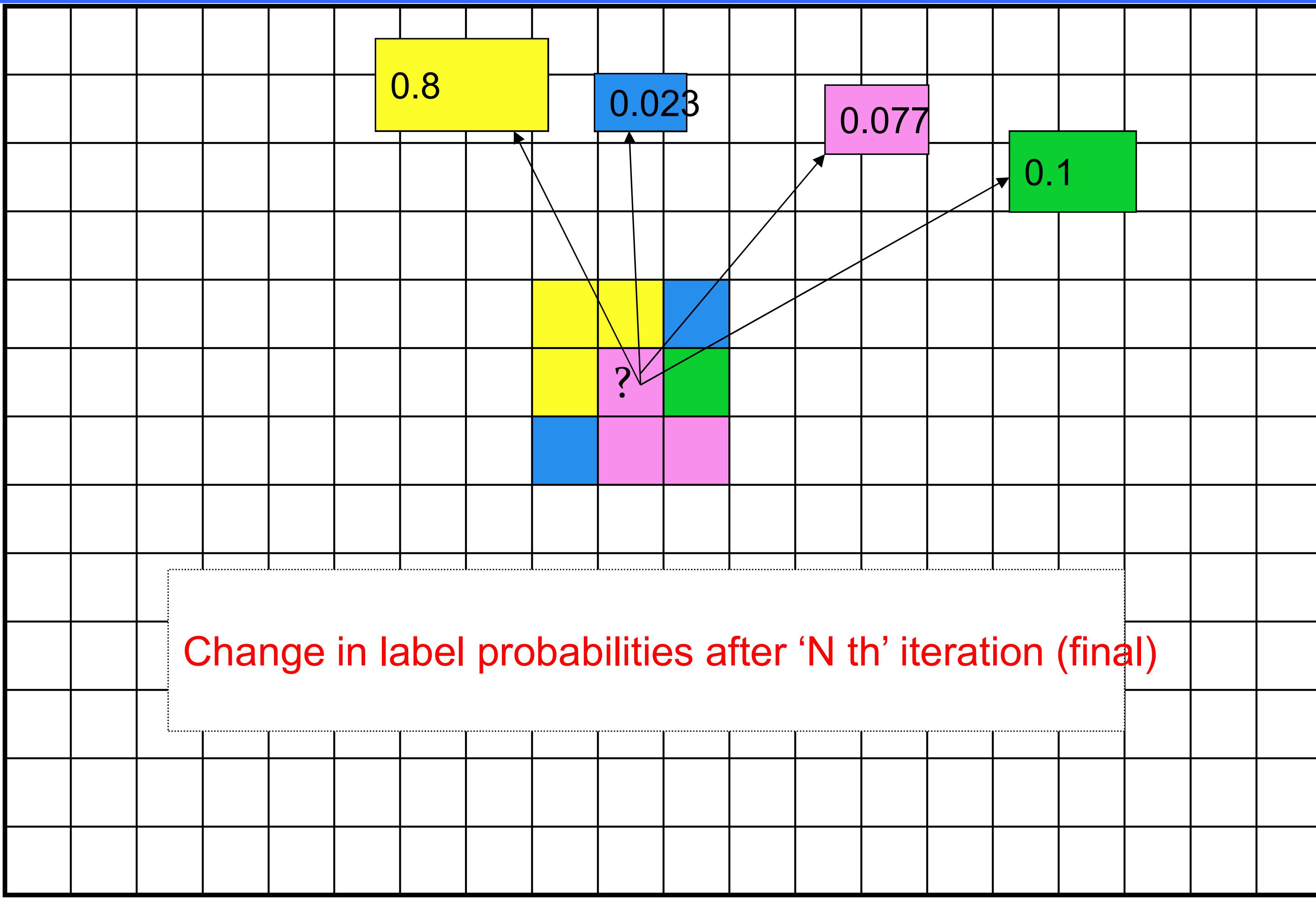
Set of label probabilities at each pixel location



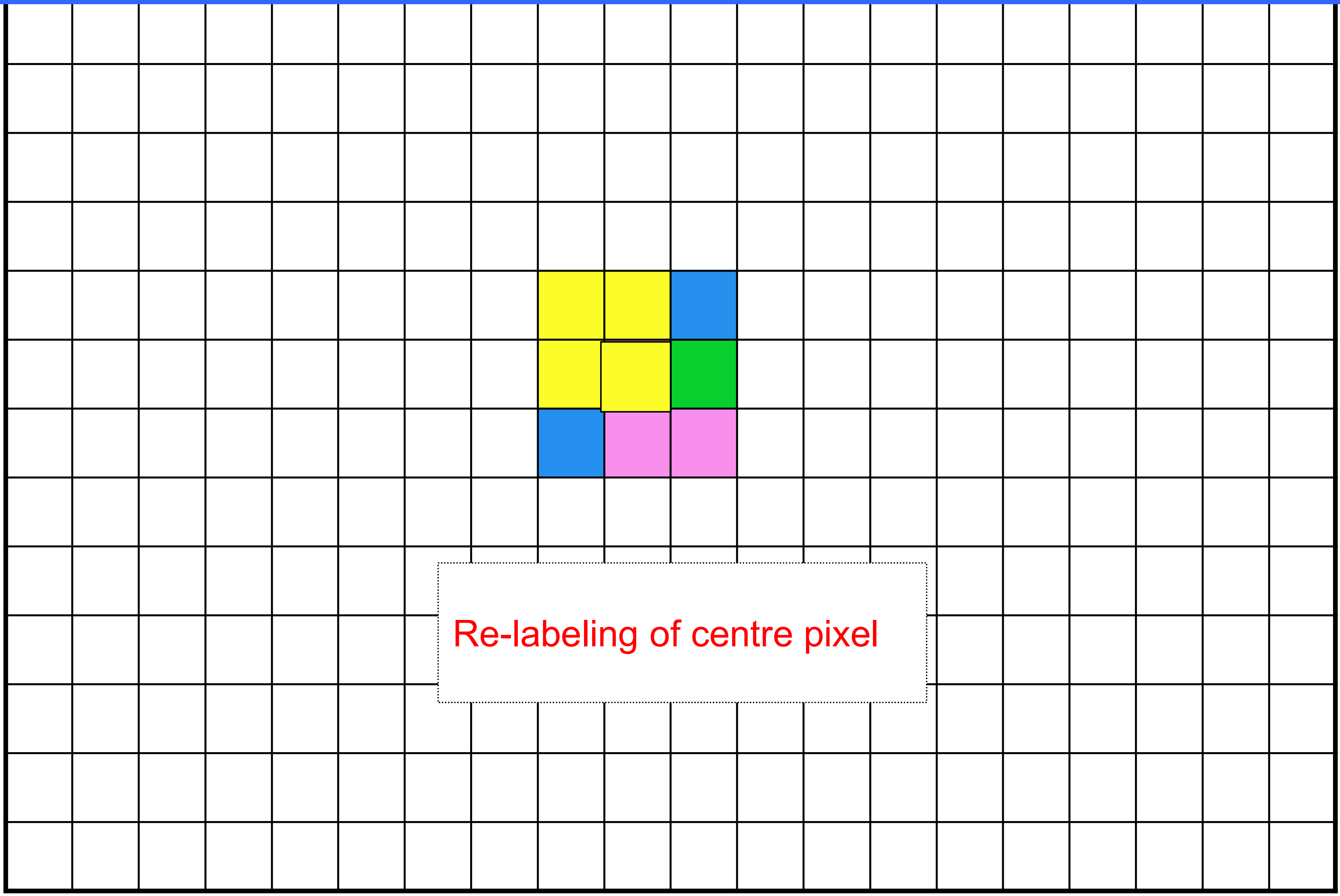
Change in label probabilities after FIRST iteration

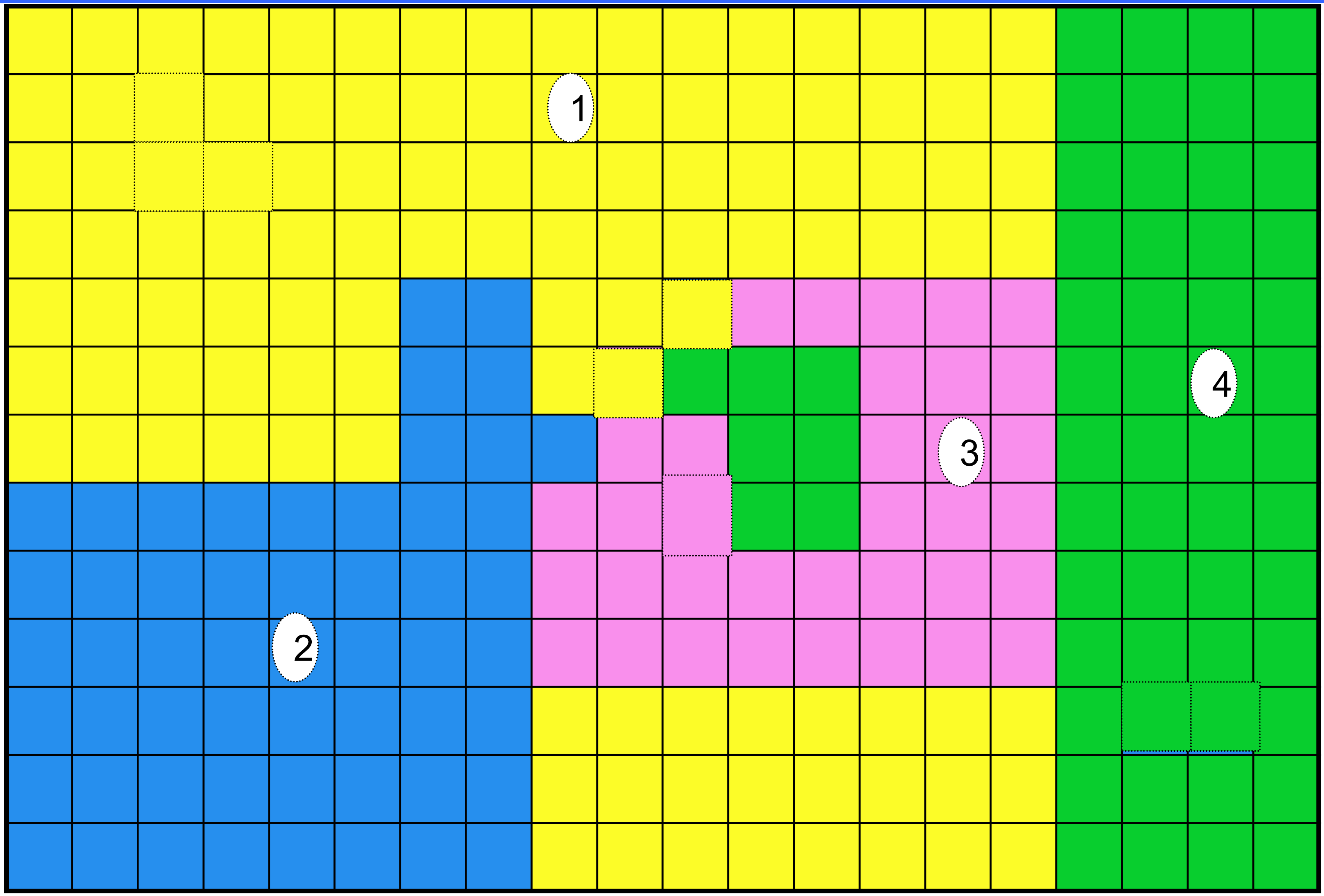


Change in label probabilities after SECOND iteration

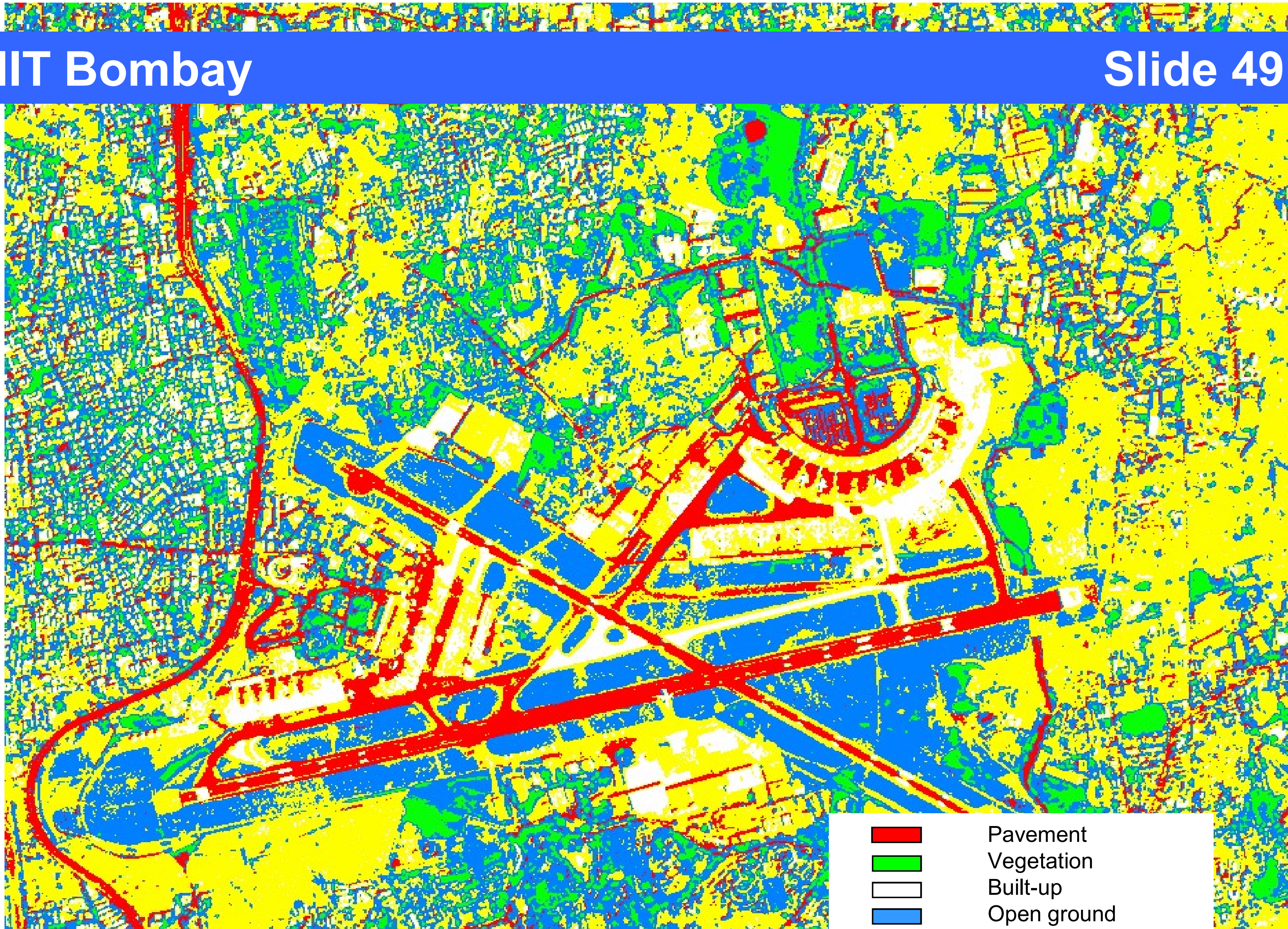


Change in label probabilities after 'N th' iteration (final)

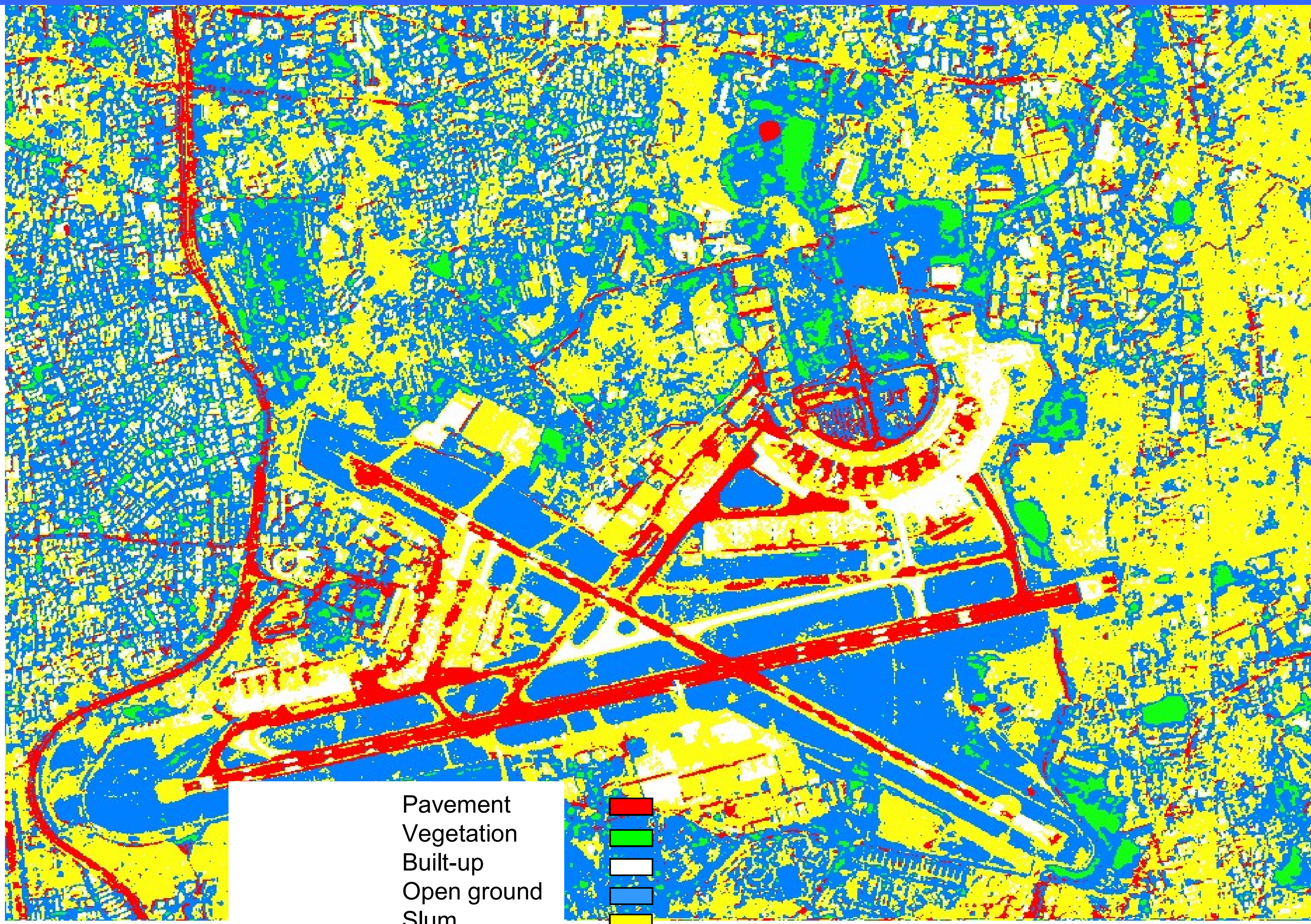

Re-labeling of centre pixel


1
2
3
4

False color composite image of International Airport, Mumbai (IRS P6 LISS IV)



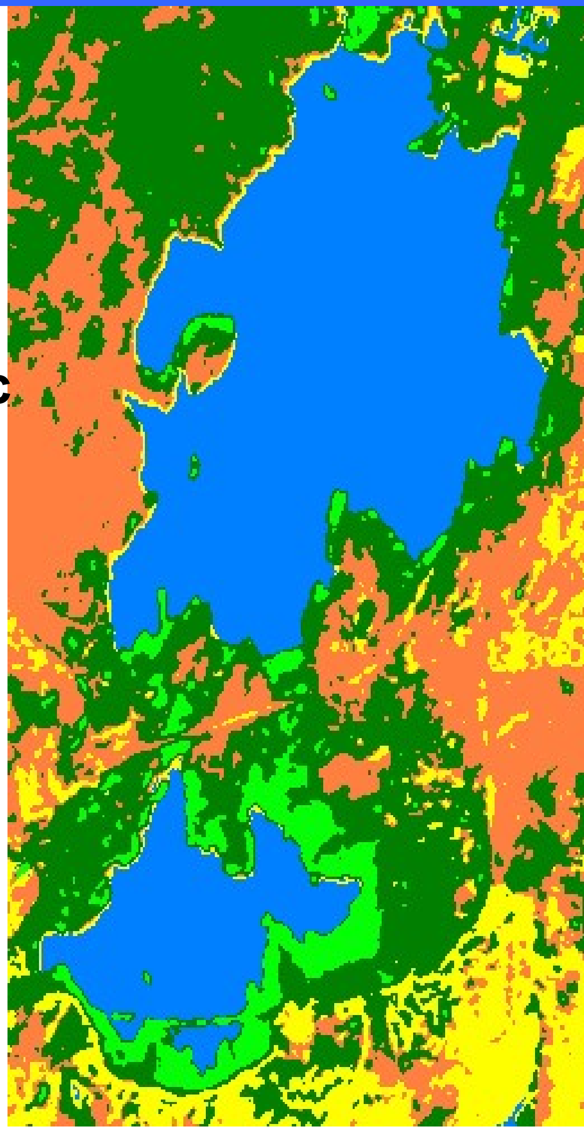
Initial classification by ANN

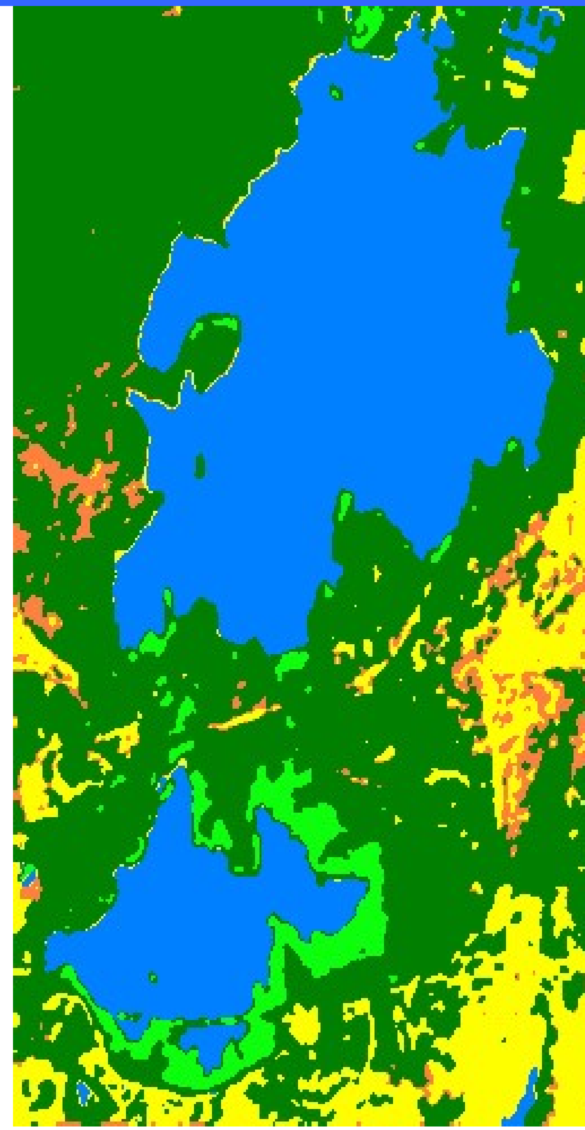


After 5 iterations of RLP

FCC image of
Powai area
(Landsat TM)

Thematic map (ANN)





Vegetation
Forest
Water
Built-up
Open ground

Refined using relaxation process (after 8 iterations)

Evaluation

- a) Change in the label probabilities – if the maximum change in label probabilities of any pixel exceeds some ε, continue the iterations

- If $| P_i^{n}(\lambda) - P_i^{n+1}(\lambda) | > \varepsilon$, for all i and λ then iterations continue

Evaluation

- b) Classify the image based on highest probabilities and check the change in classification with each iteration. Continue the iterations while the number of pixels changing class exceeds some threshold T

- If $\sum_{j=1}^{N} d_j$ > T, then continue iterations;

where

d_j = 1 if pixel j is assigned to different classes at iterations n and n+1

d_j = 0 if pixel j is assigned to the same class at iterations n and n+1

Evaluation

- Use test pixels and evaluate the accuracy of classification with each iteration. The accuracy normally peaks off and then starts going down. Stop at the point where the accuracy is maximum

Evaluation

- Distance of current label probabilities from initial assignment increases rapidly and levels off after a few iterations. Conversely, the distance between consecutive iterations starts at a high value, and decreases rapidly, and after a few iterations remains at a low value.

$$D_n = \sum_{i=1}^{N} \sum_{\lambda=1}^{L} \left[| \, p_i^n - p_i^0 \, | \right]$$

Contextual in High Resolution Image Analysis

- Context can resolve the errors in classification of certain objects
- For example, shadows and water bodies
- The need for certain objects to be adjacent shadows can help resolve the ambiguity
- Also, there will be a certain shape similarity between the object and its shadow

Can edge information be incorporated?

- At edge pixels, neighbors are split between two classes or more
- How should edge pixels be protected, since they do not have support of the entire neighborhood?

How to Include Edge Information?

- Obtain independent edge map using standard edge detector
- Apply conservative threshold so that strong edges are detected
- Scale down neighbor supports if a pixel is an edge pixel OR
- Leave edge pixels unchanged altogether

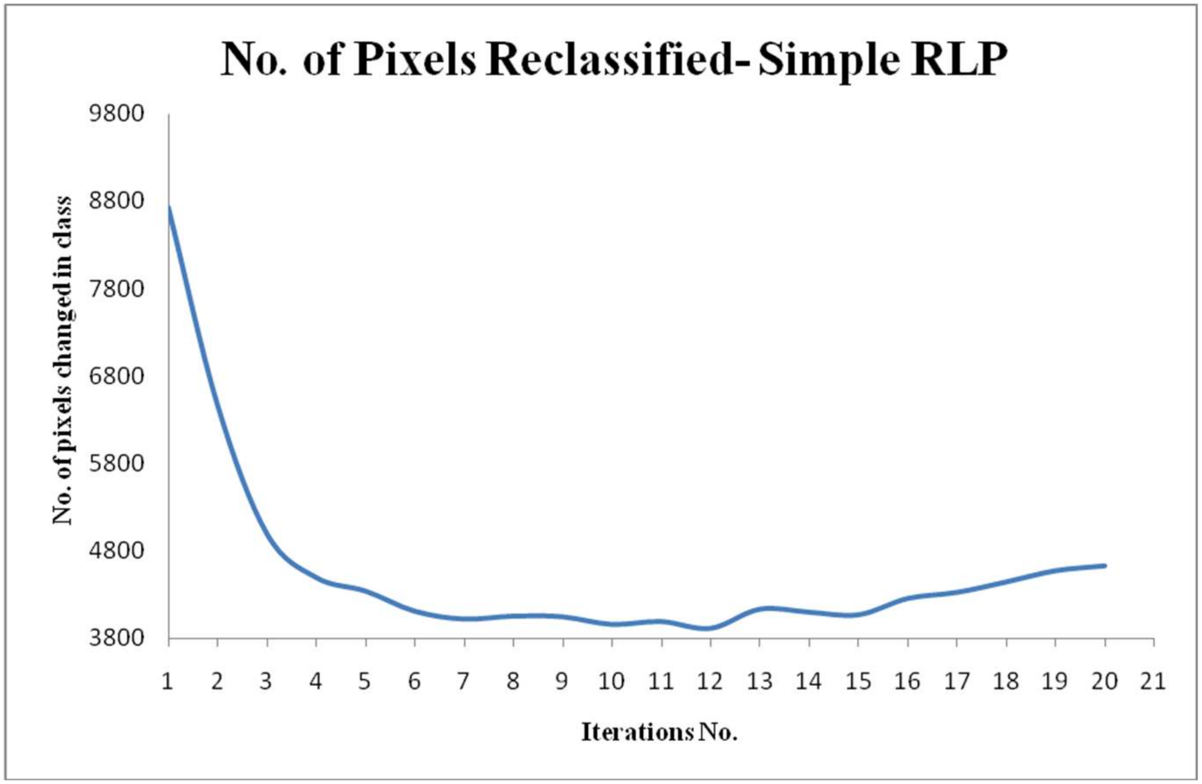
Convergence

- a) Change in the label probabilities – if the maximum change in label probabilities of any pixel exceeds some ε, continue the iterations
- b) Classify the image based on highest probabilities and check the change in classification with each iteration. Continue the iterations while the number of pixels changing class exceeds some threshold T

Convergence

- Distance of current label probabilities from initial assignment increases rapidly and levels off after a few iterations. Conversely, the distance between consecutive iterations starts at a high value, and decreases rapidly, and after a few iterations remains at a low value.
- Number of pixels whose class labels change with iteration can fluctuate, but with sufficient number of iterations, will drop towards zero.
- Usually about 10 iterations will be adequate for images without too much texture, but for images having fine details, final classification takes much longer

Convergence and over-iteration



Edge Relaxation

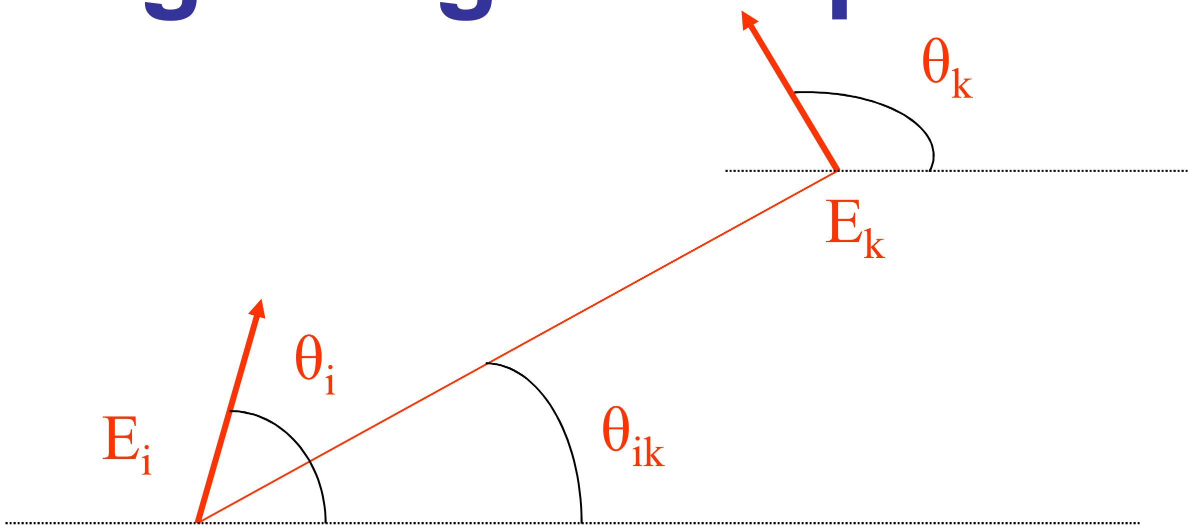
- Use an edge detector that generates an edge magnitude and edge orientation at each pixel
- Initial probabilities are computed from edge magnitude, updation employs both magnitude and orientation

- $p_i(\text{edge}) = \text{edgemag}_i \,/\, \max_j(\text{edgemag}_j)$
- $p_i(\text{noedge}) = 1.0 - p_i(\text{edge})$

Edge Relaxation

- For each point (x_i,y_i) on a Gradient image compute its probability P_i to belong to an edge
  - if point (x_j,y_j) is very close to point (x_i,y_i) and P_j is large
    - then the two events (P_i and P_j belong to the same edge) are compatible → increase P_i
  - if point (x_j,y_j) is very close to point (x_i,y_i) and P_j is low
    - then the two events are incompatible → decrease P_i

Edge-Edge Compatibilities

- 

- *Compatibility* is defined only for the nearest neighbors
- $C\ (.,.,.,.) = \cos(\theta_i - \theta_{ik})\cos(\theta_k - \theta_{ik})$
  - if $E_i, E_k \parallel E_{ik} \rightarrow C(.,.,.,.) = 1$
  - if $E_i \perp E_{ik}$ or $E_k \perp E_{ik} \rightarrow C(.,.,.,.) = 0$

Upgrade Rule

- The same upgrade rule used before can be used for edge relaxation.
- Convergence condition leads to all pixels unambiguously classified as edge or non-edge pixels
- This process can fill small gaps between edge segments and eliminate small noisy edge segments due to incompatible neighbor supports

Issues

- **How to evaluate neighbor supports?**
  - If support from a neighbor is equal for two or more classes, should it be discarded / downscaled / accepted as it is?
  - Is it possible to rank the supports by entropy or any other index?
- **Convergence issues - when the iterations should stop**
- **How to speed up the computations?**

Contd…