Automatic Image Annotation using 2D MHMM

Anuvab Chhotray
(111CS0030)

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

Automatic Image Annotation using 2D MHMM

by

Anuvab Chhotray

National Institute of Technology, Rourkela

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING
UNDER THE SUPERVISION OF
Dr. Pankaj K. Sa

May 2015

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Certificate

This is to certify that this is a bonafide record of the project presented by ***Anuvab Chhotray,111CS0030*** in partial fulfillment of the requirements of the degree of Bachelor of Technology in Computer Science and Engineering.

Date:

Dr. Pankaj K. Sa
(Project Guide)

Acknowledgement

No undertaking is possible by a sole individual. Many individuals have helped throughout for the finishing of this task and each of their inputs has been valuable. My deepest and sincerest gratitude goes to my supervisor, Pankaj Kumar Sa, Assistant Professor, Department of Computer Science & Engineering, for his guidance and encouragement throughout the period of the undertaking.

I am also thankful to NIT Rourkela for the academic resources it has supplied me through the course of the project. I want to thank the staff from the office who have been sufficiently benevolent to extend their help towards the completion of this project.

I would also like to take this opportunity to express my sincere gratitude to everyone who has provided me with inspirational words, ideas, constructive criticism, and above all, their invaluable time.

Declaration

I, Anuvab Chhotray, declare that this thesis is a presentation of my original work. Contributions of others involved, have been duly indicated wherever necessary, with due reference to the literature, and acknowledgment of collaborative research and discussions.

The work was carried out under the able guidance of Pankaj Kumar Sa, at National Institute of Technology, Rourkela.

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

Contents

List of Figures

List of Tables

1 Abstract

Automated Image Annotation is an important and challenging task in the field of computer vision and CBIR(Content-Based Image Retrieval). It has extensive use in research as well as personal fields. In this project, the same has been achieved with the help of a statistical method, namely a 2-dimensional multi-resolution hidden Markov model. Prior to classifying images by the system, it is trained using a set of images which are previously annotated using labels. Then the image to be annotated is compared against each trained model produced as a result of the previous step. This produces a parameter called likelihood. The label having the highest likelihood is assigned to the image.

Keywords: Hidden Markov Model, Content-Based Image Retrieval

2 Introduction

Classifying images is a major task in any scientific and research field. But with current photographing techniques being made cheap by the technological advancements in the field has led to the problem of sorting and classifying images even for personal purposes. An avid photographer may wish to group her images by the content in them. So the problem is no more relevant to a limited scope of biomedicine, military, commerce, digital libraries etc. like it used to be. Another major example of this system is the Image-based CAPTCHA generation system [1]. The system implemented here allows a computer to annotate images automatically with high accuracy after it has learned the concepts from several images already supplied.

The method proposed here allows a computer to annotate images automatically with high accuracy after it has learned the concepts from several images already supplied. A statistical approach is followed here, namely the 2-dimensional hidden Markov model.

3 Related Work

There are many CBIR systems in existence since the 1990s. CBIR systems perform the task of providing results in the form of images only visually similar to the query image. They are incapable of assigning 'labels' to them, i.e. linguistic indexing. There also exist multiple examples of work done in this field of image indexing and retrieval using machine learning techniques. [2, 3]. Minka and Picard, in 1997 developed a system which generated multiple collections of each images' regions based on a varied combination of their features. Then the system tried to learn the combinations which best represented the user supplied examples. The system was a supervised one, requiring the user to select the areas of interest in the image.

Bobulski, J., Adrjanowicz, L. [4] also worked on a 2-dimensional hidden Markov model for pattern recognition. Barnard et al. [5] and Duygulu et al. [6] at the University of California at Berkeley have been involved in more recent work in this field. Barnard and Forsyth were focused on annotating an entire image and Duygulu et al. were interested in labeling specific regions. Meanwhile, more recently, Oriol V. et al. [7], researchers at Google have achieved some success in annotating images with entire sentences.

4 Proposed Work

In the method proposed here, a predefined fixed number of classes of images each of which correspond to a keyword/concept are modeled by the two-dimensional multi-resolution hidden Markov model. The information generated from one image is denoted by a two dimensional matrix of features extracted from each image at varied resolutions and this information is arranged hierarchically on a logical pyramid. The MHMM model belonging to each image label performs the task of extraction of prototypical information pertinent to that label. As there are separate 2D MHMMs for all categories, an entirely new label with corresponding set of training images added to the existing list can have their representative information added to the database by computing using the algorithm, meaning that the procedure can be theoretically extended to any number of labels. Since all categories are annotated manually, a statistical mapping between the models and the set of labels can be generated. The feature vectors on the grid are computed for an input image. The likelihood for each model is then calculated. Those categories producing the highest likelihoods are selected as words to be annotated to the image.

Hidden Markov Model(HMM)s are applied extensively in data classification. Because of its effectiveness, it is used in fields like texture analysis, character recognition, face recognition etc. Hidden Markov Models, technically, are not a particular class of supervised or unsupervised learning algorithms. HMM has three problems which are used to solve the scenarios of Machine Learning. Considering machine learning algorithms like *support vector machines* or *k-means* belong to the classification and clustering paradigms. But HMM's *three* different problems deal

with theses paradigms. The use of a 2-dimensional HMM in this context is justified in the following text.

In the context of probability theory, Markov Model is a stochastic model used to model randomly changing systems where the most basic assumption is that the future state depends only on the present state and not on the sequence of events that preceded it. Markov models may be classified into four types depending on whether the sequential states are observable and whether the observations made control the system. The simplest form of a Markov Model is the Markov chain. It models a system's state across time with a stochastic variable.

Hidden Markov Model is a special case of a Markov chain for which the observations are related to the state of the system, but they are insufficient to determine the state precisely. 2D HMM is an extension of a 1D HMM to work on two dimensional data. It may be thought of as a combination of a transition matrix(i.e. the probabilities of transitions between states) and an observation matrix(some observed set of data, in our case the feature vectors), where the transition between different states is dependent on a 2D Markovian probability and each state at the same matrix position generates an independent observation.

There are three fundamental problems that can be solved using HMM. Two of them are involved in the current system.

(i) The second step in the proposed approach entails the **Learning Problem** which is stated as:
Given some observation sequences $O = O_1 O_2 \dots O_K$ and the general structure of HMM defined by count of hidden and visible states, decide HMM parameters $\lambda = (TR, EM, \pi)$ that

best fit observation sequence.

(ii) The third step is the **Evaluation Problem** which is stated as: Given the HMM $\lambda = (TR, EM, \pi)$ and the observation sequence $O = O_1 O_2 \ldots O_K$, estimate the probability that the model λ has generated the sequence O.

Approach

The system is comprised of three components, namely

(i) Feature Extraction,

(ii) Statistical Modeling *Fig. 1* and

(iii) Linguistic Indexing process *Fig. 2*.

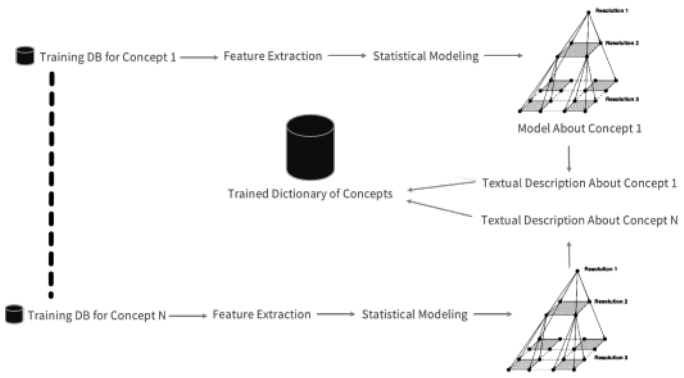

Figure 1: Creating the trained dictionary of concepts

4.1 Feature Extraction

The images are processed in blocks of 4×4 for feature extraction. This is a trade-off between computational complexity and texture detail. A total of six features from these 4×4 blocks are calculated. Three of them are the average of the RGB pixel values of the block.

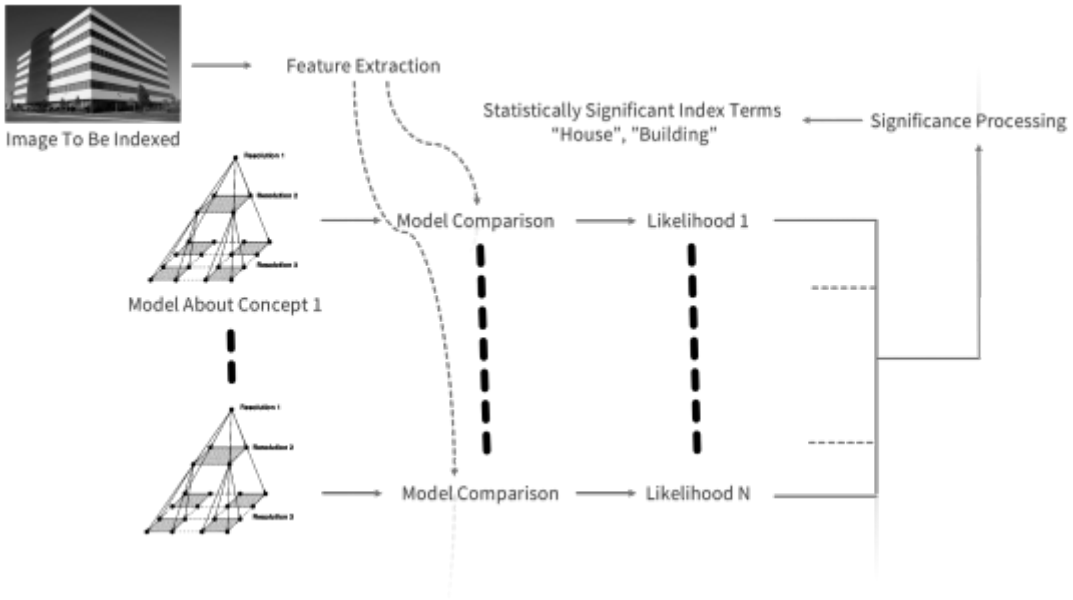

Figure 2: Automatic Linguistic Indexing

The other three are the square roots of summation of squares of the wavelet co-efficents in the high-frequency HL, LH and HH bands. Merely color features can't classify images because in certain cases like *mountains* covered with snow, the white color may be more and similar will be the case for a white *bus*. Hence texture features are necessary. LUV color space is chosen to extract the features. Images captured by cameras or CG rendering programs have a better perception and processing capability by computers in the LUV color space.

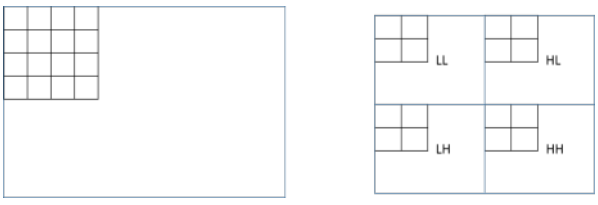

Figure 3: Applying Haar Wavelet Transform to image blocks

The block's L(Luminance) component was subjected to Haar transform to extract the remaining three texture features. Upon applying Haar transform to a 4×4 block, it is decomposed into

4 frequency bands with 2×2 coefficients as in *Fig. 3.* Assuming HL band coefficients to be $c_{x,y}, c_{x,y+1}, c_{x+1,y}, c_{x+1,y+1}$, a feature is computed as

$$f = \frac{1}{2}\sqrt{\sum_{i=0}^{1}\sum_{j=0}^{1} c_{x+i,y+j}^2}$$

This is done for HL, LH and HH bands whose wavelet coefficients imply variations in the horizontal, vertical and diagonal directions respectively.

4.2 Statistical Modeling

A feature vector is composed of features extracted from a block at a particular resolution. This in the 2D MHMM is treated as multivariate data. At each successive lower resolution, the rows' and columns' block count halve. Thus, a larger region of the image is covered by a lower resolution image's block. Thus, at a coarser resolution, a block is denoted as the parent block. At the same location, four blocks at a higher resolution are denoted as child blocks. This four-way split is what makes up the pyramid grid. Feature vectors, generated as suggested in the previous section may change state while transitioning from one block to another.

The parameters of the HMM for a label, defined as $\lambda = (TR, EM, \pi)$ are estimated using the Baum-Welch or Expectation-Maximization algorithm whose complexity is dependent on the number of states at different resolutions. Here, the number of resolutions is 3 and the number of states for the HMM are taken to be 7. There are a total of M states and the state of the block (x, y) is denoted by $u_{x,y}$. If $P(.)$ represents the probability of an event, then the assumption is as follows:

$$P(s_{x,y}|s_{x',y'}, u_{x',y'} : (x',y') < (x,y)) = a_{p,q,r}$$

where $(x',y') < (x,y)$ is true if $x' < x$ or $x' < x, y' < y$ and $p = s_{x-1,y}, q = s_{x,y-1}, r = s_{x,y}$.

This means that a block in the image is dependent on its previous blocks, i.e. its top and its left. This is represented in *Fig. 4*

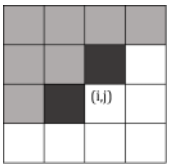

Figure 4: Markovian property of transition among states

Also, the feature vectors follow a Gaussian distribution and hence are conditionally independent of other blocks once the state of a block becomes known. Every state s has its unique covariance matrix and mean vector for the distribution. The fact that the only observable parameter in a given image are the feature vectors, is obvious that the system is named **Hidden** Markov Model. The set of resolutions for the model is denoted by $J = \{1 \dots j\}$ and $j = J$ being the original resolution. Let l and b represent the width and height of the image. Let the blocks at an intermediate resolution j be

$$V^{(j)} = \{(x,y) : 0 <= x < b/2^{J-j}, 0 <= y <= l/2^{J-j}\}$$

The number of states for an HMM is dependent on the problem and is not a fixed number.7 here is calculated using the likelihood parameter. During training or modeling the HMM, the likelihood versus the number of states is plotted. The number of states which maximizes the likelihood for all all models is selected.

To ease the computational complexity, the Viterbi training algorithm is used.

In summary, let the notations be as follows:

(1) entire image's feature vector set: $\mathbf{u} = \{u_{x,y} : (x,y)\epsilon V\}$

(2) image's set of states: $\mathbf{s} = \{s_{x,y} : (x,y)\epsilon V\}$

(3) image's set of classes: $\mathbf{c} = \{c_{x,y} : (x,y)\epsilon V\}$

(4) $C(s_{x,y})$ is the mapping from state $s_{x,y}$ to its class is $c_{x,y}$ and

(5) at iteration g, model estimated is $\phi^{(g)}$

The model is trained using the EM algorithm in the following steps as in [8, 9]:

(i) having known the model estimate $\phi^{(g)}$ at current iteration, the updation of mean vectors and covariance matrices for the model takes place as follows

$$\mu_m^{(g+1)} = \frac{\sum_{x,y} L_m^{(g)}(x,y) u_{x,y}}{\sum_{x,y} L_m^{(g)}(x,y)}$$

$$\sum_m^{(g+1)} = \frac{\sum_{x,y} L_m^{(g)}(x,y)(u_{x,y} - \mu_m^{(g+1)})(u_{x,y} - \mu_m^{(g+1)})'}{\sum_{x,y} L_m^{(g)}(x,y)}$$

where $L_m^{(g)}(x,y) = P(s_{x,y} = m | u_{x',y'}, c_{x',y'}, (x',y')\epsilon V; \phi^{(g)})$ is the *a posteriori* probability of block (x,y) of being in state m.

(ii) The transition probabilities are updated as

$$a_{p,q,r}^{(g+1)} = \frac{\sum_{x,y} H_{p,q,r}^{(g)}(x,y)}{\sum_{r=1}^{M} \sum_{x,y} H_{p,q,r}^{(g)}(x,y)}$$

where $H_{p,q,r}^{(g)}(x,y)$ is the *a posteriori* probability of block (x,y) being in state $r, (x-1,y)$ in state p and $(x,y-1)$ in state q.

4.3 The Indexing Process

For any input image, the system does a statistical comparison of the image with the trained models and the most significant label is assigned to the image. The similarity measure between the image and the models is the log likelihood and is calculated using the Forward Algorithm of the Hidden Markov Model. The procedure is as follows :
Let the forward variable be $\alpha_t(i, j, k)$ defined as

$$\alpha_t(i, j, k) = P(o_1, o_2, \ldots, o_t, q_t = ij|\lambda)$$

1. Initialization:
$$\alpha_1(i, j, k) = \pi_{ijk} b_{ij}(o_1)$$

2. Induction:
$$\alpha_{t+1}(i, j, k) = [\sum_{l=1}^{N} \alpha_t(i, j, k) a_{ijl}] b_{ij}(o_{t+1})$$

3. Termination
$$P(O|\lambda) = \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_T(i, j, k)$$

After this, the log likelihoods were sorted and e highest log likelihoods were obtained. Since the value of e is not fixed because the range of likeilhoods produced depends on the category the image belongs to, the label generating the highest log likelihood was only assigned to the image.

This also indicates the fact that any image will be compulsorily assigned to one of the classes. If e were to be fixed, it could have been possible to make a choice on the basis of the values of the log likelihoods.

5 Simulation Results

5.1 Training

To show the working of the system and demonstrating its efficiency, the proposed system was implemented in Matlab. The test images were obtained from *http://wang.ist.psu.edu/docs/related*. There were a total of 10000 test images in 10 classes, 1000 each and of sizes 384×256 or 256×384. The images were stored in subdirectories $s1 \ldots s10$ for the 10 classes. The system was then trained and the database of concepts was created, database here meaning the set of TR, EM and π for each class.

The following were the different classes taken into consideration:

ID	**Category**
1	Person
2	Beach
3	Building
4	Bus
5	Dinosaur
6	Elephant
7	Flower
8	Horse
9	Mountain
10	Food

Table 1: Different Categories taken for training and testing

Fig. 5 is a plot of log likelihood versus number of iterations for one of the models. It was observed that the more diverse or less alike the images in the training set were, the longer it took for the log likelihood to converge i.e., it took a larger number of iterations.

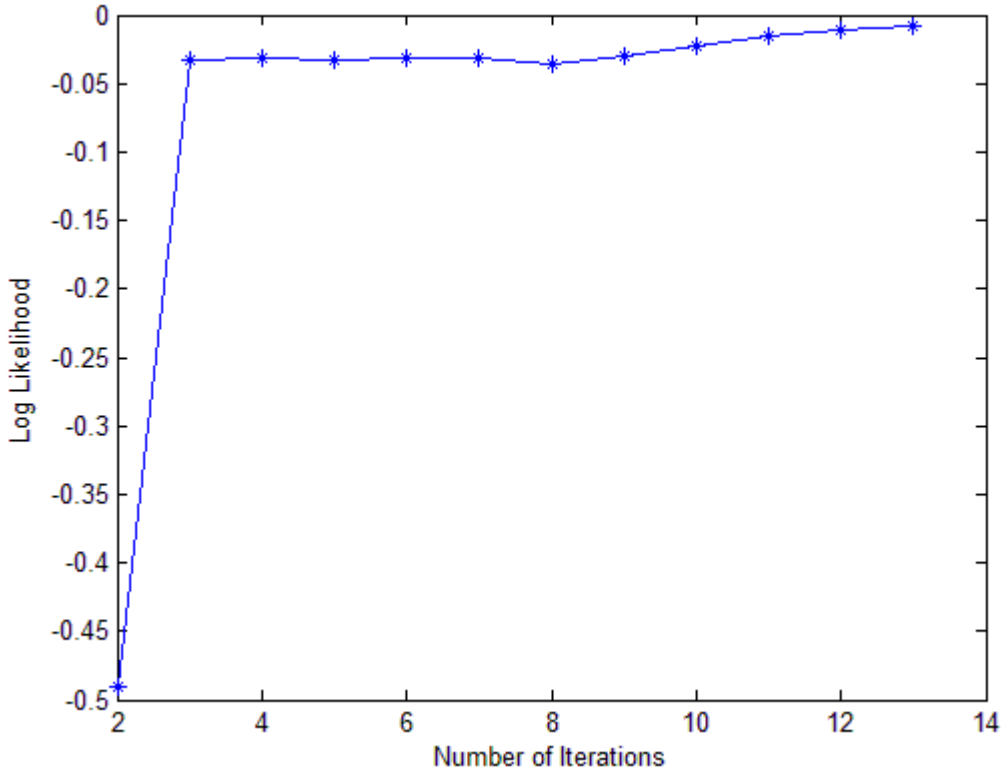

Figure 5: Log Likelihood versus Number of Iterations

5.2 Testing

Then the system was fed inputs for testing with 100 images outside the set of images which were used for training. The table displays the result of the classification. The numbers across the rows denote the number of images in a category classified into each of the ten categories by the system. Figures on the diagonals denote how many images were classified correctly for each category.

The platform was Windows 8.1 64-bit with MATLAB 2014 64-bit on Intel Core i7-2630 with 4GB of RAM. The training process took 14 minutes for the setup and testing took 4 minutes.

Initial testing produced an accuracy of 62.1%. Also, separately

	Person	Beach	Building	Bus	Dinosaur	Elephant	Flower	Horse	Mountain	Food
Person	54	20	5	0	4	1	5	6	4	1
Beach	7	73	2	1	8	1	4	3	0	1
Building	2	5	49	7	2	0	10	4	15	6
Bus	5	16	17	25	9	0	5	9	7	7
Dinosaur	4	13	0	0	81	0	1	1	0	0
Elephant	0	0	0	0	0	100	0	0	0	0
Flower	1	0	12	0	0	0	77	0	3	7
Horse	2	12	1	0	0	0	0	74	11	0
Mountain	2	8	23	0	0	0	13	9	43	2
Food	5	1	15	3	6	0	12	2	11	45

Figure 6: Classification Accuracy

other images were tested and some of the annotations are shown in *Fig. 7.*

Figure 7: Labels assigned by the system

6 Conclusion

6.1 Advantages

A statistical approach to automatic image classification was presented. The advantages of the presented approach are:

(i) models belonging to different labels/categories can be trained independently.

(ii) and because of (i), the number of categories that can be trained and stored is large.

It should also be noted that this entire system is parallelizable and hence can be made more efficient while training or testing.

6.2 Drawbacks

Following are some of the drawbacks of the system:

(1) A notable drawback of this system would be that if images which belong to a certain class label or context but contain images drastically different from most of the images in the set, this is likely to hamper the accuracy of the system. So for example, say *buses* and *trucks* belong to the same general category of *vehicles* but having a mixed set of images of *buses* and *trucks* can lead to multiple false positives thereby decreasing the classification accuracy. Hence, they should be trained differently and a manual super-class annotation may be used to classify them singularly into *vehicles*.

(2) Another drawback of this system is that it only works on color images. So monochromatic images can't be classified using this system. So, scans of old photographs or those taken with a

black and white camera like surveillance systems can not be processed in this system.

(3) Only a single label is assigned to an image. A better method may be proposed which makes it possible to assign multiple labels to a single image.

(4) Similarity between different images causes the system to classify the images incorrectly. For instance buildings with windows and buses may be classified into the same category, which is wrong. This can be avoided by making the training set larger.

References

[1] J.Z. Wang, R. Datta, and J. Li. Image-based captcha generation system, April 19 2011. US Patent 7,929,805.

[2] T.P. Minka and R.W. Picard. Interactive Learning Using a Society of Models. *Pattern Recognition*, 30(3):565 – 563, 1997.

[3] J.Z. Wang and M.A. Fischler. Visual Similarity, Judgmental Certainty and Stereo Correspondence. *Proc. DARPA Image Understanding Workshop, G. Lukes*, 2:1237 – 1248, November 1998.

[4] Adrjanowicz L. Bobulski, J. Two-dimensional hidden markov models for pattern recognition. *7894 Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, pages 515 – 523, 2013.

[5] M. Piccardi. Learning the semantics of words and pictures. In *International Conference Computer Vision*, volume 2, pages 126 – 131, 2001.

[6] N. de Freitas P. Duygulu, K. Barnard and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conference Computer Vision*, volume 4, pages 97 – 112, 2002.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014.

[8] Li Yujian. An analytic solution for estimating two-dimensional hidden markov models. *Applied Mathematics and Computation*, 185(2):810 – 822, 2007. Special Issue on Intelligent Computing Theory and Methodology.

[9] Jia Li, R.M. Gray, and R.A. Olshen. Multiresolution image classification by hierarchical modeling with two-dimensional hidden markov models. *Information Theory, IEEE Transactions on*, 46(5):1826 – 1841, Aug 2000.