Semantic Feature Extraction with Multidimensional Hidden Markov Model

Joakim Jiten, Bernard Merialdo, Benoit Huet
Multimedia Communications Department, Institut EURECOM, BP 193, 06904 Sophia-Antipolis, FRANCE
{jiten, merialdo, huet}@eurecom.fr

ABSTRACT

Conventional block-based classification is based on the labeling of individual blocks of an image, disregarding any adjacency information. When analyzing a small region of an image, it is sometimes difficult even for a person to tell what the image is about. Hence, the drawback of context-free use of visual features is recognized up front.
This paper studies a context-dependant classifier based on a two dimensional Hidden Markov Model. In particular we explore how the balance between structural information and content description affect the precision in a semantic feature extraction scenario. We train a set of semantic classes using the development video archive annotated by the TRECVid 2005 participants. To extract semantic features the classes with maximum a posteriori probability are searched jointly for all blocks. Preliminary results indicate that the performance of the system can be increased by varying the block size.

Keywords: Block-based, Image Classification, Hidden Markov Model, 2D HMM.

1. INTRODUCTION

Hidden Markov Models (HMM) have become increasingly popular for learning purposes in such diverse applications as speech recognition [1], language modeling, language analysis, and image recognition [3]. The reason for this is that they have a rich mathematical structure and therefore can form theoretical basis for many domains. A second reason is the discovery of the Baum-Welch's forward-backward algorithm [2] which allows estimating the numeric values of the model parameters from training data.
Most of the current applications involve uni-dimensional data. In theory, HMMs can be applied as well to multi-dimensional data. However, the complexity of the algorithms grows exponentially in higher dimensions, even in dimension 2, so that the usage of plain HMM becomes prohibitive in practice [4]. For this reason we use a new type of multi-dimensional Hidden Markov Model: the Dependency-Tree Hidden Markov Model [5] (DTHMM).
We explored the intrinsic ability of the DTHMM to model contextual information by running some experiments and comparing them with old TRECVid results. To study the balance of structural information and content description we parameterized with the block size.
The remainder of this paper is organized as follows: section 3 and 4 outlines our motivation and presents the DTHMM. We show how the most common algorithms for solving the necessary problems associated with HMMs can be adapted to the model and that the algorithms keep the same linear complexity as in one dimension. Section 5 will describe the experimental setup conducted on TRECVid 2005 data. Finally in section 5 we give the conclusion and suggest future work.

2. RELATED WORK

A growing trend in the field of image retrieval is region-based approaches. The Stanford SIMPLIcity system [17] uses a scalable method for indexing and retrieving images based on region segmentation. A statistical classification is done to group images into rough categories, which potentially enhances retrieval by permitting semantically adaptive search methods and by narrowing down the searching range in a database.

Motivated by the desire to incorporate contextual information, Li and Gray [3] proposed a 2D HMM for image classification based on a block-based classification algorithm using a path constrained Viterbi.
A system developed by Minka and Picard [16] included a learning component. Assuming that there is no single model that can capture everything what humans perceive in images, their system used a "society of models". The system internally generates several groupings of each image's regions based on different combinations of features, then learns which combinations best represents the semantic categories given as examples by the user. The system requires the supervised training of various parts of the image.
Recent work in associating semantics with image features was done by Barnard and Forsyth at University of California at Berkeley [19]. Using region segmentation in a pre-processing step to produce a lower number of color categories, image feature search becomes a text search. The data is modeled as being generated by a fixed hierarchy of nodes organized as a tree. The work has achieved some success for certain categories of images. But, as pointed out by the authors, one serious difficulty is that the algorithm relies on semantically meaningful segmentation which is, in general, not available to image databases. Automatic segmentation is still an unsolved problem in computer vision [12].
In previous work Agazzi et al.[4] [9] extended the 1-D HMM to a pseudo2-D HMM. The model is called "pseudo 2 D" because it is not a fully connected 2-D HMM. The assumption is that there exists a set of "superstates" that are Markovian which subsume a set of simple Markovian states. For images the superstate is decided depending on the transition probability based on the previous superstate. The superstates determine the simple Markov chain to be used by the entire row. A simple Markov chain is then used to generate observations in the row. Inherently from its structure this model is expected to perform better with structured images like documents.

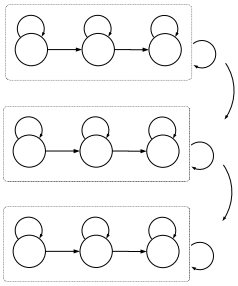

Figure 1. A Pseudo 2-D HMM

Another approach is to consider independent horizontal and vertical 1-D HMMs. For the problem of OCR Hallouli et al. [10,11] explored two different fusion schemes: decision fusion and data fusion. In the decision fusion scheme, the classifiers are assumed independent which enables to derive an approximation of the joint likelihood. In the data fusion scheme line and column features occurring at the same spatial index are considered correlated.
In other work to reduce the complexity of the HMM reestimation procedures, assumptions that simplify the original formulas have been proposed:

- select a subset of state configurations only [3],
- ignore correlation of distant states [11],
- approximate probabilities by turbo-decoding [12].

The main disadvantage of these approaches is that they only provide approximate computations, so that the probabilistic model is no longer theoretically sound. In this experiment we use the DTHMM model which leads to efficient algorithms for 2D-observations.

3. DTHMM: Dependency-Tree HMM

3.1. Spatial Coherence

For most images with reasonable resolution, pixels have spatial dependencies which should be enforced during the classification.

For the sake of computational simplicity, the identical independent distribution (I.I.D.) assumption is commonly used. However the I.I.D assumption doesn't consider the spatial contextual constraints among pixels. In most images of reasonable resolution, if all the neighbors to a pixel (or block of pixels) belong to a class 'A', it is not very likely that this pixel belongs to a completely different class 'B' since generally images have a spatial coherence.
The HMM considers observations (e.g. feature vectors representing blocks of pixels) statistically dependent on neighboring observations through transition probabilities organized in a Markov mesh, giving a dependency in two dimensions. The state process defined by this Mesh is a special case of the Markov Random Field described beneath.

3.2. Markov Random Field

The 2-D counterpart of the 1-D Markov chain is called a Markov random field (MRF) where the ordering of past, present and future in the 1-D model is replaced by spatial neighborhood.

In order to consider the neighborhood information, one approach is to estimate the joint posterior probability for the whole image's labeling configuration. With an image of 320x240 pixels there would be $M^{320x240}$ configurations to compute, where M is the number of states to be considered. In order to simplify the computation the Markov assumption is considered, which states that the label of a certain pixel (block or observation) is independent to other pixel's labels given its direct neighbors. For example on a given image, a first order neighborhood system is defined as shown in Figure 2.

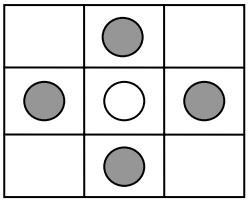

Figure 2. First order neighborhood

The Markov property forms the basis of the Markov Random Field modeling. Even though the Markov assumption can simplify significantly the estimation of the probability of the states the complexity of the problem still becomes exponential as mentioned earlier, namely $(w \times h - 1)^N$. Where w or h is the number of blocks in one row or column and N is the number of states. In our case the complexity becomes $(22 \times 16 - 1)^9 \approx 80 \times 10^{21}$, which is prohibitive for a straight forward calculation.

3.3. Dependency-Tree HMM

In this section, we briefly recall the basics of 2D-HMM and describe our proposed DT-HMM [5].

3.4. 2D-HMM

The reader is expected to be familiar with 1D-HMM. We denote by $O=\{o_{ij}, i=1,\ldots m, j=1,\ldots,n\}$ the observation, for example each o_{ij} may be the feature vector of a block (i,j) in the image. We denote by $Q = \{q_{ij}, i=1,\ldots m, j=1,\ldots,n\}$ the state assignment of the HMM, where the HMM is assumed to be in state q_{ij} at position (i,j) and produce the observation vector o_{ij}. If we denote by λ the parameters of the HMM, then, under the Markov assumptions, the joint likelihood of O and Q given λ can be computed as:

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda)$$
$$= \prod_{ij} p(o_{ij}|q_{ij},\lambda)p(q_{ij}|q_{i-1,j},q_{i,j-1},\lambda)$$

If the set of states of the HMM is $\{s_1, \ldots s_N\}$, then the parameters λ are:

- the output probability distributions $p(o \mid s_i)$
- the transition probability distributions $p(s_i \mid s_j,s_k)$.

Depending on the type of output (discrete or continuous) the output probability distribution are discrete or continuous (typically a mixture of Gaussian distribution).

3.5. DT-HMM

The problem with 2D-HMM is the double dependency of $q_{i,j}$ on its two neighbors, $q_{i-1,j}$ and $q_{i,j-1}$., which does not allow the factorization of computation as in 1D, and makes the computations practically intractable.

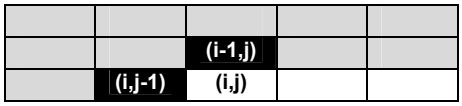

Figure 3. 2D Neighbors

Our idea is to assume that $q_{i,j}$ depends on one neighbor at a time only. But this neighbor may be the horizontal or the vertical one, depending on a random variable t(i,j). More precisely, t(i,j) is a random variable with two possible values :

$$t(i,j) = \begin{cases} (i-1,j) & \text{with prob } 0.5 \\ (i,j-1) & \text{with prob } 0.5 \end{cases}$$

For the position on the first row or the first column, t(i,j) has only one value, the one which leads to a valid position inside the domain. t(0,0) is not defined.

So, our model assumes the following simplification:

$$p(q_{i,j} | q_{i-1,j}, q_{i,j-1}, t) = \begin{cases} p_V(q_{i,j} | q_{i-1,j}) & \text{if } t(i,j) = (i-1,j) \\ p_H(q_{i,j} | q_{i,j-1}) & \text{if } t(i,j) = (i,j-1) \end{cases}$$

If we further define a "direction" function:

$$D(t) = \begin{cases} V & \text{if } t = (i-1,j) \\ H & \text{if } t = (i,j-1) \end{cases}$$

then we have the simpler formulation:

$$p(q_{i,j} | q_{i-1,j}, q_{i,j-1}, t) = p_{D(t(i,j))}(q_{i,j} | q_{t(i,j)})$$

Note that the vector **t** of the values t(i,j) for all (i,j) defines a tree structure over all positions, with (0,0) as the root. Figure 4 shows an example of random Dependency Tree.

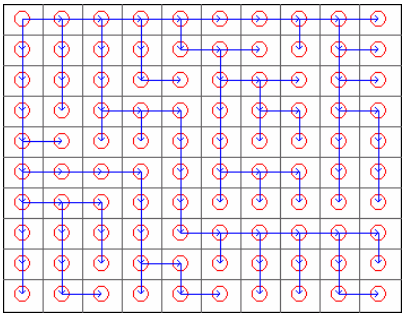

Figure 4. Example of Random Dependency Tree

The DT-HMM replaces the N^3 transition probabilities of the complete 2D-HMM by $2N^2$ transition probabilities. Therefore it is efficient in terms of storage. We will see that it is also efficient in terms of computation. Position (0,0) has no ancestor. In this paper, we assume for simplicity that the model starts with a predefined initial state s_I in position (0,0). It is straightforward to extend the algorithms to the case where the model starts with an initial probability distribution over all states.

4. SEMANTIC CLASSIFICATION

In this section, we discuss the theory on how to use the DT-HMM for the detection of images with a given semantic category, which is put in practiced in section 5.

4.1. Inside probabilities

In a manner which is very similar to the Viterbi alignment, we can easily compute the probability that the portion of the image covered by a subtree T(i,j) with root (i,j) is produced from state s in position (i,j). If we denote this probability by $\beta_{i,j}(s)$, then these values can be calculated recursively, in reverse order (starting from the last position), according to the relations:

- if (i,j) is a leaf in T(i,j):

$$\beta_{i,j}(s) = p(o_{i,j}|s)$$

- if (i,j) has only an horizontal successor:

$$\beta_{i,j}(s) = p(o_{i,j}|s)\sum_{s'} p_H(s'|s)\beta_{i,j+1}(s')$$

- if (i,j) has only a vertical successor:

$$\beta_{i,j}(s) = p(o_{i,j}|s)\sum_{s'} p_V(s'|s)\beta_{i+1,j}(s')$$

- if (i,j) has both an horizontal and a vertical successors:

$$\beta_{i,j}(s) = p(o_{i,j}|s)\left(\sum_{s'} p_H(s'|s)\beta_{i,j+1}(s')\right)\left(\sum_{s'} p_V(s'|s)\beta_{i+1,j}(s')\right)$$

Note that the formulas are similar to those appearing in the Viterbi algorithm, with maxima replaced by summations. The probability that the complete image is produced by the model is then: $P(O|t) = \beta_{0,0}(s_I)$

4.2. The Outside Probabilities

To prepare for the Maximum Likelihood reestimation algorithm, we need to define the Outside Probabilities.
Let us denote by O(i,j) the portion of the image which is not covered by the sub trees starting at the successors of (i,j). For example, if (i,j) has two successors, O(i,j) is the portion of the image outside the sub trees T(i+1,j) and T(i,j+1). { O(i,j), T(i+1,j), T(i,j+1)} is a partition of the image, as shown in 5.

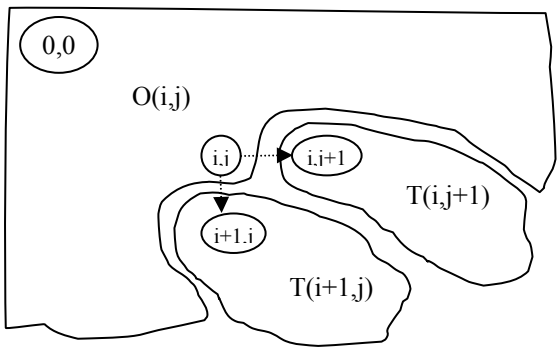

Figure 5. Schema of Outside O(i,j) and Inside T(i,j) areas

If we denote by $\alpha_{i,j}(s)$ the probability of starting from (0,0), generating the output vectors for all the positions in O(i,j), and reaching position (i,j) in state s, then the $\alpha_{i,j}(s)$ probabilities may be computed by the following recursion:

- $\alpha_{0,0}(s_I) = p(o_{0,0}|s_I)$, $\alpha_{0,0}(s) = 0$ for $s \neq s_I$,
- if (i,j+1) has an horizontal ancestor:

$\alpha_{i,j+1}(s) = p(o_{i,j+1}|s)\sum_{s'}\alpha_{i,j}(s')p_H(s|s')\sum_{s''}p_V(s''|s')\beta_{i+1,j}(s'')$ (in this case, O(i,j+1) is the union of O(i,j), (i,j) and T(i+1,j)),

- if (i+1,j) has a vertical ancestor:

$\alpha_{i+1,j}(s) = p(o_{i+1,j}|s)\sum_{s'}\alpha_{i,j}(s')p_V(s|s')\sum_{s''}p_V(s''|s')\beta_{i,j+1}(s'')$ (in this case, O(i+1,j) is the union of O(i,j), (i,j) and T(i,j+1)).

The cases where (i,j) has only one successor lead to simpler formulas:

- if (i,j+1) is the horizontal successor:

$$\alpha_{i,j+1}(s) = p(o_{i,j+1}|s)\sum_{s'}\alpha_{i,j}(s')p_H(s|s')$$

- if (i+1,j) is the vertical successor:

$$\alpha_{i+1,j}(s) = p(o_{i+1,j}|s)\sum_{s'}\alpha_{i,j}(s')p_V(s|s')$$

4.3. Maximum Likelihood training

It is now possible to perform a Maximum-Likelihood estimation of the DT HMM model, that is, with a fixed dependency tree, apply a variation of the Baum-Welch algorithm to recomputed new estimates of the probabilities that increase the likelihood of the training data. The following algorithm is inspired by the Inside-Outside algorithm [13,14,15], which is the application of the Baum-Welch algorithm to the parse trees of Probabilistic Context-Free Grammars.

In the Maximum Likelihood training, we need to estimate the number of times that a particular transition, or a particular emission, is used while generating the complete image. For example, we need to compute the probability that the Model is in state s at position (i,j). Using the inside and outside probabilities defined previously, the probability of generating the complete image is:

$$P(O|t) = \sum_s \alpha_{i,j}(s)\left(\sum_{s'}p_H(s'|s)\beta_{i,j+1}(s')\right)\left(\sum_{s''}p_V(s''|s)\beta_{i+1,j}(s'')\right)$$

For a simpler formulation we define:

$$\gamma^H_{i,j}(s) = \sum_{s'}p_H(s'|s)\beta_{i,j+1}(s')$$
$$\gamma^V_{i,j}(s) = \sum_{s''}p_V(s''|s)\beta_{i+1,j}(s'')$$

Then:

$$P(O|t) = \sum_s \alpha_{i,j}(s)\gamma^H_{i,j}(s)\gamma^V_{i,j}(s)$$

Note that this relation is valid for all positions (i,j). Note also that if we define $\gamma^H_{i,j}(s)$ (respectively $\gamma^V_{i,j}(s)$) to be equal to 1 when (i,j) has no horizontal (respectively vertical) successor, then the formula is valid whatever the number of successors of (i,j) is.

The probability for being in state s at position (i,j) while generating the complete image is therefore:

$$P(q_{i,j} = s|O,t) = \frac{1}{P(O|t)}\alpha_{i,j}(s)\gamma^H_{i,j}(s)\gamma^V_{i,j}(s)$$

The expected number of times that the system is in state s during the generation of the image is:

$$E(s) = \sum_{i,j}P(q_{i,j} = s|O,t) = \frac{1}{P(O|t)}\sum_{i,j}\alpha_{i,j}(s)\gamma^H_{i,j}(s)\gamma^V_{i,j}(s)$$

The probability for going from state s at position (i,j) to state s' in position (i,j+1) while generating the complete image is:

$$P(q_{i,j} = s, q_{i,j+1} = s' | O, t) = \frac{1}{P(O|t)} \alpha_{i,j}(s) p_H(s'|s) \beta_{i,j+1}(s') \gamma^V_{i,j}(s)$$

The expected number of times that the horizontal transition from s to s' takes place during the generation of the image is:

$$E(s \xrightarrow{H} s') = \sum_{i,j} P(q_{i,j} = s, q_{i,j+1} = s' | O, t)$$

$$= \frac{1}{P(O|t)} \sum_{i,j} \alpha_{i,j}(s) p_H(s'|s) \beta_{i,j+1}(s') \gamma^V_{i,j}(s)$$

This provides the basis for the reestimation of the horizontal transition probabilities:

$$p'_H(s'|s) = \frac{E(s \xrightarrow{H} s')}{\sum_{s''} E(s \xrightarrow{H} s'')}$$

The vertical transitions are handled in the same way. The re-estimation of the output probabilities is similar, as the probability of being in state s at (i,j) is also the probability of emitting the output vector $o_{i,j}$ from state s at position (i,j). When the output probability distribution is discrete, the re-estimate is obtained by counting:

$$p'(o|s) = \frac{E(s \to o)}{\sum_{o'} E(s \to o')}$$

When the distribution is continuous, the expected number of times that the emission is observed can be used to update the parameters of the distribution (in the case of a Mixture of Gaussian distributions, weights, means and variances).
In summary, the ML algorithm consists in iterating two steps over the whole set of training images:

- Compute the inside and the outside probabilities (for an image and dependency graph),
- Accumulate the expected number of occurrences for transitions and outputs,

When all images have been processed, the probabilities are re-estimated, and a new iteration on the set of images is performed. A stopping criterion is used to terminate the iterations

5. EXPERIMENT SETUP

We implemented an experimental system which processes the TRECVid database to extract signature features and semantic descriptions. The results were stored in an indexed database to speed up experiments. For the classifier we implemented both a discrete and a continuous DTHMM. The discrete model uses a vector quantization step to handle multivariate signature vectors, and the continuous model uses a GMM to describe the output probabilities. We use the EM algorithm to estimate the GMM and K-means to initialize the parameters. To evaluate and analyze the result we calculate a precision – recall curve and list the top 50 ranked frames. The presented results are based on the continuous model, whereas the discrete model was used for verification.

5.1. System Design

As in all block-based classification systems, an image is divided into non overlapping blocks, forming a regular grid. Feature vectors are then evaluated as statistics of the blocks, which can be regarded as a sequence of observations. The assumption in a 2-D HMM is that the observation sequence was produced by the model, i.e. $P(O | \lambda)$ where O is the observation sequence and λ the set of model parameters (see section 3.6). The number of states is a fixed parameter and is set to 16; each one with a Gaussian Mixture Model to represent the continuous observation densities, which has a fixed number of components:

$$b_j(o) = \sum_{m=1}^{M} c_{jm} \pi[o, \mu_{jm}, \Sigma_{jm}] \quad 1 \le j \le N$$

We use a variant of the Baum-Welch algorithms to estimate the model parameters in the training step. To classify an image its low-level features are extracted and then $P(O \mid \lambda)$ is computed for each model giving a score on how well the model matches the observation, and then search the model with highest a posteriori probability. A general illustration of the classification system is shown in Figure 6.

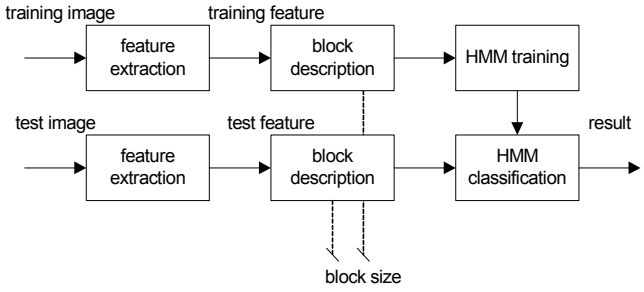

Figure 6. Image Classification Scheme

Since we are interested in exploring the balance between structural information and content description, we vary the size of the blocks. Using big blocks means that content description becomes more important while small blocks imply that structure becomes more prominent and content description simpler. To produce the observations for training we decompose the image into n_x by n_y non-overlapping blocks as shown in figure 7. This gives us a vector field describing the whole picture, $O = \{o_{ij}\}$ where o_{ij} is the feature vector extracted at position (i,j). We let the block size vary in the range of 176 x 120 to 2 x 2 pixels.

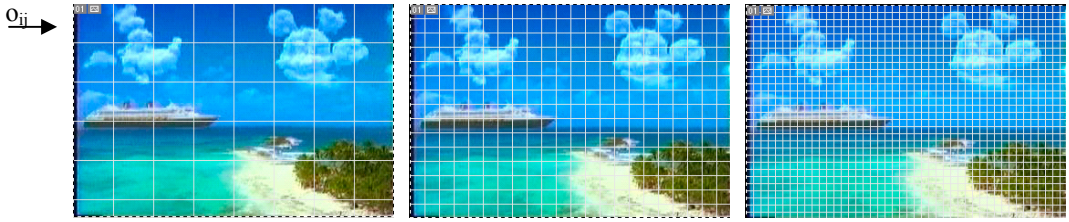

Figure 7. Example of images decomposed into blocks with different sizes; 44x40, 16x15 and 8x8 pixels

5.2. Extracted Low-Level Features

The visual features where extracted from the decompressed key-frames in the source videos of the TRECVid 2005 development archive. As aforementioned the image was split into blocks before extraction.

We designed and implemented feature extraction algorithms for color moments and DCT coefficients. In the light of the fact that we use Gaussian Mixture Models it was desirable to use features which are Gaussian distributed and are as much uncorrelated as possible. Further as it is well known that histogram output as features has highly skewed probability distributions, we decided to use HSV means and variances for color descriptors and DCT coefficients for their discriminative ability of energies in the frequency domain. Hence for each block, there are six color features and 16 DCT coefficients $\{ H_\mu, S_\mu, V_\mu, H_\sigma, S_\sigma, V_\sigma, D_{i,j} : i,j \in (0,1,2,3) \}$. Since our classifier is trained over a sequence of observations; the number of dimensions is the same of that of the block (22-dimensions). With this order of dimensionality we disregard the possibility of dimension reduction.

5.3. The Dataset

Algorithms were tested on data from TRECVid 2004 during development, and then the experiments were carried out on TRECVid 2005 development when it became available. The archive consists of 137 annotated video files corresponding to 63 GB. To manage the extensive amount of data and numerous files we developed a *Sample Parser*. Given a header file and a label, the Sample Parser creates an indexed subset of frames to speed up the training and test process.

Each video has a common shot boundary reference that was provided by the TRECVid organization. To create the shot reference, the videos was segmented and keyframes defined according to a scheme that guarantees that a shot is at least two seconds in length. We decompress all annotated keyframes within each shot, crop it to a standard size of 352x240 pixels, and then compute image signatures on features extracted from those representative images. The classifier was jointly trained on 100 frames annotated as "Waterscape_Waterfront". Some sample images are shown in Figure 8.

Figure 8. Images annotated "Waterscape_Waterfront" from the TRECVid 2005 archive

During training we measure the average probability of the model for different number of Gaussians per mixture (gpm) to investigate how that affects the performance of the model. The graph below shows the evolution of likelihood of the training data during the training of a model based on blocks with the size 88x60 pixels.

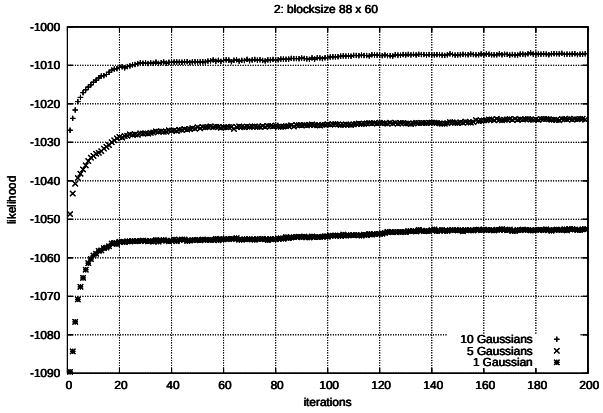

Figure 9. Likelihood of the training data for three different gpm's

We choose to use five Gaussians per mixture in our further experiments since they provide a good compromise between computational complexity and performance.

5.4. Result

In our initial experiments we have already noticed the problem of one known drawback of the HMMs, that the output probability plays a more important role than the transition probability. The output distribution ranges over greater dispersion than the transition probability which range over 16 states only, with a majority of transitions from a state to itself. This explains why an image which has an almost uniform color and gets a high output probability from one of the states is likely to get a very large emission probability.

In order to reduce this effect, we have introduced a modified algorithm: during the training phase, we compute the probability that a state is used in a given position p(s | (i,j)). We use this as prior probability providing some knowledge on the position of states in the image. This allows to describe, for example, that states representing sky colors are more likely to appear in the top area of the image. We then compute the inside probabilities with prior, which are given by the formula:

$$\beta'_{i,j}(s) = p(o_{i,j}|s)p(s|(i,j))\left(\sum_{s'} p_H(s'|s)\beta'_{i,j+1}(s')\right)\left(\sum_{s'} p_V(s'|s)\beta'_{i+1,j}(s')\right)$$

To explore how the block-size affects the precision we trained seven models; each one based on observations with a different partitioning, and then the models were tested against a common test set resulting in a ranked list. The graph below shows the average precision for seven different block sizes. We noticed that a block size of 16x15 pixels (model #4) gives the highest average precision.

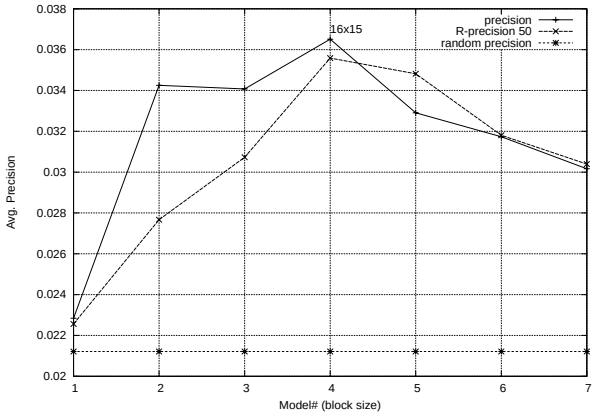

Figure 10. Avg. precision for block size 1: 176x120, 2: 88x60, 3: 44x40, 4: 16x15, 5: 8x8, 6: 4x4, 7: 2x2

6. Conclusion and Future Work

In an effort to improve semantic classification by examining local context in an image, we have studied a new multidimensional Hidden Markov Model – the Dependency-Tree HMM. We have shown that the most common algorithms for solving the necessary problems associated with HMMs, can be adapted to DT-HMM, which allows reestimation of the HMM parameters in the same linear complexity as in the one dimensional HMM case.

To investigate the performance of the model and to find its point of operation, we have studied the importance of number of Gaussians per mixture during training and the effect of varying the block size. The results indicate that during the training process a model with a larger number of Gaussians per mixture (gpm) needs more iteration to reach an asymptote which happens after about 30 iterations. A model trained with five gpm achieves the best performance with a block size of 16x15 pixels.

So far, only preliminary experiments have been carried-out, more time is needed to understand how to best exploit the model. We have already identified many interesting experiments to conduct using our novel approach. Some issues among those we ought to investigate are the use of restricted state models and the number of hidden states. Since in speech recognition several hundred states are often used, it seems obvious that an efficient model for images should also contain a larger number of states.

REFERENCES

[1] Rabiner, L.R., S.E. Levinson, and M.M. Sondhi, (1983). On the application of vector quantization and hidden Markov models to speaker independent, isolated word recognition. B.S.T.J.62,1075-1105

[2] LE. Baum and T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, Annual Math., Stat., 1966, Vol.37, pp. 1554-1563.

[3] J. Li, A. Najmi, and R. M. Gray, Image classification by a two-dimensional hidden markov model, IEEE Trans. Signal Processing, vol. 48, no. 2, pp. 517–533, 2000.

[4] Levin, E.; Pieraccini, R.; Dynamic planar warping for optical character recognition, IEEE International Conference on Acoustics, Speech, and Signal Processing, , Volume 3, 23-26 March 1992 Page(s):149 - 152

[5] Merialdo, B; Dependency Tree Hidden Markov Models, Research Report RR-05-128, Institut Eurecom, Jan 2005

[6] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. IEEE Trans. on Image Processing (IP), 11(4):467-476, Apr 2002.

[7] Smith J R, Integrated spatial and feature image system: Retrieval, analysis and compression [Ph D dissertation], Columbia University, New York 1997

[8] Zhang, L., Lin, F.Z., Zhang, B. "A CBIR method based on color-spatial feature". IEEE Region 10 Annual International Conference 1999:166-169

[9] O. Agazzi, S. Kuo, E. Levin, and R. Pieraccini. Connected and degraded text recognition using planar hidden Markov models. In Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), volume 5, pages 113-116, 1993.

[10] K. Hallouli, L. Likforman-Sulem, and M. Sigelle. A comparative study between decision fusion and data fusion in Markovian printed character recognition. In Proc. of the IEEE Int. Conf. on Pattern Recognition (ICPR), volume 3, pages 147-150, 2002.

[11] Merialdo, B.; Marchand-Maillet, S.; Huet, B.; Approximate Viterbi decoding for 2D-hidden Markov models, IEEE International Conference on , Acoustics, Speech, and Signal Processing, Volume 6, 5-9 June 2000 Page(s):2147 - 2150 vol.4

[12] Perronnin, F.; Dugelay, J.-L.; Rose, K.; Deformable face mapping for person identification, International Conference on Image Processing, Volume 1, 14-17 Sept. 2003 Page(s):I - 661-4

[13] J.K. Baker, Trainable grammars for speech recognition ,; In Jared J.Wolf and Dennis H. Klatt, editors, Speech communication papers presented at the 97th Meeting of the Acoustical Society of America, MIT, Cambridge, MA, June 1979.

[14] F. Jelinek, J. D. Lafferty, and R. L. Mercer; Basic methods of probabilistic context free grammars Technical Report RC 16374 (72684), IBM, Yorktown Heights, New York 10598. 1990.

[15] F. Jelinek, *Statistical Methods for Speech Recognition* Cambridge, MA: MIT Press, 1997.

[16] T. P. Minka and R.W. Picard, "Interactive learning using a 'society of models'," Submitted for Publication, 1995. Also appears as MIT Media Lab Perceptual Computing TR#349

[17] J.Z. Wang, Integrated Region-Based Image Retrieval. Dordrecht: Kluwer Academic, 2001.

[18] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[19] K. Barnard and D. Forsyth, "Learning The Semantics of Words and Pictures," Proc. Int'l Conf. Computer Vision, vol 2, pp. 408-415, 2001.