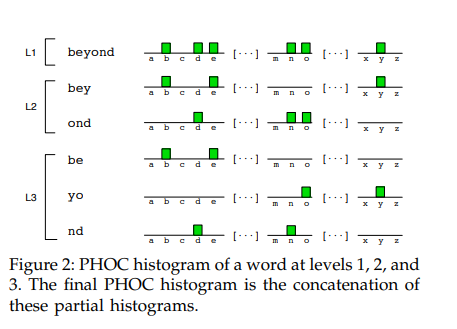


Image are first projected into an attributes space with the embedding function ɸi after being encoded with f. Labels are embedded into a label space of the same dimensionality using function ɸy at the same time. Then, we project embedded labels and attributes in a learned common subspace by minimizing a dissimilarity( khác nhau) function 

In this common subspace representations are comparable and labels and images that are relevant to each other are brought together.

By having images and text strings share a common subspace with a defined metric, word spotting and recognition become a simple nearest neighbor problem in a low-dimensional space. We can perform QBE and QBS (or even a hybrid QBE+S, where both an image and its text label are provided as queries) using exactly the same retrieval framework. The recognition task simply becomes finding the nearest neighbor of the image word in a text dictionary embedded first into the PHOC space and then into the common subspace. Since we use compact vectors, compression and indexing techniques such as Product Quantization [20] could now be used to perform spotting in very large datasets.

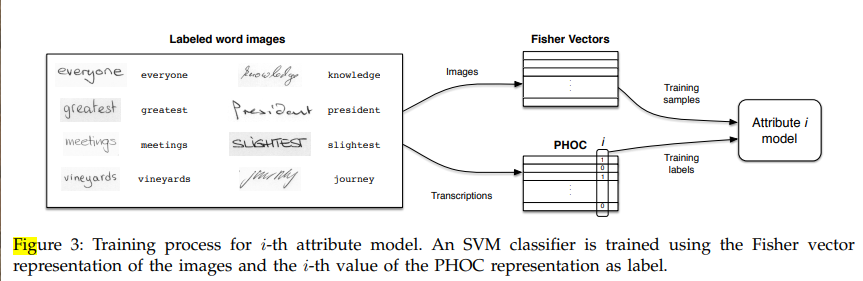


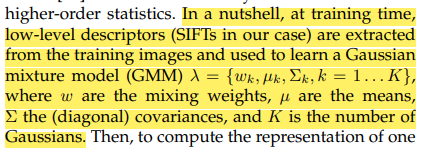
Instead of doing this, we focus on different regions of the word. At level 2, we define attributes such as “word contains character x on the first half of the word” and “word contains character x on the second half of the word”. Level 3 splits the word in 3 parts, level 4 in 4, etc. In practice, we use levels 2, 3, 4, and 5, leading to a histogram of (2 + 3 + 4 + 5) × 36 = 504 dimensions. Finally, we also add the 50 most common English bigrams at level 2, leading to 100 extra dimensions for a total of 604 dimensions. For that, we first define the normalized occupancy of the k-th character of a word of length n as the interval, where the postion k is zero-based



2. Learning attributes with PHOCS

we use linear SVMs. Word images are first encoded into feature vectors, and these feature vectors are used together with the PHOC labels to learn SVM-based attribute models

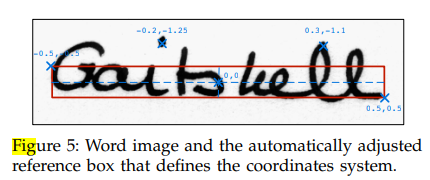




Lỗi có thể xảy ra: “since weights add little extra information) evaluated at those points. This leads to a highly discriminative, high-dimensional signature”.

3. Adding Spatial Information

In a nutshell, the SIFT descriptors of the image are enriched by appending the normalized x and y coordinates and the scale they were extracted at. Then, the GMM is learned not on the original SIFT features but on these enriched ones. Next we automatically and approximately find the begining and end, as well as the baseline and the median line of the word (by greedily finding the smallest region that contains 95% of the density of the binarized image) and use that for our reference coordinates system The center of that box corresponds to the origin, and the limits of the box are at [−0.5, 0.5

Ex: 

the GMM vocabulary is learned using the whole image. We propose to improve these representations by learning region-specific GMMs. At training time, we split the images in regions similar to spatial pyramids, and learn an independent, specialized vocabulary on each region. These GMMs are then merged together and their weights are renormalized to sum 1.

Correctly predicting the attributes is of paramount importance, since it is deeply correlated with the final performance at retrieval and recognition tasks. We used the training set of IAM to learn the attributes, and then evaluate the average precision of each attribute on the test set and report the mean average precision