Mid Evaluation
Computer Vision
Project Report
Word Spotter

Praveen Balireddy (2018201052)
Aman Joshi (2018201097)
Abhijeet Panda (2018201044)

Contents

Abstract 1

Challenges with previous works 2

Objective 2

Approach 2

Encoding Formats 4

Word Representation using Fisher Vectors 4

Supervised Word Representation with PHOC Attributes 6

Algorithm to Develop 7

Work So Far 7

Next Steps 7

Abstract

The project's focus is to provide an approach to multi-writer word spotting, where the goal would be to find a query word in a dataset comprised of document images. It is an attributes-based approach that leads to a low-dimensional, fixed-length representation of the word images that is fast to compute and, especially, fast to compare. This approach would lead to a unified representation of word images and strings, which seamlessly allow one to indistinctly perform query-by-example, where the query is an image, and query-by- string, where the query is a string.

Challenges with previous works

- Out of Vocabulary words(words not there in training images, but exist in the test images)
- Time taken for the search
- Same word, different handwriting

Objective

To find all instances of a given word in a potentially large dataset of document images. The various types of Queries to be handled are:

- Query by example(Image)
- Query by string(Text)

Approach

Build a unified classifier to predict the attribute representation(PHOC), given an FV descriptor representation calculated over SIFT features of the query.

Maintain a PHOC dictionary for each image in the document dataset.

Given a QBS/QBE, get the matches with the cosine similarity score above a certain threshold among all the document images

Training

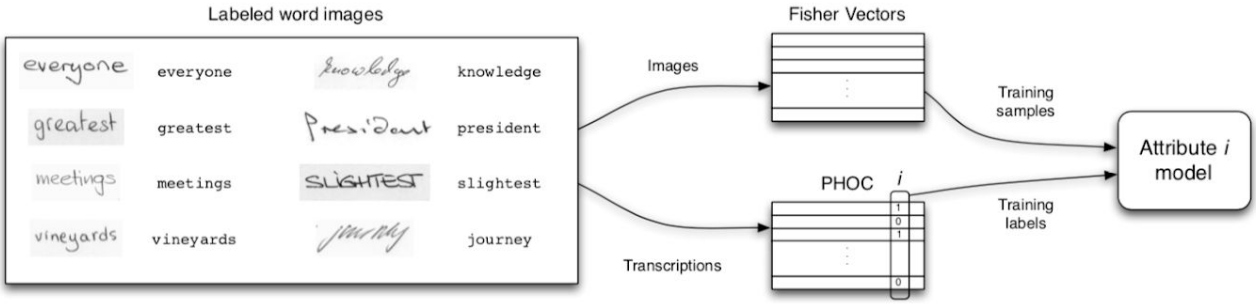

Fig-1 Training process for i-th attribute model. A classifier is trained using the FV representation of the images and the i-th value of the PHOC representation as label.

Testing:

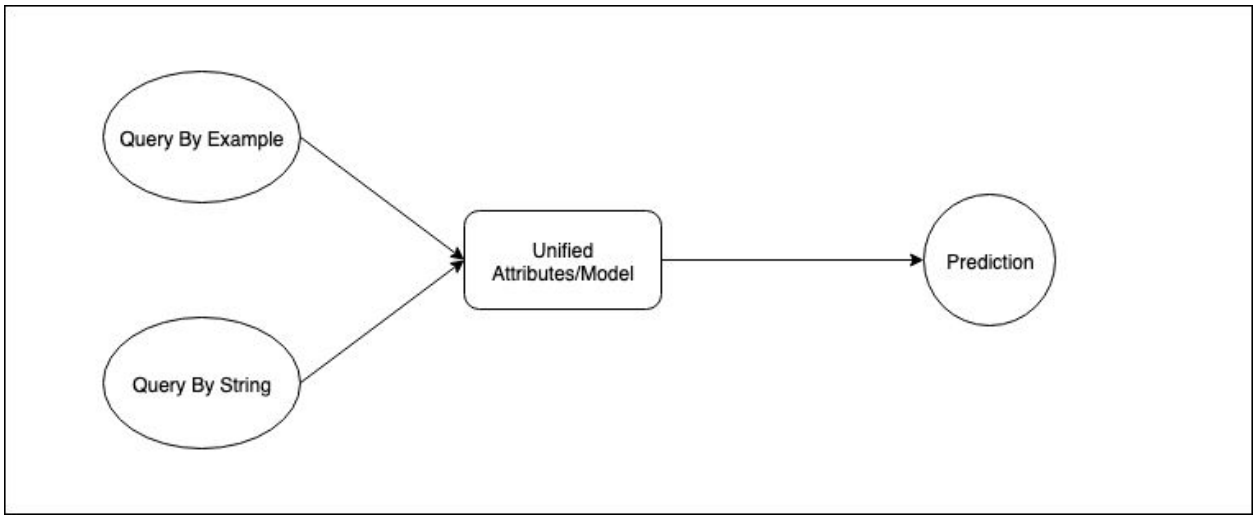

Fig-2 Unified Model Representation

Encoding Formats

Word Representation using Fisher Vectors

In this project, the Fisher vector (FV) representation is computed over SIFT descriptors extracted densely from the word image. The dimension of the SIFT Descriptors has been reduced using Principal Component Analysis (PCA). The Fisher vector can be understood as a bag of words that also encodes higher order statistics, and has been shown to be a state-of-the-art encoding method for several computer vision tasks such as image classification and retrieval. First, fixed-length word image representations are reviewed and FV is introduced as the reference representation. Then labeled training data is used to embed these FV representations in a more discriminative and low-dimensional space by means of attributes.

Algorithm 1 Compute Fisher vector from local descriptors

Input:

- Local image descriptors $X = \{x_t \in \mathbb{R}^D, t = 1, \ldots, T\}$,
- Gaussian mixture model parameters $\lambda = \{w_k, \mu_k, \sigma_k, k = 1, \ldots, K\}$

Output:

- normalized Fisher Vector representation $\mathcal{G}_\lambda^X \in \mathbb{R}^{K(2D+1)}$

1. **Compute statistics**
   - For $k = 1, \ldots, K$ initialize accumulators
     - $S_k^0 \leftarrow 0, \quad S_k^1 \leftarrow 0, \quad S_k^2 \leftarrow 0$
   - For $t = 1, \ldots T$
     - Compute $\gamma_t(k)$
     - For $k = 1, \ldots, K$:
       - $S_k^0 \leftarrow S_k^0 + \gamma_t(k)$,
       - $S_k^1 \leftarrow S_k^1 + \gamma_t(k) x_t$,
       - $S_k^2 \leftarrow S_k^2 + \gamma_t(k) x_t^2$
2. **Compute the Fisher vector signature**
   - For $k = 1, \ldots, K$:

$$\begin{aligned} \mathcal{G}_{\alpha_k}^X &= \left(S_k^0 - T w_k\right) / \sqrt{w_k} \\ \mathcal{G}_{\mu_k}^X &= \left(S_k^1 - \mu_k S_k^0\right) / \left(\sqrt{w_k}\sigma_k\right) \\ \mathcal{G}_{\sigma_k}^X &= \left(S_k^2 - 2\mu_k S_k^1 + (\mu_k^2 - \sigma_k^2) S_k^0\right) / \left(\sqrt{2w_k}\sigma_k^2\right) \end{aligned}$$

   - Concatenate all Fisher vector components into one vector $\mathcal{G}_\lambda^X = \left(\mathcal{G}_{\alpha_1}^X, \ldots, \mathcal{G}_{\alpha_K}^X, \mathcal{G}_{\mu_1}^{X\prime}, \ldots, \mathcal{G}_{\mu_K}^{X\prime}, \mathcal{G}_{\sigma_1}^{X\prime}, \ldots, \mathcal{G}_{\sigma_K}^{X\prime}\right)'$
3. **Apply normalizations**
   - For $i = 1, \ldots, K(2D+1)$ apply power normalization
     - $\left[\mathcal{G}_\lambda^X\right]_i \leftarrow \text{sign}\left(\left[\mathcal{G}_\lambda^X\right]_i\right) \sqrt{\left|\left[\mathcal{G}_\lambda^X\right]_i\right|}$
   - Apply ℓ_2-normalization: $\mathcal{G}_\lambda^X = \mathcal{G}_\lambda^X / \sqrt{\mathcal{G}_\lambda^{X\prime} \mathcal{G}_\lambda^X}$

Supervised Word Representation with PHOC Attributes

The Project defines attributes such as “word contains an a” or “word contains a k”, leading to a histogram of 26 dimensions when using the English alphabet. Then, at training time, models are learned for each of the attributes using the image representation of the words (FVs in our case) as data and set their labels as positive or negative according to whether those images contain that particular character or not (see Figure 2). During testing time, given the FV of a word, its attribute representation can be computed by concatenating the scores that those models yield on that particular sample. After calibrating the scores (using, e.g., Platts scaling), these attribute representations can be compared using measures such as the Euclidean distance or the cosine similarity. However, this model is not a word- discriminative: words such as “listen” and “silent” share the same representation. Therefore, a pyramid version of this histogram of characters is used, which is called PHOC. Instead of finding characters in the whole word, the focus is on different regions of the word.

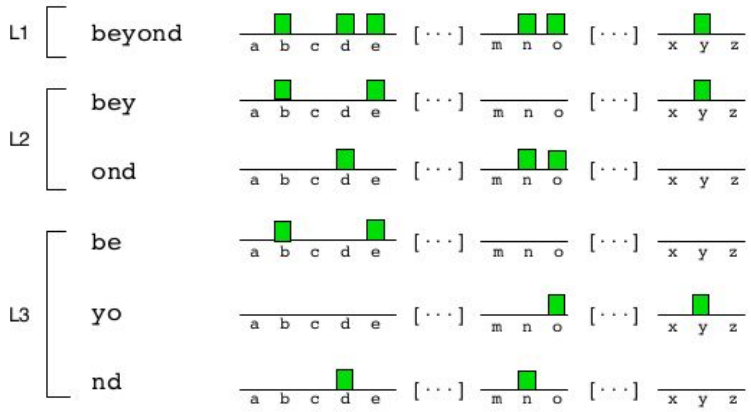

Fig-3 PHOC histogram at levels 1, 2, and 3. The final PHOC histogram is the concatenation of these partial histograms.

Algorithm

1. Get the SIFT features of the image and find their Fisher vectors
2. Predict/train the PHOC attributes using a classifier(SVM), given the FV
3. Since we have the image and string for a word, actual PHOC attributes can be found by using the string input
4. Using the CCA tool, get the projections of the predicted scores and the ground truth values
5. Use distance/cosine similarity to compute the mean average precision

Work So Far

1) Able to get FV representation from SIFT vectors
2) Used PCA to reduce dimensions of each SIFT vector from 128 to 64
3) Trained a GMM with 16 gaussians using ~35k examples on the FVs
4) Did a benchmark test, by getting the cosine similarity scores between the test data FVs and the training dataset FVs
5) MAP score on seen test dataset(~10k examples): 0.20044
6) MAP score on unseen test dataset(~10k examples): 0.24188

Next Steps

1) Implement a PHOC representation for a given word string
2) Train a SVM to predict the PHOC representation given a Fisher Vector
3) Understand and implement Canonical correlation analysis to project the predicted PHOC representation of the image and the ground truth representation of the actual word string into the same subspace
4) Compare the MAP score with the benchmark score