



Document Specific Sparse Coding for Word Retrieval

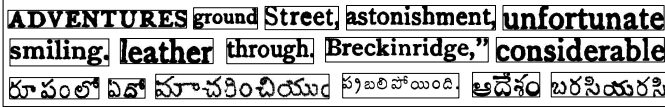
Ravi Shekhar and C.V. Jawahar

CVIT, IIIT-Hyderabad, INDIA



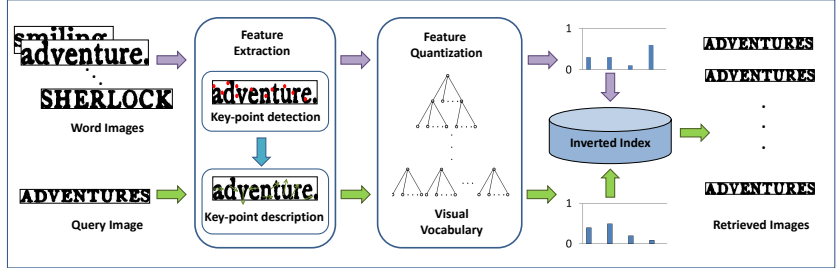
Introduction

- Robust OCRs are unavailable for many non-Latin languages
- Recognition free approach based on Bag of Words outperforms OCR based methods
- Performance is further improved by defining document specific sparse coding
- Coding is provided by using locality constraint at word level
- Next level coding is provided at character level representation
- Text query support is provided using pre-learned character representation
- Sample images from dataset are shown below:



Bag of Words

- Represents word image as a histogram of visual words
- Histogram is indexed using inverted file index

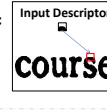


Document Specific Sparse Coding

Coding and Pooling in BoW

- Generates code from raw descriptor
- Single descriptor is assigned to single visual word
- Has codeword uncertainty and codeword plausibility problems

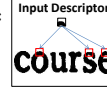
$$\arg \min_c \sum_{i=1}^N \|x_i - Bc_i\|^2 \text{ s.t. } \|c_0\|_0 = 1, \|c_i\|_0 = 1 \forall i$$



Sparse Coding

- Single descriptor is assigned to one or more visual words
- Has less quantization error but slow in computation

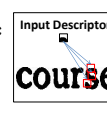
$$\arg \min_c \sum_{i=1}^N \|x_i - Bc_i\|^2 + \lambda \|c_i\|_1$$



Sparsity due to Locality in Coding

- Locality constrained linear coding (LLC) learns representation using nearest codewords (Wang et al. CVPR 2010)
- Locality of character is more important than sparsity
- Similar patches will have similar codewords

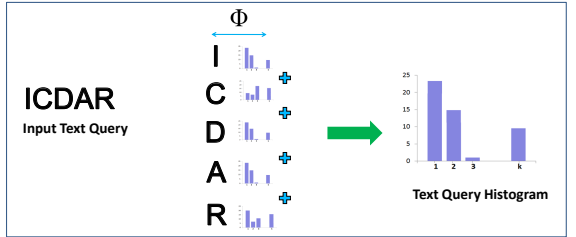
$$\arg \min_c \sum_{i=1}^N \|x_i - Bc_i\|^2 + \lambda \|d_i \odot c_i\|_1 \text{ s.t. } 1^T i = 1, \forall i$$



* where x_i : i^{th} descriptor of image $X=[x_1, \dots, x_N]$, c_i : i^{th} code for image X from $C=[c_1, \dots, c_N]$, N : #data points, B : codebook, ℓ^0 : L0 norm, ℓ^1 : L1 norm, $\|\cdot\|_0$: norm, $\odot$: element-wise multiplication, d_i : i^{th} locality adapter.

Text Query Support

- Originally, retrieval is performed by example
- Character specific representation is learned offline
- Word representation is calculated using learned representation



Retrieved Results

Data set Details

Book	# Pages	# Words
English-1601	363	113008
Telugu-1718	100	21345
Telugu-1716	102	4121

mAP of Text Query Support

Language	mAP
English	0.87
Telugu	0.90

Sample Outputs

Query Image	Retrieved Images				
ADVENTURES	ADVENTURES	ADVENTURES	ADVENTURES	ADVENTURE	
beggar	beggar	beggar.	beggar,	begging	
Holmes	Holmes	Holmes,	Holmes.	Holmes,"	
కలిగి	కలిగి	కలిగి	కలిగి	కలిగి	
భూస్వామ్య	భూస్వామ్య	భూస్వామ్య	"భూస్వామ్య	భూస్వాములు.	

mAP of Different Methods

Book	BoW	BoW + SIFT Re-ranking	BoW + LCS Re-ranking	Doc. Coding	Doc. Coding + LCS Re-ranking
English-1601	0.8015	0.8645	0.92	0.8765	0.9451
Telugu-1718	0.7834	0.8861	0.918	0.92	0.96
Telugu-1716	0.8173	0.8531	0.9036	0.91	0.95

Implementation Details

- Retrieval is performed based on similarity and re-ranking score
- Similarity score is given by
$$Sim_Score(Q, I) = \frac{\sum_{i=1}^{|Q|} w_i}{\sum_{j=1}^{|I|} w_j} \text{ where } w_i = \frac{1}{\log(f_i + 1)} \quad \forall$$
- Re-ranking is performed based on Longest common sub-sequence of visual words and given by
$$LCS_Score(Q, I) = \frac{\sum_{i=1}^{lcs(Q, I)} w_i}{\sum_{j=1}^{|Q|} w_j} \quad \forall$$
- Final score is given by
$$Score(Q, I) = [\gamma \times Sim_Score(Q, I)] + [(1 - \gamma) \times LCS_Score(Q, I)] \quad \forall$$

$\forall$ where Q : visual words corresponding to query image, I : visual words corresponding to candidate image, w_i : weight of i^{th} visual word, f_i : frequency of i^{th} visual word, γ : weighting parameter

Achievements

- An efficient document specific coding technique is proposed
- Text query support is provided using pre-learned representation
- Experimentation is performed on English and Telugu scripts to achieve state of the art performance

Future Work

- Selecting codebook specific to particular documents
- Designing document specific descriptor
- Learning character representation by using automatic mining of codebook

