Supplementary Material for Paper 523

July 9, 2012

In this Supplementary Material,we show all of experiments's results for different Encodeing,Normalization,pooling method.

1 Normalization

Then pooled feature $\mathbf{p}$ is further normalized by some methods. Generally, there are three common normalization techniques:

- ℓ_1**-Normalization.** In ℓ_1 normalization , the feature $\mathbf{p}$ is divided by its ℓ_1-norm: $\mathbf{p} = \mathbf{p}/\sum_{k=1}^{K}|p_k|$
- ℓ_2**-Normalization.** In ℓ_2 normalization , the feature $\mathbf{p}$ is divided by its ℓ_2-norm: $\mathbf{p} = \mathbf{p}/\sqrt{(\sum_{k=1}^{K}p_k^2)}$
- **Power Normalization.** In power normalization , we apply in each dimension the following function

  $$f(p_k) = \text{sign}(p_k)|p_k|^\alpha$$

  where $0 \leq \alpha \leq 1$ is a parameter of the normalization. We can combine power normalization with ℓ_1-normalization or ℓ_2-normalization.

2 Encodeing Method and Results

- $\ell 1$ is ℓ_1-Normalization.
- $\ell 2$ is ℓ_2-Normalization.
- P+$\ell 1$ is Power Normalization and ℓ_1-Normalization.
- P+$\ell 2$ is Power Normalization and ℓ_2-Normalization.

$0 \leq \alpha \leq 1$ is a parameter of the Power normalization.

codebook size	Normalization							
-	$\ell 1$	$\ell 2$	P+$\ell 1$			P+$\ell 2$		
-	-	-	α					
-	-	-	0.25	0.5	0.75	0.25	0.5	0.75
1k	16.84	19.22	17.45	18.47	17.93	2002	20.92	20.68
2k	18.87	20.92	19.76	21.29	20.48	22.11	23.33	22.72
3k	17.76	21.11	20.17	20.61	20	23.38	23.68	22.85
4k	18.78	21.87	20.5	20.98	20.63	23.42	24.6	24.07
6k	19.67	22.44	20.81	21.87	21.37	24.86	25.45	24.68
8k	18.89	22.27	21.18	21.53	20.85	24.34	24.79	23.88

Table 1: Results of different codebook size and Normalization Method for Vector Quantization(VQ) on HMDB51.

2.1 Vector Quantization

VQ is also known as *Hard-assignment coding*. For each local feature descriptor $\mathbf{x}_n$, it is represented by its nearest visual word in the dictionary:

$$u_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_k \|\mathbf{x}_n - \mathbf{d}_k\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

2.2 Soft-assignment Encoding

For each local feature, the k^{th} coefficient represents the degree of membership of the local feature $\mathbf{x}_n$ being to the k^{th} visual word:

$$u_{nk} = \frac{\exp(-\beta\|\mathbf{x}_n - \mathbf{d}_k\|^2)}{\sum_{j=1}^{K}\exp(-\beta\|\mathbf{x}_n - \mathbf{d}_j\|^2)} \tag{2}$$

where β is the smoothing factor controlling the softness of the assignment. Note that all the K visual words are used in computing u_{nk}. Recently developed a *localized soft-assignment coding*. They only considered the k nearest visual words into encoding, and conceptually set its distances to the remaining words as infinity,

$$u_{nk} = \frac{\exp(-\beta\hat{d}(\mathbf{x}_n, \mathbf{d}_k))}{\sum_{j=1}^{K}\exp(-\beta\hat{d}(\mathbf{x}_n, \mathbf{d}_j))} \tag{3}$$

where $\hat{d}(\mathbf{x}_n, \mathbf{d}_k)$ is defined as follows:

$$\hat{d}(\mathbf{x}_n, \mathbf{d}_k) = \begin{cases} \|\mathbf{x}_n - \mathbf{d}_k\|^2 & \text{if } \mathbf{d}_k \in N_k(\mathbf{x}_n) \\ \infty & \text{otherwise.} \end{cases} \tag{4}$$

where $N_k(\mathbf{x}_n)$ denotes the k-nearest neighborhood of $\mathbf{x}_n$ defined by the distance $\|\mathbf{x}_n - \mathbf{d}_k\|^2$.

codebook size	KNN	β	pooling	Normalization							
-	-	-	-	$\ell 1$	$\ell 2$	P+$\ell 1$			P+$\ell 2$		
-	-	-	-	-	-	α					
-	-	-	-	-	-	0.25	0.5	0.75	0.25	0.5	0.75
1k	5	1	max	17.52	22.68	16.27	17.52	18.37	18.82	20.61	21.11
2k	5	1	max	21.24	23.92	18	19.78	20.11	21.24	23.07	23.36
3k	5	1	max	22.37	26.14	20.41	21.5	22.18	24.23	25.4	25.97
4k	5	1	max	23.16	27.21	21.26	21.72	22.7	25.21	26.07	27.06
6k	5	1	max	24.31	28.82	22	22.59	23.62	26.8	27.65	28.67
8k	5	1	max	25.27	28.98	22.72	24.2	24.53	27.54	28.24	28.74
1k	5	1	sum	18.95	21.02	19.04	20.96	20.04	22.92	23.46	22.37
2k	5	1	sum	20.31	22.05	20.94	22.18	21.35	24.66	25.51	24.14
3k	5	1	sum	20.54	22.72	21.61	23.36	22.53	26.14	26.75	24.79
4k	5	1	sum	21.48	23.97	22.48	23.73	23.33	27.49	27.25	26.6
6k	5	1	sum	21.83	24.81	23.57	24.2	23.86	28.8	28.71	26.93
8k	5	1	sum	22.27	24.95	23.79	24.51	24.31	28.71	28.71	27.58

Table 2: Results of different codebook size , Normalization Method and pooling for Soft-assignment Encoding(SA-K) on HMDB51.

codebook size	β	pooling	Normalization
-	-	-	$\ell 2$
1k	1	max	18.69
2k	1	max	20.13
3k	1	max	21.7
4k	1	max	21.55
6k	1	max	20.59
8k	1	max	20.94

Table 3: Results of different codebook size for Soft-assignment Encoding(SA-all) on HMDB51.

codebook size	β	pooling	Normalization
-	-	-	P+$\ell 2$
-	-	-	$\alpha = 0.5$
1k	1	sum	20.28
2k	1	sum	21.81
3k	1	sum	21.96
4k	1	sum	21.9
6k	1	sum	22.68

Table 4: Results of different codebook size for Soft-assignment Encoding(SA-all) on HMDB51.

codebook size	λ	pooling	Normalization
-	-	-	$\ell 2$
1k	0.15	max	22.92
2k	0.15	max	25.56
3k	0.15	max	26.45
4k	0.15	max	27.15
6k	0.15	max	29.2
8k	0.15	max	29.97

Table 5: Results of different codebook size for Sparse Encoding on HMDB51.

codebook size	λ	pooling	Normalization			
-	-	-	$\ell 1$	$\ell 2$	P+$\ell 1$	P+$\ell 2$
-	-	-	-	-	$\alpha = 0.5$	
8k	0.15	max	28.02	29.97	27.26	28.3
8k	0.15	sum	26.28	29.07	29.77	31.82

Table 6: Results of different Normalization Method size for Sparse Encoding on HMDB51.

2.3 Sparse Encoding

SPC represents a local feature $\mathbf{x}_n$ by a sparse linear combination of basis vectors. The coefficient vector $\mathbf{u}_n$ is obtained by solving an ℓ_1-norm regularized approximation problem,

$$\mathbf{u}_n = \arg\min_{\mathbf{u}\in\mathbb{R}^K} \|\mathbf{x}_n - \mathbf{D}\mathbf{u}\|^2 + \lambda\|\mathbf{u}\|_1. \tag{5}$$

2.4 Locality-constrained Linear Encoding

Unlike the sparse coding, LLC enforces locality instead of sparsity and this leads to smaller coefficient for the basis vectors far away from the local feature $\mathbf{x}_n$. The coding coefficients are obtained by solving the following optimization,

$$\begin{aligned} &\mathbf{u}_n = \arg\min_{\mathbf{u}\in\mathbb{R}^K} \|\mathbf{x}_n - \mathbf{D}\mathbf{u}\|^2 + \lambda\|\mathbf{s}_n \odot \mathbf{u}\|^2 \\ &\text{s.t.} \quad \mathbf{1}^T\mathbf{u}_n = 1 \end{aligned} \tag{6}$$

where $\odot$ denotes the element-wise multiplication and $\mathbf{s}_n$ is the locality adaptor that gives weights for each basis vector proportional to its similarity to the input descriptor $\mathbf{x}_n$,

$$\mathbf{s}_n = \exp\left(\frac{\text{dist}(\mathbf{x}_n, \mathbf{D})}{\sigma}\right) \tag{7}$$

where $\text{dist}(\mathbf{x}_n, \mathbf{D}) = [\text{dist}(\mathbf{x}_n, \mathbf{d}_1), \cdots, \text{dist}(\mathbf{x}_n, \mathbf{d}_K)]^T$ and $\text{dist}(\mathbf{x}_n, \mathbf{d}_k)$ is the Euclidean distance between $\mathbf{x}_n$ and $\mathbf{d}_k$. σ is used for adjusting the weighted decay speed for the locality adaptor. The constraint $\mathbf{1}^T\mathbf{u}_n = 1$ follows the shift-invariant requirements of the LLC code. In practice,

codebook size	KNN	pooling	Normalization
-	-	-	$\ell 1$
1k	5	max	20.89
2k	5	max	23.71
3k	5	max	25.80
4k	5	max	26.54
6k	5	max	27.76
8k	5	max	28.68

Table 7: Results of different codebook size for Locality-constrained Linear Encoding on HMDB51.

codebook size	KNN	pooling	Normalization			
-	-	-	$\ell 1$	$\ell 2$	P+$\ell 1$	P+$\ell 2$
-	-	-	-	-	$\alpha = 0.5$	
8k	5	max	23.53	28.68	23.23	28.11
8k	5	sum	21.79	21.78	25.1	28.16

Table 8: Results of different Normalization Method size for Locality-constrained Linear Encoding on HMDB51.

an approximation is proposed to improve its computational efficiency. Ignoring the second term in Equation, it directly selects the k nearest basis vectors of $\mathbf{x}_n$ to minimize the first term by solving a much smaller linear system. This gives the coding coefficient for the selected k basis vectors and other coefficients are simply set to zero.

2.5 Fisher Kernel Encoding

Fisher kernel is introduced for large-scale image categorization. Unlike previous coding methods based on a codebook, the fisher kernel is a generic framework which combines the benefits of generative and discriminative approaches. Suppose we has a generative model $p(\mathbf{x};\theta)$ in feature space. Let $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ be the set of T local features extracted from an video. Then the video can be described by the gradient vector:

$$G_\theta^{\mathbf{X}} = \frac{1}{T}\nabla_\theta \log p(\mathbf{X};\theta) \tag{8}$$

Note that the dimensionality of this vector depends only on the number of parameters in θ, not on the number of local features T. A natural kernel on these gradients is:

$$K(\mathbf{X},\mathbf{Y}) = G_\theta^{\mathbf{X}T} F_\theta^{-1} G_\theta^{\mathbf{Y}} \tag{9}$$

where F_θ is the Fisher information matrix of $p(\mathbf{x};\theta)$:

$$F_\theta = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x};\theta)}[\nabla_\theta \log p(\mathbf{x};\theta) \nabla_\theta \log p(\mathbf{x};\theta)^T] \tag{10}$$

PCA	GMM	org	Normalization							
-	-	-	$\ell 2$	P+$\ell 2$						
-	-	-	-	α						
-	-	-	-	0.125	0.25	0.375	0.5	0.625	0.75	0.875
40	32	17.02	19.04	18.3	21.02	22.35	23.36	22	21.59	20.26
-	64	20.22	20.54	21.76	24.64	25.49	25.95	24.88	23.07	22.09
-	128	20.54	20.61	25.45	27.6	27.93	27.17	25.58	24.44	22.81
60	32	17.76	20.07	19.06	21.044	23.18	23.2	22.64	21.76	21.63
-	64	20.31	22.05	22.42	24.18	25.71	25.88	25.45	24.23	23.38
-	128	21.57	22.35	26.43	28.47	29.46	29.08	28	25.86	23.86
80	32	18.67	21.44	20.57	22.75	24.49	24.14	23.53	23.31	22.75
-	64	20.52	22.11	24.16	16.1	26.27	26.43	25.4	24.23	23.01
-	128	22.09	21.74	27.58	29.02	29.22	28.08	26.45	25.19	23.77
100	32			21.94	23.68	24.29	24.73	23.31	22.37	21.5
-	64			25.36	26.36	26.86	26.58	25.57	24.18	22.7
120	32			21.2	21.98	22.09	21.94	21.94	21.15	19.91
-	64			25.21	25.9	25.8	25.12	24.23	22.51	21.29
162	32	18.52	21.72	20.7	22.46	23.36	22.85	22.77	22.42	22.07
-	64	20.02	19.41	19.67	20.28	21.72	21.02	20.44	20.52	20.52
-	128	18.5	17.04	15.34	16.67	18.43	18.71	18.91	18.13	17.67

Table 9: Results of different PCA and GMM size for Fisher Kernel Encoding on HMDB51.

As F_θ is symmetric and positive definite, then we can define the *Fisher Vector* as:

$$\mathcal{G}_\theta^{\mathbf{X}} = F_\theta^{-1/2} G_\theta^{\mathbf{X}} \tag{11}$$

Here we use Gaussian Mixture Model for $p(x;\theta)$, and assume that the covariance matrices Σ_k are diagonal. Then fisher coding can be derived as,

$$\mathcal{G}_{\mu,k}^{\mathbf{X}} = \frac{1}{T\sqrt{\pi_k}} \sum_{t=1}^{T} \gamma_t(k) \left(\frac{\mathbf{x}_t - \mu_k}{\sigma_k} \right) \tag{12}$$

$$\mathcal{G}_{\sigma,k}^{\mathbf{X}} = \frac{1}{T\sqrt{\pi_k}} \sum_{t=1}^{T} \gamma_t(k) \left[\frac{(\mathbf{x}_t - \mu_k)^2}{\sigma_k^2} - 1 \right] \tag{13}$$

where$\gamma_t(k)$ is the soft assignment of local feature $\mathbf{x}_t$ to Gaussian i:

$$\gamma_t(k) = \frac{\pi_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)}{\sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x};\mu_i,\Sigma_i)} \tag{14}$$

The final gradient vector $\mathbf{u}$ is the concatenation of the $\mathcal{G}_{\mu,k}^{\mathbf{X}}$ and $\mathcal{G}_{\sigma,k}^{\mathbf{X}}$ and its total dimension is $2KD$.

PCA	GMM	org	Normalization			
-	-	-	$\ell 1$	$\ell 2$	P+$\ell 1$	P+$\ell 2$
-	-	-	-	-	$\alpha = 0.5$	
100	128	20.22	17.04	21.87	23.86	28.23

Table 10: Results of different Normaliztion Method for Fisher Kernel Encoding on HMDB51.