Video Semantic Indexing using Image Classification

Ming Yang, Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Kai Yu
NEC Laboratories America, Inc.
10080 N. Wolfe Road, SW-350, Cupertino, CA 95014
{myang, ylin, flv, zsh, kyu}@sv.nec-labs.com

Mert Dikmen, Liangliang Cao, Thomas S. Huang
Dept. of ECE, University of Illinois Urbana-Champaign
405 N. Mathews Ave, Urbana, IL 61801
{mdikmen, cao4, huang}@ifp.uiuc.edu

Abstract

This notebook paper summarizes Team NEC-UIUC's approaches for TRECVid 2010 Evaluation of Semantic Indexing. Our submissions mainly take advantage of advanced image classification methods using linear coordinate coding (LCC) of local features powered by the distributed computing software Hadoop. For every video shot, we evenly sample key frames and extract dense local features including DHOG and LBP, which are encoded by linear coordinate coding. Then, for every concept large-scale linear SVM classifiers are trained based on spatial pyramid of LCC features. Finally, we employ multiple instance learning to rank the video shots according to the SVM scores of individual frames. Our systems achieve mean extended inferred average precision (mean xinfAP) 7.40% for the 30 concepts evaluated by NIST and mean average precision 28.63% using 1/5 of the development data as the validation set for the total 130 concepts.

1. Introduction

Content-based video semantic indexing for large scale internet video footages is a challenging problem for computer vision and machine learning, which is of great interests to video hosting and archiving services. The challenges include the huge amount of data to analyze compared to static images and the diverse nature of different concepts, *e.g.*, human attributes, scenes, objects, and actions, *etc*.

TRECVid 2010 Evaluation of Semantic Indexing [5] uses a new dataset called Internet Archive Videos provided by NIST, which are real user uploaded video clips. Both development and test sets include 200 hours videos with duration between 10 seconds to 3.5 minutes. Given the test collection, master shot reference, and the concept definitions, this evaluation task requires to return for each of the 130 concepts a list of at most 2000 shot IDs from the test collection ranked according to the likelihood of presence of that concept in video shots. To our best knowledge, this is the largest public available internet video dataset so far.

The system developed by Media Analytic group at NEC laboratories America mainly utilizes frame-based image classification methods using linear coordinate coding (LCC) [10] of local features and linear SVM classifiers. To handle the huge computation of video data, we extensively leverage the distributed computing software Hadoop [1] to perform feature extraction, SVM learning, and classification. To optimize the rank of related video shots of a concept, we employ multiple instance learning [8] to generate the rank of shots from SVM scores of frames. The details of the system will be present in Sec. 2 with the experiments in Sec. 3.

2. System Overview

The understanding of video contents ideally demands for fusion of multiple modalities from meta data, closed captions, audio, and visual features. The audio features may be particular useful to distinguish some concepts, such as *singing*, however, the subset of concepts concerning with audio features may need manual selection. In our system, we apply the same approach to all concepts based on image classification. The correlation or interaction among frames are considered merely in the pooling stage to generate the likelihood of shots.

In our system, for one frame sampled from a shot, we conduct local feature extraction, feature encoding, and SVM classification. Afterwards, given the master shot references, we employ multiple instance learning to generate the likelihood of a shot from the corresponding frame-based SVM scores. For one frame, we densely extract two kinds

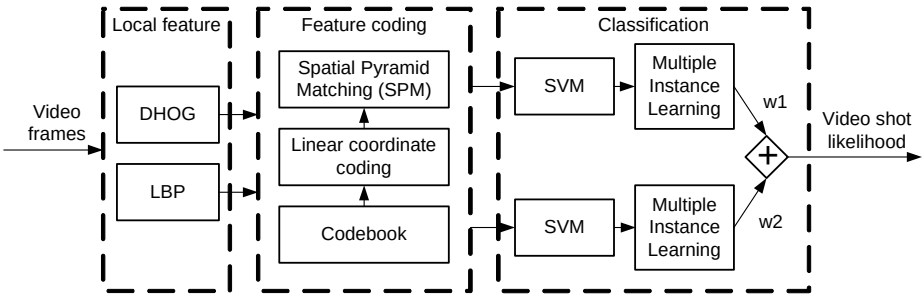

Figure 1. The system diagram of NEC-UIUC's approach on video semantic indexing.

of local features, the DHOG features, where DHOG is essentially a fast implementation of the SIFT descriptor [4], and the local binary pattern (LBP) features [6]. For each type of feature, we train a 4096-dimensional codebook and conduct linear coordinate coding [10]. Each frame is divided to 10 cells, the LCC codes in all cells are concatenated to generate the frame-based features which are fed to the SVM learner. With the frame-based SVM scores of a concept, we maximize the *noisy-OR* probability of the shots in the training set to generate the shot-based likelihood. The entire procedure is summarized in Fig. 1.

2.1. Feature Extraction

Given the shot boundary definition in the .ssb files [5], we evenly sample the key frames every 7 frames in a shot. For the short shots, we try to ensure there are at least 7 key frames, although there are quite a few very short shots with even less than 7 frames. Using this sampling scheme, we end up to process 2.01 millions frames for the development set and 2.28 millions frames for the test set.

For each frame, we calculate both dense DHOG [4] and LBP [6] features every 8 pixels with 4 patch sizes, *i.e.*, 7×7, 16×16, 25×25, and 31×31. For each type of feature, we learn a 4096-dimensional codebook using clustering, which is used in the linear coordinate coding [10] to encode the dense local features.

The spatial pyramid matching (SPM) [3] of interest point descriptors demonstrates superb performance in object and scene categorization due to its power to delineate the spatial layout of images. Following the idea, we divide the frame to 10 cells, *i.e.*, 1×1 and 3×3, and concatenate the LCC codes of each cell to represent the frame. Thus, the frame-level feature is 40960-dimensional. Using float numbers to store the features, the sizes of the frame-level features are 306G bytes ($2.01M\times40960\times4$) and 348G bytes ($2.28M\times40960\times4$) for the development and test set respectively.

2.2. SVM Training

For each of the 130 concepts, we learn a linear binary SVM classifier from the frame-level features. Since the subset of labeled shots for each concepts are different from each other, we have to generate separate training sets for different concepts. The manipulation of the 306G bytes training features turns out to be a challenge for us. After creating the data, an even bigger challenge is how to train SVM classifiers on such huge amount of data.

We utilize Hadoop to train the 130 binary SVM classifiers in parallel. The challenge in training lies in the fact that training data sets are often way too big to be fitted into memory. Although there exist some off-the-shelf packages that handles memory issues [2, 9], we found them computationally expensive – they typically need to go through data many times. To resolve the difficulties, we developed a novel averaged stochastic gradient descent (ASGD) method for train linear binary SVM classifier. The ASGD algorithm is advantageous. First, it is memory efficient – it only needs to load one data sample at a time, although in practice we load data chunk by chunk to speed up loading. Second, because of the memory efficiency, it is easy to run multiple ASGD algorithms in parallel on multi-core machines. Third, it can be proved that, by simple averaging schemes, the ASGD algorithm is able to achieve similar convergence rate as sophisticated second-order stochastic gradient descent [7], but it avoids the expensive computation of inverse Hessian matrix (or its approximation). Because of its fast convergence property, the ASGD method often achieves fairly good classification after a single pass of data especially when training data are not too few. In our experiments, we set the maximum pass for ASGD to be 5.

2.3. Ranking with Multiple Instance Learning

Even one frame in a shot is related to a concept, this shot shall be regarded as positive to the concept. Therefore, the video shot indexing is essentially a multiple instance learning problem. Thus, given the SVM scores of the frame-level features, we employ the multiple instance learning method [8] to generate the shot-level likelihood for each concept.

Denote the training shot set of a concept by $S = \{s_i\}, i = 1, \cdots, N$ and the SVM score of a frame in the shot s_i by x_{ij}, where j is the index of frames. We calculate

the probability of one frame being positive to a concept by

$$p_{ij} = 1 - (\tanh(-\frac{ax_{ij} + b}{2}) + 1)/2, \quad (1)$$

where a and b are two scalars to optimize. Then, the probability of a shot being positive is a *noise OR* of the probability of frames, that is

$$p_i = 1 - \prod_j (1 - p_{ij}). \quad (2)$$

Thus, we can compute the log likelihood of the training set

$$\log(L(S)) = \log(\prod {p_i}^{t_i} (1 - p_i)^{1-t_i}), \quad (3)$$

where $t_i \in \{0, 1\}$ is the label of the shot. After maximizing $\log(L(S))$ with respect to a and b, we employ Eq. 2 to rank the video shots and return the top 2000 shot IDs.

We have also tried the mean pooling and max pooling methods to rank the shots with the frame-based SVM scores. By using the multiple instance learning, we observe 1% improvement in terms of the mean average precision on the validation set.

2.4. Distributed Computing with Hadoop

For both development and testing set, there are more than 2 millions frames to process, moreover, we need to train 130 linear SVMs. Efficient computation and management are critical issues for this video semantic indexing task. Thus, we extensively leverage the distributed computing software Hadoop [1] to manage the computation, which implements the Map Reduce framework and a distributed file system, *i.e.*, Hadoop File System (HDFS). In addition, Hadoop is capable of managing a large number of data intensive jobs and HDFS can tolerate storage node failure without suffering data loss.

The experiments are mainly performed on a cluster of 64bit blade servers with Intel Xeon 2.5GHz CPU (8 cores) and 16GB RAM, which are managed by the Hadoop. Using about 10 servers in the Hadoop, we finish the feature extraction of DHOG features for the development set in around 40 hours, which significantly accelerates the computation by 20-40 times. It turns out the network bandwidth becomes the bottleneck when the computation nodes need to frequently access data on the HDFS. Taking advantage of the distributed computing infrastructure, we manage to finish the video semantic indexing task in two weeks.

3. Experiments

For both the development set and test set, there are 200 hours of short video clips with over 110K and 140K shots respectively. The development set is partially annotated for the 130 concepts. The statistics of the video shots and the number of labeled positive shots are summarized in Table 1 and Table 2, where we observe that there are about 30% shots with less than 10 frames and for more than one half of the concepts the numbers of positive shots are less than 500.

Table 1. The statistics of the video shots in the development and test set.

	Development set		Test set	
# frame	# shots	ratio	# shots	ratio
≤ 1	2376	2.00%	3066	2.12%
≤ 3	14907	12.57%	21640	14.93%
≤ 10	32389	27.31%	42068	29.02%
≤ 30	53415	45.05%	68343	47.15%
≤ 60	72626	61.25%	89852	61.98%
≤ 90	83779	70.65%	102682	70.83%
≤ 200	101658	85.73%	123717	85.34%
≤ 500	112194	94.61%	137026	94.52%
≤ 1000	115752	97.61%	141538	97.64%
>1000	118581	100%	144963	100%

Table 2. The statistics of the labeled positive shots.

# positive shots	5-100	101-500	501-1K
# concepts	31	41	17
# positive shots	1K-5K	5K-10K	$>$10K
# concepts	30	8	3

Table 3. Validation performance of *DHOG-SVM*, *LBP-SVM*, and their combination

	DHOG-SVM	LBP-SVM	DHOG+LBP-SVM
meanAP	27.58%	18.46%	28.63%

3.1. Validation Performance

We employ 4/5 of the development set as the training set and the remaining 1/5 as the validation set. For the validation set, we calculate the mean average precision (meanAP) of top 2000 shots without sampling. As shown in Table 3, the results of the DHOG features is superior to that of the LBP features. The combination of two features improves the meanAP by about 1%. The accuracy is varying a lot across different concepts, some are very good. The detailed results are shown in Table 5 in the Appendix.

Note this validation set is much smaller than the test set. We return 2000 shots from around 10K shots, in contrast, we need to submit top 2000 shots out of 140K shots from the test set. Thus, the meanAP in Table 5 may be an optimistic estimation.

3.2. Evaluation Results from NIST

NIST selectively evaluated 30 concepts using the mean extended inferred average precision (mean xinfAP). Our system, which combines the DHOG and LBP features, achieves mean xinfAP 7.40%, as shown in Table 4.

Our mean xinfAP 7.40% outperforms most of the sys-

Table 4. Evaluation results of *F_A_NEC-UIUC-4_4* by NIST

Concept	xinfAP	Concept	xinfAP	Concept	xinfAP	Concept	xinfAP
Airplane-flying	0.106	Animal	0.052	Asian-people	0.013	Bicycling	0.05
Boat-ship	0.117	Bus	0	Car-racing	0.019	Cheering	0.023
Cityscape	0.117	Classroom	0.022	Dancing	0.032	Dark-skinned-people	0.124
Demonstration-or-protest	0.11	Doorway	0.071	Explosion-fire	0.03	Female-face-closeup	0.117
Flowers	0.037	Ground-vehicles	0.13	Hand	0.037	Mountain	0.234
Nighttime	0.083	Old-people	0.035	Running	0.019	Singing	0.076
Sitting-down	0	Swimming	0.347	Telephones	0.006	Throwing	0.007
Vehicle	0.149	Walking	0.058				
Overall	0.074						

tems (or 90% of the systems) in the evaluation. The results are fairly good considering we combine limited image features. The best system in the evaluation gives mean xinfAP 9.0%. However, there are two main issues. First, the SVM training suffers from the extremely unbalanced training data. For example, for the concept *Bus*, there are only 31 positive shots with 68980 negative shots, and for *Car-racing*, there are 21 positive samples with 101343 negative samples, which poses severe challenges to the training algorithm. Second, the lack of motion features makes the system can hardly detect some actions not contingent to certain scenes, such as *Sitting-down*.

4. Conclusions

We test our advanced image classification methods on the video semantic indexing task. Our experiments interestingly show that using only image classification we achieve fairly good results. In addition, for large scale image and video analysis task, it is critical to take advantage of the emerging distributed computing software and infrastructure to ease the management of computation. Our future work includes incorporation of motion features and different treatments to concepts related to static objects and scenes or actions.

References

[1] Apache Hadoop. http://hadoop.apache.org/. 1, 3

[2] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 2

[3] S. Lazebnik, C. Achmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR'06*, volume 2, pages 2169–2178, New York City, June17 - 22 2006. 2

[4] D. G. Lowe. Distinctive image features from scale invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004. 2

[5] National Institute of Standards and Technology (NIST): TRECVID 2010 Evaluation for Semantic Indexing. http://www-nlpir.nist.gov/projects/tv2010/tv2010.html#sin, 2010. 1, 2

[6] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(7):971–987, 2002. 2

[7] B. Polyak and A. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992. 2

[8] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS'06*, 2006. 1, 2

[9] H.-F. Yu, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin. Large linear classification when data cannot fit in memory. In *ACM'KDD'10*, 2010. 2

[10] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *NIPS'09*, 2009. 1, 2

Appendix

The detailed performance on the validation set is shown in Table 5. The concepts are ranked according to the number of positive shots. The average precision is calculated for the top 2000 shots on the 1/5 of the original development set. The mean average precision is 28.63%.

Table 5. Validation results on 1/5 of the development set.

Concept	# Pos Shot	# Neg Shot	Avg. Precision	Concept	# Pos Shot	# Neg Shot	Avg. Precision
Person	36305	18841	23.71%	Outdoor	18800	12723	53.45%
Face	11084	33850	38.08%	Male-person	7945	37152	31.73%
Indoor	7574	29150	53.49%	Adult	7463	37099	28.15%
Vegetation	7051	13268	48.12%	Daytime-outdoor	6570	24580	41.98%
Overlaid-text	6431	11217	56.68%	Trees	6081	11186	60.14%
Single-person	5723	41352	33.67%	Female-person	4888	42903	33.93%
Entertainment	4734	14904	34.40%	Computer-screens	4549	12901	66.70%
Sky	4491	26885	56.13%	Plant	4180	27910	30.90%
Building	3893	27424	44.70%	Vehicle	3657	51050	29.18%
Suburban	3386	28166	26.76%	Singing	3151	45244	17.52%
Road	2956	29299	39.60%	Streets	2929	29235	36.32%
Dark-skinned-people	2590	46244	37.13%	Celebrity-entertainment	2426	46226	22.87%
Landscape	2387	29076	53.06%	Walking-running	2379	51778	20.15%
Ground-vehicles	2236	60415	22.57%	Actor	1883	48418	5.45%
Instrumental-musician	1739	48940	36.62%	Waterscape-waterfront	1659	31270	42.94%
Scene-text	1547	16084	23.78%	Sitting-down	1505	48167	17.53%
Athlete	1212	49831	10.12%	Car	1208	70006	31.49%
Two-people	1206	49397	20.12%	Explosion-fire	1194	16600	38.49%
Hand	1105	16336	17.25%	Beards	1081	55031	9.30%
Walking	1061	62130	14.08%	Crowd	1044	50173	32.14%
Politics	1024	15905	37.23%	Doorway	986	19175	33.82%
News-studio	961	45855	71.15%	Female-face-closeup	892	61077	24.72%
Sports	882	53363	4.93%	Animal	786	58699	9.46%
Beach	781	32075	47.64%	Reporters	740	50298	68.67%
Girl	651	60487	7.01%	Teenagers	612	49206	18.10%
Nighttime	575	36579	16.76%	Computers	572	16964	24.59%
Boy	561	56827	3.00%	Cityscape	558	32794	46.67%
Chair	555	16955	41.77%	Anchorperson	551	49945	78.29%
Mountain	519	35915	29.12%	Charts	518	17077	41.43%
Politicians	483	59790	46.42%	Snow	476	36714	40.85%
Dancing	466	52750	4.33%	Highway	371	32538	58.62%
Flowers	359	39583	11.11%	Roadway-junction	353	32838	13.56%
Kitchen	342	45846	20.11%	Demonstration-or-protest	312	50977	10.70%
Office	247	45125	29.28%	Indoor-sports-venue	240	44720	31.88%
Science-technology	235	17175	58.83%	Running	234	63888	17.19%
Press-conference	232	55943	32.64%	Stadium	226	17263	25.64%
Driver	222	51355	17.01%	Asian-people	221	49865	5.09%
Old-People	218	50762	9.23%	Laboratory	211	45713	2.25%
Government-leader	210	64151	23.32%	People-marching	207	53840	4.58%
Military	203	18828	5.97%	Bicycles	201	79143	24.08%
Meeting	199	50542	25.59%	Maps	186	20245	24.64%
Bridges	178	33741	13.43%	Boat-ship	172	68984	26.84%
Airplane	163	65363	19.14%	Cats	155	71797	39.94%
Cheering	152	51402	14.83%	Military-base	151	34728	11.69%
Dogs	146	71052	28.13%	Classroom	139	44777	15.50%
Birds	130	72836	35.18%	House-of-worship	130	17600	22.89%
US-Flags	120	17348	16.86%	Industrial-setting	117	20138	20.55%
Bicycling	109	94340	0.80%	Swimming	109	56342	88.60%
Construction-vehicles	106	78470	2.61%	Telephones	103	17419	28.80%
Desert	101	34127	37.56%	Eaters	91	51759	20.29%
Scientists	89	62186	12.48%	Soccer-player	88	89069	36.55%
Corporate-leader	73	50627	33.60%	Police-	70	55430	26.72%
Airplane-flying	66	80785	11.80%	Conference-room	62	48330	30.37%
Infants	62	50197	5.14%	Shopping-mall	61	33417	14.38%
Tent	54	24003	3.77%	Harbors	53	33379	15.42%
Horse	42	73560	7.16%	Greeting	39	53883	7.14%
Motorcycle	38	66783	2.27%	Bus	31	68980	67.51%
Natural-disaster	31	33558	0.46%	Throwing	31	53170	14.61%
Weather	26	17650	0.29%	Prisoner	25	58122	20.25%
Hospital	24	50067	0.23%	Basketball	19	87731	89.18%
Helicopter-hovering	18	70528	12.47%	Truck	17	68010	34.69%
Handshaking	15	55806	0.01%	Car-racing	14	101343	6.72%
Golf	13	102673	100.00%	Canoe	11	88917	36.81%
Court	8	59518	0.56%	Emergency-vehicles	7	79546	100.00%
Cows	6	87266	0.00%	Tennis	5	99160	100.00%