Visual Event Recognition

December 9, 2013

Supervisor: A/P Xu Dong

Examiner: A/P Cham Tat Jen

Presented by Gong Li

Introduction

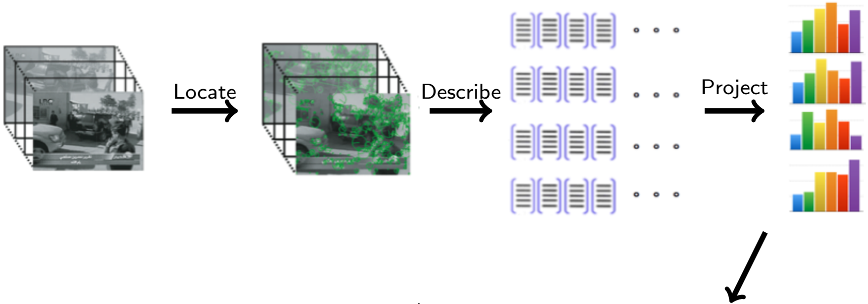
- Goal - recognize various videos based on their content

- Youtube statistics
  - 100 hours of videos are uploaded every minute
  - Over 6 billion hours of video are watched each month
- Vine and Instagram

- Currently, video search is heavily relying on texts associated with videos
- Explore other approaches to index and retrieve videos

Outline

Naive Bag of Words



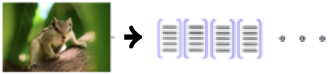
- M: number of sampled frames
- V: size of visual vocabulary

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1(V-1)} & A_{1V} \\ A_{21} & A_{22} & \cdots & A_{2(V-1)} & A_{2V} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{M(V-1)} & A_{MV} \end{pmatrix}$$

Better Bag of Words - Soft Assignment

- Straightforward approach [5]
  - Assign top N nearest words to each feature
  - Update the histogram by adding different weights to these N words
- Gaussian Mixture Models Assignment [1]

Training Videos → SIFT Features → Gaussian Mixture Models



$$\begin{matrix}[p_1, p_2, \cdots, p_K] \\ [p_1, p_2, \cdots, p_K] \\ \vdots \\ [p_1, p_2, \cdots, p_K]\end{matrix} + [P_1, P_2, \cdots, P_K]$$

Better Bag of Words - Weighting Schemes [8]

- Inverse document frequency of visual word t_i

$$idf(t_i) = \log(N/n_i) \tag{1}$$

  - N is the total number of images in the corpus
  - n_i is the number of images having visual word t_i

- Various different weighting schemes

Name	**Factors**	**Value for t_i**
bxx	*binary*	1 if t_i presents, 0 if not
txx	*tf*	tf_i
txc	*tf*, *normalization*	$\frac{tf_i}{\sum_i tf_i}$
tfx	*tf*, *idf*	$tf_i \cdot \log(^N/_{n_i})$
tfc	*tf*, *idf*, *normalization*	$\frac{tf_i \cdot \log(^N/_{n_i})}{\sum_i tf_i \cdot \log(^N/_{n_i})}$

Table : Weighting schemes for visual-word feature [8]

Aligned Space-Time Pyramid Matching [4] at level 0

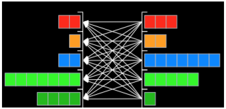
- Incorporate Earth Mover's Distance (EMD) [7]
- Given two videos P and Q

$$P = \{(p_1, 1/m), ..., (p_m, 1/m)\}, Q = \{(q_1, 1/n), ..., (q_n, 1/n)\}$$

- Solve the below optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij} \\ \text{subjective to} \quad & f_{ij} \geq 0 \qquad 1 \leq i \leq m, 1 \leq j \leq n \\ & \sum_{j=1}^{n} f_{ij} \leq 1/m \quad 1 \leq i \leq m \\ & \sum_{i=1}^{m} f_{ij} \leq 1/n \quad 1 \leq j \leq n \\ & \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = 1 \end{aligned} \tag{2}$$

- d_{ij}: distance between p_i and q_j
- f_{ij}: flow between p_i and q_i



- Distance between P and Q

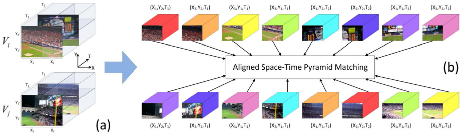
$$D_{PQ} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \tag{3}$$

Aligned Space-Time Pyramid Matching [4] at level l

- Each video is divided into 8^l non-overlapped sub-videos
- Given two videos P and Q

$$P = (p_1, \cdots, p_R), Q = (q_1, \cdots, q_R), \text{where } R = 8^l$$

Aligned Space-Time Pyramid Matching [4] at level l



1. Calculate pairwise distance matrix D
   - D_{ij} is the EMD distance between sub-video p_i and q_i
2. Align sub-videos in P with sub-videos in Q

$$\hat{F}_{ij} = \arg\min_{F_{ij}} \sum_{i=1}^{R} \sum_{j=1}^{R} F_{ij} D_{ij}$$

$$\text{subject to} \quad \sum_{i=1}^{R} F_{ij} = 1, \quad \forall j$$
$$\sum_{j=1}^{R} F_{ij} = 1, \quad \forall i \tag{4}$$

$$D_l(P, Q) = \frac{\sum_{i=1}^{R} \sum_{j=1}^{R} \hat{F}_{ij} D_{ij}}{\sum_{i=1}^{R} \sum_{j=1}^{R} F_{ij}} \tag{5}$$

Experiments of Aligned Space-Time Pyramid Matching

- Data set

	Wedding	**Sports**	**Show**	**Picnic**	**Parade**	**Birthday**	**Total**
Kodak	27	75	57	6	14	16	195

Table : Number of videos in each class from Kodak

- Kernel types

Kernel type	**Kernel function**
Gaussian	$\exp(-\gamma D^2(I_i, I_j))$
Laplacian	$\exp(-\sqrt{\gamma} D(I_i, I_j))$
ISD	$\frac{1}{\gamma D^2(I_i, I_j)+1}$
ID	$\frac{1}{\sqrt{\gamma} D(I_i, I_j)+1}$

- $D(I_i, I_j)$ represents the distance between I_i and I_j
- $\gamma = \frac{1}{A}$, A is the mean value of the squared distances between training samples
- Fused scores

$$f^{Fuse} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{1 + \exp(-f_i)} \tag{6}$$

Experiments of Aligned Space-Time Pyramid Matching

- Division of training and testing samples
  - randomly select 3 videos from each class as training samples
  - the rest of videos act as testing samples
- Evaluation metric: Mean Average Precision
- Experimental results at different levels using histograms built by naive bag of words

	Gaussian	**Laplacian**	**ISD**	**ID**	**Fused scores**
Level 0	44.38 ± 2.13	44.90 ± 2.73	44.01 ± 2.13	45.36 ± 3.13	44.33 ± 2.61
Level 1 (Unaligned)	43.08 ± 3.14	43.85 ± 3.84	43.22 ± 3.11	43.85 ± 3.56	43.55 ± 3.46
Level 1 (Aligned)	43.61 ± 2.97	43.40 ± 3.18	43.46 ± 2.97	43.22 ± 3.11	44.08 ± 3.25

Table : Means and standard deviations (percent) of MAPs at different levels

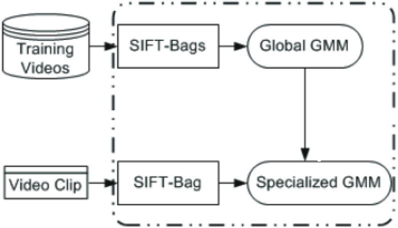
Experiments of Aligned Space-Time Pyramid Matching

- Experimental results using histograms built by better bag of words at level 0 distance

	Gaussian	Laplacian	ISD	ID	Fused scores
bxx	40.20 ± 2.57	38.35 ± 2.31	39.93 ± 2.58	38.23 ± 2.08	39.34 ± 2.55
txx	44.28 ± 2.14	44.90 ± 2.73	44.01 ± 2.13	45.36 ± 3.13	44.33 ± 2.61
txc	42.15 ± 4.73	45.01 ± 3.45	43.47 ± 4.56	45.38 ± 3.20	44.11 ± 3.90
tfx	43.76 ± 2.99	44.14 ± 3.36	43.61 ± 3.03	44.05 ± 3.51	44.18 ± 3.22
tfc	43.71 ± 1.37	$\mathbf{46.02 \pm 1.84}$	$\mathbf{44.93 \pm 1.64}$	$\mathbf{46.21 \pm 1.83}$	$\mathbf{45.28 \pm 1.62}$
Easy soft	43.54 ± 2.12	44.77 ± 2.41	43.52 ± 2.08	45.24 ± 2.47	44.79 ± 2.55
Gaussian soft	$\mathbf{44.77 \pm 2.80}$	45.23 ± 2.76	44.90 ± 3.01	45.23 ± 2.87	$\mathbf{45.20 \pm 3.04}$

Table : Means and standard deviations (percent) of MAPs using different mechanisms to build histograms at level 0

Gaussian Mixture Models to Represent Videos [10]



- Global GMM built from training data
- Specialized GMM by adaption

- Specialized GMM of video P and Q

$$P = (\mu_1^p, \cdots, \mu_K^p), Q = (\mu_1^q, \cdots, \mu_K^q)$$

- Given global GMM as $\Theta = \{w_1, \mu_1, \Sigma_1, \cdots\}$, the distance of P and Q

$$d(P, Q) = \frac{1}{2} \sum_{k=1}^{K} w_k (\mu_k^p - \mu_k^q)^T \Sigma_k^{-1} (\mu_k^p - \mu_k^q) \tag{7}$$

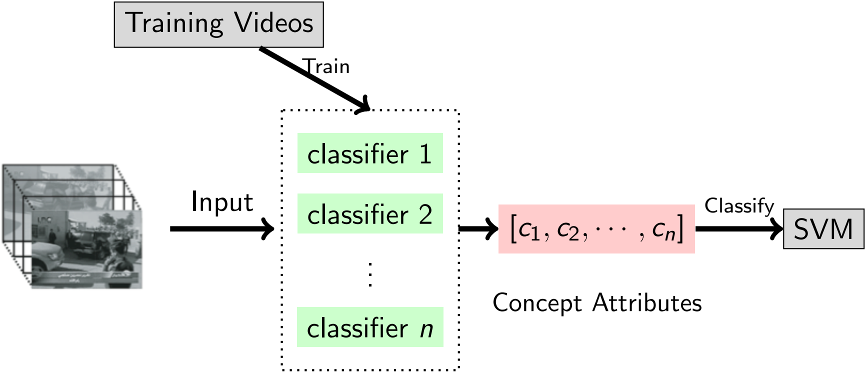
Experiments on GMM

	Gaussian	Laplacian	ISD	ID	Fused scores
spherical 128	24.70 ± 1.41	43.04 ± 1.61	26.92 ± 1.00	**43.64 ± 0.96**	32.91 ± 2.20
spherical 64	23.99 ± 1.40	42.35 ± 1.64	25.62 ± 1.11	43.42 ± 1.18	29.01 ± 1.10
full 128	25.69 ± 7.57	21.39 ± 7.32	26.49 ± 8.38	21.93 ± 7.75	21.79 ± 7.29
full 64	25.23 ± 0.94	29.69 ± 1.81	25.68 ± 1.34	30.74 ± 1.67	26.74 ± 1.63

Table : Means and standard deviations (percent) of MAPs using different GMMs

- Spherical covariance using ID kernel performed the best
- Spherical covariance performed better than full covariance

Concept Attributes to Represent Videos [6]



1. Recognize complex events
2. Make use of pre-trained detectors (even by others)

Experiments on Concept Attributes

	Wedding	Sports	Show	Picnic	Parade	Birthday	Total
Kodak	27	75	57	6	14	16	195
Youtube	91	260	200	85	119	151	906

Table : Number of videos in each class from Kodak and Youtube

	Recognition Accuracy
Kodak $\rightarrow$ Kodak	38.5 ± 12.7
Youtube $\rightarrow$ Kodak	30.0 ± 6.9
Baseline	41.6 ± 11.5

Table : Means and standard deviations (percent) of recognition accuracies

Compress Videos

1. Run K-Means on sampled frames of a video
2. Build a graph for each cluster
3. Choose representative frames from each graph

Experiments on Compressed Videos

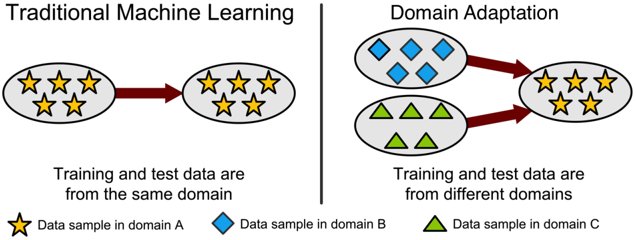
- Size of Youtube videos is compressed from 4.42 GB to 3.17 GB. (28.41% reduced)

Training videos	**Testing videos**	**Original videos**	**Compressed videos**
60	846	38.9 ± 2.9	38.6 ± 2.8
120	786	45.7 ± 2.2	44.5 ± 1.6
180	726	49.5 ± 1.8	48.3 ± 1.9
240	666	52.0 ± 2.1	50.6 ± 2.1

Table : Means and standard deviations (percent) of MAPs over six events

1. Compressed videos performed slightly worse than original videos
2. With more training samples, the performance increases

Domain Adaptations



- Feature Replication [2] (FR)
- Adaptive Support Vector Machine [9] (A-SVM)
- Domain Transfer Support Vector Machine [3] (DTSVM)
- Adaptive Multiple Kernel Learning [4] (A-MKL)

Feature Replication (FR)

- Mapping functions to augment samples $\{x\}$ from different domains

$$\Phi^T(\mathbf{x}) = (\mathbf{x}, \mathbf{x}, \mathbf{0}), \quad \Phi^A(\mathbf{x}) = (\mathbf{x}, \mathbf{0}, \mathbf{x}) \tag{8}$$

- Kernelized meaning of the above augmenting
  - If x_i and x_j come from the same domain,

$$\begin{aligned} \hat{K}(x_i, x_j) &= \theta(x_i)^T \cdot \theta(x_j) + \theta(x_i)^T \cdot \theta(x_j) \\ &= 2K(x_i, x_j) \end{aligned} \tag{9}$$

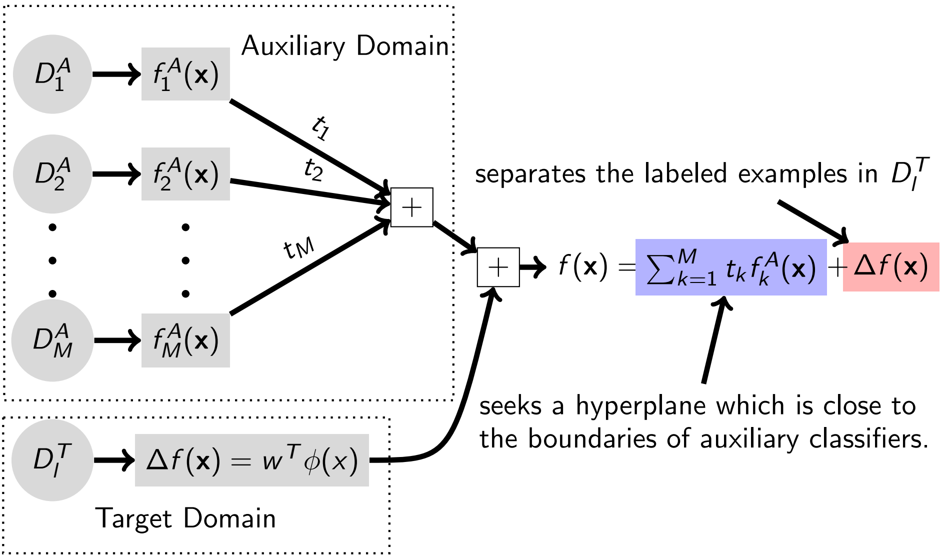
  - If x_i and x_j come from different domains,

$$\begin{aligned} \hat{K}(x_i, x_j) &= \theta(x_i)^T \cdot \theta(x_j) \\ &= K(x_i, x_j) \end{aligned} \tag{10}$$

- To summarize,

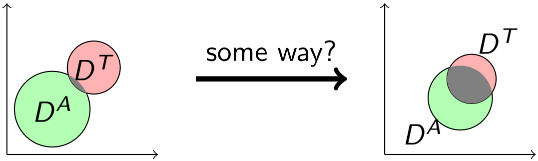
$$\hat{K}(x_i, x_j) = \begin{cases} 2K(x_i, x_j) & \text{if } x_i \text{ and } x_j \text{ come from the same domain} \\ K(x_i, x_j) & \text{otherwise} \end{cases}$$

Adaptive Support Vector Machine (A-SVM)

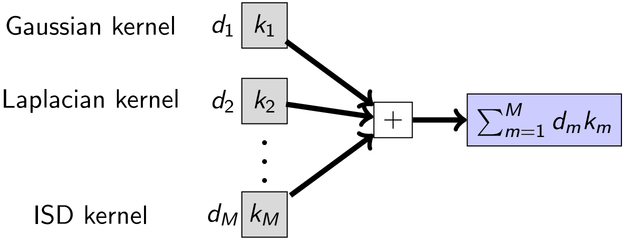


Domain Transfer Support Vector Machine (DTSVM)

- Some way to reduce the difference in distribution of D^T and D^A



- Seek a special kernel function $\varphi(x)$ which minimizes the difference
  - Find the optimal weights $(d_1, d_2, ..., d_M)$ of multiple kernels



- Define the mismatch between D^A and D^T as

$$DIST_k(\mathcal{D}^A, \mathcal{D}^T) = \| \frac{1}{n_A} \sum_{i=1}^{n_A} \varphi(x_i^A) - \frac{1}{n_T} \sum_{i=1}^{n_T} \varphi(x_i^T) \| \qquad (11)$$

- Simplify the square of equation (4) to

$$DIST_k^2(\mathcal{D}^A, \mathcal{D}^T) = \mathrm{tr}(\mathbf{KS}) \qquad (12)$$

where $\mathbf{s} = [\underbrace{\frac{1}{n_A}, ..., \frac{1}{n_A}}_{n_A}, \underbrace{\frac{-1}{n_T}, ..., \frac{-1}{n_T}}_{n_T}]^T$, $\mathbf{S} = \mathbf{ss^T}$, $\mathbf{K} = \begin{bmatrix} K^{A,A} & K^{A,T} \\ K^{T,A} & K^{T,T} \end{bmatrix}$

- Incorporate $\mathbf{d} = [d_1, d_2, ..., d_M]^T$ into equation (5)

$$DIST_k^2(\mathcal{D}^A, \mathcal{D}^T) = \Omega(\mathbf{d}) = \mathbf{h^T d} \qquad (13)$$

where $\mathbf{h} = [tr(\mathbf{K_1 S}), \cdots, tr(\mathbf{K_M S})]^T$, and $\mathbf{K_m} = [\varphi(x)^T \varphi(x)]$ is the mth base kernel matrix

- Optimization problem of DTSVM:
  1. Distribution mismatch
  2. SVM structural risk function

$$\text{minimize} \quad G(\mathbf{d}) = \boxed{\tfrac{1}{2}\Omega^2(\mathbf{d})} + \theta \boxed{J(\mathbf{d})} \tag{14}$$

where $J(\mathbf{d}) = \max_{\boldsymbol{\alpha}} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j (\sum_{m=1}^{M} d_m\, \varphi_m(x_i)^T \varphi_m(x_j))$

- Iteratively update coefficient $\mathbf{d}$ and the dual variable $\boldsymbol{\alpha}$
  1. Update the dual variable $\boldsymbol{\alpha}$
  2. Update the coefficient $\mathbf{d}$ using gradient descent method

$$\mathbf{d}_{t+1} = (1-\eta_t)\mathbf{d}_t + \eta_t \mathbf{d}_t^{new} \tag{15}$$

where $\mathbf{d}_t^{new} = \theta(\mathbf{h}\mathbf{h}^{\mathbf{T}} + \varepsilon \mathbf{I}_{\mathbf{M}})^{-1}\mathbf{q}$, $\mathbf{q} = [\frac{1}{2}(\boldsymbol{\alpha}_t \diamond \mathbf{y})^T \mathbf{K}_1(\boldsymbol{\alpha}_t \diamond \mathbf{y}), \cdots, \frac{1}{2}(\boldsymbol{\alpha}_t \diamond \mathbf{y})^T \mathbf{K}_M(\boldsymbol{\alpha}_t \diamond \mathbf{y})]$,
η_t is the learning rate.

- Final decision function

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{m=1}^{M} d_m \mathbf{K}_m(x_i, x) \right) + b \tag{16}$$

Adaptive Multiple Kernel Learning [4] (A-MKL)

- Adaptive SVM

$$f(x) = \sum_{k=1}^{M} t_k f_k^A(x) + \Delta f(x) \tag{17}$$

- Domain Transfer SVM

$$f(x) = \sum_{m=1}^{M} d_m w_m^T \varphi_m(x) + b \tag{18}$$

- Adaptive MKL

$$f(x) = \sum_{p=1}^{P} \beta_p f_p(x) + \sum_{m=1}^{M} d_m w_m^T \varphi_m(x) + b \tag{19}$$

  - Seek a hyperplane which is close to that of all labeled samples
  - Reduce the mismatch of different domains

Experiments of Domain Adaptation Approaches

- Data set

	Wedding	Sports	Show	Picnic	Parade	Birthday	Total
Kodak	27	75	57	6	14	16	195
Youtube	91	260	200	85	119	151	906

Table : Number of videos in each class from Kodak and Youtube

- Distances using various approaches

Setting Name	Content
MAP(1)	Level 0 distance in Aligned Space-Time Pyramid Matching
MAP(2)	unaligned Level 1 distance in Aligned Space-Time Pyramid Matching
MAP(3)	aligned Level 1 distance in Aligned Space-Time Pyramid Matching
MAP(4)	Level 0 distance using histograms built in "tfc" weighting scheme
MAP(5)	Level 0 distance using histograms built by straightforward soft assignment
MAP(6)	Level 0 distance using histograms built by GMM soft assignment
MAP(7)	distances calculated by specialized GMMs built on 128 dimensional SIFT features with spherical covariance
MAP(8)	two set of distances: MAP(3) + MAP(6)

Table : Experimental distance set

- Base kernel matrices

Kernel type	Kernel function
Gaussian	$\exp(-\gamma D^2(I_i, I_j))$
Laplacian	$\exp(-\sqrt{\gamma} D(I_i, I_j))$
ISD	$\frac{1}{\gamma D^2(I_i,I_j)+1}$
ID	$\frac{1}{\sqrt{\gamma} D(I_i,I_j)+1}$

- $D(I_i, I_j)$ represents the distance between I_i and I_j
- $\gamma = 2^l \gamma_0$
  - $\gamma_0 = \frac{1}{A}$, A is the mean value of the squared distances between training samples
  - $l \in (-3, -2, \cdots, 1)$
- 4×5 combinations $\rightarrow$ 20 base kernel matrices

- Division of training and testing samples
  - 3 videos for each class in Kodak domain as $\mathcal{D}_l^T$, and the left
  - the left Kodak videos as $\mathcal{D}_u^T$
  - all Youtube videos as fully labeled $\mathcal{D}^A$
- Evaluation metric: Mean Average Precision

Experimental Results

	SVM_T	SVM_AT	FR	A-SVM	DTSVM	A-MKL
MAP(1)	44.33 ± 2.61	52.21 ± 2.54	52.33 ± 2.20	47.03 ± 3.26	47.14 ± 3.26	54.29 ± 2.21
MAP(2)	43.55 ± 3.46	55.37 ± 2.26	55.95 ± 3.79	45.86 ± 4.39	50.97 ± 1.38	54.26 ± 3.46
MAP(3)	44.08 ± 3.25	57.56 ± 3.02	53.91 ± 1.48	45.42 ± 3.62	53.32 ± 2.56	$\mathbf{57.45 \pm 1.64}$
MAP(4)	45.27 ± 1.63	51.83 ± 2.27	52.55 ± 2.00	45.94 ± 1.70	52.31 ± 2.56	53.05 ± 2.21
MAP(5)	44.79 ± 2.55	47.80 ± 1.67	51.89 ± 1.99	47.41 ± 3.13	45.05 ± 4.07	51.08 ± 2.87
MAP(6)	45.20 ± 3.04	56.90 ± 2.79	54.03 ± 4.02	46.62 ± 3.14	53.41 ± 3.29	$\mathbf{59.16 \pm 3.38}$
MAP(7)	32.91 ± 2.20	33.15 ± 1.78	41.78 ± 3.98	37.07 ± 3.52	$\mathbf{46.61 \pm 2.41}$	35.88 ± 1.98
MAP(8)	44.69 ± 2.84	60.21 ± 1.94	55.29 ± 3.00	46.28 ± 4.23	57.01 ± 2.45	$\mathbf{61.40 \pm 1.91}$

Table : Means and standard deviations (percent) of MAPs over six events

1. In all cases, SVM_AT performed better than SVM_T
2. Gaussian soft assignment outperformed the other approaches
3. DTSVM performed amazingly in MAP(7)
4. A-MKL performed the best by selecting two distance matrices smartly

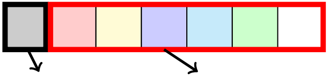
Recognition Accuracy as Evaluation Metric

- Recognition accuracy is defined as

$$\text{recognition accuracy} = \frac{\text{correct predictions}}{\text{number of testing samples}} \tag{20}$$

- SVM multi-class classification

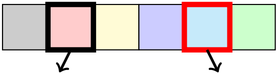
one-vs-all



positive negative

n classes $\rightarrow$ n classifiers

one-vs-one



positive negative

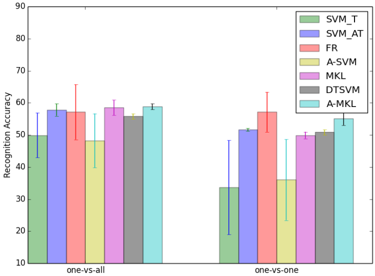
n classes $\rightarrow \frac{n(n-1)}{2}$ classifiers

	SVM_T	SVM_AT	FR	A-SVM	DTSVM	A-MKL
one-vs-all	49.94 ± 6.96	57.85 ± 1.91	57.18 ± 8.59	48.25 ± 8.43	55.82 ± 0.83	$\mathbf{58.87 \pm 0.90}$
one-vs-one	33.67 ± 14.67	51.64 ± 0.45	57.18 ± 6.21	36.05 ± 12.62	50.85 ± 0.80	55.14 ± 2.10

Table : Means and standard deviations (percent) of average recognition accuracies

	SVM_T	SVM_AT	FR	A-SVM	DTSVM	A-MKL
one-vs-all	0.98	8.54	10.53	12.52	11.33	27.87
one-vs-one	1.49	4.08	5.68	7.16	5.05	10.94

Table : Average running time (seconds)



- One-vs-all outperformed one-vs-one
- Trade-offs between running time and accuracy
  1. One-vs-all requires more time
  2. Domain adaptations require more time

Conclusion

- What have been done in this FYP
  1. Successfully implemented a recognition system to recognize videos
  2. Explored 4 various approaches to recognize videos
  3. Studied and implemented 4 domain adaptation methods to boost the performance
  4. Designed and developed a web application to demonstrate the work
- Future recommendations
  1. Incorporate more types of features: space-time feature and acoustic feature
  2. Employ more attribute detectors
  3. Combine various domain adaptation methods

Demo

- A. Agarwal and B. Triggs, "Hyperfeatures–multilevel local coding for visual recognition," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 30–43.
- H. Daumé III, "Frustratingly easy domain adaptation," in *ACL*, vol. 1785, no. 1786, 2007, p. 1787.
- L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1375–1381.
- L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 9, pp. 1667–1680, 2012.
- Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval,"

in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 494–501.

J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. S. Sawhney, "Video event recognition using concept attributes." in *WACV*, 2013, pp. 339–346.

Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM, 2007, pp. 197–206.

J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 188–197.

X. Zhou, X. Zhuang, S. Yan, S.-F. Chang, M. Hasegawa-Johnson, and T. S. Huang, "Sift-bag kernel for video event analysis," in *Proceedings of the 16th ACM international conference on Multimedia*. ACM, 2008, pp. 229–238.