# ITI-CERTH in TRECVID 2015 Multimedia Event Detection

Christos Tzelepis, Damianos Galanopoulos, Stavros Arestis-Chartampilas, Nikolaos Gkalelis, Vasileios Mezaris

Information Technologies Institute / Centre for Research and Technology Hellas

TRECVID 2015 Workshop, Gaithersburg, MD, USA, November 2015

# Highlights

- For detecting events without training examples
  - Use web resources such as Google search and Wikipedia to enrich the textual information of visual concepts
- For learning from training examples, use KSDA+LSVM
  - Greatly reduces feature dimensionality
  - Achieves KSVM precision at a fraction of state-of-the-art KSVM time (1-2 orders of magnitude faster)
  - GPU version (not used in this year's MED experiments): further time reduction, much faster than state-of-the-art Linear SVM
- For learning from very few positive training examples, use Relevance Degree SVM (RDSVM)
  - Exploits "near-miss" samples, by assigning a relevance degree to each training sample

# Video representation

- Three kinds of descriptors
  - **Static visual features**
    - Local descriptors (SIFT, OpponentSIFT, RGB-SIFT, RGB-SURF) from 1 keyframe/6 sec, VLAD encoding, random projection (results in 16.000-element feature vector); averaging the feature vectors of all keyframes of the video
  - **Motion features**
    - Improved dense trajectories, Fisher vector encoding (feature vector in $\mathbb{R}^{101376}$)
  - **DCNN-based features**
    - 16-layer pre-trained DCNN (16-layer deep ConvNet network) applied on 2 keyframes/sec of video; the two last hidden layers (fc7, fc8) and the output are averaged across all keyframes to represent the video

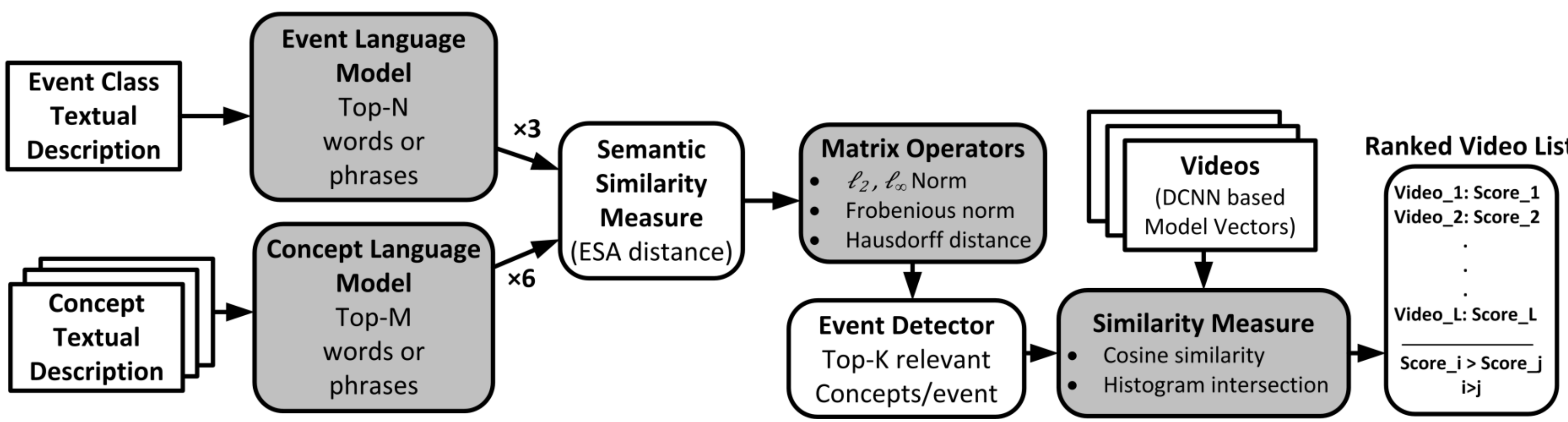
# 000Ex task: system overview

- Fully automatic system
- Links textual information with the visual content using
  - The textual descriptions from the event kits
  - A pool of 1000 concepts along with their titles and subtitles
  - A pre-trained detector (16-layer deep ConvNet pre-trained on the ImageNet data) for these concepts
- Visual modality only

# 000Ex task: system overview

Algorithm:

1. Create Event Language Model (ELM)
2. Create Concept Language Model (CLM)
3. Calculate semantic similarity between every ELM and every CLM
4. Find the most relevant visual concepts per event (event detector)
5. Calculate the distances between event detector and each video's model vector (concept detectors output scores)



# 000Ex task: language models

- Event Language Model
  - Top-N words or phrases most closely related to an event
  - Three types of ELMs (depending on the information used)
    - Title of the target event
    - Title AND visual cues of the target event
    - Title AND visual cues AND audio cues of the target event
- Concept Language Model
  - Top-M words or phrases most closely related to a visual concept
  - Three different information sources
    - Title and subtitles of the visual concept
    - Top-20 articles returned by Google Search (searching by concept title, subtitles)
    - Top-20 articles returned from Wikipedia (searching by concept title, subtitles)
  - Bag-of-Words approach in these corpora, using two weighting techniques (Tf-Idf; no weighting), leads to six different CLMs

# 000Ex task: event detector

- Semantic similarity between concepts and events
  - Each ELM and CLM is a ranked list of words
  - For an ELM, CLM pair, calculate the Explicit Semantic Analysis (ESA) measure between each word in the ELM and each word in the CLM → $N * M$ matrix $S$ with scores
- Building an event detector
  - Transform each matrix $S$ to a scalar value
    - Use one of: $\ell_2$ norm; $\ell_\infty$ norm; Frobenious norm; Hausdorff distance
    - In all cases scores normalized to [0,1]
  - The 1000 concepts of our concept pool are ordered in descending order
  - The top-K concepts and corresponding weights constitute our event detector

# 000Ex task: event detection

- Matching videos to an event detector
  - Each video is represented in $\mathbb{R}^{1000}$ using the DCNN-based concept detector output scores (model vector)
  - The scores for the *K* event-specific concepts (normalized to [0,1]) are retained
  - **Cosine similarity** and **histogram intersection** distances are used as distance functions; the videos are ordered according to distance (in ascending order) for each event

# 010Ex, 100Ex tasks: overview

- Our runs are based on KSDA and RDKSVM methods.
- Our KSDA method:
  - Tackles the problem of high dimensionality
  - Uses all available features: required to get a good video description
  - Is very fast to train: can be cross-validated thoroughly
- Our RDKSVM method:
  - Tackles the lack of sufficient number of positive training samples
  - Uses related (“near-miss”) videos as weighted positive or negative to extend the training set

# KSDA+LSVM

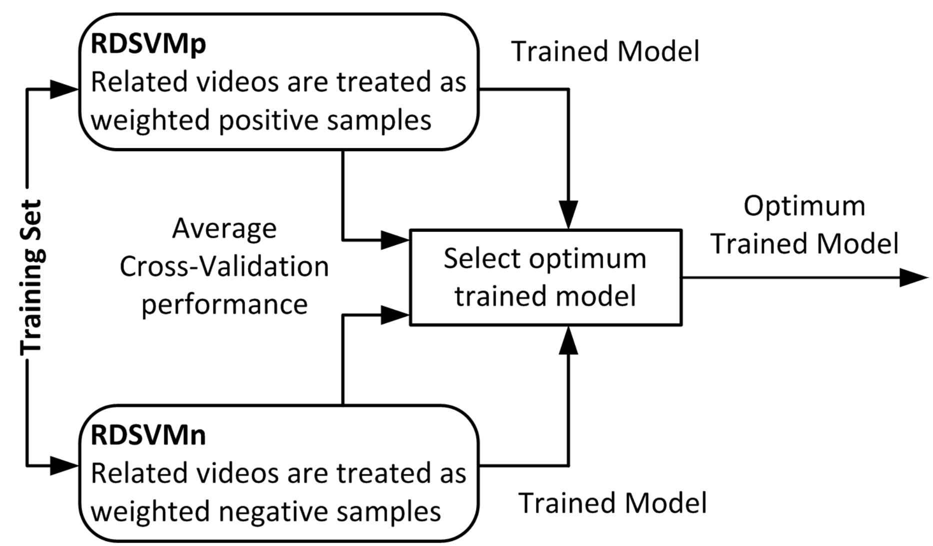
- Partition a training set $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{L \times N}$ in sub-classes, where $\mathbf{X}_{i,j}$ contains the samples of the $j$th subclass of class $i$
- Use a vector-valued function $\phi(\cdot): \mathbb{R}^L \to \mathbb{R}^F$, $\phi = \phi(\mathbf{x})$ as a kernel (map data from the input space to a higher-dimensional space): $\phi_r^\top \phi_q = k(\mathbf{x}_r, \mathbf{x}_q) = k_{r,q}$
- AGSDA seeks the coefficient matrix $\mathbf{\Gamma} \in \mathbb{R}^{N \times D}$ solving $\mathbf{KAK\Gamma} = \mathbf{KK\Gamma\Delta}$ (1):
  - $\mathbf{K} = \Phi^\top\Phi$, with $\mathbf{K} \in \mathbb{R}^{N \times N}$ being the Gram matrix. $\mathbf{\Delta} \in \mathbb{R}^{D \times D}$ $(D \ll F)$ is a diagonal matrix with the eigenvalues of the generalized eigenvalue problem in (1) on its main diagonal
  - $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the between subclass factor matrix

$$A_{r,q} = \begin{cases} p_{i,j}(1-p_i)/(N_{i,j}N_{i,j}), & \text{if } (i,j) = (k,l), \\ 0 & \text{if } i = k, j \neq l, \\ -p_{i,j}p_{k,l}/(N_{i,j}N_{k,l}), & \text{otherwise,} \end{cases}$$

- Each element $A_{r,q}$ corresponds to samples $\mathbf{x}_r \in \mathbf{X}_{i,j}$ and $\mathbf{x}_q \in \mathbf{X}_{k,l}$ where:
  - $p_i$, $p_{i,j}$ are the estimated priors of $i$th class and $(i,j)$th subclass
  - $N_{i,j}$ is the number of samples of $(i,j)$th subclass
- The problem above can be solved by:
  - Identifying the eigenpairs ($\mathbf{V} \in \mathbb{R}^{N \times D}$, $\mathbf{\Lambda} \in \mathbb{R}^{D \times D}$) of $\mathbf{A}$,
  - Solving $\mathbf{K\Gamma} = \mathbf{V}$ for $\mathbf{\Gamma}$

# RDKSVM

- Relevance Degree SVM (RDSVM) extends the standard SVM formulation such that a relevance degree can be assigned to each training sample
  - Relevance degree is a confidence value indicating the relevance of each sample with its respective class
  - It is used to exploit “near-miss” samples
- All “near-miss” samples are assigned with one global relevance degree, optimized with cross-validation during training
  - Considering the samples both as if they were all weighted positive and weighted negative
  - Automatically decide a global relevance degree for all samples



# 000Ex: experiments

- 72 different event detectors: 3 ELMs x 6 CLMs x 4 matrix operators
- Based on experiments on previous MED datasets, two detectors are chosen:
  - The best of the 72 (*best detector*)
  - A new one created by fusion of the top-10 (fusion of concept lists & averaging of weights) (*top-10 detector*)
- 5 submitted runs
  - **c-1oneCosine:** The *best detector*; cosine similarity
  - **c-2avgCosine**: The *top-10 detector*; cosine similarity
  - **c-3oneHist:** The *best detector*; histogram intersection
  - **c-4avgHist:** The *top-10 detector*; histogram intersection
  - **p-1Fusion:** The late fusion (arithmetic mean) of the results of the above four runs

# 000Ex: results & conclusions

- The fusion of the top-10 detectors, combined with histogram intersection, gives a boost to performance
- Late fusion of scores leads to better detection results

| Run ID | mInfAP@200 |
|---|---|
| p-1Fusion | **0.0617** |
| c-1oneCosine_1 | 0.0478 |
| c-2avgCosine_1 | 0.0473 |
| c-3oneHist_1 | 0.0474 |
| c-4avgHist_1 | 0.0592 |

# 010Ex, 100Ex: experiments & results

- 4 submitted runs
  - **c-1KDALSVM**: Based on KSDA+LSVM, using visual, motion and fc7+fc8 DCNN descriptors
  - **c-2RDKSVM:** Based on RDKSVM, using fc8 DCNN descriptors
  - **c-3RDKSVM:** Based on RDKSVM, using fc7+fc8 DCNN descriptors
  - **p-1Fusion**: Late fusion of all the above

(b) 010Ex

| Run ID | mInfAP@200 |
|---|---|
| p-1Fusion | 0.211 |
| c-1KDALSVM | **0.2493** |
| c-2RDKSVM | 0.1588 |
| c-3RDKSVM | 0.2026 |

(c) 100Ex

| Run ID | mInfAP@200 |
|---|---|
| p-1Fusion | 0.3649 |
| c-1KDALSVM | **0.4111** |
| c-2RDKSVM | 0.2894 |
| c-3RDKSVM | 0.2367 |

# 010Ex, 100Ex: conclusions

- In both training conditions, our KSDA+LSVM method achieved the best results (24.93% and 41.11%, respectively), compared to RDSVM, late fusion of multiple runs
  - The use of all features (DCNN, dense trajectories, static visual) makes the difference
- The runs that exploited “near-miss” samples using RDSVM achieve better results than what traditional SVM would achieve using the same features
  - Approximately +4,5%, based on non-submitted experiments
- Our run based on KSDA+LSVM, using all the features (run c-1KDALSVM) achieved mInfAP@200=0.4111: second-best result among all participants' runs on the MED15-EvalSub set

# 010Ex, 100Ex: conclusions

- KSDA+LSVM allows for very fast learning from high-dimensional data and increased accuracy, compared to SVM
- RDSVM can exploit “near-miss” videos, but at present there are limitations in feature vector dimensions (cannot be used with very high-dimensional data)

# Questions?

More information and contact:
Vasileios Mezaris, http://www.iti.gr/~bmezaris, bmezaris@iti.gr

KSDA+LSVM software for download: http://mklab.iti.gr/project/gpu-agsda

**TRECVID 2015 paper:**
F. Markatopoulou, A. Ioannidou, C. Tzelepis, T. Mironidis, D. Galanopoulos, S. Arestis-Chartampilas, N. Pittaras, K. Avgerinakis, N. Gkalelis, A. Moumtzidou, S. Vrochidis, V. Mezaris, I. Kompatsiaris, I. Patras, "ITI-CERTH participation to TRECVID 2015", Proc. TRECVID 2014 Workshop , Gaithersburg, MD USA, November 2015.