The INRIA-LIM-VocR and AXES submissions to Trecvid 2014 Multimedia Event Detection

Matthijs Douze, Dan Oneata, Mattis Paulin, Clément Leray, Nicolas Chesneau, Danila Potapov, Jakob Verbeek, Karteek Alahari, Lori Lamel, Jean-Luc Gauvin, et al.

The INRIA-LIM-VocR and AXES submissions to TrecVid 2014 Multimedia Event Detection

Matthijs Douze[1], Dan Oneata[1], Mattis Paulin[1], Clément Leray[1], Nicolas Chesneau[1], Danila Potapov[1], Jakob Verbeek[1], Karteek Alahari[1], Lori Lamel[2], Jean-Luc Gauvin[2], Christoph Andreas Schmidt[3], Cordelia Schmid[1]

[1]Inria* – email: firstname.lastname@inria.fr
[2]Spoken Language Processing group, LIMSI, CNRS – email: lastname@limsi.fr
[3]Fraunhofer Sankt Augustin – email: Christoph.Andreas.Schmidt@iais.fraunhofer.de

Abstract

This paper describes our participation to the 2014 edition of the TrecVid Multimedia Event Detection task. Our system is based on a collection of local visual and audio descriptors, which are aggregated to global descriptors, one for each type of low-level descriptor, using Fisher vectors. Besides these features, we use two features based on convolutional networks: one for the visual channel, and one for the audio channel. Additional high-level features are extracted using ASR and OCR features. Finally, we used mid-level attribute features based on object and action detectors trained on external datasets. Our two submissions (INRIA-LIM-VocR and AXES) are identical in terms of all the components, except for the ASR system that is used. We present an overview of the features and the classification techniques, and experimentally evaluate our system on TrecVid MED 2011 data.

1 Introduction

Our participation to the 2014 edition of the TrecVid Multimedia Event Detection task [22] included two runs, namely, INRIA-LIM-VocR and AXES, with the difference between them being the ASR system that is used: one provided by the CNRS LIMSI laboratory, and the other by the Fraunhofer Institute. The low-level features we used in 2014 are similar to the ones we used in 2013 [2]. The various types of local visual and audio descriptors are aggregated to global descriptors, one for each type of low-level descriptor, using Fisher vectors [25]. This year, in addition to these features, we used two types of features learned with convolutional networks: one for the visual channel and one for the audio channel.

The task with 10 training examples was the mandatory one in this year's edition — a task in which the use of external training data to produce "attribute" descriptors is more attractive as compared to the 100 training example setting. To this end, we used attribute-based descriptors in our system, which is another novelty in comparison to our last year's submission. Figure 1 provides a schematic overview of our event detection system.

The rest of this paper is organized as follows. In Section 2 we give a detailed description of the features that are used in our system. Section 3 presents the classification techniques we use to map the features to event detection scores. We present experimental results for the TrecVid MED 2011 dataset in Section 4. Finally, we present our conclusions in Section 5.

2 Feature extraction for multimedia event detection

We extract a collection of features for each video, which provide a set of high-dimensional signatures of the video. The features belong to three categories:

- Low-level descriptors that do not rely on supervised training.
- Mid-level attribute descriptors that use a supervised training stage to obtain a signature in terms of confidence scores for various concepts (e.g., for object presence, or action detection).
- High-level textual descriptors based on optical character recognition (OCR) and automatic speech recognition (ASR) that output semantically meaningful text features, rather low-level signatures, or mid-level visual concept detections.

*LEAR team, Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.

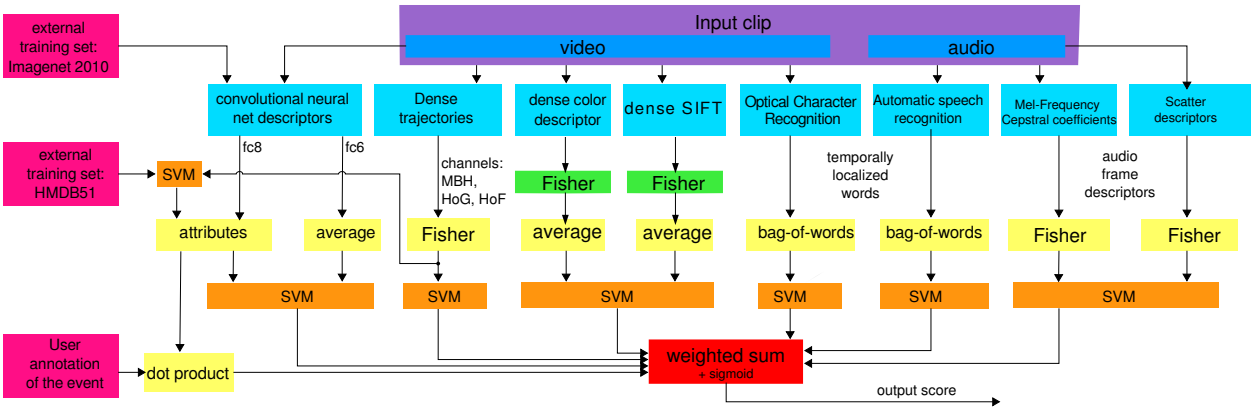

Figure 1: Schematic overview of the processing of a video clip for MED, for the 10-example case. For the 100-example case we used the same system with minor modifications, see text in Section 4.2 for details.

Descriptor	Dimension	Real-time factor
Dense trajectories	434,176	7.1
SIFT	276,479	3.1
Color	72,703	2.6
MFCC	80,895	0.04
Scatter transform	65,663	0.18
CNN	4,096	0.33
Attributes	2,102	(7.43)
OCR	110,000 (sparse)	1.5
ASR LIMSI	110,000 (sparse)	2
ASR Fraunhofer	110,000 (sparse)	1.1

Table 1: Descriptor dimension and processing time as a factor w.r.t. video duration on one CPU core. The real-time factor in parentheses (for the descriptor Attributes) is derived from other features at a negligible additional cost.

This year, we included two types of features based on multi-layer architectures, one for images and one for audio. The visual one is a type of mid-level feature, while the audio one is a low-level feature.

In Table 1, we provide a brief overview of these features, listing their final signature dimension, and run-time as a real-time factor. The real-time factor expresses how long the computation takes on a single CPU core with respect to the video length: a factor two means that two minutes of computation are needed for one minute of video.

2.1 Low-level features

The low-level features we used in 2014 are similar to the ones from 2013 [2]. For each type of low-level feature, we aggregate the local descriptors into a global signature by means of a Fisher vector (FV) [25]. The number of Gaussians chosen for the FVs are a trade-off between the accuracy of the representation and computational constraints. Visual frame-based FVs are averaged together to produce a signature for the complete video. We used the following visual low-level features:

- **Dense trajectories**: We used the improved dense trajectory features of [27]. We re-scale the videos to be at most 320 pixels wide, and skip every second frame when decoding the video. The four descriptors (MBHx, MBHy, HOG, and HOF) are reduced to half their original dimension by performing PCA, and each is encoded with a FV based on a vocabulary of 512 Gaussians. We also included the spatial Fisher vector of [12] and a horizontal binning into three regions [20] to encode the spatio-temporal layout of the low-level features.

- **SIFT:** We extract SIFT features [17] on a dense multi-scale grid, and encode these in a FV using a vocabulary of size 2,048. The 128 dimensional SIFT descriptors are first projected to 64 dimensions using PCA. We extract SIFT on every 60-th frame. The video signature is then the average of these FVs.

- **Color:** We extract color features based on local mean and variance of the color channels [5] every 60-th frame, and encode them in a single FV with a vocabulary size 1,024. The 96 dimensional color descriptors are first projected to 32 dimensions using PCA. Similar to SIFT, these color descriptors computed at the frame level are averaged to form a descriptor for the video. We also included the spatial Fisher vector of [12] to encode the spatial layout of these low-level features.

In addition to these three visual features, we used the following low-level audio features:

- **MFCC:** We down-sample the original audio track to 16 kHz with 16 bit resolution and then compute Mel-frequency cepstral coefficients (MFCC) with a window size of 25 ms and a step-size of 10 ms, keeping the first 12 coefficients of the final cosine transformation and the energy of the signal. We enhance the MFCCs with their first and second order derivatives. The MFCC features are then aggregated into a FV with a vocabulary size of 256.

- **Scattering transform:** We resample the audio track to 44.1 kHz and then compute scattering coefficients [3] with a window size of 500 ms and a step size of 185 ms. The scattering transform is based on several layers of a modified wavelet decomposition. It is designed to capture longer-range audio structures as compared to the standard MFCC descriptor. We used first and second order coefficients, with quality factors $Q_1 = 8$ and $Q_2 = 1$, which results in 526 dimensions. The scattering coefficients are then aggregated into a FV with a vocabulary size of 128.

2.2 Mid-level features

In order to cope with the restricted positive training data, we implemented mid-level representations. These representations rely on detectors trained for a set of object and action classes that are not directly related to the MED events. The classes are chosen such that they are basic, and a sufficient amount of training data is available to train classifiers for them. For example, the action "stand up" is more basic than the event "townhall meeting". This is inspired by similar representations used for attribute-based and zero-shot image classification [1]. The mid-level feature vector of a video clip is built from the confidence scores of the clip for each of the chosen classes. In the case of the CNN features described below, we do not directly use the detection confidences, but rather an internal representation that is used by the convolutional network to detect object classes. The three mid-level representations we used are:

- **HMDB51 attributes:** HMDB51 [13] is a dataset of 7,000 video clips of 51 basic action classes (like "dive", "jump"). We compute dense trajectory features, aggregated with a FV, and classify them with an SVM to produce the attribute scores.

- **ImageNet attributes:** The training set of ImageNet 2010 [24] is a dataset of 1.2 million images, each prominently representing one object from a total of 1,000 object classes (such as "bear", "hook", "restaurant"). We used the classification scores produced as output of the 8th layer of a convolutional neural network trained on this data with the Caffe software package [**?**]. This results in a 1,000 dimensional feature vector representing class confidences.

- **CNN:** We extract another 4,096 dimensional feature using the same convolutional network trained on the ImageNet 2010 data. This feature is obtained from the layer six of the network, after applying the linear rectification which clips negative values to zero. We extract these CNN features in every 10-th frame, and average them into a single video-wide feature vector. This 4,096 dimensional feature vector is not directly related to the category confidence scores, but is an intermediate representation.

Each of the two attributes features were separately power-normalized [25] and ℓ_1-normalized. Then, the two sets of attributes were concatenated, yielding a 1,051 dimensional vector. Finally, we double the size of this vector by appending the squares of the elements of the 1,051 dimensional attribute vector. This enables a linear classifier to implement non-linear quadratic decision surfaces over the original 1,051 dimensional attribute vector. The CNN feature is not normalized.

2.3 High-level features

The high-level features temporally localize words in the video. They can come from on-screen text transcribed by optical character recognition (OCR) or from speech recognition. The transcripts are aggregated to sparse feature vectors using a bag-of-words representation based on a 110k-word dictionary consisting mainly of English words. We have used following textual descriptors:

- **OCR:** We used the same OCR system from our 2012 and 2013 MED submissions [2, 19]. In each video frame (sampling rate of 5Hz), MSER [18] regions are extracted from the luminance channel. Regions that do not have a suitable aspect ratio or weak gradients on their boundary are eliminated. The remaining regions are grouped into text lines, and then further segmented into words. Each region is expressed in terms of a HOG-based descriptor [6]. An RBF kernel SVM classifier (trained on standard Windows fonts) predicts the probability of each character. These probabilities are combined using an English language model based on 4-grams over characters to yield the final OCR results at the word level.

- **LIMSI ASR:** The state-of-the-art speech transcription systems from LIMSI are available for 16 languages [14, 15]. The files were processed by first performing speaker diarization, followed by language identification (LID) and then transcription.

The system corresponding to the language identified was used if the LID confidence score was above 0.8, else the English system was used. See the next section for details on the LIMSI ASR system.

- **Fraunhofer ASR:** The Fraunhofer ASR is the output of a large-vocabulary continuous speech recognition system. The underlying acoustic models are trained on about 100h of American English broadcast data which was manually transcribed. The language model includes online news and newswire articles as well as patents. The vocabulary uses the most frequent 130k words and provides multiple pronunciations. Decoding is performed with the open-source speech recognition toolbox Kaldi [23] with deep neural network based models on automatically generated segments with model-based speech activity detection. Only an English ASR system with no language detection is used.

The INRIA-LIM-VocR runs include LIMISI ASR system, while the AXES runs include Fraunhofer ASR system. This is the only difference between the runs.

2.4 The LIMSI ASR system

The speech-to-text transcription systems used to process the data for this evaluation were developed jointly by LIMSI and Vocapia research and evaluated within the Quaero program [14]. The systems make use of statistical modeling techniques as described in [9]. The acoustic, language and pronunciation models are specific to each language and trained on large audio and text corpora [14, 16].

Prior to transcription, the audio files are partitioned: identifying the portions containing speech to be transcribed [8], and associating segment cluster labels, where each segment cluster ideally represents one speaker.

Two types of acoustic features are used. The first type is PLP-like [10], with cepstral normalization carried out on a segment-cluster basis [9]. A 3-dimensional pitch feature vector (pitch, Δ, and $\Delta\Delta$ pitch) is added to the original PLP feature, resulting in a 42-dimension vector. The second type are probabilistic features produced by a Multi-Layer Perceptron (MLP) from raw TRAP-DCT features [26], which have been shown to improve system performance when concatenated with cepstral features [7]. The MLP networks were trained using the simplified training scheme proposed in [28] with phone state targets. The feature vector formed by concatenating the MLP, PLP and pitch features has 81 elements.

The acoustic models are trained on the data distributed in Quaero (50-100 hours per language) as well as on data from other sources mainly from previous European or national projects. For some languages the acoustic models were developed in an unsupervised manner, since no manually transcribed training data was available, as described in [21].

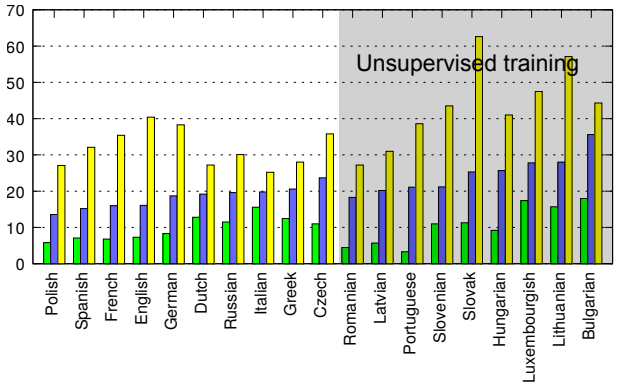

Figure 2: Word error rates (WER) on Quaero test data for 19 European languages. The blue bars show the average WER for the complete test set (4-5 hours per language), as well as the highest (yellow) and lowest (green) WER on individual files.

Large n-gram language models are obtained by interpolating multiple unpruned component language models trained on subsets of the training texts and used for both decoding and lattice rescoring. Language model training is performed with LIMSI STK toolkit which allows efficient handling of huge language models without any pruning or cutoff.

Since the language of the audio data is unknown, this was automatically identified using the Vocapia Research language identification (LID) system v4.2. It is assumed that the audio file contains speech in only one language. This system identifies the language and gives a language confidence score. If the confidence was above 0.8, and a transcription system exists for the language, the audio file was transcribed with the detected language. If not, the audio was transcribed with the English system. The audio files with a low LID confidence score were also transcribed with the detected language (if a transcription system exists). The transcription with the higher average word confidence score was selected. For all languages the transcripts include alternate hypotheses obtained from a consensus network.

Taking into account only files with LID scores above 0.8, English is by far the predominant language, selected for 95% of the files. The second language is Spanish (just under 3%), with French, German and Russian accounting for about 1%. If only files with a confidence score of 1.0 are considered, 98% of them are identified as English. Therefore, we expect that adding bag-of-words dictionaries in other languages would probably not improve the performance much.

Figure 2 shows word error rates (WER) obtained on

Quaero test data for 19 European languages. The languages shown in the left part of the figure were trained using standard supervised methods, with at least 50-100 hours of transcribed audio data for all languages. The systems on the right were all trained in an unsupervised manner, providing indirect supervision via a language model and using only unlabeled audio for acoustic model training. This procedure is described for the Latvian language in [21]. It can be seen that some of the systems trained without any transcribed audio have similar average WERs as those trained with annotated data. It can also be noticed that there is a large variation in WER across the audio files, with the higher errors on files containing more interactive speech, speech from non-native speakers and produced in noisy environments.

3 Classifier training and experimental setup

In this section we briefly present the training of our event detection models, and the validation setup based on the TrecVid 2011 data.

3.1 Training classifiers for individual feature channels

Each of the feature vectors (low, mid, or high level) is used to train a linear SVM classifier with the LIBSVM software package [4]. To determine the hyper-parameters of the SVM we used different strategies, depending on the number of training examples.

- For the 100 example case, we used the same classification approach as in 2013 [2]: 10-fold cross-validation to estimate the SVM's regularization parameter C and the weighting factor for the positive samples.
- For the 10 example case, we observed that cross-validation per event and per channel resulted in unstable parameters. Therefore, we globally cross-validated these parameters across events and channels, which led to an SVM cost parameter of $C = 9$ and a weight for the positive training samples of $1/16 \times N_{\text{neg}}/(N_{\text{neg}} + N_{\text{pos}})$.

3.2 Valdiation setup using TrecVid 2011

To validate our system we used the TrecVid 2011 dataset. NIST provided a TV14Test set for this kind of experiments, but we found that it contained too few positive test examples, which makes evaluation results unstable. TrecVid 2011 training set contains 2,650 positive clips for 10 classes (E006 to E015), and 9,600 background clips that belong to none of these classes. The test set comprises 31,820 clips, which represents roughly 15 % of the TrecVid 2014 test set. The main difference in the data corpus of TrecVid 2011 and 2014 is that several near-miss examples are provided with TrecVid 2011.

For each of the ten classes, we generated a 100 example training subset to validate the 100 example training regime. Since evluation results may be unstable when using small training sets, we generated 10 sets of 10 positive examples each (which we refer to as "versions"), to validate the 10 positive example training regime. Evaluating the classification performance on the 10 versions allows to report an average performance as well as to estimate a standard deviation.

4 Experiments

We conducted several experiments on the TrecVid MED 2011 dataset to assess the effectiveness of different features, and to gain insight in the differences between the regimes with 100 and 10 training examples (100Ex and 10Ex respectively).

4.1 Performance of individual features

The experiments below are presented in the chronological order they were carried out, which explains why some cases of interest might not have been tested.

In our first set of experiments we evaluated the performance of individual features. Table 2 gives an overview of these results for the 100Ex and 10Ex scenarios. For 10Ex, we present results for two cases, where the SVM hyper-parameters (positive weighting and regularization parameter) are: (i) set by cross-validation; and (ii) fixed globally.

The best feature channels are the dense trajectories, CNN, and SIFT features, together with the 1,051 dimensional attribute features derived from them. Our reference results are from the previous year, 2013, and are reported for SIFT and dense trajectories. This year's results are slightly better, due to computationally more expensive settings, e.g., 2,048 GMM components for SIFT *vs*. 1,024 in 2013.

For the 10-example version of the training, it is hard to perform cross-validation to estimate the SVM hyper-parameters, due to the small training set. Indeed, the mAP is better, and also more stable (i.e., it reduces the standard deviation) when we fix these parameters instead (see Table 2). Therefore, we used the fixed hyper-parameter settings in all subsequent experiments.

Going from 100 to 10 training examples reduces the mAP performance by a factor $1.6\times$ to $2.1\times$. We visualize the relation between the 10 and 100 example setting per-

feature	100Ex	10Ex	
cross-val. parameters	yes	yes	no
SIFT	33.38	15.24 ± *1.31*	15.62 ± *1.05*
Color	27.89	12.15 ± *1.11*	12.79 ± *0.80*
CNN fc6	37.87	19.42 ± *1.33*	20.34 ± *1.10*
Dense trajectories	45.11	24.09 ± *0.91*	24.26 ± *0.82*
MFCC	17.52		8.32 ± *0.82*
Scatter descriptor	13.05		6.25 ± *0.54*
OCR	10.54		4.10 ± *1.20*
ASR	14.75		8.20 ± *1.36*
ASR2	9.82		5.33 ± *1.51*
Attributes	33.66		20.73 ± *0.95*
User attributes		5.89	
Previous results			
ICCV'13 SIFT [20]	29.04		
AXES'13 SIFT	29.62		
AXES'13 dense traj.	44.94		

Table 2: mAP of the individual channels for 100 and 10 training examples. For the 10 example case, we also show results for the case where SVM hyper-parameters were set by cross-validation.

formance in Figure 3. Attribute descriptors are the ones that resist best to this reduction — compare, e.g., with SIFT — probably because they encode higher-level information. The dense trajectory features also have a relatively robust performance.

4.2 Early fusion and late fusion

We employ two kinds of techniques to combine individual features: early and late fusion. In early fusion, the combined feature vector is a concatenation of the (scaled) individual vectors. The SVM classifier is then trained on this concatenated feature vector. In the case of late fusion, we linearly combine the scores of the SVM classifiers trained on the individual features (with appropriate weights).

Table 3 presents results obtained with fusion of various channels. The first row of the table gives the results when using late fusion on all channels without doing any early fusion. The next five rows show results of performing early fusion with two channels. Here, we first report the performance of using only this combination of two channels, and then the result when this early fusion combination is late fused with all the remaining channels. Although early fusion generally improves over the individual channel outputs, the early fusion generally does not improve performance significantly when combining it with late fusion of the other channels. One exception to this trend is the CNN+Attributes combination in the

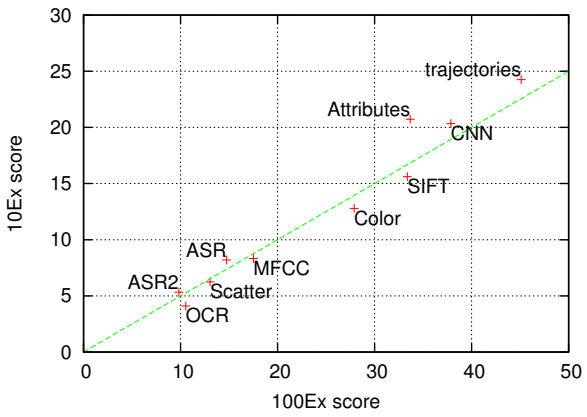

Figure 3: Comparison of performance in the 100Ex and 10Ex scenarios of the individual features.

100Ex case. The following four rows in the table show similar results when using early fusion to combine different pairs of features. All the results in this part of the table improve over the late fusion baseline.

The late fusion weights are estimated with 30-fold cross-validation for 100Ex, and set uniformly for 10Ex, see below for more discussion about this choice. The best settings for the 100Ex and 10Ex case (marked in bold in the table) are different, but share most early fusion channels, except that in the 100Ex case SIFT and Color are input separately to the late fusion. These are the combinations we retained for all further experiments. For 100Ex this was the system that was submitted, further details on the submitted system for 10Ex follow below.

4.3 Handling of near-miss examples

This year, many of the training examples were "near-miss" rather than positive examples. To evaluate the impact of these near-miss examples, we used folds of cross-validation on the TV14Test dataset, since there were almost no near-miss training examples in the TrecVid 2011 dataset.

We considered three options to treat the near-miss examples: use them as negative examples, ignore them, or use them as positive examples. Table 4 shows the results with these three strategies. The standard deviations are high, especially for the 10Ex case, due to the few positive test examples in TV14Test. Considering near-miss examples as negatives is similar to ignoring them, because they get absorbed in the background videos. In the 10Ex case, it is often beneficial to use near-miss examples as positive training data. In contrast, for 100Ex it is never optimal to use them as positives. This difference is explained by the lack of positive training data in the 10Ex case: here it is

feature combined channels	100Ex EF	 LF with rest	10Ex EF	 LF with rest
LF only		53.07		31.47 ± *0.97*
Color+SIFT	34.07	53.09	15.83 ± *0.84*	31.61 ± *1.00*
SIFT+trajectory	46.22	52.68		
MFCC+Scatter	18.66	53.05	8.68 ± *0.69*	31.51 ± *0.97*
ASR+OCR	18.57	53.22	8.20 ± *1.45*	31.33 ± *0.93*
CNN+Attributes	39.15	54.17	23.35 ± *1.03*	31.47 ± *1.02*
MFCC+Scatter, CNN+Attributes		**54.27**		31.55 ± *0.97*
MFCC+Scatter, CNN+Attributes, Color+SIFT		54.02		**31.71** ± *0.99*
MFCC+Scatter, ASR+OCR		53.22		31.37 ± *0.99*
all possible EFs		53.95		31.47 ± *1.02*
AXES'13 submitted combination		52.58		

Table 3: Early and late fusion (EF and LF respectively). The "EF" columns report results combining just the 2 features with early fusion (e.g., Color+SIFT: 34.07). The columns "LF with rest" are obtained with a LF on all the channels, where some of them are combined with EF (e.g., EF of Color+SIFT and LF of the other channels: 53.09). See Section 4.2 for more details.

better to get more examples, despite the fact that they are not perfect. In the 100Ex case this effect does not apply since a sufficient number of positives is available.

4.4 User annotations

User annotations were integrated in two different stages and only in the 10Ex runs. For 100Ex the sufficiency of training data made this unnecessary.

User attributes. We refer to a classification score that is obtained as the scalar product of the attribute vector with a manually designed weight vector as user attributes. Each component of the attribute vector is semantically meaningful: it corresponds to a class from the external training sets. The user reads the event definition and annotates each of these classes with a $+1$ score that indicates that the class is relevant for the event, producing a sparse weight vector. This is done during the Semantic Query Generation phase. The user attributes are combined with the other classification scores with a fixed weight that is optimized on TrecVid 2011, see Figure 4. The user annotations do bring an improvement, be it a small one from from 31.5% to 31.6% mAP when combined with the LF-only baseline system.

Vocabulary for text descriptors. The annotator extracts a set of relevant words from the event description text, and assigns each word a weight between 1 and 3. This is used as a bag-of-words vector that is included as another positive training example in the training of the text-based features, besides the others text vectors extracted from videos. No specific weighting was applied to this additional example. Table 5 shows that this improves the performance for the 10Ex case.

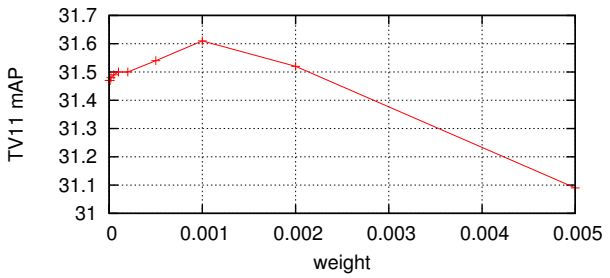

Figure 4: Improvement of the LF score on 10Ex (baseline of Table 3) when combined with the user attributes with different weightings.

4.5 Late fusion weights for the 10Ex case

Until now, we have used a fixed weight of 1 for all the inputs when performing late fusion in the 10Ex case. This is because cross-validation for computing the weights, as we do for 100Ex, is not beneficial, see Table 6. However, if we had access to the optimal weights, the results would be very good (see the row "optimal LF weights").

In order to obtain late fusion weights for the 10Ex setting we use the TrecVid 2011 data. We estimate late fusion weights for each of the classes (using 10 random 10Ex sets). We then use the average as a fixed set of late fusion weights for the TrecVid 2014 10Ex setting. To validate this approach, we estimate late fusion weights for all but one TrecVid 2011 class, average these weights, and test on the remaining class. We do this in turn for each

channel	100Ex negative	100Ex ignore	100Ex positive	10Ex negative	10Ex ignore	10Ex positive
trajectory	**63.96** ± *3.3*	63.48 ± *3.6*	60.43 ± *3.9*	32.80 ± *11.4*	**33.02** ± *11.5*	30.88 ± *10.0*
SIFT	62.28 ± *3.6*	**62.61** ± *3.5*	60.52 ± *3.7*	33.92 ± *13.0*	34.40 ± *12.7*	**36.92** ± *12.2*
CNN	**62.87** ± *3.5*	62.40 ± *3.5*	59.41 ± *3.5*	39.98 ± *10.8*	**40.27** ± *11.2*	38.21 ± *10.4*
Color	55.67 ± *3.1*	**56.40** ± *3.3*	53.67 ± *3.2*	15.30 ± *10.0*	15.51 ± *10.1*	**21.44** ± *11.6*
MFCC	**34.94** ± *3.2*	34.90 ± *3.2*	33.55 ± *3.3*	4.99 ± *4.8*	**5.24** ± *5.4*	4.31 ± *4.1*

Table 4: Cross-validation performance on TV14Test folds (5 classes), depending on the handling of near-miss training examples (considered as positive, negative or ignore).

augmentation	100Ex ASR	100Ex OCR	100Ex LF	10Ex ASR	10Ex OCR	10Ex LF
no	14.75	10.54	54.27	8.20 ± *1.36*	4.10 ± *1.20*	31.61 ± *0.89*
yes	15.52	11.09	54.17	10.88 ± *0.90*	5.48 ± *1.14*	**32.55** ± *0.94*

Table 5: Effect of augmenting the training set of text-based features with a manually selected set of words. LF baselines: optimal values obtained in Table 3 (100Ex) and Figure 4 (10Ex).

LF method	mAP
cross-validation of weights	30.34 ± *1.92*
uniform weights	32.50 ± *0.98*
optimal weights (based on test labels!)	37.43 ± *0.88*
leave-one-out, average	34.08 ± *1.23*
leave-one-out, median	34.00 ± *1.45*
leave-one-out generalized mean, $p = 0.4$	**34.16** ± *1.28*

Table 6: Late fusion approaches for the 10Ex case.

class and obtain a classification score. This result can then be compared to a uniform weighting late fusion baseline, and the cross-validation procedure to estimate the late fusion weights used in the 100Ex setting.

We investigated a couple of alternatives to averaging, namely the median and the generalized mean. The generalized mean is defined as

$$w = \frac{1}{n}\left(\sum_{i=1}^{n} w_i^p\right)^{\frac{1}{p}},$$

where p is a parameter that controls the type of mean (for example, we obtain the geometric mean when $p = 0$ or the averaging when $p = 1$). We varied p from 0 to 1 in steps of 0.1. We found that the best way of aggregating the weights across the events was using the generalized mean with a parameter $p = 0.4$.

5 Conclusions

In the final evaluation, results were submitted by twelve teams for the 10 and 100 example tasks, in the pre-specified and ad-hoc tracks. The results for the INRIA-LIM-VocR submission were 0.5 to 1 mAP point above those for the AXES submission, which is consistent with our experiments on TrecVid 2011. The INRIA-LIM-VocR ranked first with CMU on the 10 example ad-hoc track. For the other tasks, our submissions ranked between 4th and 5th place, behind CMU, BBNVISOR, and Sesame.

In terms of processing speed, our feature computation time is similar to other participants, at about $0.26\times$ real-time (on a 24-core computer). The time our system takes for training – 17 min. per event – is also comparable to that reported by other teams. The time taken to compute the event detection scores for the 200,000 test videos – 90 min. per event – is longer than the average. Most time at this stage is spent on reading data from disk, which for our features represents a volume of 1 TB. Scoring the test videos for several events in one pass would take 90 min. in total, but this was not allowed by the TrecVid guidelines.

The usage of the I/O server was new this year, and worked without any major problems. The main impact on our workflow was that it restricted our flexibility to order the computations as we found most convenient. For example, there was a tradeoff on the check-out time of the metadata generation tasks: later was better to get a shorter processing time measurement, but required to deprive other users of access to our cluster during the processing. A minor inconvenience was the time taken by the server to validate submitted results (10 to 20 minutes).

Acknowledgements This work was partly supported by the European integrated project AXES and the ERC advanced grant ALLEGRO. The LIMSI ASR system was partially developed within the Quaero program.[1]

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Attribute-based classification with label-embedding. In *NIPS 2013 Workshop on Output Representation Learning*, 2013.

[2] R. Aly, R. Arandjelovic, K. Chatfield, M. Douze, B. Fernando, Z. Harchaoui, K. Mcguiness, N. O'Connor, D. Oneata, O. Parkhi, D. Potapov, J. Revaud, C. Schmid, J. Schwenninger, D. Scott, T. Tuytelaars, J. Verbeek Jakob, H. Wang, and A. Zisserman. The AXES submissions at TrecVid 2013. In *TRECVID Workshop*, 2013.

[3] J. Andén and S. Mallat. Multiscale scattering for audio classification. In *ISMIR*, 2011.

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[5] S. Clinchant, J.-M. Renders, and G. Csurka. Transmedia pseudo-relevance feedback methods in multimedia retrieval. In *Advances in Multilingual and Multimodal Information Retrieval*, 2008.

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[7] P. Fousek, L. Lamel, and J.-L. Gauvain. On the Use of MLP Features for Broadcast News Transcription. In *Text, Speech and Dialogue*, number 5246/2008 in Lecture Notes in Computer Science, pages 303–310, 2008.

[8] J.-L. Gauvain, L. Lamel, and G. Adda. Partitioning and transcription of broadcast news data. *ICSLP*, 98(5):1335–1338, 1998.

[9] J.-L. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System. *SPCOM*, 37(1-2):89–108, 2002.

[10] H. Hermansky. Perceptual linear prediction (PLP) analysis for speech. *Journal of the Acoustical Society of America*, 87:1738–1752, April 1990.

[11] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org, 2013.

[12] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*, 2011.

[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[14] L. Lamel. Multilingual Speech Processing Activities in Quaero: Application to Multimedia Search in Unstructured Data. In *The Fifth International Conference: Human Language Technologies - The Baltic Perspective*, pages 1–8, Tartu, Estonia, October 4-5 2012.

[15] L. Lamel and J.-L. Gauvain. Speech processing for audio indexing. In *Advances in Natural Language Processing*, 2008.

[16] L. Lamel and J.-L. Gauvain. Speech processing for audio indexing. In *Proceedings of the 6th International Conference on Natural Language Processing, GoTAL 2008 - Advances in Natural Language Processing*, number 5221/2008 in Lecture Notes in Computer Science, pages 4–15, 2008.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002.

[19] D. Oneata, M. Douze, J. Revaud, J. Schwenninger, D. Potapov, H. Wang, Z. Harchaoui, J. Verbeek, C. Schmid, R. Aly, K. Mcguiness, S. Chen, N. O'Connor, K. Chatfield, O. Parkhi, R. Arandjelovic, A. Zisserman, F. Basura, and T. Tuytelaars. AXES at TrecVid 2012: KIS, INS, and MED. In *TRECVID Workshop*, 2012.

[20] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *ICCV*, 2013.

[21] I Oparin, L. Lamel, and J-L. Gauvain. Rapid Development of a Latvian Speech-to-Text System. In *ICASSP*, Vancouver, Canada, 2013.

[22] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. Smeaton, and G. Quénot. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, Petr P. Schwarz, et al. The Kaldi Speech Recognition Toolkit. In *Proc. Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 1–4, 2011.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. arXiv:1409.0575, 2014.

[25] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.

[26] Petr Schwarz, Pavel Matějka, and Jan Černocký. Towards lower error rates in phoneme recognition. In *Text, Speech and Dialogue*, volume 3206 of *Lecture Notes in Compek, Ivan and Pala, Karel*, pages 465–472. Springer Berlin Heidelberg, 2004.

[27] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[1] http://www.quaero.org

[28] Qifeng Zhu, Andreas Stolcke, Barry Y Chen, and Nelson Morgan. Using mlp features in sri's conversational speech recognition system. In *Interspeech*, pages 2141–2144, 2005.