# TNO at TRECVID 2013:
# Multimedia Event Detection and Instance Search

Henri Bouma, George Azzopardi, Martijn Spitters, Joost de Wit, Corné Versloot, Remco van der Zon, Pieter Eendebak, Jan Baan, Johan-Martijn ten Hove, Adam van Eekeren, Frank ter Haar, Richard den Hollander, Jasper van Huis, Maaike de Boer, Gert van Antwerpen, Jeroen Broekhuijsen, Laura Daniele, Paul Brandt, John Schavemaker, Wessel Kraaij, Klamer Schutte

TNO, P.O. Box 5050, 2600 GB Delft, The Netherlands

## ABSTRACT

We describe the TNO system and the evaluation results for TRECVID 2013 Multimedia Event Detection (MED) and instance search (INS) tasks. The MED system consists of a bag-of-word (BOW) approach with spatial tiling that uses low-level static and dynamic visual features, an audio feature and high-level concepts. Automatic speech recognition (ASR) and optical character recognition (OCR) are not used in the system. In the MED case with 100 example training videos, support-vector machines (SVM) are trained and fused to detect an event in the test set. In the case with 0 example videos, positive and negative concepts are extracted as keywords from the textual event description and events are detected with the high-level concepts. The MED results show that the SIFT keypoint descriptor is the one which contributes best to the results, fusion of multiple low-level features helps to improve the performance, and the textual event-description chain currently performs poorly. The TNO INS system presents a baseline open-source approach using standard SIFT keypoint detection and exhaustive matching. In order to speed up search times for queries a basic map-reduce scheme is presented to be used on a multi-node cluster. Our INS results show above-median results with acceptable search times.

Table 1: Overview of the submitted MED-runs and the mean average precision (MAP).

| Run | Label for MED | Method description | MAP (%) |
|---|---|---|---|
| 1 | TNO_MED13_FullSys_PROGSub_PS_100Ex_1 | Complete system on PS | 10.3 |
| 2 | TNO_MED13_VisualSys_PROGSub_PS_100Ex_1 | SIFT only on PS | 5.2 |
| 3 | TNO_MED13_FullSys_PROGSub_PS_0Ex_1 | Semantics only on PS | 0.4 |
| 4 | TNO_MED13_FullSys_PROGSub_AH_100Ex_1 | Complete system on AH | 8.2 |
| 5 | TNO_MED13_VisualSys_PROGSub_AH_100Ex_1 | SIFT only on AH | 5.2 |
| 6 | TNO_MED13_FullSys_PROGSub_AH_0Ex_1 | Semantics only on AH | 0.3 |

Table 2: Overview of the submitted INS-runs and the mean average precision (MAP).

| Run | Label for INS | Method description | MAP (%) |
|---|---|---|---|
| 1 | F_NO_TNOM3-SHOTBFSIFT_1 | SIFT key point descriptors, exhaustive search using approximate nearest-neighbor key point matching. | 14.1 |

## 1. INTRODUCTION FOR MED

TNO has performed a TRECVID Multimedia Event Detection (MED) submission [22] as part of its ongoing GOOSE project [27]. Goal of this project is to allow users to execute arbitrary queries on live sensor data, similar to how internet search engines allow queries on web pages. Key GOOSE challenges include scalability (in amount of users, domains, simultaneous queries and sensors) and the semantic gap between sensor data and user queries. Main design paradigm for GOOSE to reach scalability is to perform non-query specific processing on its incoming sensor data, and have a query performed on this generic meta data. This nicely fits the MED paradigm of having a meta data store independent of the actual events. The basic design elements within GOOSE to close the semantic gap is by using a semantic analysis of the user query, use external crowd-sourced knowledge sources (e.g. semantic web, ImageNet, and Youtube) to obtain specific understanding of domains not specifically considered at design time, and rely on user interaction to disambiguate concepts.

Participants of the MED task develop an automated system that determines whether an event is present in a video clip. The participants receive a test set of videos and a training set (event kit) consisting of a textual description and example videos describing the event. The system computes an event probability for each video in the test set. The 2013 MED evaluation consists of 20 “pre-specified” (PS) and 10 unseen “ad-hoc” (AH) event kits containing 100, 10 or 0 example event videos. In the development phase, the PS event kits could be used and metadata was extracted from the test set. For the AH submission, the metadata generator was locked and no new data was extracted from the test videos. TRECVID provides a complete test set of 98,000 video clips and a subset of 32,000 clips. The TNO system is evaluated on the subset.

The outline of this paper is as follows. In Section 2 we describe the proposed TNO MED system and in Section 3 we describe the MED experiments and report the results. Section 4 describes the system and results for the instance-search (INS) task. Finally, in Section 5, we draw our conclusions.

## 2. THE TNO MED SYSTEM

The TNO system elements are inspired by TRECVID 2012 MED systems of: CMU [31], SESAME [1], ECNU [30], BNNVISER [19], SRIAURORA [8], MediaMill [28], AXES [2], Tokyo [12], GENIE [23], IBM [6]. Figure 1 depicts an overview of the system.

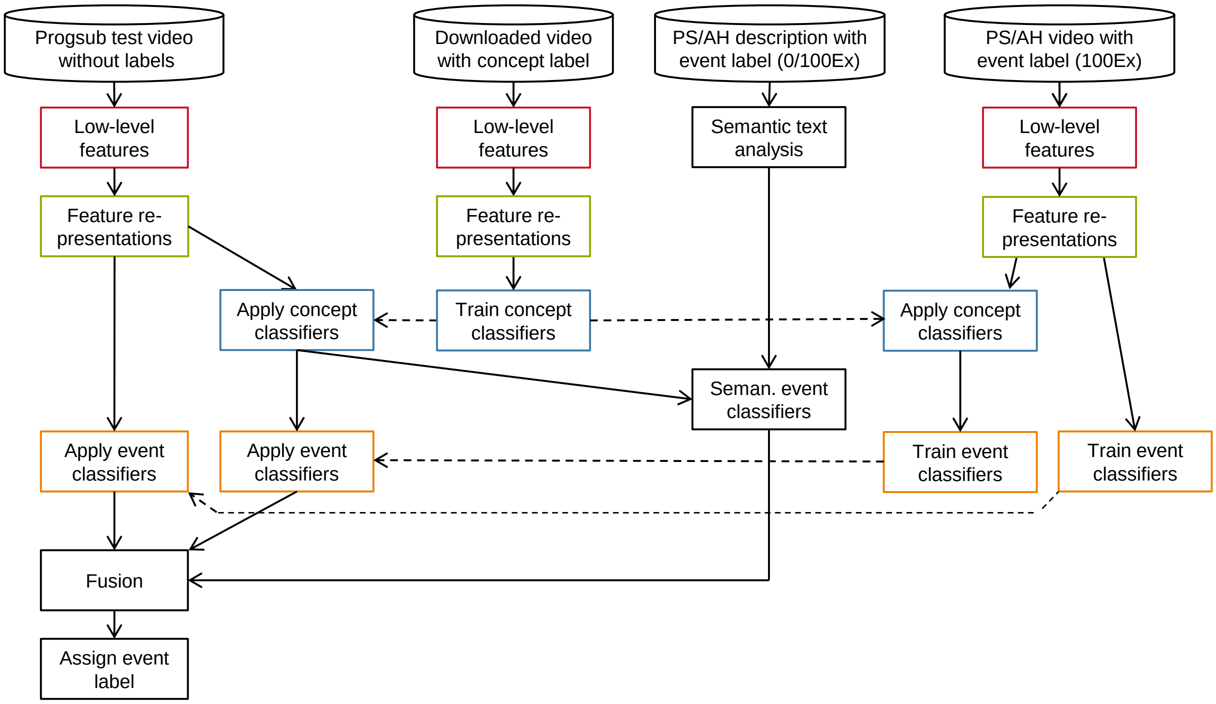

Figure 1: Overview of the TNO architecture. For each event and each feature, a separate classifier is trained.

The system uses low-level feature descriptors that extract local information from given video clips, BOW feature representations that store the data efficiently, high-level concept classifiers that recognize concepts such as objects and actions in the video, semantic text analysis of textual description of the event kits, classifiers that are trained on the event-kit example videos and applied to the test set (progsub), and fusion that combines different features.

The outline of this section is as follows. Sec. 2.1 describes the low-level features, Sec. 2.2 the feature representation, Sec. 2.3 high-level concepts, Sec. 2.4 on-the-fly video download, Sec. 2.5 semantic text analysis, and finally, Sec. 2.6 describes classification and fusion.

## 2.1 Low-level features

We use three categories of low-level features: static visual features, dynamic visual features and audio. Our implementation does not use speech (ASR) or text (OCR) recognition. For static visual features, we use scale-invariant feature transform (SIFT) for structure, Opponent-SIFT for color, and local binary patterns (LBP) for texture. For dynamic visual features, we used spatio-temporal interest points (STIP) and for audio we used the Mel-frequency cepstral coefficients (MFCC) and its derivatives.

- SIFT [17]: We use the implementation of OpenCV[1] that detects keypoints from a Difference-of-Gaussian scale space. We use the sparse keypoint detection with the default parameters as set on OpenCV: 3 octave layers, a contrast threshold of 0.04, edge threshold of 10 and a sigma of 1.6. The resulting SIFT vector has a length of 128. SIFT descriptors are computed on one frame per second for all given video clips.
- Opponent-SIFT [25]: Here, we use the implementation of UvA[2]. We use sparse keypoint detection based on Harris-Laplace scale space with the following default parameters as suggested by [25]: Harris threshold = 1e-9, k = 0.06, Laplace threshold = 0.03. It is computed on one frame per second for short videos and a maximum of 100 frames (equidistant on the time scale) for long videos. We also rescale the image frames in such a way that the maximum dimension is 512 pixels. The descriptor results in a vector of length (3x128=) 384 elements.
- LBP [21][11][32]: The implementation of Oulu[3] is used. The LBP is computed on a fixed grid, where each element consists of 16x16 pixels. We used the following parameters: radius R = 2, number of samples P = 16 and a uniform rotation-invariant mapping.
- STIP [14]: We use the implementation of Toyon[4]. The video frames are rescaled to 320 pixels in width, and STIPs are computed on three 1-second segments in a video. The descriptor consists of 162 elements, containing 72 gradient (HOG) and 90 flow (HOF) features. STIP appears to be a good dynamic feature for action recognition [4][5]. Computing the camera-motion compensated dense STIP (CC-D-STIP) [12] and generating BOW at one frame per second resulted in a much better performance than STIP on only three small segments in the video, but due to computation time constraints we use the former approach.
- MFCC [33]: The VOICEBOX[5] implementation is used with its default values. The original sample rate (in Hz) of the audio file is retained and a mono signal is forced by averaging the stereo frequencies. The MFCC is computed on segments of 10 msec. The descriptor consists of 39 elements containing the MFCC (12 coefficients and the zero-order cepstral coefficient) and its first- and second-order derivatives.

## 2.2 Feature representation

We use the classical Bag-of-Words (BOW) approach to describe the frames of given video clips. This is achieved as follows. First, we randomly select 2000 video clips from the event kit and compute all the descriptors of SIFT, Opponent-SIFT, LBP, STIP and MFCC. Then, we use K-means clustering to generate a vocabulary of 300 words for each of the concerned descriptors. Finally, we process a given video clip by first computing the feature vectors of each descriptor and then assign these vectors to the nearest word (prototype) in the Euclidean space. This process results in a histogram that contains the number of occurrences of each word in a given frame. Further improvements of this approach have been proposed recently, such as VLAD [13] and Fisher Vector [24] that use soft assignments. Due to time constraints and simplicity reasons, we use the classical BOW approach (see Table 3).

A disadvantage of the BOW approach is that spatial information is not encoded. This information can be preserved to a certain extent by using some form of spatial tiling (e.g., spatial pyramid) [15]. We use different configurations for different features (see Table 3) and concatenate the representations of the individual tiles to create the complete representation. The complete BOW representations of different moments in time are summed, to create one BOW representation per clip. Different types of normalization were tried, such as normalization per tile with different weights per level in the spatial pyramid. In the end, L1-normalization over the whole vector was chosen because it appeared to give the best result.

[1] http://www.opencv.org
[2] http://koen.me/research/colordescriptors/
[3] http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab
[4] http://www.toyon.com/
[5] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

Table 3: BOW representations for each feature.

| Feature | Vocabulary size | Spatial tiling | Histogram size |
|---|---|---|---|
| SIFT | 300 | Spatial pyramid: 1x1 + 2x2 + 4x4 | 6300 |
| Opp. SIFT | 300 | Spatial pyramid: 1x1 + 2x2 | 1500 |
| LBP | 300 | 2x2 | 1200 |
| STIP | 300 | 3x3 | 2700 |
| MFCC | 300 | N / A | 300 |

### 2.3 High-level concepts

Our high-level concepts are trained on example images and/or videos. We downloaded images and videos from ImageNet [10], Google and Youtube, based on the keyword describing the concept. No manual check is performed for actual correspondence of the imagery to the concept. We identified 546 general concepts, in various categories, such as objects, actions, scenes and sounds. The identification was based on event descriptions of last years, analysis of the ImageNet structure and an analysis of missing concepts in every category. These concepts are used by semantic text analysis and classification. Videos and images have been downloaded for the identified 546 concepts. The total number of videos was 6744 (on average 14.9 ± 6.0 videos per concept) and the total number of images was 185069 (on average 410.3 ± 132.9 per concept). The distribution over the different concepts was not uniform, due to the availability of the different concepts on the public datasets.

Concept classifiers have been trained on the example data for three features (LBP, SIFT, MFCC). Not all concept classifiers could be trained. For example the MFCC feature requires audio, which is not available for images and some of the videos. If the total number of training objects for a certain object is too low (the threshold was placed on 12 objects), then the classifier is not trained for that particular combination of feature and concept. Finally, we used 442 concepts for LBP, 418 for SIFT and 86 for MFCC.

### 2.4 On-the-fly downloaded images and videos

Besides the submitted version of our MED system, we have developed a non-compliant version with automatic on-the-fly video download from Youtube based on the semantic text analysis of the textual description of the event. These downloaded videos are subsequently used as training material for event classifiers – similar to the labeled event videos – in effect providing an alternative zero-example video MED system.

### 2.5 Semantic text analysis

The semantic reasoning is performed by a fully automated component that takes the textual description of an event kit as input and generates a system query as output. This textual description consists of an event name, definition, explication and evidential description fields. Any new event kit can be processed without further modification of the system and without human interaction. An overview of this component is given in Figure 2.

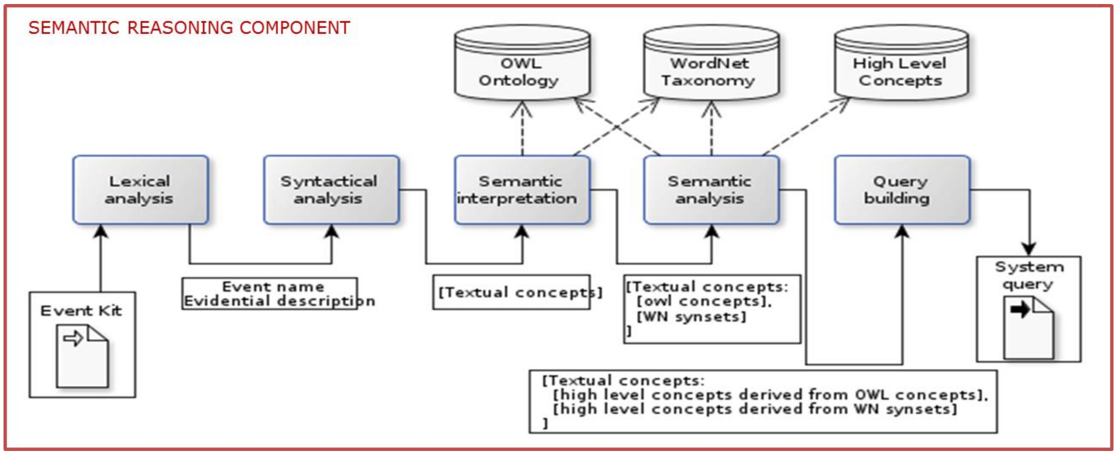

Figure 2: The semantic reasoning component uses the textual description in the event kit to select high-level concepts.

The lexical analysis module selects the event name and evidential description fields of an event kit. The other fields are ignored because they might create ambiguities and misinterpretations. The event name and evidential description are further analyzed by the syntactical analysis module, which is implemented using the Stanford Parser [18] and detects whether a concept should be used as positive or negative for retrieval. For example, the event "winning a race without a vehicle" contains both positive and negative concepts ("race" and "vehicle", respectively). Since the syntactical analysis returns a set of so called textual concepts without actual semantics, the semantic interpretation and semantic analysis modules use an OWL ontology language[6] and WordNet[7] to add semantics to these textual concepts. The OWL ontology models a set of recurring concepts that are particularly relevant for the application domain (i.e., the MED task), while WordNet is used as a complementary source for semantic reasoning. In particular, the semantic interpretation module maps the textual concepts to WordNet and OWL concepts, while the semantic analysis applies the following two types of query expansions:

1. Selection of hyponyms (specific subclasses) of a certain concept in the OWL ontology. For example, in the event "grooming an animal" the semantic analysis provides "Grooming a pet" as an expansion, where pet is a hyponym of animal.
2. Selection of hypernym (general super class) and hyponym relations in WordNet. For a certain concept, hypernyms and hyponyms of three levels deep in the WordNet taxonomy are selected. These hypernyms and hyponyms are then matched against a list of high level concepts that are known by our system. If a direct match is found, then this concept is provided as an expansion of the original concept and its semantic distance is set as the Lin-measure [16].

The query building sub-component generates a system query in which high-level concepts known by the system are combined using logical operators. In particular, an AND-operator connects high level concepts that are derived from the same textual concept, while the OR-operator connects expansions (i.e., hypernyms and hyponyms) of the same concept. The AND-operator is implemented as a summation (summation of log-probabilities equals multiplication of probabilities) and the OR-operator is implemented as maximum. The NOT-operator is used to identify negative concepts that should be excluded for retrieval. The semantic distance values set during the semantic analysis are added to the system query.

### 2.6 Classification and fusion

This section describes five components (A-E) for event classification, concept classification and fusion.

*A: Event classification on each low-level feature*
For each event and for each low-level feature we train an SVM classifier using the LiBSVM implementation [7][8]. The input to an SVM is a set of features vectors generated using the BoW model (see section 2.2). We tested the following kernels: radial basis function (RBF), chi-squared ($\chi^2$) and the histogram-intersection. In our experiments, it turned out that the histogram-intersection is the most effective one, and therefore we only use this metric in our submission. To handle the unbalanced data, multiple balanced classifiers were created where all positives were reused for every classifier and the same number of negatives were randomly sampled for every classifier (bagging with undersampled negatives to obtain an equal amount).

*B: Concept classification on each low-level feature*
Similarly, we trained an SVM-based classifier for each concept and for each low-level feature with a histogram-intersection kernel. In the training phase, each SVM is trained on example images and videos (Sec. 2.3) for a single concept. When applied to a TRECVID video clip, a vector of probabilities is generated indicating the presence of multiple concepts.

*C: Event classification based on concepts without semantic text analysis.*
Also here, we use an SVM-based classifier with histogram intersection to model each event based on concepts without semantic text analysis. The input for the SVM consists for each clip of a vector of confidences indicating concept presence that is generated by the concept classifiers. This component was only used for internal evaluation and it was not included in the submission.

[6] http://www.w3.org/2004/OWL/
[7] http://wordnet.princeton.edu/
[8] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

*D: Event classification based on concepts and semantic text analysis*

Event classification based on the concept classifiers and semantic text analysis is necessary to handle the case when no example videos can be used for event training. On one hand, the concept classifiers (component B) compute a probability for the concept presence for each video clip. On the other hand, the semantic text analysis identifies positive and negative concepts from the textual description of the video. The computed probabilities of the identified concepts are normalized over all pre-specified events and subsequently multiplied by a weighting factor in order to create a concept score. The weight is a multiplication of three elements: semantic distance (explained in Sec. 2.5), positive or negative concept (value 1 or -1, respectively) and the detectability value. The detectability value is an estimate of the predictive quality of a certain concept and it is implemented as the average probability estimate of the concept classifier, measured over all events in which the concept is identified by the semantic text analysis. The probability estimates of the concept classifier are z-scored over all videos for normalization, which implies that a negative detectability value does not indicate the presence of an event. The negative detectability values are clipped to zero, resulting in a weight and concept score of zero, and therefore the concept is not taken into account for the total score. The total score per event is the sum of the highest concept score of each OR-group (i.e. the expanded concepts including hypernyms and hyponyms). The total score is adjusted to a value between zero and one – with the inverse function of the tangent.

*E: Fusion of the event classifiers from multiple features, concepts and/or semantic analysis*

Fusion of event and concept probabilities is performed as late fusion. We tested the following functions: arithmetic mean, geometric mean, accuracy weighted average, and threshold-distance weighted average. Each of them is described in more detail in the paper of Natarajan [20]. In order to compare scores from multiple classifiers, we first normalize the scores, as each classifier has its own optimal threshold and score distribution. The fuser applies a double sigmoid function [20] to normalize the scores for the computed threshold. Based on our experiments, accuracy weighted average [20] was chosen, where the weighting with accuracy was replaced by a weighting by (1 – pMiss@TER). The 'pMiss@TER' is described in Sec. 3.

# 3. EXPERIMENTS AND RESULTS FOR MED

## 3.1 Experimental setup

The proposed 2013 MED system was applied to the 20 "pre-specified" (PS) and 10 "ad-hoc" (AH) event kits for 100 or 0 example event videos. The metadata was generated for all videos in the test set of 32k before the ad-hoc events were processed. The different runs are shown in Table 1.

- Run 1 (TNO_MED13_FullSys_PROGSub_PS_100Ex_1): The complete system consists of all elements in Sec. 2, including the low-level features and semantic text analysis that uses high-level concepts.
- Run 2 (TNO_MED13_VisualSys_PROGSub_PS_100Ex_1): Our 'visual system' is actually a SIFT-only implementation, which includes the SIFT feature and excludes other low-level features, high-level concepts and semantic text analysis.
- Run 3 (TNO_MED13_FullSys_PROGSub_PS_0Ex_1): The system for zero-example clips (0Ex) uses only semantic text analysis and high-level concepts.
- Run 4-6 are similar to Run 1-3, but on the ad-hoc event kit.

TRECVID uses the following definitions to measure the performance:

- pFA = fp/(fp+tn)
- pMiss = fn/(tp+fn)
- pMiss @ TER = pMiss at the target error rate, where ratio pFA/pMiss = 12.5
- precision = tp / (tp + fp)
- recall = tp / (tp + fn)
- AP = average precision over all positives
- MAP = mean AP over all events.
- Ro = recall – 12.5 ((tp+fp) / (tp+fp+tn+fn))

## 3.2 Internal performance estimation

Before the official submission, we estimated the performance of system components with cross-validation on the 20 pre-specified events. The results are shown in Figure 3 and Table 4 for five features (LBP, SIFT, Opp. SIFT, MFCC, STIP), semantic analysis using the high-level concepts based on three features (which we call semantic LBP, semantic MFCC and semantic SIFT), and the fusion of these features. Besides the components that are part of our full system submission, we performed an additional analysis of the feature CC D-STIP (which is computed at one frame per second), an SVM trained on the high-level concept sets for the SIFT feature (which we call Concept SIFT), and a fusion of five event classifiers based on five features that are trained using on-the-fly downloaded data.

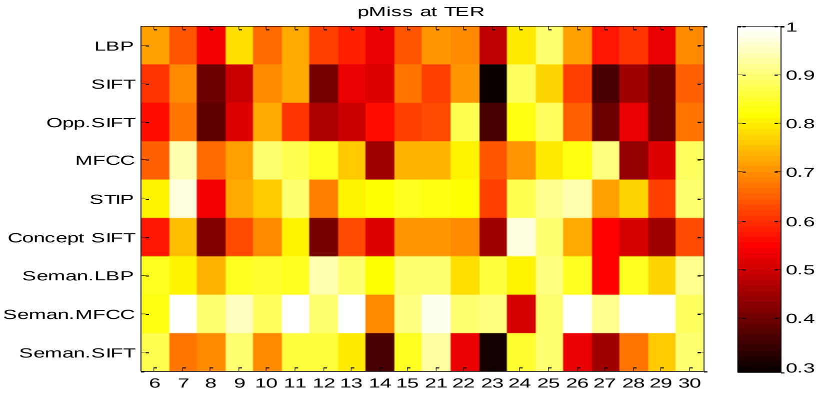

Figure 3: Performance (pMiss@TER) for different feature classifiers (vertical axis) on the pre-specified events (horizontal axis). Lower number indicates better performance.

The results show that on average SIFT performs better than other feature descriptors, indicated by the dark/reddish color. Figure 3 shows that MFCC performs well in event E014, semantic-MFCC in E024, and semantic-SIFT in E014, E022 and E026. The performance of the Concept-SIFT appears to be very similar to the SIFT-feature, while the Semantic-SIFT is really different.

Table 4: Overview of cross-validation results aiming at TER = 12.5 sorted by pMiss on the pre-specified events. To show the variation in TER, both pMiss and pFA are reported.

| | Pre specified (PS) | | Ad hoc (AH) | |
|---|---|---|---|---|
| Method | pMiss @ TER (%) | pFA @ TER (%) | pMiss @ TER (%) | pFA @ TER (%) |
| Semantic: MFCC | 90.2 | 7.3 | 91.0 | 7.2 |
| Semantic: LBP | 85.6 | 6.5 | 92.3 | 7.5 |
| Semantic: SIFT | 79.9 | 6.2 | 89.2 | 7.1 |
| Feature: STIP | 79.3 | 6.4 | 78.5 | 6.3 |
| **Fusion:** 3 semantic *(run 3,6 = 0Ex)* | 78.0 | 6.1 | 87.1 | 7.4 |
| Feature: MFCC | 74.0 | 6.0 | 74.3 | 6.0 |
| Feature: LBP | 65.7 | 5.2 | 63.6 | 5.1 |
| Feature: Opponent-SIFT | 59.1 | 4.7 | 55.8 | 4.5 |
| Feature: SIFT *(run 2,5)* | 57.6 | 4.5 | 55.4 | 4.4 |
| **Fusion:** 5 features | 48.5 | 3.7 | 46.0 | 3.4 |
| **Fusion:** 5 features + 3 semantic *(run 1,4)* | 47.8 | 4.0 | 46.4 | 3.4 |
| Feature: CC D-STIP 1FPS *(not used)* | 68.4 | ±5.5 | --- | --- |
| Concept: SVM SIFT *(not used)* | 63.7 | 5.1 | --- | --- |
| Fusion**:** 5 features trained using on-the-fly downloaded video *(not allowed)* | --- | --- | 63.0 | 5.1 |

The results reported in Table 4 show that fusion of multiple features performs much better than the best single feature (SIFT), with an improved pMiss@TER from 57% to 47.8%. Semantic features show lower performance on the ad-hoc (AH) events compared to the pre-specified (PS) events. This is most likely due to the fact that the used concepts are partly based on the keywords found in the PS events kit and because the actual concept detectability (Sec. 2.6D) could only be trained for concepts that are present in the PS event kit. Moreover, the detectability is thus not well defined for concepts that are not present in the PS event kit.

The results on the PS event kits indicated that it would have been beneficial to use the CC-DSTIP at one frame-per-second instead of the STIP descriptor at only three segments in the whole video (improved pMiss@TER from 79.3% to 68.4%). However, time restrictions did not allow us to generate metadata on the test set in time.

Besides the submitted version of our MED system, we developed a system version with automatic on-the-fly video download from Youtube based on the semantic text analysis of the textual description of the event. The purpose of this system is to explore what can be done if some of the MED system restrictions are weakened. The downloaded videos are used as training material for event classifiers, providing an alternative zero-example video MED system. The results show that the performance is much better than the compliant zero-example case (fusion of 3 semantic), with an improved pMiss@TER from 87.1% to 63.0%. However, it is still worse than the system that is trained on 100 examples (46.0%). This is probably because the downloaded videos have not been manually checked and because they may be biased to a specific instance of the event.

Computation times are as follows: computation of metadata on the Progsub would have taken 8000 hours (av. 15 min. per clip) if it would have been computed on a single core, training event classifiers took 77 minutes per event (using max. 3 cores), and applying event classifiers to the Progsub took 33 minutes per event (av. 61 msec per clip, using 1 core). Times for training and applying the event classifiers excluded data transfer over the network.

### 3.3 Official MED results

The official results are shown in Table 1. They show that our internally used pMiss@TER scores do not linearly correlate with this year’s TRECVID MAP scores.

### 3.4 Discussion for MED

The MED results show that SIFT is our main feature. The performance of the visual features benefit from using spatial tiling. However, we realize that better per-feature performance should be possible, since our SIFT-only implementation has a pMiss@TER of 57%, while others (e.g. CMU in 2012 [31]) obtained a pMiss@TER of 42%. We observed that none of the features is best for all events, rather they are complementary, and the accuracy-weighted fusion of multiple features significantly helps to improve performance. The results on the event kits indicated that it would have been beneficial to use the dense CC-DSTIP at one frame-per-second instead of the STIP-feature at only three segments in the whole video. Yet, we were unable to generate CC_DSTIP metadata on the test set in the available time.

Literature suggests that elements that may improve the system are VLAD or Fisher vectors instead of the BOW representation and a better combination of BOW’s from different tiles. Furthermore, the classification could be modified to handle the unbalanced data better, so that more negatives can be used to train the classifiers. Due to time limitations we were unable to pursue these options in this first year submission.

The current TRECVID MED guidelines and clarifications do not allow the use of automatically on-the-fly downloaded external material during the zero-example cases. Within our GOOSE philosophy [27] we foresee that to obtain a truly domain independent system you can rely on external (crowd-sourced) data sources such as Google Image and Youtube to obtain at query time knowledge on the specific user-query domain. As such this TRECVID imposed restriction does not allow us to test our foreseen system concept. Indeed the on-the-fly downloaded videos – as shown in the bottom row of Table 4 – show a huge increase in performance compared to our next best zero-example case, i.e. the fusion of 3 semantic features.

The textual event description currently has a poor performance. The current system assumes that the event description can be summarized as an AND of multiple OR-groups consisting of concepts. Any additional relation to the concepts, such as when or where they should occur, is not used in the system, limiting the expressive power of the current semantic system.

In the current semantic analysis only hypernym and hyponym relations are used for query expansion. However, the system would benefit from adding different type of relations, so that concepts can be selected with more precision. The following three examples illustrate possible relations that can improve the system. First, the event “Changing a vehicle tire” (event E007) will currently lead to the selection of all possible vehicles, including vehicles that do not have tires, such a ship. Selecting only the subset ‘road vehicles’, e.g. by creating a relation between tire and these vehicles, would give better results. Second, if it is known that a concept is related to an outdoor scenery, the semantic analysis sub-component can expand this concept with more concepts related to outdoor scenery (e.g., sky, grass). Third, in the current implementation, if there are multiple meanings for a certain textual concept, all these meanings are evaluated and further processed in the system. Adding word sense disambiguation would allow more precise reasoning algorithms.

In the current system, some concept classifiers do not have a precise semantics and, as a consequence, multiple examples are mixed in the same concept, creating ambiguities. For example, for the concept hospital, different classifiers could be generated for different sceneries in a hospital (outside view, operating room, or bedroom) allowing the selection of the more specific concepts. Furthermore, we foresee that a richer user interaction, such as concept disambiguation by a human operator, as a promising way to increase system performance for the case of 0 examples.

# 4. THE TNO SYSTEM AND RESULTS FOR INS

## 4.1 The TNO system for INS

Our INS approach is an extension of the original object recognition idea outlined by David Lowe in his paper on SIFT [17]. In our approach (as in our submissions of 2011-2012: [26]) we follow the same scheme of detecting key points, computing and matching local descriptors but instead of a few objects in the recognition database we have descriptors from every video. In order to handle that large amount of videos and subsequent feature descriptors we propose to use a MapReduce [9] scheme for video pre-processing (feature computation) as well as for query matching on a cluster of machines.

## 4.2 MapReduce scheme for ingestion and matching

Our approach consists of two parts, the first part is ingestion of all videos of the dataset. Processing of every video is mapped on a single worker thread (the Map step). This worker thread will decode a video, sample one video frame per second, compute SIFT descriptors (using OpenCV) on that sampled video frame until the video ends. The result is one XML file with SIFT descriptors per frame. Meanwhile, worker threads are started to perform the Reduce step. Every Reduce worker thread collects SIFT descriptors from video frames into a chunk of features. When a chunk reaches its size limit, the SIFT descriptors of that chunk are written to a binary file with corresponding video, frame and key point indices to XML. The number of Reduce worker threads can be chosen significantly smaller than the number of Map (ingestion) workers as ingestion is much more computational expensive.

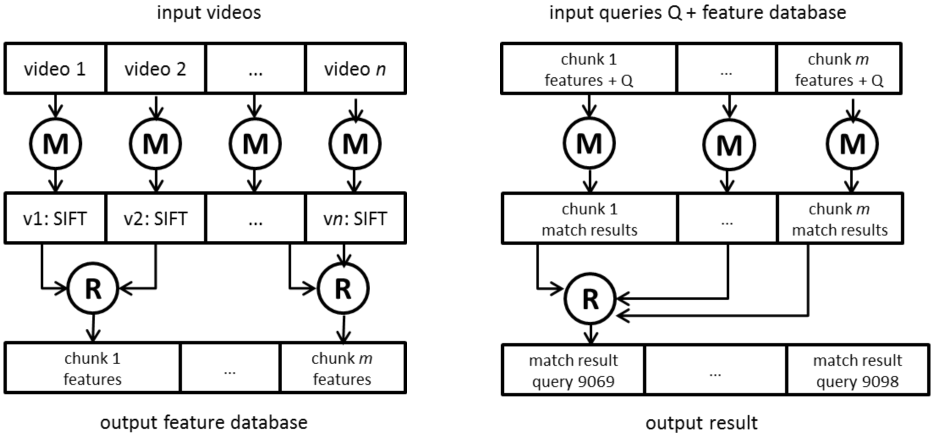

Figure 4: (left) MapReduce scheme for ingestion. (right) MapReduce scheme for matching.

The second part is matching: for every chunk in the feature database a Map worker thread is started. Every Map matcher loads one chunk of the database features from a binary file, loads the pre-computed SIFT descriptors for the query set of images and matches the query descriptors with the descriptors of the chunk. The reduce step here performs collection of the match results per query from every matched chunk taking into account shot boundaries. The end result is per query a sorted list on score of matching video shots. Here we take one Reduce thread per query.

### 4.3 Matching and scoring

Key point matching is performed between a single query image and descriptors from a single video frame of the chunk. Matching uses the distance ratio of the 1st and 2nd key point match to select only 'strong' matches [17]. The score of a match is computed using the query image mask and dilated derivatives, see Figure 5. If a matching key-point pair is within the original provided query mask it scores 1.0 (blue in Figure 5), key-point pairs outside the mask score less depending on the distance to the original mask (green, red, light blue in Figure 5). In this way a mechanism is built in that prefers matches in the original mask but can match background outside mask as well. The total match of a video shot is the sum of all scores of matching key-point pairs in that shot with the query image set.

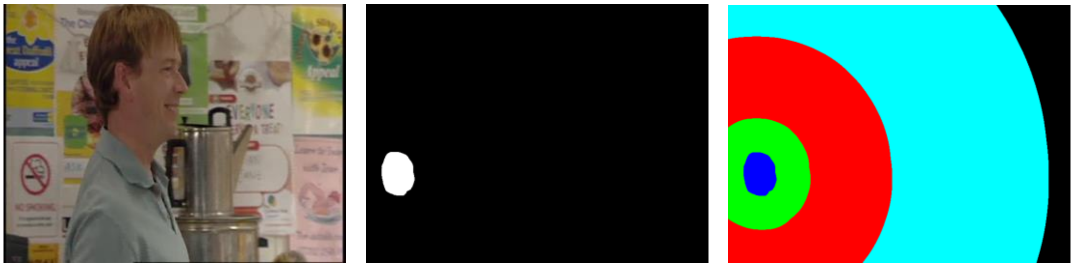

Figure 5: Using masks in scoring, matches in blue score 1, in green score 0.1, in red score 0.01 and in light blue score 0.001. Programme material © BBC.

### 4.4 The results for INS

Figure 6 shows the TNO INS results in average precision per query. In general the results follow the median run scores of all parties  and for most queries are more than average (median).

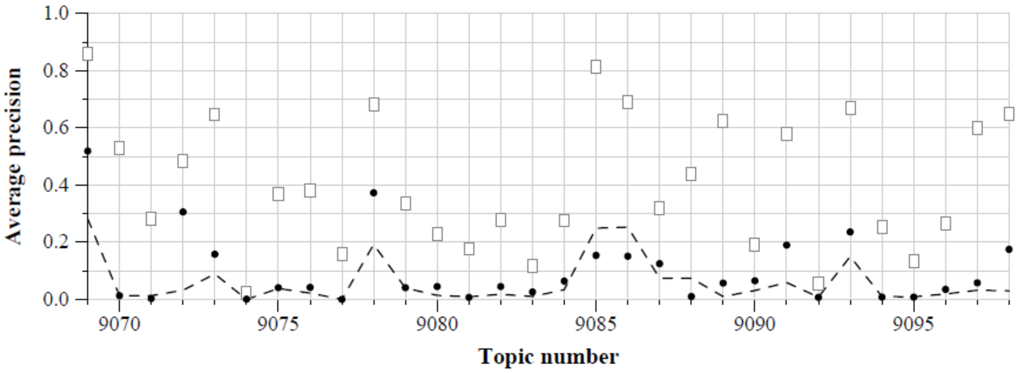

Figure 6: TNO INS results in average precision per query. The figure shows the run score (dots) versus median (dashed curve) versus best (boxes) by topic.

The processing time for performing a single INS query with a set of five images is for implementation on a single PC (Intel i7 CPU with multiple cores and 8Gb memory) in the order of 60 minutes. In our experiments we also implemented the proposed MapReduce scheme on our high-performance cluster. The cluster consists of 34 Linux nodes with in total about 280 cores and sufficient memory (8 - 128GB) per node. All binary chunks with SIFT descriptors used for matching are stored on a local, fast-accessible file server (250 Mb/s) to warrant quick access times. With MapReduce, processing times for queries increase with an estimated factor of ten times when compared with a single PC solution. Because all binary chunks are read from disk, multiple Map worker matching threads become I/O bound instead of the CPU bound on a single PC.

Figure 7  shows the TNO INS results in mean average precision over all queries with respect to the best runs of all other parties. The results show a ranking just above the middle (rank 11 of 22 parties).

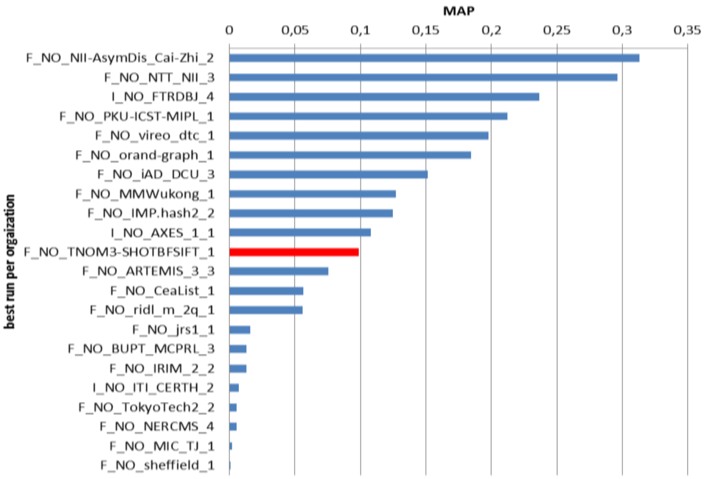

Figure 7: TNO INS results w.r.t. other participants (only best run per organization is shown).

## 5. CONCLUSIONS

In this paper, we described the TNO systems for MED and INS.

The MED results show that best low-level keypoint descriptor is SIFT, spatial tiling of each feature and fusion of multiple features help to improve the performance, and the textual event description currently has a poor performance.

The INS results show that with a baseline open-source approach using standard SIFT and brute-force matching one can obtain above median results. Using a basic map-reduce scheme for video pre-processing and matching can reduce processing times significantly.

## ACKNOWLEDGEMENT

This research for the MED submission was performed in the GOOSE project, which is jointly funded by the enabling technology program Adaptive Multi Sensor Networks (AMSN) and the MIST research program of the Dutch Ministry of Defense. The INS submission was partly supported by the MIME project of the creative industries knowledge and innovation network CLICKNL.

## REFERENCES

[1] Akbacak, M, Bolles, R., Burns, J., et al., "The 2012 SESAME Multimedia Event Detection (MED) and Multimedia Event Recounting (MER) Systems," TRECVID, (2012).

[2] Aly, R., McGuinness, K., Chen, S., et al., "AXES at TRECVid 2012: KIS, INS, and MED," TRECVID, (2012).

[3] Ayache, S., Quenot, G., "Video corpus annotation using active learning," ECIR, 187-198 (2008).

[4] Bouma, H., Burghouts, G., Penning, L. et al., "Recognition and localization of relevant human behavior in videos," Proc. SPIE 8711, (2013).

[5] Burghouts, G.J., Schutte, K., Bouma, H., Hollander, R. den, "Selection of negative samples and two-stage combination of multiple features for action detection in thousands of videos," Machine Vision and Applications MVA, (2013).

[6] Cao, L., Chang, S., Codella, N., et al., "IBM Research and Columbia University TRECVID-2012 Multimedia Event Detection (MED) Multimedia Event Recounting (MER) and Semantic Indexing (SIN) Systems," TRECVID, (2012).

[7] Chang, C., Lin, C., "LIBSVM: A library for support vector machines," ACM Trans. Intell. Systems and Technology 2(3), (2011).

[8] Cheng, H., Liu, J., Ali, S., et al., “SRI-Sarnoff AURORA System at TRECVID 2012; Multimedia Event Detection and Recounting,” TRECVID, (2012).

[9] Dean, J., Ghemawatt, S., “MapReduce: Simplified Data Processing on Large Clusters”, OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, USA, (2004).

[10] Deng, J., e.a., “Imagenet: A large-scale hierarchical image database,” IEEE CVPR, (2009).

[11] Heikkila, M., Pietikainen, M., Schmid, C., “Description of interest regions with center-symmetric local binary patterns”, Proc. ICVGIP LNCS 4338, 58-69 (2006).

[12] Inoue, N., Kamishima, Y., Mori, K., Shinoda, K., “TokyoTechCanon at TRECVID 2012”, TRECVID, (2012).

[13] Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., Schmid, C., “Aggregating local image descriptors into compact codes,” IEEE Trans. PAMI 34(9), 1704-1716 (2012).

[14] Laptev, I., “On space-time interest points,” Int. J. Computer Vision IJCV 64(2/3), 107-123 (2005).

[15] Lazebnik, S., Schmid, C., Ponce, J., “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” IEEE Conf. Computer Vision and Pattern Recognition CVPR, 2169-2178 (2006).

[16] Lin, D., “An information-theoretic definition of similarity,” ICML 98, 296-304 (1998).

[17] Lowe, D., “Object recognition from local scale-invariant features,” Int. Conf. Computer Vision, (1999).

[18] Marneffe, M., MacCartney, B., Manning, C., “Generating typed dependency parses from phrase structure parses,” LREC, (2006).

[19] Natarajan, P., Natarajan, P., Wu, S., et al., “BBN VISER TRECVID 2012 Multimedia Event Detection and Multimedia Event Recounting Systems,” TRECVID, (2013).

[20] Natarajan, P., Wu, S., Vitaladevuni, S., et al., “Multimodal feature fusion for robust event detection in web videos”, IEEE CVPR, 1298-1305 (2012).

[21] Ojala, T., Pietikainen, M., Maenpaa, T, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”, IEEE Trans. PAMI 24(7), 971-987 (2002).

[22] Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A., Quenot, G., “TRECVID 2013 - An overview of the goals, tasks, data, evaluation mechanisms and metrics,” TRECVID, (2013).

[23] Perera, A., Oh, S., Pandey, M., et al., “TRECVID 2012 GENIE: Multimedia Event Detection and Recounting,” TRECVID, (2012).

[24] Perronnin, F., Sanchez, J., Mensink, T., “Improving the Fisher kernel for large-scale image classification”, European Conf. Computer Vision ECCV, (2010).

[25] Sande, K., Gevers, T., Snoek, C., “Evaluating color descriptors for object and scene recognition,” IEEE Trans. PAMI 32(9), 1582-1596 (2010).

[26] Schavemaker, J., Versloot C., de Wit J., and Kraaij W., “TNO instance search submission 2012,” Proc. TRECVID 2012, (2012).

[27] Schutte, K., Bomhof, F., Burghouts, G. et al., “GOOSE: semantic search on internet connected sensors”, Proc. SPIE 8758, (2013).

[28] Snoek, C., Sande, K., Habibian, A., et al., “The MediaMill TRECVID 2012 Semantic Video Search Engine”, TRECVID, (2012).

[29] S. Strassel, A. Morris, J. Fiscus, et al., “Creating HAVIC: Heterogeneous audio visual internet collection,” LREC, (2012).

[30] Wang, F., Sun, Z., Zhang, D., Ngo, C., “Semantic Indexing and Multimedia Event Detection: ECNU at TRECVID 2012”, TRECVID, (2012).

[31] Yu, S., Xu, Z., Ding, D., “Informedia E-Lamp @ TRECVID 2012; Multimedia Event Detection and Recounting”, TRECVID, (2012).

[32] Zhao, G., Ahonen, T., Matas, J., Pietikainen, M., “Rotation-invariant image and video description with local binary pattern features”, IEEE Trans. Im. Proc. 21(4), 1465-1477 (2012).

[33] Zheng, F., Zhang, G., Song, Z., “Comparison of different implementations of MFCC,” Journal of Computer Science and Technology 16(6), 582-589 (2001).