Informedia @ TRECVID 2010

Huan Li[1,2], Lei Bao[1,3], Zan Gao[1,4], Arnold Overwijk[1], Wei Liu[1], Long-fei Zhang[1,5], Shoou-I Yu[1], Ming-yu Chen[1], Florian Metze[1] and Alexander Hauptmann[1]
[1]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]School of Computer Science and Engineering, Beihang University, Beijing, 100191, China
[3]Graduate University of Chinese Academy of Sciences, Beijing 100049, China
[4]School of Information and Telecommunication Engineering,
Beijing University of Posts and Telecommunications, Beijing 100876, China
[5]School of Software, Beijing Institute of Technology, Beijing, 100081, China
{lihuan0611,lei.bao.cn,zangaonsh4522,lwbiosoft,kevinzlf}@gmail.com
{arnold.overwijk,iyu,mychen,fmetze,alex}@cs.cmu.edu

Abstract

The Informedia group participated in four tasks this year, including Semantic indexing, Known-item search, Surveillance event detection and Event detection in Internet multimedia pilot. For semantic indexing, except for training traditional SVM classifiers for each high level feature by using different low level features, a kind of cascade classifier was trained which including four layers with different visual features respectively. For Known Item Search task, we built a text-based video retrieval and a visual-based video retrieval system, and then query-class dependent late fusion was used to combine the runs from these two systems. For surveillance event detection, we especially put our focus on analyzing motions and human in videos. We detected the events by three channels. Firstly, we adopted a robust new descriptor called MoSIFT, which explicitly encodes appearance features together with motion information. And then we trained event classifiers in sliding windows using a bag-of-video-word approach. Secondly, we used the human detection and tracking algorithms to detect and track the regions of human, and then just focus on the MoSIFT points in the human regions. Thirdly, after getting the decision, we also borrow the results of human detection to filter the decision. In addition, to reduce the number of false alarms further, we aggregated short positive windows to favor long segmentation and applied a cascade classifier approach. The performance shows dramatic improvement over last year on the event detection task. For event detection in internet multimedia pilot, our system is purely based on textual information in the form of Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR). We submitted three runs; a run based on a simple combination of three different ASR transcripts, a run based on OCR only and a run that combines ASR and OCR. We noticed that both ASR and OCR contribute to the goals of this task. However the video collection is very challenging for those features, resulting in a low recall but high precision.

1 Semantic Indexing (SIN)

In SIN task, we submit 4 runs this year. The first, the second and the forth runs are the full submissions whose results include all the 130 high level features. The third run is the light submission which submits the results for 10 high level features predifined.

1.1 Description of submissions

- CMU1_1: MoSIFT feature only, trained with χ^2 kernel for each high level feature.
- CMU2_2: Select the low level feature which has best performance on training data and then train a classifier based on it.

- CMU3_3: Cascade classifier is trained with four layers, and different layer is trained by using different visual feature.
- CMU4_4: Linearly combine the prediction results of the classifiers trained on MoSIFT feature, SIFT feature, color feature, audio feature and face feature.

1.2 Details of submissions

1.2.1 CMU1_1

In this run, we use MoSIFT [9] feature to train a SVM(Support Vector Machines) classifier for each high level feature. MoSIFT feature is a combination of SIFT feature with motion information. First MoSIFT points are detected for each keyframe, and then 1000 visual vocabularies are generated by using K-means. Each keyframe will be represented by a 1000 dimensional feature vector by mapping the MoSIFT point to its most similar vocabulary. In the process of mapping, we consider the N (N = 4) nearest neighbor vocabularies for each point and assign different weights to them according to their distance rank. In the process of training, we do a two-fold cross-validation on the development set for finding the best parameter and then train a model by using all the training dataset. χ^2 kernel is used in SVM because it shows better performance for calculating histogram distance [35].

1.2.2 CMU2_2

As same as last year [10], this year we extract 5 different kinds of low level features for each keyframe, including MoSIFT, SIFT, Grid-based color moments(GCM), Face, Mel-frequency cepstral coefficients (MFCCs). The development set are separated to 3 folders. The first two folders are used do a two-fold cross-validation for finding the best parameter of each low level feature for each high level feature. Then a model is trained on these two folders and tested on the other unused folder. For each high level feature, we use average precision to evaluate the classifiers trained on different low level features and use the low level feature that has the best performance to train a classifier on the whole development set and then test it on the evaluation set for submission.

1.2.3 CMU3_3

This run is for the light submission which submits 10 high level features predefined. Cascade method is used. Following are the details.

The key idea of cascade is inherited from AdaBoost [15] which combines a collection of weak classifiers to get a strong classifier. The classifiers are called weak because they are not expected to have the best performance in classifying the whole training data. In order to boost weak classifiers, each classifier emphasizes the examples which are incorrectly classified by the previous weak classifiers. In our task, simpler classifiers are first used to reject the majority of negative samples before more complex classifiers are called upon to achieve low false positive rates. Each weak classifier keeps most of the positive examples but rejects a good number of negative examples. Face detection has shown that the cascade architecture can reduce false positives rapidly but keep a high detection rate.

Firstly, we divide the negative samples into four parts, named Part-I, Part-II, Part-III and Part-IV, where each part has the same positive samples. For each part, the MoSIFT, SIFT, GCM and Texture are extracted in each frame or its neighbor frame. After that, the MoSIFT cascade SVM is trained on Part-I, and then test it on Part-II. In the Part-II, if a negative sample is predicted as positive sample, we will keep this negative sample, otherwise we will discard it. Thus, the Part-II has been filtered by the MoSIFT model. The SIFT model will be trained on the Part-II, and we will test it on the Part-III. For the GCM and Texture models will be trained by the same way. The training processing is illustrated in Figure 1.

1.2.4 CMU4_4

In order to consider the influence of all low level features, we use constant weights for each feature and linearly combine them to get a prediction score for each keyframe, and then rank them to get the top ones for each high level feature as the submision result.

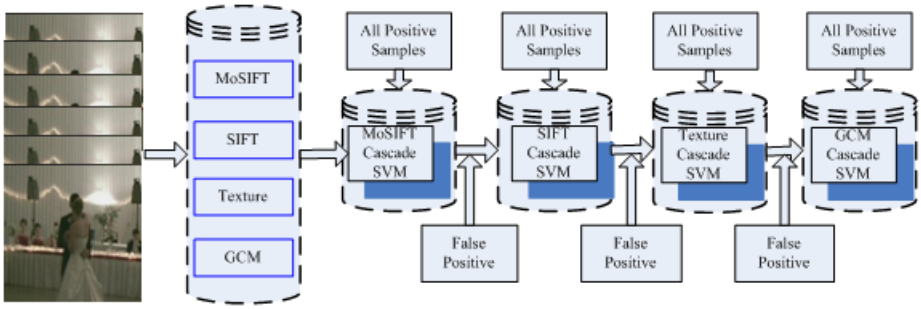


Figure 1: Framework of cascade method for semantic indexing.

2 Known-item search (KIS)

2.1 Description of submissions

In Known Item Search (KIS) task, we submitted 4 runs this year.

- CMU1_1: We classified all queries into 5 classes and optimized the weights of different query types and different text fields for each query class.
- CMU2_2: We classified all queries into 5 classes and optimized the weights of different text fields in keyword query.
- CMU3_3: We took all queries as 1 class and optimized the weights of different text fields in keyword query.
- CMU4_4: We linearly combined runs from text-based retrieval and visual-based retrieval.

2.2 System introduction

- Text-based Retrieval with Lemur: the availability of the metadata make text-based retrieval can be the most effective solution. In addition to released metadata, we also extracted the Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR) from videos. Furthermore, to get more informative query description, we used Flickr API [1] to filter and expand the query. Finally, with Lemur [3], we could weight the results from different query type in different fields.
- Visual-based Retrieval: since the image/video examples are unavailable this year, we expand the image examples from Google Images [2]. That allowed us to build a content-based video retrieval with Bipartite Graph Propagation Model [6]. In addition, we also added 12 color concept detectors in addition to the SIN 130 concept detectors to improve the concept-based retrieval, as the color information is also important in the KIS queries. Furthermore, we used Latent Dirichlet Allocation (LDA) [7] to exploit the correlations between texts (metadata) and visual feature (video). This is our multimodal-based retrieval.
- Late Fusion: because the performance of different runs varies over the queries, optimizing the fusion weights for all the queries was not sufficient. This year, we automatically clustered the queries into different classes, based on their relevant scores from different runs. Finally we optimized the fusion weights for each query class.

2.3 Text-based Retrieval with Lemur

We used Lemur [3] to build our Text-based Retrieval system. The most important things in Lemur is the field index and query creativity.

For field index, firstly we analyzed the metadata and discovered that there are 74 different fields in total. Most of those fields do not contain relevant information. They contain non relevant information such as the upload date, non discriminative information such as whether the video is in color and rare fields that only occur in a small number of videos. We decided that the most informative fields were *description*, *keywords* and *title*. In addition, we added Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR). For all those fields we used Lemur [3] to create a model $M_{f_1}, \ldots, M_{f_5}$ for each field and also a model M_{f_6} that is a combination of all those fields.

For query creativity, in addition to the released keyword query and visual cues,we also used Flickr API [1] to filter and expand keywords query and visual cues. Therefore, we get two kinds of filtered queries: keyword query filtered by Flickr and visual cues filtered by Flickr, which only keeps the word that appears in Filckr tag. we also get two kinds of expansion queries: keyword query expansion by Flickr and visual cues expansion by Flickr, which only expand each word with the top 10 related tags in Flickr.

We calculate the optimal weights for each field and each query type based on an exhaustive search and cross fold validation. Then the weighted beliefs for all the terms are combined resulting in a score for the document. This is illustrated in Figure 2

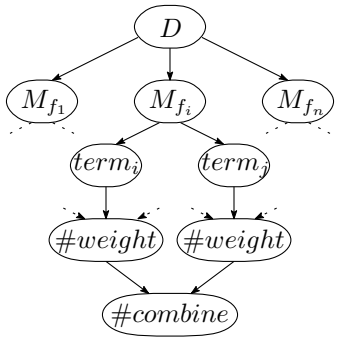


Figure 2: Indri Retrieval Model

2.4 Visual-based Retrieval

To make use of the visual information, we tried three different approaches: concept-based retrieval, content-based retrieval, and multimodal-based retrieval.

2.4.1 Concept-based Video Retrieval

The flow of concept-based video retrieval is: lexicon construction, concept detection, query-to-concept mapping.

For lexicon construction and concept detection, we first choose the 130 concepts in SIN task. Then we noticed that a lot of queries contain color information; therefore we also trained 12 color concept detectors. The training images are downloaded from Google Images. For query-to-concept mapping, we simply check whether the concept appears in the query keywords or not.

The 12 color concept detectors improved our concept-based retrieval from 0.0043 to 0.0061 on the evaluation queries.

2.4.2 Content-based Video Retrieval

Since the image/video examples are unavailable for KIS task, it is hard to make use of the visual information. Therefore we used visual cues as query in Google Images and took the top 20 relevant images as our image examples for KIS.

We discovered the latent topic in testing collection by Latent Dirichlet Allocation (LDA) [7], where the SIFT bag-of-word feature was used and the topic number is set to 200. For the image examples and keyframes in each testing video, we get their predicting score in each latent topic. Then using the Bipartite Graph Propagation Model in [6], we can get the relevance scores between query and latent topics. Finally, we can use these scores to linearly combine their predicting scores in videos, and get the relevant scores between query and videos.

The Mean Inverted Rank on the evaluation queries is 0.0047.

2.4.3 Multimodal-based retrieval

To exploit the correlations between texts (metadata) and visual features (video). We use LDA to describe the joint distribution of text bag of words and the SIFT bag of words feature in the video

collection. This allows us to represent each video in a latent topic subspace as well as the query and build a bridge between text and visual feature with the latent topic subspace.

The Mean Inverted Rank on the evaluation queries is 0.0032.

2.5 Late Fusion

We used linearly combination to fusion to results from text-based system and visual-based system. However, as the performance varies over the queries, it wouldn't be effective to optimize the fusion weights for all the queries. As the previous works in [33, 34],query-class dependent weights optimization is a more reasonable strategy.

The query classifications in [33,34] are both from human perspective. In [34], four query classes are predefined by human. They are Named person, Named object, General object and Scene. In [33], each query is described by some binary query features, i.e. if the query topic contains 1) specific person names, 2) specific object names, 3) more than two noun phrase, and so on. Since that, when the characteristics of different runs don't correspond to these query features or predefined class, the query classification will fail. This year, we describe each query by its results from different runs to find the corresponding query class.

For weight optimization, Logistical Regression was used in [33, 34]. However, for KIS, since the number of answers of each query is only one, the ratio of positive samples to negative samples is 1/8282. The training data is very unbalanced. That leads to the fail of logistical regression in KIS task. Therefore, we used an exhaustive search and cross fold validation to find the optimized weights.

2.6 Results and Discussion

In text-based system, with different query type and different field, we can get different runs. Considering the computational cost of the exhaustive search, we optimized the weights of different fields in each query type and then optimized the weights for different query type. This is our submitted CMU1_1 run.

To evaluate our performance the fusion of different fields and query-class dependent fusion, we submitted two runs: one optimized weights of different text fields for keyword query in 5 classes, the other optimized in 1 class. The former is CMU2_2 run. The latter is CMU3_3 run.

The evaluation results of these three runs are in Table 1.

- **Average Fusion vs. Optimized Fusion:** Comparing the row 1 and row 2 in Tabel 1, we can find the Optimized Fusion increases the performance of the Average Fusion from 0.234 to 0.243. The optimized weights that found by an exhaustive search and cross fold validation in 122 training queries did work in 300 testing queries.
- **Querytype Fusion vs. Field Fusion:** Comparing the row 2 with row 4 in Tabel 1, we can find that the performance decreases from 0.243 to 0.234 after Querytype Fusion. Since we did two level weights optimization, it could make the parameters over-fitting in training queries. This also can explain the decreasing from 0.253 (row 3) to 0.214 (row 6).
- **Single class vs. Multi-class:** Comparing the performance of 1 class optimization with 5 classes optimization(as shown in the row 2 and row in Table 1), we can find the query-classes improves the performance from 0.243 to 0.253. The demonstrate the affectivity of query-class optimization.

The last submitted CMU4_4 run is a linear combination of the text based retrieval and the visual based retrieval. Since the performance of visual-based runs are very low (less than 0.01) in training queries, and for most of queries, the inverted rank is 0. We just fixed very small weights for visual-based runs. Finally, the mean inverted rank of CMU4_4 is 0.231, in comparison to 0.243, that is the performance of text-based fusion in CMU3_3. That means we didn't get any improvement from the visual part.

Since the text-based runs significantly outperform the visual-based runs, it is hard to effectively fusion their results. As the computational cost of the exhaustive search, we could not find the best weights to combine them in single query class or multi query classes. However, based on the query-class dependent fusion results in text-based runs, we can see it is a promising pproach to perform

late fusion. In the future work, we will design a more practical algorithm to optimize fusion weights on query classes.

Table 1: The Mean Inverted Rank of Different Runs.

	Training Queries	Testing Queries
Average fusion of different text fields for keyword query	0.263	0.234
CMU3-3:Optimized fusion of different text fields for keyword query in 1 class	0.279	0.243
CMU2-2: Optimized fusion of different text fields for keyword query in 5 classes	**0.338**	**0.253**
Optimized fusion of different query types and different text fields in 1 class	0.297	0.234
CMU1-1: Optimized fusion of different query types and different text fields in 5 classes	**0.354**	**0.214**

3 Surveillance event detection (SED)

Surveillance video recording is becoming ubiquitous in daily life for public areas such as supermarkets, banks, and airports. Thus it attracts more and more research interests and experiences rapid advances in recent years. A lot of schemes have been proposed for the human action recognition, among them, local interest points algorithm have been widely adopted. Methods based on feature descriptors around local interest points are now widely used in object recognition. This part-based approach assumes that a collection of distinctive parts can effectively describe the whole object. Compared to global appearance descriptions, a part-based approach has better tolerance to posture, illumination, occlusion, deformation and cluttered background. Recently, spatio-temporal local features [17, 18, 24, 27, 31, 32] have been used for motion recognition in video. The key to the success of part-based methods is that the interest points are distinctive and descriptive. Therefore, interest point detection algorithms play an important role in a part-based approach.

The straightforward way to detect a spatio-temporal interest point is to extend a 2D interest point detection algorithm. Laptev et al. [18] extended 2D Harris corner detectors to a 3D Harris corner detector, which detects points with high intensity variations in both spatial and temporal dimensions. On other words, a 3D Harris detector finds spatial corners with velocity change, which can produce compact and distinctive interest points. However, since the assumption of change in all 3 dimensions is quite restrictive, very few point results and many motion types may not be well distinguished. Dollar et al. [13] discarded spatial constraints and focused only on the temporal domain. Since they relaxed the spatial constraints, their detector detects more interest points than a 3D Harris detector by applying Gabor filters on the temporal dimension to detect periodic frequency components. Although they state that regions with strong periodic responses normally contain distinguishing characteristics, it is not clear that periodic movements are sufficient to describe complex actions. Since recognizing human motion is more complicated than object recognition, motion recognition is likely to require with enhanced local features that provide both shape and motion information. So MoSIFT algorithm [9] are proposed, which detects spatially distinctive interest points with substantial motions. They first apply the well-known SIFT algorithm to find visually distinctive components in the spatial domain and detect spatio-temporal interest points with (temporal) motion constraints. The motion constraint consists of a "sufficient" amount of optical flow around the distinctive points.

However, in the local interest point algorithms, most of them [13, 17, 18, 24, 27, 31, 32] did not care where the interest points located, as their experiment scenes are relative simple and clear, and most of conditions, just one or two people have some actions. However, these conditions seldom hold in real-world surveillance videos. Even the same type of actions may exhibit enormous variations due to cluttered background, different viewpoints and many other factors in unconstrained real-world environment, such as TREC Video Retrieval Evaluation (TRECVID) [4]. To our best knowledge, TRECID has made the largest effort to bridge the research efforts and the challenges in real-world conditions by providing an extensive 144-hour surveillance video dataset recorded in London Gatwick Airport. In this dataset, the cameras are fixed, but the scenes are very complex, and there are a lot of people walking through on the scenes. Thus, if we just adopt the local inter-

est points to detect the events on the scene, there are a lot of noise interest points for some events. In TRECVID 2010 Evaluation, there are 7 required events such as CellToEar, Embrace, ObjectPut, Pointing, PeopleMeet, PeopleSplitUp and PersonRuns. All of them are relative to the human. Therefore, we will use some human detection and tracking approaches to locate these interest points, and filter the noise interest points. Finally, we also adopt the results of human detection to estimate the correctness of detection.

3.1 System introduction

For the tasks in TRECVID 2010 Event Detection Evaluation, we focus on human-related events. We mainly follow the framework we employed in TRECVID 2009 Evaluation, which incorporates interesting point extraction, clustering and classification modules. In TRECVID 2009 Evaluation, the MoSIFT interesting points are extracted for each video firstly, and then bag-of-features are adopted. After that, the cascade SVM will be trained. The details can be viewed in [10].

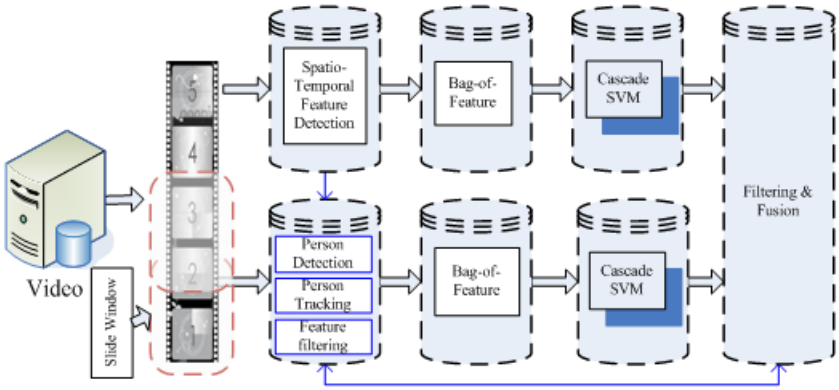


Figure 3: Framework of surveillance event detection.

However, we extend our framework by three kinds of ways. Firstly, for the classification modules, different numbers of layers cascade SVM are trained. Secondly, for each frame, the MoSIFT points are extracted, but they maybe have activities in these frames, and we can not discriminate them. Thus, the human detection and tracking are adopted. We split the activities into many parts according to the results of human detection and tracking, and just use these MoSIFT points located in the region of human. Thirdly, when making the decision, there are a lot of false alarms, so we will filter the decision according to results from the human detection and tracking. If there are no human in the frame, but the decision shows there are some activities, and we will think this is a false alarm. After getting the probabilities, we will fuse these results. In addition, to reduce the number of false alarms further, we aggregated short positive windows to favor long segmentation. The system framework is illustrated in the Figure 3.

3.2 MoSIFT Feature Based Action Recognition

For action recognition, there are three major steps: detecting interest points, constructing a feature descriptor, and building a classifier. Detecting interest points reduces the video from a volume of pixels to compact but descriptive interest points.

This section outlines our algorithm [9] to detect and describe spatio-temporal interest points. It was shown [9] to outperform the similar Laptev's method [18]. The approach first applies the SIFT algorithm to find visually distinctive components in the spatial domain and detects spatio-temporal interest points through (temporal) motion constraints. The motion constraint consists of a "sufficient" amount of optical flow around the distinctive points.

3.2.1 Motion Interest Point Detection

The algorithm takes a pair of video frames to find spatio-temporal interest points at multiple scales. Two major computations are applied: SIFT point detection [19] and optical flow computation matching the scale of the SIFT points.

SIFT was designed to detect distinctive interest points in still images. The candidate points are distinctive in appearance, but they are independent of the motions in the video. For example, a cluttered background produces interest points unrelated to human actions. Clearly, only interest points with sufficient motion provide the necessary information for action recognition.

Multiple-scale optical flows are calculated according to the SIFT scales. Then, as long as the amount of movement is suitable, the candidate interest point contains are retained as a motion interest point.

The advantage of using optical flow, rather than video cuboids or volumes, is that it explicitly captures the magnitude and direction of a motion, instead of implicitly modeling motion through appearance change over time.

Motion interest points are scale invariant in the spatial domain. However, we do not make them scale invariant in the temporal domain. Temporal scale invariance could be achieved by calculating optical flow on multiple scales in time.

3.2.2 Person Area Detection Based Feature Filter

MoSIFT feature does a great job in human behavior representation for human action recognition. However, Are the MoSIFT interesting points caused by human? The MoSIFT points might be caused by moving, light shaking, or shadow. If we could sample the MoSIFT points from human body or area containing people, we might get much more accurate results. Thus, in this section, we use person detection and tracking method to filter the MoSIFT point, and only keep the MoSIFT point in human area for further use.

- Person Detection
  Person detection is the most direct method to detect the area of human. Histogram of Oriented Gradient (HOG) feature [11] and Haar like feature [30] are the most popular features used in person detection. Locally normalized HOG descriptors are computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalizations for improved performance. Haar like feature person detection used in VJ(Viola and Jones) works is using AdaBoost to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences. It consists of a filter that takes image windows from n consecutive frames as input, a threshold and a positive and negative vote. Since there are too many people in Gatwick surveillance video(especially camera 2, 3 and 5) , full body person detection is very limited in detecting the person blinded by some background objects, such as showed in Figure 4. In our experiments, both HOG person upper/full body detectors and Haar person upper/full body detectors are trained on the development videos in Dev08 and INRIA dataset, and then the person detection results are adopted to initialize the tracking objects, and finally we get the effective regions of people according to tracking result.
- People tracking
  However, detection people is a challenging problem, especially in complex real world scenes, such as the Gatwick surveillance video, commonly involved multiple people, complicated occlusions, and cluttered or varied illuminate backgrounds. High false positive and low recall rate in human detection make the detection unreliable and cannot help much to filter the MoSIFT point.

  Tracking the person is another challenging problem in computer vision, but we use multiple objects tracking to increase the recall of the person detection results. We use an ensemble tracking algorithm which is based on particle filter tracker [20] and Multiple Instance tracker [5] to track all the persons detected by person detection procedure.

  For example, suppose the duration of an event is from 1 to 20 frames. From the first frame, we can detect the people, but miss detection in the second frame. Then, we use tracking method to track the person detected in the first image. And then, we add the person detected in the second frame into the tracking object list, and track both objects detected in the first frame and the second frame to generate the region of people. We maintain the tracing object list and the temporary person detection objects list and determine which one in detection objects list should be add to the tracking objects and which one should be remove from the tracking object list. The tracking object list should be the basis of feature filter for further use.

  However, this is just a forward detection and tracking procedure to find where the people are, we also perform the backward tracking to improve the recall of detection and tracking.

After that, we can find the human region in the very frame as many as we can. The same way has been used in decreasing the high false positive rate.

Figure 4: Illustration of SIFT (left), MoSIFT (middle) and People detection (right).

- Motion and Appearance Feature Description
  After getting the MosSIFT interest points, we need describe these points. Appearance and motion information together are the essential components for an action classifier. Since an action is only represented by a set of spatio-temporal point descriptors, the descriptor features critically determine the information available for recognition.

  The motion descriptor adapts the idea of grid aggregation in SIFT to describe motions. Optical flow detects the magnitude and direction of a movement. Since, optical flow has the same properties as appearance gradients, the same aggregation can be applied to optical flow in the neighborhood of interest points to increase robustness to occlusion and deformation.

  The main difference to appearance description is in the dominant orientation. For human activity recognition, rotation invariance of appearance remains important due to varying view angles and deformations. Since our videos are captured by stationary cameras, the direction of movement is an important (non-invariant) vector to help recognize an action. Therefore, our method omits adjusting for orientation invariance in the motion descriptors.

  Finally, the two aggregated histograms (appearance and optical flow) are combined into the descriptor, which now has 256 dimensions.

3.3 Experiments and Discussion

Our event detection using a sliding window framework is applied to extend the MoSIFT recognition algorithm to a detection task. Our submission started with MoSIFT interest points in each window, clustered them into visual keywords, and used a classifier to detect events based on trained SVM models. Figure 4 shows our MoSIFT features in a Gatwick video key frame. It shows that MoSIFT feature is able to clearly focus on areas with human activity.

We assume that an event can be described though a combination of these different types of small motions. MoSIFT is a scale invariant local feature which is less affected by global appearance, posture, illumination and occlusion. After getting the MoSIFT, we try to use bag-of-words (BoW) to quantify MoSIFT feature to a fixed number vector feature of each key frame. We use K-means clustering to find the conceptual meaningful clusters and each cluster is treated as a visual word in BoW approach. All the visual words consist of a visual word vocabulary. Then key points in each key frame are assigned to clusters in the visual vocabulary which are their nearest neighbors. In the end, each key frame is presented by a visual word histogram feature. In our experiments, the vocabulary size is 2000, and a soft boundary to form our bag-of-word features is applied. We also apply a kernel SVM [8] and one-against-all strategy to construct action models.

In our experiments, the size of the window is 25 frames (1 second) and it repeats every 5 frames. In the training set, annotations are distributed to each window to mark it as positive or negative. This creates a highly unbalanced dataset (positive windows are much less frequent than negative windows). Therefore, we build a one, five and ten layers cascade classifier to overcome this imbalance in the data and reduce false alarms. For each layer, we choose an equal ratio of (positive v.s. negative) training data to build a classifier to favors to positive examples. This leads the classifier with high detection rates. In the training process, the cross-validation is adopted. By cascading five or ten layers of these high detection rate classifiers, we can efficiently eliminate a good amount of false positives without losing too many detections. We also aggregate consecutive positive predictions to achieve multi-resolution.

In the Table 2, it was from our TRECVID2009, and Table 3, 4 and 5 are from TRECVID2010. When we training one layer cascade SVM, four of seven events are less than 1 in MinDCR, but when five or ten layers cascade SVM are trained, five of seven events are less than 1 in MinDCR. Compared with our result from last year, MoSIFT and the cascade classifier significantly improved our performance. In addition, five and ten layers cascade SVM can eliminate a good amount of false positives, but the performance of ten layers cascade SVM is not much better than that in five layers cascade SVM. Thus, in the future, we do not need train more than five layers cascade SVM for the task.

In TRECVID 2009 Event Detection Evaluation [4], they provide 99 hours videos in the development set and about 44 hours videos in the evaluation set, where the videos were captured using 5 different cameras with image resolution 720576 at 25 fps. From the statistics of events in the development set, we find out there are hardly any events in the videos of CAM4, so we exclude those videos from our experiments to save some computation power. Even though, it will be very difficult to compute so huge dataset. For some reasons, we can not finish all the experiments we design, but our performance still has some improvement comparing to our TRECVID2009. In the following table, RFA denotes Rates of False Alarms. PMiss denotes probability of missed event. DCR denotes Detection Cost Rate.

Table 2: Our SED results of TRECVID2009.

Analysis Report	# Ref	# Sys	# CorDet	# FA	# Miss	Act.RFA	Pmiss	Act.DCR	MinRFA	MinPMiss	MinDCR
CellToEar	194	22658	100	22558	94	1479.483	0.484	7.882	0.066	1	1
Embrace	175	20080	146	19934	29	1307.386	0.166	6.703	1.377	0.989	0.996
ObjectPut	621	2353	42	2311	579	151.569	0.932	1.69	0.066	1	1
PeopleMeet	449	854	58	796	391	52.206	0.871	1.132	0	0.998	0.998
PeopleSplitUp	187	9351	28	9323	159	611.456	0.85	3.907	0.721	0.995	0.998
PersonRuns	107	20799	87	20712	20	1358.411	0.187	6.979	0.066	1	1
Pointing	1063	6968	230	6738	833	441.917	0.784	2.993	0.066	0.999	0.999

Table 3: SED results of TRECVID2010, using one layer cascade SVM.

Analysis Report	# Ref	# Sys	# CorDet	# FA	# Miss	Act.RFA	Pmiss	Act.DCR	MinRFA	MinPMiss	MinDCR
CellToEar	194	1787	14	1773	180	116.284	0.928	1.509	0.066	1	1
Embrace	175	5890	113	5777	62	378.889	0.354	2.249	28.005	0.846	0.986
ObjectPut	621	1961	45	1916	576	125.662	0.927	1.556	0.066	1	1
PeopleMeet	449	5814	197	5617	252	368.395	0.561	2.403	2.164	0.969	0.98
PeopleSplitUp	187	2784	42	2742	145	179.836	0.775	1.675	2.755	0.984	0.998
PersonRuns	107	5741	61	5680	46	372.527	0.43	2.292	7.214	0.925	0.961
Pointing	1063	2992	180 2812	883	184.427	0.831	1.753	0.066	1	1	

Table 4: SED results of TRECVID2010, using five layers cascade SVM.

Analysis Report	# Ref	# Sys	# CorDet	# FA	# Miss	Act.RFA	Pmiss	Act.DCR	MinRFA	MinPMiss	MinDCR
CellToEar	194	39	0	39	194	2.558	1	1.013	0.066	1	1
Embrace	175	410	16	394	159	25.841	0.909	1.038	1.574	0.983	0.991
ObjectPut	621	20	1	19	620	1.246	0.998	1.005	0.066	1	1
PeopleMeet	449	305	24	281	425	18.43	0.947	1.039	0.525	0.987	0.989
PeopleSplitUp	187	31	2	29	185	1.902	0.989	0.999	1.443	0.989	0.997
PersonRuns	107	583	19	564	88	36.99	0.822	1.007	1.049	0.944	0.949
Pointing	1063	183	25	158	1038	10.363	0.977	1.028	0	0.999	0.999

4 Event detection in Internet multimedia(MED)

4.1 System introduction

We developed a general system that is independent of the concept. Our system classifies each video into two classes, the video either belongs to the given concept or not. This classification is solely based on ASR and OCR. We extracted speech transcripts with 3 different segmentations as described in Section 4.1.1. In addition we extracted OCR as described in Section 4.1.2.

We combined the three different speech transcripts into one bag of words on which we trained a SVM classifier for each concept. For this we used the LIBSVM implementation [8]. We used 2 fold cross validation to optimize the parameters for the Normalized Detection Cost [28]. Similarly we trained a SVM classifier for the OCR as well as a combined SVM on both of the outputs of our two SVM classifiers.

Table 5: SED results of TRECVID2010, using ten layers cascade SVM.

Analysis Report	# Ref	# Sys	# CorDet	# FA	# Miss	Act.RFA	Pmiss	Act.DCR	MinRFA	MinPMiss	MinDCR
CellToEar	194	57	0	57	194	3.738	1	1.019	0.066	1	1
Embrace	175	551	26	525	149	34.432	0.851	1.024	0.262	0.989	0.99
ObjectPut	621	26	1	25	620	1.64	0.998	1.007	0.328	0.998	1
PeopleMeet	449	388	27	361	422	23.676	0.94	1.058	0.197	0.989	0.99
PeopleSplitUp	187	42	3	39	184	2.558	0.984	0.997	2.23	0.984	0.995
PersonRuns	107	532	19	513	88	33.645	0.822	0.991	2.23	0.925	0.936
Pointing	1063	219	26	193	1037	12.658	0.976	1.039	0	0.999	0.999

4.1.1 Automatic Speech Recognition

An automatic transcription of the audio track of the videos was generated by a simple speech-to-text (STT) system. This consisted of the first pass of a "Rich Transcription" system developed for and successfully evaluated in the NIST RT-04S "Meeting" evaluation.

The system used a robust front-end and was trained on a variety of sources, including Broadcast News (BN) and "Meeting" audio, but no Web-, Youtube, or home-made material. The system is built for American English and uses a vocabulary of about 40 k words, plus models for silence, human, and non-human noises, plus entries for mumbled words and multi-words. The language model was trained on BN and Meetings. It is implemented in Janus [14], using the Ibis decoder [29]. A detailed description of the original STT system can be found in [21, 22]. Prior to transcription, the signal is segmented, using audio information alone. For this, three approaches were evaluated:

- a simple segmentation using fixed 10 s segments as "baseline"
- a first segmentation and clustering into silence, noise, music, and speech classes [16], of which only the speech part is processed by the STT system
- processing all segments with STT, neglecting the clustering information, so that the noise models can handle non-speech events

The output of the recognizer consists of a word string (cluster information is not currently used), including "gamma" word confidences [26]. Table 6 shows the influence of the segmentation on the generated output, and Table 7 shows overall characteristics of the generated segmentation.

Table 6: Characteristics of different segmentations. "speech-only" consists of 48.9 h from the overall corpus.

Segmentation	# segments	# lexical	# non-lex	type
test-1	42 247	427 393	125 460	10 s fixed
test-2	66 763	315 516	58 021	speech-only
test-3	327 811	978 657	367 593	all

The total database consists of 3290 "speakers", or clips, with a total duration of 123h. Audio was extracted and processed using publicly available tools. The system runs in about real-time on "clean" audio; the use of a Condor scheduler, plus the large proportion of mismatched audio however provide for a significantly larger wall-clock processing time.

Table 7: Overall audio characteristics and segmentation.

Type	silence	music	noise	speech
average duration	1.0 s	6.5 s	2.4 s	2.6 s
number	16 428	16 778	59 388	66 763

Word error rates (WER), or similar measures have not been computed for the speech recognition output, however it appears that the accuracy of audio analysis is reasonable for some clips (WER $<$ 20 %), while for others, notably those far from the original domain of the STT system, they are very bad. Confidence measures and the availability of non-lexical output of the STT system may provide additional useful input to the overall classification module.

4.1.2 Optical Character Recognition

We used the Informedia system [12] to extract the OCR. Our assumption is that text needs to be visible at least one second for humans to be readable. Therefore we sample one frame per second and consider this frame for text extraction. A text localization step extracts candidate text lines from each image by applying a series of filters. First text blocks are identified, where a text block is a rectangular region containing one or more lines of text, based on the assumption that text blocks consist of short edges in vertical and horizontal orientations. Moreover it assumes that those edges are connected to each other, because of the connection between character strokes. Edge detection is done using a Canny filter and morphological operators then perform edge dilation in both vertical and horizontal direction. Then a single image is created by combining both the horizontal and vertical edges after dilation as illustrated in Figure 5.

(a) Horizontal Edge Dilation (b) Vertical Edge Dilation (c) Combined Image

Figure 5: Illustration of Edge Dilation.

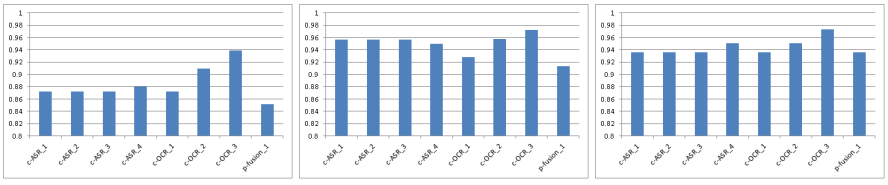
In the next step text blocks are extracted by doing a connected component analysis and computing the external contour of a region. At this point the system has a high recall, but also quite some false alarms mainly caused by slanting stripes and small areas of the background or human faces consisting of sharp edges. In order to reduce the number of false alarms and refine the location of text strings in candidate regions that contain text connected with background objects, individual text lines are identified. Then the system classifies extracted text lines into actual text regions, which is called text verification.

The final phase is the recognition, which is performed by a commercial OCR system. Before the text line can be processed by such a system, the image needs to be binarized. In this binarization step the text is extracted from the background by using Otsu's algorithm [25], which creates a histogram of the image and selects a threshold to maximize interclass variance. The resulting binary image is given to a commercial OCR system, which in this case is Textbridge OCR [23].

4.2 Results and Discussion

We submitted 3 runs. One based on ASR, one run based on OCR and a combination run. This resulted in 8 runs with different thresholds: c-ASR-1,2,3,4, c-OCR-1,2,3 and p-fusion-1. For the combination run, we used a meta fusion strategy which takes the component probability output as input and outputs an overall prediction.

Figure 6 shows an overview of all the submitted runs for the three different events. Our best run is p-fusion-1, it detects more positive samples than all the other runs. Moreover it has zero false alarms in contrast to the best ASR and OCR runs. This indicates that a late fusion helps improving the detection and even makes the system more robust against noise.



(a) Battling in Run (b) Assembling Shelter (c) Making Cake

Figure 6: All runs's NDC score.

Table 8: Best NDC, Average NDC, and Our Best NDC for three events.

	Best NDC	Average NDC	Our Best NDC
Battling in Run	0.4074	0.8285	0.8511
Assembling Shelter	0.7284	1.3100	0.913
Making Cake	0.6298	1.2330	0.9362

Table 9: Our system's performance on the 1724 videos of the test collection.

	battling in run			assembling shelter			making cake		
	#CorDet	#FA	#Miss	#CorDet	#FA	#Miss	#CorDet	#FA	#Miss
c-ASR_1	6	0	41	2	0	44	3	0	44
c-ASR_2	6	0	41	2	0	44	3	0	44
c-ASR_3	6	0	41	2	0	44	3	0	44
c-ASR_4	6	1	41	3	2	43	3	2	44
c-OCR_1	6	0	41	4	2	42	3	0	44
c-OCR_2	6	5	41	4	6	42	3	2	44
c-OCR_3	6	9	41	4	8	42	3	5	44
p-fusion_1	7	0	41	4	0	42	3	0	44

Table 8 shows the Best NDC, Average NDC, and Our Best NDC for the three events. The performance of our system is above average according to the evaluation criteria.

Table 9 shows the detection results and false alarms for all the runs for the event "battling in run". Obviously our system is very reserved, this can be explained by the fact that we depend on ASR and OCR only. Both those features are challenging to extract from internet videos, because there is a huge variation in quality. Therefore the ASR and OCR are very noisy and often contain noise only. We noticed that both the ASR and OCR either perform very well or very poor, i.e. we did not find many recognized words with small errors. This was also reflected in the confidence scores of our SVM classifiers.

Interesting to see is that there were ingredients such as corn, butter and sauce among the most frequent words recognized by our OCR. Those words would strongly indicate that the video belongs to the 'making a cake' event. The ASR on the other hand contains much more stopwords and less words that strongly indicate one of the events. This is also reflected in the results, hence the OCR runs are slightly better than the ASR runs.

4.3 Conclusion & Future Work

We noticed that our system is only able to detect a relatively small number of positive samples, but with high precision. This is caused by the fact that this video collection is very challenging for ASR and OCR systems. However there is plenty of room for improvement to make the most out of ASR and OCR. The most obvious would be improving the performance of the OCR and ASR systems itself.

However there are alternatives that we did not explore yet, one of them is the possibility of detecting certain sounds that might indicate a certain event. Consider for example the 'battling in run' event, here sounds like a cheering crowd and shouting would distinguish those videos from the other events. Another unexplored area is using the event descriptions, currently we ignored those and only made use of the development videos and their labels. Furthermore we did not use any visual features such as SIFT, color histograms, gist and motion features such as optical flow.

We conclude that both ASR and OCR showed their contribution to being able to recognize videos of certain events. In particular those features are able to detect videos with high precision. In the future we plan to incorporate visual features as well as improving upon our ASR and OCR performance.

5 Acknowlegments

This work was supported by the Nation Science Foundation under Grant No. IIS-0205219 and Grant No. IIS-0705491.

References

[1] Flickr api. http://www.flickr.com/services/api/.

[2] Google images. http://www.google.com/imghp?hl=entab=wi.

[3] Lemur. http://www.lemurproject.org/.

[4] Trecvid 2009 evaluation for surveillance event detection. http://www.nist.gov/speech/tests/trecvid/2009/.

[5] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009.

[6] L. Bao, J. Cao, Y. Zhang, M. Chen, J. Li, and A. G. Hauptmann. Explicit and implicit concept-based video retrieval with bipartite graph propagation model, acm international conference on multimedia. Firenze, Italy, 2010. ACM.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/, 2001.

[9] M. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. *Computer Science Department*, page 929, 2009.

[10] M. Chen, H. Li, and A. Hauptmann. Informedia@ TRECVID 2009: Analyzing Video Motions. *Computer Science Department*, page 927, 2009.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[12] D. Das, D. Chen, and A. G. Hauptmann. Improving multimedia retrieval with a video OCR. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6820 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, January 2008.

[13] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2006.

[14] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal. The Karlsruhe Verbmobil Speech Recognition Engine. In *Proc. ICASSP 97*, München; Germany, Apr. 1997. IEEE.

[15] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[16] Q. Jin, K. Laskowski, T. Schultz, and A. Waibel. Speaker Segmentation and Clustering in Meetings. In *Proc. ICASSP-2004 Meeting Recognition Workshop*, Montreal; Canada, May 2004. NIST.

[17] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. Citeseer.

[18] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[20] H. Medeiros, J. Park, and A. Kak. A parallel color-based particle filter for object tracking. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[21] F. Metze, C. Fügen, Y. Pan, T. Schultz, and H. Yu. The ISL RT-04S Meeting Transcription System. In *Proceedings NIST RT-04S Evaluation Workshop*, Montreal; Canada, May 2004. NIST.

[22] F. Metze, Q. Jin, C. Fügen, K. Laskowski, Y. Pan, and T. Schultz. Issues in Meeting Transcription – The ISL Meeting Transcription System. In *Proc. INTERSPEECH2004-ICSLP*, Jeju Island; Korea, Oct. 2004. ISCA.

[23] Nuance. Textbridge ocr. http://www.nuance.com/textbridge/.

[24] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal saliency for human action recognition. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, page 4. IEEE, 2005.

[25] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

[26] T. Schaaf and T. Kemp. Confidence measures for spontaneous speech. In *Proc. ICASSP 97*, München; Bavaria, Apr. 1997. IEEE.

[27] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[28] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[29] H. Soltau, F. Metze, C. Fügen, and A. Waibel. A One-pass Decoder based on Polymorphic Linguistic Context Assignment. In *Proc. Automatic Speech Recognition and Understanding (ASRU)*, Madonna di Campiglio, Italy, Dec. 2001. IEEE.

[30] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[31] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *Computer Vision–ECCV 2008*, pages 650–663, 2008.

[32] S. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[33] R. Yan and A. G. Hauptmann. Probabilistic latent query analysis for combining multiple retrieval sources. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 324–331, New York, NY, USA, 2006. ACM.

[34] R. Yan, J. Yang, and A. G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 548–555, New York, NY, USA, 2004. ACM.

[35] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, page 13. IEEE, 2006.