NEWS WEB VIDEO EVENT MINING USING BAG OF FEATURES

A PROJECT REPORT
Submitted by

Aggie Varghese
(Candidate Code:PRP15CSCE02)

to

the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology
In
Computer Science and Engineering

Department of Computer Science and Engineering

COLLEGE OF ENGINEERING & MANAGEMENT PUNNAPRA

PUNNAPRA, ALAPPUZHA

May 2017

DECLARATION

I undersigned hereby declare that the project report News Web Video Event Mining using Bag of Features , submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Mrs.Huda Noordean. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Alappuzha

08-05-2017 Aggie Varghese

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COLLEGE OF ENGINEERING AND MANAGEMENT
PUNNAPRA

CERTIFICATE

This is to certify that the report entitled ”**News Web Video Event Mining using Bag of Features**” submitted by **AGGIE VARGHESE** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Mrs. Huda Noordean
Assistant Professor, CSE
(Internal Supervisor)

External Supervisor

Mrs. Sheena N.
Assistant Professor, CSE
(PG Coordinator)

Mr.Suresh Kumar N.
Associate Professor, CSE
(Head of the DEPT)

CONTENTS

ACKNOWLEDGEMENT

First and foremost I thank God Almighty for his blessings for this project. I take this opportunity to express my gratitude to all those who have guided in the successful completion of this Endeavor. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Principal Mr. SURESH KUMAR N. for providing infrastructural facilities and for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the department, CSE, Mr. SURESH KUMAR N. , Associate Professor for his full-fledged support. I am also deeply indebted to our project coordinator Mrs. SHEENA N. , Assistant Professor, CSE and my project guide Mrs. HUDA NOORDEAN , Assistant Professor, CSE for their keen interest and ample guidance throughout the project.

I am indebted to my beloved teachers whose cooperation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the project. I convey my hearty thanks to my parents for the moral support, suggestions and encouragement to make this venture a success.

AGGIE VARGHESE

ABSTRACT

Due to the explosive growth of web videos, it becomes great challenge of how to efficiently browse lakhs or even millions of videos at a glance. Given a user query, social media web sites usually return a large number of videos that are diverse and not relevant. Exploring such results will be time-consuming and thus degrades user experience. Based on these observations, an event mining solution should provide the users a quick overview for each topic with a reduced browsing time.In the proposed system, for the classification of videos into events, first the videos are converted into keyframes.After shot boundary detection, the middle frame in each video shot is extracted as the keyframe for the shot.The training and validation keyframe sets are prepared.The bag-of-words technique was adapted to computer vision from the world of natural language processing.Speeded-up robust features (SURF) detector is used to find interesting points in the images and encode information about the area around the points as a feature vector.The visual vocabulary is constructed by reducing the number of features through quantization of feature space using K-means clustering.The histogram created forms the basis for training a classifier and for the actual keyframe classification.The classifier performance is evaluated and the newly trained classifier is applied to categorize new keyframes.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

With the rapid advancement of the World Wide Web, social media websites have emerged as a convenient platform for communication and publication for the past few years. The users are able to get information about on going incidents and events through several video sharing websites or search engines, such as YouTube, Google, Youku and Baidu. In addition, newswires like BBC, CCN and CCTV also publish news videos. While searching a hot topic online, the website may return thousands of videos which are not relevant. Based on these observations news web video event mining approaches have been developed which provide the users a quick overview of the topic and a reduced browsing time.

News web videos are mainly composed of visual and textual information. Visual information contains semantic gap and user subjectivity problems, and therefore, using either visual or textual information alone for news web video event mining may lead to unsatisfactory results. In order to overcome these shortcomings, both visual and textual features are utilized for web news video event mining.

For visual information, some important shots are frequently inserted into videos as a support of viewpoints, which carry useful information. Since there is unique role of near-duplicate keyframes (NDK) in the news search, topic detection and tracking (TDT) and copyright infringement detection, these duplicate keyframes /shots are clustered to form different groups according to visual content. Such groups are similar to the hot terms in the text field. Here, each cluster is called an NDK group, which can be used to group videos with similar content to the same events.

1.1 DATA MINING

Data mining is the computing process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. It is an interdisciplinary subfield of computer science.The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Data mining is the analysis step of the knowledge discovery in databases process, or KDD.

The term is a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (mining) of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing(collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence, machine learning, and business intelligence.

The actual data mining task is the automatic or semiautomatic analysis of large quantities of data to extract previously unknown, interesting patterns such as groups of data records(cluster analysis),unusual records (anomaly detection), and dependencies (association rule mining, sequential pattern mining). This usually involves using database techniques such as spatial indices. These patterns can then be seen as a kind of summary of the input data, and may be used in further analysis or, for example, in machine learning and predictive analytics. For example, the data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system. Neither the data collection, data preparation, nor result interpretation and reporting is part of the data mining step, but do belong to the overall KDD process as additional steps.

The related terms data dredging, data fishing, and data snooping refer to the use of data mining methods to sample parts of a larger population data set that are (or may be)too small for reliable statistical inferences to be made about the validity of any patterns discovered. These methods can, however, be used in creating new hypotheses to test against the larger data populations.

The knowledge discovery in databases (KDD) process is commonly defined with the stages:

1. Selection
2. Pre-processing
3. Transformation
4. Data mining
5. Interpretation/evaluation

It exists, however, in many variations, such as the Cross Industry Standard Process for Data Mining (CRISP-DM) which defines six phases:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

or a simplified process such as

1. pre-processing
2. data mining
3. results validation

1.1.1 Pre-Processing

Before data mining algorithms can be used, a target data set must be assembled. As data mining can only uncover patterns actually present in the data, the target data set must be large enough to contain these patterns while remaining concise enough to be mined within an acceptable time limit. A common source for data is a data mart or data warehouse. Pre-processing is essential to analyze the multivariate data sets before data mining. The target set is then cleaned. Data cleaning removes the observations containing noise and those with missing data.

1.1.2 Data Mining

Data mining involves six common classes of tasks:

- Anomaly detection (outlier/change/deviation detection) The identification of unusual data records, that might be interesting or data errors that require further investigation.
- Association rule learning (dependency modelling) Searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.
- Clustering is the task of discovering groups and structures in the data that are in some way or another similar, without using known structures in the data.
- Classification is the task of generalizing known structure to apply to new data. For example, an email program might attempt to classify an e-mail as legitimate or as spam.
- Regression attempts to find a function which models the data with the least error.
- Summarization providing a more compact representation of the data set, including visualization and report generation.

1.1.3 Results Validation

Data mining can unintentionally be misused, and can then produce results which appear to be significant; but which do not actually predict future behaviour and cannot be reproduced on a new sample of data and bear little use. Often this results from investigating too many hypotheses and not performing proper statistical hypothesis testing. A simple version of this problem in machine learning is known as overfitting, but the same problem can arise at different phases of the process and thus a train/test split - when applicable at all - may not be sufficient to prevent this from happening.

The final step of knowledge discovery from data is to verify that the patterns produced by the data mining algorithms occur in the wider dataset. Not all patterns found by the data mining algorithms are necessarily valid. It is common for the data mining algorithms to find patterns in the training set which are not present in the general dataset. This is called overfitting. To overcome this,the evaluation uses a test set of data on which the data mining algorithm was not trained. The learned patterns are applied to this test set, and the resulting output is compared to the desired output. For example, a data mining algorithm trying to distinguish spam from legitimate e-mails would be trained on a training set of sample e-mails. Once trained, the learned patterns would be applied to the test set of e-mails on which it had not been trained. The accuracy of the patterns can then be measured from how many e-mails they correctly classify. A number of statistical methods may be used to evaluate the algorithm, such as ROC curves.

If the learned patterns do not meet the desired standards, subsequently it is necessary to re-evaluate and change the pre-processing and data mining steps.If the learned patterns do meet the desired standards, then the final step is to interpret the learned patterns and turn them into knowledge.

1.2 BAG OF FEATURES

In order to do the visual similarity measure, the videos are converted into keyframes which are equivalent to images. For classifying these images into events, Bag of Features approach is used.Bag of Features (BoF) methods have been applied to image classification, object detection, image retrieval, and even visual localization for robots. BoF approaches are characterized by the use of an orderless collection of image features. Lacking any structure or spatial information, it is perhaps surprising that this choice of image representation would be powerful enough to match or exceed state-of-the-art performance in many of the applications to which it has been applied. Due to its simplicity and performance, the Bag of Features approach has become well-established in the field.

1.2.1 Bag Of Features Image Representation

A Bag of Features method is one that represents images as orderless collections of local features. There are two common perspectives for explaining the BoF image representation. The first is by analogy to the Bag of Words representation. With

Bag of Words, one represents a document as a normalized histogram of word counts. Commonly, one counts all the words from a dictionary that appear in the document. The term vector that represents the document is a sparse vector where each element is a term in the dictionary and the value of that element is the number of times the term appears in the document divided by the total number of dictionary words in the document.

The Bag of Features image representation is analogous. A visual vocabulary is constructed to represent the dictionary by clustering features extracted from a set of training images. The image features represent local areas of the image, just as words are local features of a document. Clustering is required so that a discrete vocabulary can be generated from millions (or billions) of local features sampled from the training data. Each feature cluster is a visual word. Given a novel image, features are detected and assigned to their nearest matching terms (cluster centers) from the visual vocabulary. The term vector is then simply the normalized histogram of the quantized features detected in the image.

The second way to explain the BoF image representation is from a codebook perspective. Features are extracted from training images and vector quantized to develop a visual codebook. A novel images features are assigned the nearest code in the codebook. The image is reduced to the set of codes it contains, represented as a histogram. The normalized histogram of codes is exactly the same as the normalized histogram of visual words, yet is motivated from a different point of view.

The BoF term vector is a compact representation of an image which discards largescale spatial information and the relative locations, scales, and orientations of the features. A contemporary large-scale BoF-based image retrieval system might have a dictionary of 100,000 visual words and 5,000 features extracted per image. Thus in an image where there are no duplicate visual words (unusual), the term vector will have 95% of its elements as zeros.

At a high level, the procedure for generating a Bag of Features image representation is shown in Figure 1.1 and summarized as follows:

- Build Vocabulary: Extract features from all images in a training set. Vector quantize, or cluster, these features into a visual vocabulary, where each cluster represents a visual word or term.

- Assign Terms: Extract features from a novel image. Use Nearest Neighbors or a related strategy to assign the features to the closest terms in the vocabulary.
- Generate Term Vector: Record the counts of each term that appears in the image to create a normalized histogram representing a term vector. This term vector is the Bag of Features representation of the image.

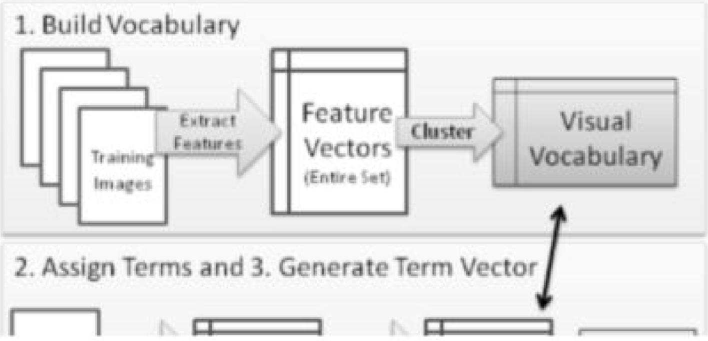


Figure 1.1: Process for Bag of Features Image Representation

There are a number of design choices involved at each step in the BoF representation. One key decision involves the choice of feature detection and representation. Many use an interest point operator, such as the Harris-Afne detector or the Maximally Stable Extremal Regions (MSER) detector .At every interest point, often a few thousand per image, a high-dimensional feature vector is used to describe the local image patch. Lowes 128-dimension SIFT descriptor is a popular choice .

Another pair of design choices involve the method of vector quantization used to generate the vocabulary and the distance measure used to assign features to cluster centers. A distance measure is also required when comparing two term vectors for similarity, but this measure operates in the term vector space as opposed to the feature space. Quantization issues and the choice of distance measure can impact term assignment and similarity scoring.

1.2.2 Feature Detection And Representation

There are many possible approaches to sampling image features, including Interest Point Operators, Visual Saliency, and random or deterministic grid sampling.

Feature Detection: Feature detection is the process of deciding where and at what scale to sample an image. The output of feature detection is a set of keypoints that specify locations in the image with corresponding scales and orientations. These keypoints are distinct from feature descriptors, which encode information from the pixels in the neighborhood of the keypoints. Thus, feature detection is a separate process from feature representation in BoF approaches.

There is a substantial body of literature that focuses on detecting the location and extent of good features from at least two different sub-fields of computer vision. The first developed from the goal of finding keypoints useful for image registration that are stable under minor affine and photometric transformations. These feature detection methods are referred to as Interest Point Operators. The second group detects features based on computational models of the human visual attention system. These methods are concerned with finding locations in images that are visually salient. In this case, fitness is often measured by how well the computational methods predict human eye fixations recorded by an eye tracker.

Finally, there is research that suggests generating keypoints by sampling the images using a grid or pyramid structure, or even by random sampling.

- Interest Point Operators: While there are many variations, an interest point operator typically detects keypoints using scale space representations of images. A scale space represents the image at multiple resolutions, and is generated by convolving the image with a set of guassian kernels spanning a range of values. The result is a data structure which is, among other things, a convenient way to efficiently apply image processing operations at multiple scales. Interest point operators detect locally discriminating features, such as corners, blob-like regions, or curves. Responses to a filter designed to detect these features are located in a three dimensional coordinate space, (x,y,s), where (x,y) is the pixel location and s is the scale. Extremal values for the responses over local (x,y,s) neighborhoods are identified as interest points.

  Perhaps the most popular keypoint detector is that developed by Lowe , which employs a Difference-of-Gaussians (DoG) filter for detection. DoG responses can be conveniently computed from a scale space structure, and extremal response values within a local (x,y,s) region used as interest points. Kadir and Brady designed a keypoint detector called Scale Saliency to find regions which have high entropy within a local scale-space region. Another popular keypoint detector, called the Harris-Affine detector , extends the well-known Harris Corner Detector to a scale space representation with oriented elliptical regions. The Maximally Stable Extremal Regions (MSER) keypoint detector finds elliptical regions based on a watershed process . These are but a handful of examples of the many interest point operators designed to be robust to small affine and photometric image transformations, with the goal of being able to find the same keypoints in two similar but distinct images.

- Visual Saliency: Bag of Features methods often rely on interest point operators to detect the location and scale of localized regions from which image features are extracted. Similarly, many biologically-inspired, or biomimetic, computer vision systems use localized regions as well. In the biomimetic computer vision literature, the interest point operator is based on computational models of visual attention.

  Itti and Koch proposed a popular model which builds upon a thread of research started by Koch and Ullman in the 1980s .The Itti and Koch saliency model, at a high level, looks for extrema in center-surround patterns across several feature channels, such as color opponency, orientation (Gabor filters), and intensity. The center-surround extrema can be detected using Laplacian-of-Gaussian (LoG) or Difference-of-Gaussian (DoG) filters, similar to the models of Lindeberg and Lowe.

  A local region is interesting not just based upon a certain pattern or filter response, but also because it differs significantly from the surrounding context of neighboring pixels.

- Deterministic and Random Sampling: A more fundamental question than which interest point detector to use, is whether or not to use an interest point detector at all.

  Maree et al. describe an image classification algorithm featuring random multiscale subwindows and ensembles of randomized decision trees . While the algorithm is not strictly a BoF approach, it illustrates the efficacy of random sampling. Nowak, Jurie, and Triggs explored sampling strategies for BoF image classification . They show that when using enough samples, random sampling exceeds the performance of interest point operators. They present evidence that the most important factor is the number of patches sampled from the test image, and thus claim dense random sampling is the best strategy.

Spatial Pyramid Matching uses SIFT descriptors extracted from a dense grid with a spacing of 8 pixels. K-means clustering is used for constructing the vocabulary. But instead of directly forming the term vector for a given image, the terms are collected in a pyramid of histograms, where the base level is equivalent to the standard BoF representation for the complete image. At each subsequent level in the pyramid, the image is divided into four subregions, in a recursive manner, with each region at each pyramid level having its own histogram (term vector). The distance between two images using this spatial pyramid representation is a weighted histogram intersection function, where weights are largest for the smallest regions. Doing so, Lazebnik captures a degree of location information beyond the standard orderless BoF representation.

Feature Descriptors: In addition to determining where and to what extent a feature exists in an image, there is a separate body of research to determine how to represent the neighborhood of pixels near a localized region, called the feature descriptor. The simplest approach is to simply use the pixel intensity values, scaled for the size of the region, or an eigen space representation thereof. Normalized pixel representations, however, have performed worse than many more sophisticated representations and have largely been abandoned by the BoF research community.

The most popular feature descriptor in the BoF literature is the SIFT (Scale Invariant Feature Transform) descriptor . In brief, the 128 dimensional SIFT descriptor is a histogram of responses to oriented gradient filters. The responses to 8 gradient orientations at each of 16 cells of a 4x4 grid generate the 128 components of the vector. The histograms in each cell are block-wise normalized. At scale 1, the cells are often 3x3 pixels.

An alternative to the SIFT descriptor that has gained increasing popularity is SURF (Speeded Up Robust Features) . The SURF algorithm consists of both feature detection and representation aspects. It is designed to produce features akin to those produced by a SIFT descriptor on Hessian-Laplace interestpoints, but using efficient approximations. Reported results indicate that SURF provides a significant speed-up while matching or improving performance.

1.2.3 Quantization And Distance Measures

Vector Quantization (Clustering) is used to build the visual vocabulary in Bag of Features algorithms. Nearest-neighbor assignments are used not only in the clustering of features but also in the comparison of term vectors for similarity ranking or classification. Thus, it is important to understand how quantization issues, and the related issues involving measuring distances in feature and term vector space, affect Bag of Features based applications.

Many BoF implementations are described as using K-means , or an approximation thereof for large vocabularies . Given any clustering method, there will be points that are equally close to more than one centroid. These points lie near a Voronoi boundary between clusters and create ambiguity when assigning features to terms. With K-means and similar clusterin gmethods,the choice of initial centroid positions affects the resultant vocabulary. When dealing with relatively small vocabularies, one can run K-means multiple times and select the best performing vocabulary during a validation step. This becomes impractical for very large data sets. When determining the distance between two features, as required by clustering and term assignment, common choices are the Manhattan (L1), Euclidean (L2), or Mahalanobis distances. A distance measure is also needed in term vector space for measuring the similarity between two images for classification or retrieval applications. Euclidean and Manhattan distances over sparse term vectors can be computed efficiently using inverted indexes, and are thus popular choices. However, the relative importance of some visual words leads to the desire to weight term vectors during the distance computation. The following methods provide more details on these and other issues.

- Term Weights: One of the earliest strategies for handling quantization issues at a gross level is to assign weights to the terms in the term vector. This can be viewed as a mitigation strategy for quantization issues that occur when the descriptors are distributed in such a way that simple clustering mechanisms over-represent some descriptors and under represent others. With term weights, one can penalize terms found to be too common to be discriminative and emphasize those that are more unique. This is the motivation behind the popular Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme used in text retrieval .

TF-IDF is defined as: $tf_i * log\frac{N}{N_i}$ where tf_i is the term frequency of the $i'th$ word, N is the number of documents (images) in the database, and N_i is the number of documents in the database containing the $i'th$ word. The log term is called the inverse document frequency and it serves to penalize the weights of common terms.

Term vectors can also be represented as binary strings. A 1 is assigned for any term that appears in the image, 0 otherwise. This might humorously be called anti term weighing, as it has the effect of making the terms have equal weight no matter how often they occur in a particular image or in the corpus as a whole. Distribution issues are thus handled by simply erasing the frequency information from the term vector so a few oversampled terms can not dominate the distance computations between term vectors. Binary representations have the benet of speed and compactness.

BoF image retrieval implementations typically use TF-IDF weights, due to evidence that this method is superior to binary and term frequency representations. When using very large vocabularies, the term vectors become extremely sparse, and the term counts (prior to any normalization) are mostly zeros or ones. In this case, binary representations tend to perform similarly to term frequencies.

- Soft Assignment: A given feature may be nearly the same distance from two cluster centers, but with a typical hard assignment method, only the slightly nearer neighbor is selected to represent that feature in the term vector. Thus, the ambiguous features that lie near Voronoi boundaries are not well-represented by the visual vocabulary. To address this problem, researchers have explored multiple assignments and soft weighting strategies.

  Multiple assignment is where a single feature is matched to k nearest terms in the vocabulary. Soft weights are similar, but the k nearest terms are multiplied by a scaling function such that the nearest term gets more weight than the $k'th$nearest term. These strategies are designed to mitigate the negative impact when a large number of features in an image sit near a Voronoi boundary of two or more clusters.

Jegou et al. show that multiple assignment causes a modest increase in retrieval accuracy. The cost of the improved accuracy is higher search time, due in part to the impact on term vector sparsity. The authors report that a k = 3 multiple assignment implementation requires seven times the number of multiplications of simple assignment.

Soft weights have been explored by Jiangetal and Philbin et al. InJiangs work, the soft weights are computed as shown below. The computed weight for term n in the term vector w, denoted w_n is dened as:

$$w_n = \sum_{i=1}^{k} \sum_{j=1}^{M_i} \frac{1}{2^{i-1}} sim(j, n)$$

where k is the number of neighbors to use in the soft weighting strategy, M_i is the number of features in the image whose $i'th$ nearest neighbor is term n, and sim(j,n) is the similarity measure between feature j and term n. Jiang et al. suggest that k=4 works well. In experimental evaluations, over a variety of vocabulary sizes, Jiangs soft weighting strategy bests binary, term frequency, and TF-IDF schemes, with one marginal exception.

Philbin et al. propose an approach that scales the term weight according to the distance from the feature to the cluster center . The weights are assigned proportionally to a Gaussian decay on the distance, $exp(\frac{d^2}{2\sigma^2})$, where d is the distance to the cluster center, and σ is the spatial scale, selected such that a relatively small number of neighbors will be given signicant weight. A problem with this continuous formulation is that all terms get a non-zero weight, so clipping very small values is prudent. Philbin et al. only compute the soft weights to a pre-determined number of nearest neighbors, which was three in their evaluations. Even with clipping, soft weights decrease the sparsity of the term vectors, and thus increase the index size and query retrieval times. After generating term vectors using the soft weighting strategy, Philbin et al. perform an L_1 normalization so that the resulting vector looks like a term frequency vector. TF-IDF is then applied, ignoring soft assignment issuesi.e., the IDF is computed as if the input vector were created by the normal hard-assignment process. Evaluation results show a strong improvement in query accuracy.

This result is consistent with the earlier observations by Jegou et al. and Jiang et al. that multiple assignment/soft weighting improves retrieval accuracy by mitigating some of the quantization errors for border line features.

- Non-uniform Distributions: Jurie and Triggs show that the distribution of cluster centers is highly non-uniform for dense sampling approaches. When using k-means clustering, high-frequency terms dominate the quantization process, yet these common terms are less discriminative than the medium-frequency terms that can get lost in quantization. Instead, they propose an online, fixed-radius clustering method that they demonstrate produces better codebooks. Jegou et al. discuss the burstiness of visual elements, meaning that a visual word is more likely to appear in an image if it has appeared once before. Thus visual words are not independent samples of the image. Various weighting functions and strategies are proposed and evaluated.

  Similar to distribution issues within the feature space used to construct the vocabulary, term vectors can be non-uniformly distributed in the gallery set. Weighting strategies can be used to compensate for non-uniform term vector distributions, as discussed above, or a distance measure can be created that scales with local distributions in an attempt to regularize the space.

- Intracluster Distances: There is a trade-off involved in choosing the granularity of the quantization, where finer-grained clustering leads to potentially more discriminative information being preserved at the cost of increased storage and computational requirements. A possible compromise is to use a courser-grained quantization, but compensate for the lack of discrimination by employing an efficient method for incorporating intracluster distances to weight terms. To this end, Jegou et al. developed a technique called Hamming Embedding, which efficiently represents how far a feature lies from the cluster center and thus how much weight to assign to that term for the detected feature.

1.2.4 Image Classification Using Bag Of Features

BoF based image classification is the process of representing a training set of images as term vectors and training a classifier over this representation. A probe image can be encoded with the same dictionary and given to the classifier to be assigned a label.

Similar to the image classification task, people discuss BoF-based object detection, object recognition, and scene classification. When the data set has essentially one object class per image, then the difference between image classification and object detection is blurred. Scene classification is essentially synonymous with image classification. That is, the entire image is classified with no attempt to detect or localize specific objects it may contain.

To help put the BoF image classification literature in context, first investigate a set of related approaches. The first is a brief discussion of the similarity between BoF image classification and texture classification. As with Bag of Features, texture classification methods commonly sample features from the image (a texture image), quantize those features, and build representative histograms, which are then used in a classification task . In the texture recognition literature, textons refer to representative image patches and are analogous to visual words in the Bag of Features paradigm. Zhang et al. performed a study of local features and classification kernels for Support Vector Machine-based texture and image classification, which indirectly illustrates how similar Bag of Features image classification is to the earlier body of work on texture representation and classification.

A second body of related work is object detection using a part-based model. Unlike the orderless Bag of Features, a part-based model (also called a constellation) learns a deformable arrangement of features that represent an object class. Similar to BoF, part-based models often start with feature detection and extraction stages. Diverging from a BoF approach, part-models typically employ a maximum likelihood estimation technique to determine which of several previously learned models best explain the detected arrangement of features. Part-based methods have shown strong robustness in the face of background clutter, partial occlusions, and variances in the objects appearance . A key weakness of part-based models is computational complexity. There is a combinatorial search required to determine which subset of features matches a given model. To reduce computation time, often only a small number of features are extracted from the image (one or two orders of magnitude smaller than with BoF methods), thus making the algorithms more sensitive to feature detection errors.

Another related technique is the use of a gist descriptor for scene classification, such as the Holistic Spatial Envelope . Gist-based approaches attempt to create a low-dimensional signature of an image that is computationally cheap and contains enough information for gross categorization. Both gist and BoF representations attempt to reduce an image to a low dimensional representation, but while BoF is a histogram of quantized local features, gist is a single global descriptor of the entire image. Gist representations are thus smaller and faster to compute, but with a commensurate loss of discriminatory information. The Holistic Spatial Envelope is akin to using a single 512 dimensional SIFT descriptor to represent the entire image.

Visual localization is the problem of determining approximate location based only on visual information. It is a related task to image classification, because input images must be classified as being part of a known location. Researchers have been using SIFT features as natural landmarks for many years. Recently, researchers have applied BoF representations directly to the task.

A well-known example is scene classification using the Neuromorphic Vision Toolkit. Siagian and Itti detect local color, intensity, and gradient orientation features using non-maxima suppression over a scale space, but in contrast to BoF representations, they impose a spatial structure by dividing each image into a 4x4 grid, over which features are aggregated. More recently, Song and Tao presented a biologically inspired feature manifold representation of an image. With the assumption that the features are sampled from a low dimensional manifold embedded in a high dimensional space, the authors propose a manifold-based projection to reduce dimensionality while preserving discriminative information. They show SVM-based scene classification on this representation yields much faster and more accurate results than that of Siagian and Itti.

Chapter 2

LITERATURE SURVEY

Several methods for news web video event mining exists: Q. He, K. Chang, and E.-P. Lim, in [1] Analyzing feature trajectories for event detection, proposed a method based on the feature trajectory using text words . In this paper, they 1) spectral analysis is applied first to categorize features for dierent event characteristics: periodic and aperiodic, important or less-reported. 2) modeled periodic features with Gaussian mixture densities and aperiodic features with Gaussian density and, and detected each features burst by the truncated Gaussian approach; 3) proposed an unsupervised greedy event detection algorithm to detect both aperiodic and periodic events. The highly sets to the correlated word features are grouped to events by mapping the word video sets. The performance of FT_T is poor. This c is a challenging mission to mine the events under the noisy and diverse social web scenario.

J. Yao, B. Cui, Y. Huang, and Y. Zhou, in [2] Bursty event detection from collaborative tags, proposed a technique based on visual Near Duplicate Keyframe -level clustering. In this paper, they proposed a new method to detect bursty tagging event, which captures the relations among a group of correlated tags. These tags are either bursty or associated with bursty tag co-occurrence. This kind of bursty tagging event generally corresponds to a real life event. The events are profiled with more comprehensible and representative clues. The proposed method is divided into three stages. As the first step,they exploit the sliding time intervals to extract bursty features, and graph clustering techniques is adopted to group bursty features into meaningful bursty events. It helps to detect the relationship among tags with bursty tag cooccurrence. In this method, the number of NDKs among videos is limited. Each event is composed of different kinds of visual scenes, while CC_V just groups one scene.

X. Wu, Y.-J. Lu, Q. Peng, and C.-W. Ngo, in [3] Mining event structures from web videos, proposed T+V which is a fusion method. This paper explores the issues of mining event structures from Web video search results using burst detection, text analysis, clustering, and other techniques It first applies feature trajectory to the visual field. Then it tries to mine events on the basis of text cooccurrence and feature trajectory. In this method the visual near-duplicate feature trajectories of NDKs are not consistent. It misses those low-frequency terms and NDKs which are common for a large number of videos.

C. Zhang, X. Wu, M.-L. Shyu, and Q. Peng, in [4] A novel web video event mining framework with the integration of correlation and co-occurrence information, proposed a method inorder to mine the correlation between events and NDKs by using the distribution characteristics of the terms. In this paper, in order to improve the performance of web video event mining they propose a novel four-stage framework. The first stage is data preprocessing. Multiple Correspondence Analysis (MCA) is then applied to explore the correlation between terms and classes. It targets to bridge the gap between NDKs and high-level semantic concepts. Next, the similarity between NDKs and classes are detected using co-occurrence information . Finally, through negative NDK pruning and positive NDK enhancement both of them are integrated for web video event mining. It can bridge the gap between NDKs and terms. In this method, multiple languages, synonyms, and the number of videos in each NDK are problems.

Chengde Zhang, Xiao Wu, Mei-Ling Shyu, and Qiang Peng in [5] Integration of visual temporal information and textual distribution information for news web video event mining, proposed a method which uses both the neighbor stabilization process and MCA similarity measure to generate the textual distribution information. It can better explore the degree of correlation between different terms and events. The visual near-duplicate feature trajectory, i.e., the time distribution information of an NDK,is integrated with the NDK-within-video information (cooccurrence) as the visual temporal information to cluster more NDKs belonging to the same event.

X. Wu, C.-W. Ngo, and A. G. Hauptmann, in [6] Multimodal news story clustering with pairwise visual near-duplicate constraint, offers a new perspective by exploring the pairwise visual cues deriving from near-duplicate keyframes (NDK) for constraint-based clustering. They propose a constraint-driven co-clustering algorithm (CCC), which utilizes the near-duplicate constraints built on top of text, to mine topic-related stories and the outliers. With CCC, the duality between stories and their underlying multimodal features is exploited to transform features in low-dimensional space with normalized cut. The visual constraints are added directly to this new space, while the traditional DBSCAN is revisited to capitalize on the availability of constraints and the reduced dimensional space. They modify DBSCAN with two new characteristics for story clustering: 1) constraint-based centroid selection and 2) adaptive radius.

J. Cao, C.-W. Ngo, Y.-D. Zhang, and J.-T. Li, in [7] Tracking web video topics: Discovery, visualization, and monitoring, studies the problems in news web video event mining from three aspects. First, given a large set of videos collected over months, an efcient algorithm based on salient trajectory extraction on a topic evolution link graph is proposed for topic discovery. Second, topic trajectory is visualized as a temporal graph in 2-D space, with one dimension as time and another as degree of hotness, for depicting the birth, growth, and decay of a topic. Finally, giving the previously discovered topics, an incremental monitoring algorithm is proposed to track newly uploaded videos, while discovering new topics and giving recommendation to potentially hot topics.

X. Zhang, C. Xu, J. Cheng, H. Lu, and S. Ma, in [8] Effective annotation and search for video blogs with integration of context and content analysis, propose a novel vlog management model which is comprised of automatic vlog annotation and user-oriented vlog search. For vlog annotation, they extract informative keywords from both the target vlog itself and relevant external resources; besides semantic annotation, they perform sentiment analysis on comments to obtain the overall evaluation. For vlog search, they present saliency-based matching to simulate human perception of similarity, and organize the results by personalized ranking and category-based clustering. An evaluation criterion is also proposed for vlog annotation, which assigns a score to an annotation according to its accuracy and completeness in representing the vlogs semantics.

C.Xu, Y.-F. Zhang, G.Zhu, Y.Rui, H. Lu, and Q. Huang, in [9] Using web-cast text for semantic event detection in broadcast sports video, present a novel approach for sports video semantic event detection based on analysis and alignment of webcast text and broadcast video. Webcast text is a text broadcast channel for sports game which is co-produced with the broadcast video and is easily obtained from the web. They first analyze webcast text to cluster and detect text events in an unsupervised way using probabilistic latent semantic analysis (pLSA). Based on the detected text event and video structure analysis, they employ a conditional random field model (CRFM) to align text event and video event by detecting event moment and event boundary in the video. Incorporation of webcast text into sports video analysis significantly facilitates sports video semantic event detection. They conducted experiments on 33 hours of soccer and basketball games for webcast analysis, broadcast video analysis and text/video semantic alignment.

L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, in [10] Correlation-based video semantic concept detection using multiple correspondence analysis, propose a novel framework that utilizes the ability of multiple correspondence analysis (MCA) to explore the correlation between different items (feature-value pairs) and classes (concepts) to bridge the gap between the extracted low-level features and high-level semantic concepts. Using the concepts and benchmark data identified and provided by the TRECVID project, they have shown that their proposed framework demonstrates promising results and performs better than the Decision Tree (DT), Support Vector Machine (SVM), and Naive Bayesian (NB) classifiers that are commonly applied to the TRECVID datasets.

L.Lin,C.Chen,M.-L.Shyu,andS.-C.Chen, in [11] Weighted subspace filtering and ranking algorithms for video concept retrieval, proposed a framework, with weighted subspace filtering and ranking components, which is the first attempt in multimedia research to apply multiple correspondence analysis to selected features while pruning data instances.

W.-L. Zhao, X. Wu, and C.-W. Ngo, in [12] On the annotation of web videos by efficient near-duplicate search, they investigate techniques which allow effective annotation of web videos from a data-driven perspective. A novel classifier-free video annotation framework is proposed by first retrieving visual duplicates and then suggesting representative tags. The significance of this paper lies in the addressing of two timely issues for annotating query videos. First, they provide a novel solution for fast near-duplicate video retrieval. Second, based on the outcome of near-duplicate search, they explore the potential that the data-driven annotation could be successful when huge volume of tagged web videos is freely accessible online. Experiments on cross sources (annotating Google videos and Yahoo! videos using YouTube videos) and cross time periods (annotating YouTube videos using historical data) show the effectiveness and efciency of the proposed classier-free approach for web video tag annotation.

X. Li, C. G. Snoek, and M. Worring,in [13] Learning social tag relevance by neighborvoting, propose a neighbor voting algorithm which accurately and efciently learns tag relevance by accumulating votes from visual neighbors. Under a set of well-dened and realistic assumptions, they prove that their algorithm is a good tag relevance measurement for both image ranking and tag ranking. Three experiments on 3.5 million Flickr photos demonstrate the general applicability of their algorithm in both social image retrieval and image tag suggestion. Their tag relevance learning algorithm substantially improves upon baselines for all the experiments.The results suggest that the proposed algorithm is promising for real-world applications.

Stephen OHara And Bruce A. Draper,in [14] Introduction to the Bag Of Features paradigm for image classification and retrieval presents an introduction to BoF image representations, describes critical design choices, and surveys the BoF literature. Emphasis is placed on recent techniques that mitigate quantization errors, improve feature detection, and speed up image retrieval.

To evaluate the performance of the web event mining, the Precision (P), Recall (R), and F1 measure (F1) are used ,which are defined as:

$$Precision = \frac{|B_i^+|}{|A_i|}$$

$$Recall = \frac{|B_i^+|}{|B_i|}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

where B_i^+ is the number of correctly grouped positive videos for cluster A_i, and B_i is the number of positive samples in the ground truth. Since F1 considers both Precision and Recall values, it is mainly used to evaluate the performance.

Table 2.1: Comparison between Related Works

Methods	Precision	Recall	F1
Analyzing feature trajectories for event detection [FT_T]	low	low	low
Bursty event detection from collaborative tags [CC_V]	high	low	low
Mining event structures from web videos [T+V]	Better than FT_T	Better than FT_T and CC_V	Better than FT_T and CC_V
A novel web video event mining framework with the integration of correlation and co-occurrence information [MCA]	low	Sometimes high	Can be low or high compared to FT_T,CC_V and T+V.
Integration of visual temporal information and textual distribution information for news web video event mining	Better than FT_T, T+V and MCA	Most of the time high	high.

Chapter 3

METHODS

To develop the proposed system using bag of features, several methods are used for feature detection and extraction, clustering and classification. Speeded up robust features (SURF) is used for feature detection and description.K-means clustering is used for clustering.Support vector machine(SVM) is used for classification.

3.1 SPEEDED UP ROBUST FEATURES(SURF)

In computer vision, speeded up robust features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor. The standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT.

To detect interest points, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. These can also be computed with the aid of the integral image.

SURF descriptors have been used to locate and recognize objects, people or faces, to reconstruct 3D scenes, to track objects and to extract points of interest.

SURF was first presented by Herbert Bay, et al., at the 2006 European Conference on Computer Vision. An application of the algorithm is patented in the United States.An upright version of SURF (called U-SURF) is not invariant to image rotation and therefore faster to compute and better suited for application where the camera remains more or less horizontal.

The image is transformed into coordinates, using the multi-resolution pyramid technique, to copy the original image with Pyramidal Gaussian or Laplacian Pyramid shape to obtain an image with the same size but with reduced bandwidth. This achieves a special blurring effect on the original image, called Scale-Space and ensures that the points of interest are scale invariant.

The SURF algorithm is based on the same principles and steps as SIFT; but details in each step are different. The algorithm has three main parts: interest point detection, local neighborhood description and matching.

3.1.1 Detection

SURF uses square-shaped filters as an approximation of Gaussian smoothing. (The SIFT approach uses cascaded filters to detect scale-invariant characteristic points, where the difference of Gaussians (DoG) is calculated on rescaled images progressively.) Filtering the image with a square is much faster if the integral image is used:

$$S(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$

The sum of the original image with in a rectangle can be evaluated quickly using the integral image, requiring evaluations at the rectangles four corners.

SURF uses a blob detector based on the Hessian matrix to find points of interest. The determinant of the Hessian matrix is used as a measure of local change around the point and points are chosen where this determinant is maximal. In contrast to the Hessian-Laplacian detector by Mikolajczyk and Schmid, SURF also uses the determinant of the Hessian for selecting the scale, as is also done by Lindeberg.

3.1.2 Descriptor

The goal of a descriptor is to provide a unique and robust description of an image feature, e.g., by describing the intensity distribution of the pixels within the neighbourhood of the point of interest. Most descriptors are thus computed in a local manner, hence a description is obtained for every point of interest identified previously.

The dimensionality of the descriptor has direct impact on both its computational complexity and point-matching robustness/accuracy. A short descriptor may be more robust against appearance variations, but may not offer sufficient discrimination and thus give too many false positives.

The first step consists of fixing a reproducible orientation based on information from a circular region around the interestpoint. Then we construct a square region aligned to the selected orientation, and extract the SURF descriptor from it.

3.1.3 Matching

By comparing the descriptors obtained from different images, matching pairs can be found.

3.2 K-MEANS CLUSTERING

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with k-means because of the k in the name. One can apply the 1-nearest neighbor classifier on the cluster centers obtained by k-means to classify new data into the existing clusters.

This is known as nearest centroid classier or Rocchio algorithm.

Given a set of observations(x_1,x_2,,x_n),where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k($\leq n$) sets S=$\{S_1$,S_2,,$S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\underset{s}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \epsilon S_i} \|x - \mu_i\|^2 = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{k} \mid S_i \mid VarS_i$$

where μ_i is the mean of points in S_i.

3.2.1 Standard Algorithm

The most common algorithm uses an iterative renement technique. Due to its ubiquity it is often called the k-means algorithm; it is also referred to as Lloyds algorithm, particularly in the computer science community.

Given an initial set of k means $m_1^{(1)}$,,$m_k^{(1)}$ the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least within cluster sum of squares(WCSS). Since the sum of squares is the squared Euclidean distance, this is intuitively the nearest mean.

Update step: Calculate the new means to be the centroids of the observations in the new clusters. Since the arithmetic mean is a least squares estimator, this also minimizes the within-cluster sum of squares(WCSS) objective.

The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a nite number of such partitionings, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm.

The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by least sum of squares, which is exactly equivalent to assigning by the smallest Euclidean distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging. Various modications of k-means such as spherical k-means and k-medoids have been proposed to allow using other distance measures.

Commonly used initialization methods are Forgy and Random Partition. The Forgy method randomly chooses k observations from the data set and uses these as the initial means. The Random Partition method rst randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean to be the centroid of the clusters randomly assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. According to Hamerly et al., the Random Partition method is generally preferable for algorithms such as the k-harmonic means and fuzzy k-means. For expectation maximization and standard k-means algorithms, the Forgy method of initialization is preferable. A comprehensive study by Celebi et al., however, found that popular initialization methods such as Forgy,Random Partition,and Maximin often perform poorly, whereas the approach by Bradley and Fayyad performs consistenty in the best group and K-means++ performs generally well.

- Demonstration of the standard algorithm.

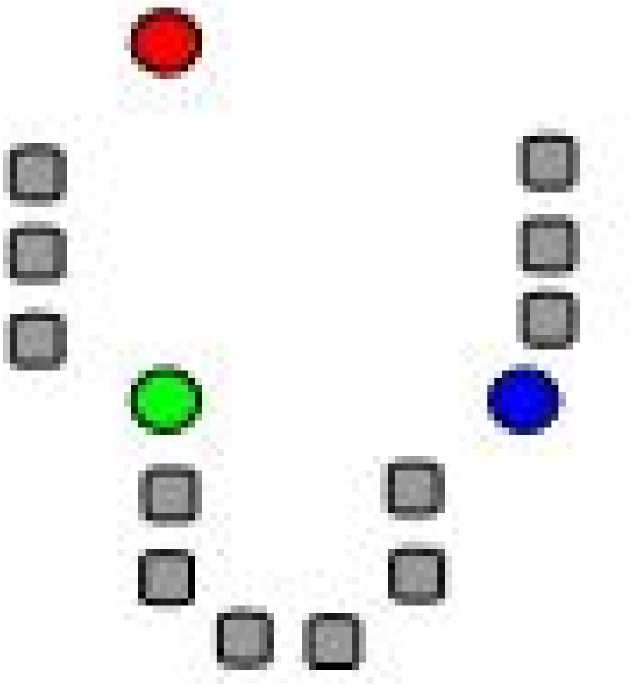


Figure 3.1: Step 1 of k-means standard algorithm

1. k initialmeans(inthis case k=3) are randomly generated within the data domain(shown in color).

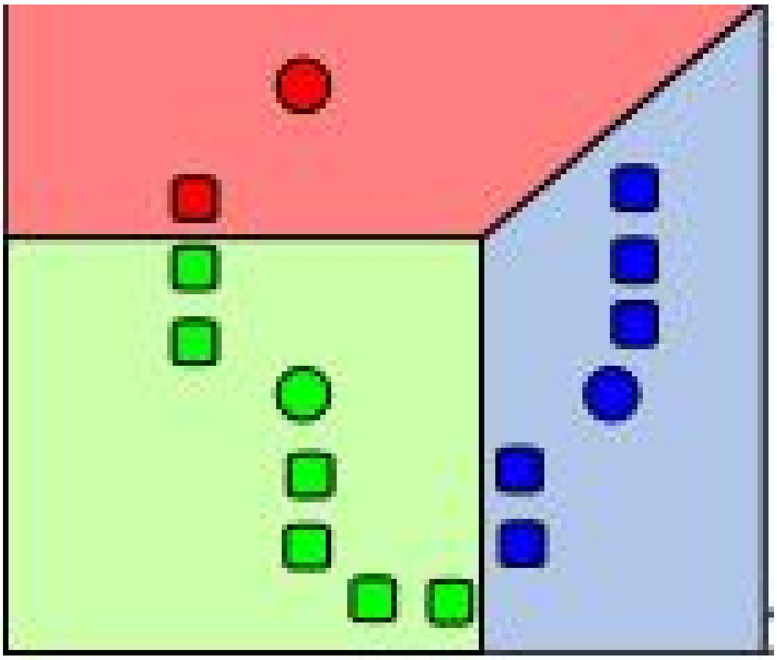


Figure 3.2: Step 2 of k-means standard algorithm

2. k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

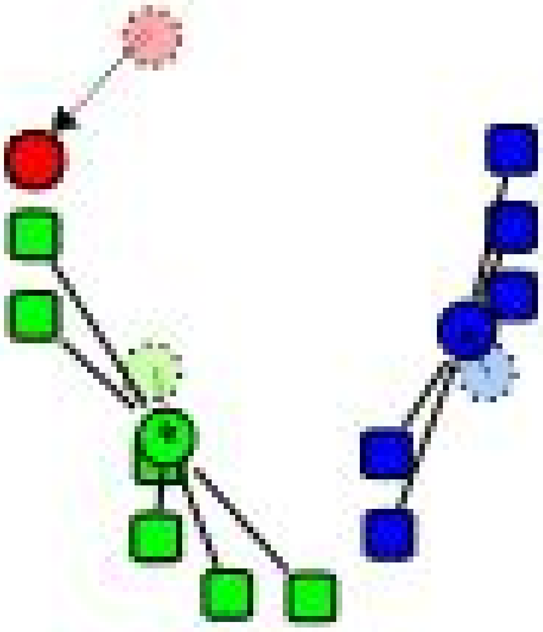


Figure 3.3: Step 3 of k-means standard algorithm

3. The centroid of each of the k clusters becomes the new mean.

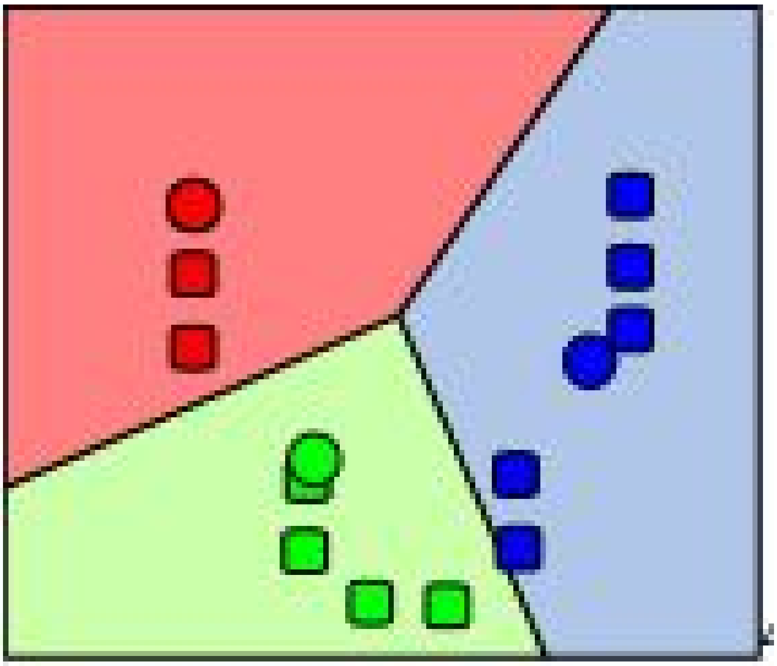


Figure 3.4: Step 4 of k-means standard algorithm

4. Steps 2 and 3 are repeated until convergence has been reached.

As it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial clusters. As the algorithm is usually very fast, it is common to run it multiple times with different starting conditions.However, in the worst case, k-means can be very slow to converge: in particular it has been shown that there exist certain point sets, even in 2 dimensions, on which k-means takes exponential time, that is $2^{(n)}$, to converge.These point sets do not seem to arise in practice: this is corroborated by the fact that the smoothed running time of k-means is polynomial.

The assignment step is also referred to as expectation step, the update step as maximization step, making this algorithm a variant of the generalized expectation maximization algorithm.

3.3 SUPPORT VECTOR MACHINE

In machine learning, support vector machines(SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering and is often used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass.

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p1)-dimensional hyperplane. This is called a linear classier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classier it denes is known as a maximum margin classier; or equivalently, the perceptron of optimal stability.

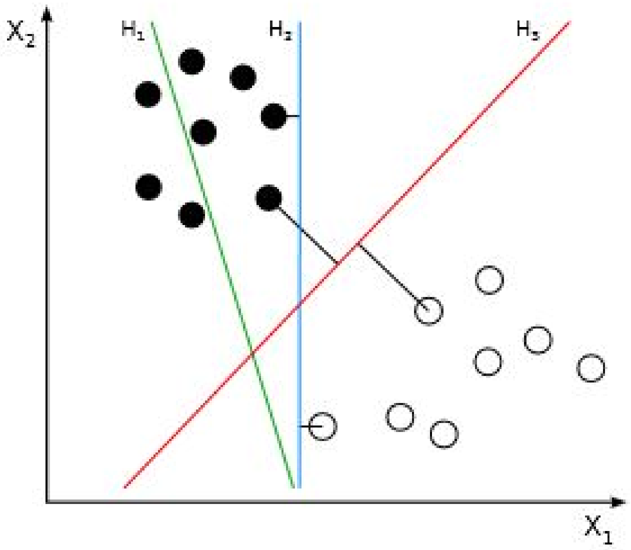


Figure 3.5: Different hyperplanes in a linear classifier

H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximum margin.

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or innite dimensional space, which can be used for classication, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classier.

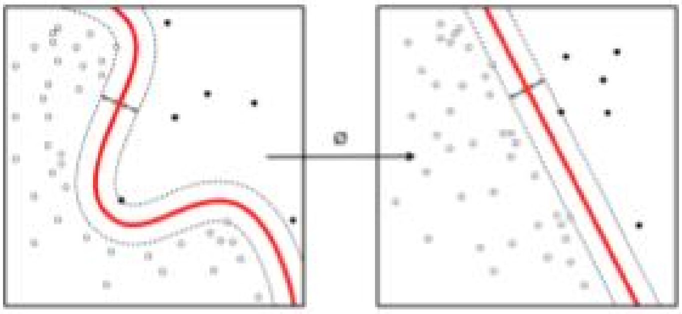


Figure 3.6: Kernel machine

Where as the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space,presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function k(x,y) selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α_i of images of feature vectors x_i that occur in the data base. With this choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation: $\sum_i \alpha_i k(x_i, x)$ = constant. Note that if k(x,y) becomes small as y grows further away from x, each term in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i . In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space.

3.3.1 Applications

SVMs can be used to solve various real world problems:

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.

- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true of image segmentation systems, including those using a modified version SVM that uses the privileged approach.

- Hand-written characters can be recognized using SVM.

- The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models.Support vector machine weights have also been used to interpret SVM models in the past.Posthoc interpretation of support vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

Chapter 4

PROPOSED SYSTEM

The proposed framework consists of the following stages: data preprocessing, text detection in keyframes, set up keyframe event sets,create bag of features,train a keyframe classifier with bag of visual words and classify a keyframe.

4.1 DATA PREPROCESSING

The videos are converted to keyframes. Key-frame refers to the image frame in the video sequence which is representative and able to reflect the summary of a video content. By using the key-frame it is able to express the main content of video data clearly and reduce the amount of memory needed for video data processing and complexity greatly. Keyframe extraction is an efficient method for video summarization. Depending on the content complexity of the shot one or more key-frames can be extracted from a single shot. A shot is defined as unbroken sequence of frames recorded from a single camera, which forms the building blocks of video. In video data which contains multiple shots it is necessary to identify individual shots for key-frame extraction.

Key-frame extraction can be classified in to sequential based approaches and cluster based approaches. In sequential based approaches visual features and temporal information are used to determine key-frames. That is the variation between the visual content of frames is estimated and the key-frame is selected whenever there is a considerable change. In cluster based approaches the basic idea is to produce the key-frame by clustering frames of a shot. The frames corresponding to representative of each cluster are selected as a key-frame. It should be noted that the clustering should preserve the temporal order of frames in video data. Also, the extracted key-frames may be static or dynamic in nature. The static key-frames are those frames that are extracted from the video which hold the important content of the video. Thus they are representative of video. Dynamic key-frames preserve dynamic nature of video in the sense that they are temporal ordered sequence of key-frames extracted.

4.1.1 Key-Frame Extraction Using Absolute Difference Of Histogram Of Consecutive Frames

The keyframe extraction algorithm is based on absolute difference of histogram of consecutive image frames.It is a two phase method in which first phase compute threshold using mean and standard deviation of histogram of absolute difference of consecutive image frames.Second phase extract key-frames comparing the threshold against absolute difference of consecutive image frames.The algorithm starts by extracting video frames one by one.After preprocessing each video frames histogram difference between two consecutive frames are calculated.The mean and standard deviation of absolute difference of histogram is calculated to fix a threshold point.The threshold(T) is computed using following equation.

$$T = \mu_{adh} + \sigma_{adh}$$

where μ_{adh} is mean of absolute difference and σ_{adh} is the standard deviation of absolute difference. Once the threshold is obtained next phase determine the key-frames by comparing the absolute difference of histogram against threshold.

4.2 TEXT DETECTION IN KEYFRAMES

The MSER feature detector works well for finding text regions [1]. It works well for text because the consistent color and high contrast of text leads to stable intensity profiles.

Although the MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text.A rule-based approach is used to remove non-text regions. For example, geometric properties of text can be used to filter out non-text regions using simple thresholds. Alternatively, a machine learning approach can be used to train a text vs. non-text classifier. Typically, a combination of the two approaches produces better results.Here a simple rule-based approach is used to filter non-text regions based on geometric properties.

There are several geometric properties that are good for discriminating between text and non-text regions, including:

- Aspect ratio
- Eccentricity
- Euler number
- Extent
- Solidity

Another common metric used to discriminate between text and non-text is stroke width. Stroke width is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations.If the stroke width image has very little variation over most of the region,then it indicates that the region is more likely to be a text region because the lines and curves that make up the region all have similar widths, which is a common characteristic of human readable text.In order to use stroke width variation to remove non-text regions using a threshold value, the variation over the entire region must be quantified into a single metric.Then, a threshold can be applied to remove the non-text regions.This threshold value may require tuning for images with different font styles.

All the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carry more meaningful information than just the individual characters. For example, recognizing the string 'EXIT' vs. the set of individual characters 'X','E','T','I', where the meaning of the word is lost without the correct ordering.

One approach for merging individual text regions into words or text lines is to first find neighboring text regions and then form a bounding box around these regions. To find neighboring regions, expand the bounding boxes computed earlier with regionprops. This makes the bounding boxes of neighboring text regions overlap such that text regions that are part of the same word or text line form a chain of overlapping bounding boxes. The overlapping bounding boxes can be merged together to form a single bounding box around individual words or text lines. To do this, compute the overlap ratio between all bounding box pairs. This quantifies the distance between all pairs of text regions so that it is possible to find groups of neighboring text regions by looking for non-zero overlap ratios. Once the pair-wise overlap ratios are computed, use a graph to find all the text regions ”connected” by a non-zero overlap ratio.

The indices to the connected text regions to which each bounding box belongs are used to merge multiple neighboring bounding boxes into a single bounding box by computing the minimum and maximum of the individual bounding boxes that make up each connected component.Finally, before showing the final detection results, suppress false text detections by removing bounding boxes made up of just one text region. This removes isolated regions that are unlikely to be actual text given that text is usually found in groups (words and sentences).After detecting the text regions, the text can be recognized within each bounding box. Without first finding the text regions, the output would be considerably more noisy.

4.3 SET UP KEYFRAME EVENT SETS

Organize and partition the images into training and test subsets.Organizing keyframes into events makes handling large sets of images much easier.Separate the sets into training and test keyframe subsets.

4.4 CREATE BAG OF FEATURES

Create a visual vocabulary, or bag of features, by extracting feature descriptors from representative images of each category.The bagOfFeatures object defines the features, or visual words, by using the k-means clustering algorithm on the feature descriptors extracted from trainingSets. The algorithm iteratively groups the descriptors into k mutually exclusive clusters. The resulting clusters are compact and separated by similar characteristics. Each cluster center represents a feature, or visual word.We can extract features based on a feature detector, or we can define a grid to extract feature descriptors. The grid method may lose fine-grained scale information. Using speeded up robust features (or SURF) detector provides greater scale invariance.

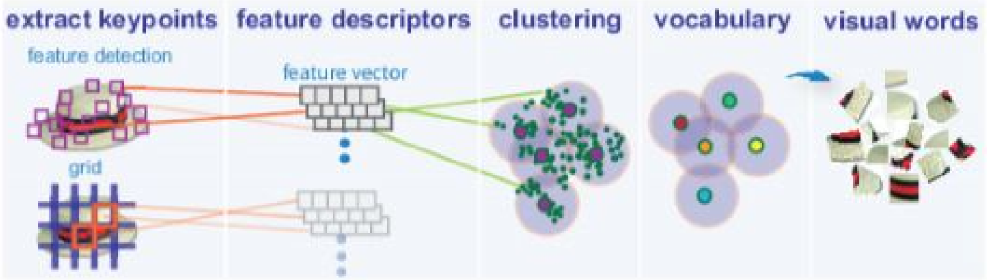


Figure 4.1: Extracting visual words from training keyframes.

The algorithm workflow analyzes keyframes in their entirety. Keyframes must have appropriate labels describing the class that they represent. The workflow does not rely on spatial information nor on marking the particular objects in a keyframe. The bag-of-visual-words technique relies on detection without localization.

4.5 TRAIN A KEYFRAME CLASSIFIER WITH BAG OF VISUAL WORDS

The keyframe classifier function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with binary support vector machine (SVM) classifiers. Thekeyframe classifier function uses the bag of visual words returned by the bagOfFeatures object to encode keyframes in the keyframe set into the histogram of visual words. The histogram of visual words are then used as the positive and negative samples to train the classifier.

1. Use the bagOfFeatures encode method to encode each image from the training set. This function detects and extracts features from the keyframe and then uses the approximate nearest neighbor algorithm to construct a feature histogram for each keyframe. The function then increments histogram bins based on the proximity of the descriptor to a particular cluster center. The histogram length corresponds to the number of visual words that the bagOfFeatures object constructed. The histogram becomes a feature vector for the keyframe.

2. Repeat the above step for each keyframe in the training set to create the training data.

3. Evaluate the quality of the classifier. The evaluate method is used to test the classifier against the validation keyframe set. The output confusion matrix

represents the analysis of the prediction. A perfect classification results in a normalized matrix containing 1s on the diagonal. An incorrect classification results fractional values.

4.6 CLASSIFY A KEYFRAME

The predict method is used on a new keyframe to determine its event. Every estimator exposes a score method that can judge the quality of the fit (or the prediction) on new data.To get a better measure of prediction accuracy (which we can use as a proxy for goodness of fit of the model), we can successively split the data in folds that we use for training and testing:This is called a KFold cross-validation.

Scikit-learn has a collection of classes which can be used to generate lists of train/test indices for popular cross-validation strategies.They expose a split method which accepts the input dataset to be split and yields the train/test set indices for each iteration of the chosen cross-validation strategy.

Chapter 5

SYSTEM DESIGN

The proposed work is news web video event mining by converting videos into keyframes and classifing them by creating bag of features.The overview of the proposed system is shown in Figure 5.1.

The input of this framework is the news web videos returned from a user query. The output is the classied events.Videos are converted to keyframes during the data preprocessing stage.Terms are detected during the text detection stage.The keyframe event sets are built and bag of features are created.Then train a keyframe classifier with bag of visual words.Finally, classify a keyframe and it is classified into events.

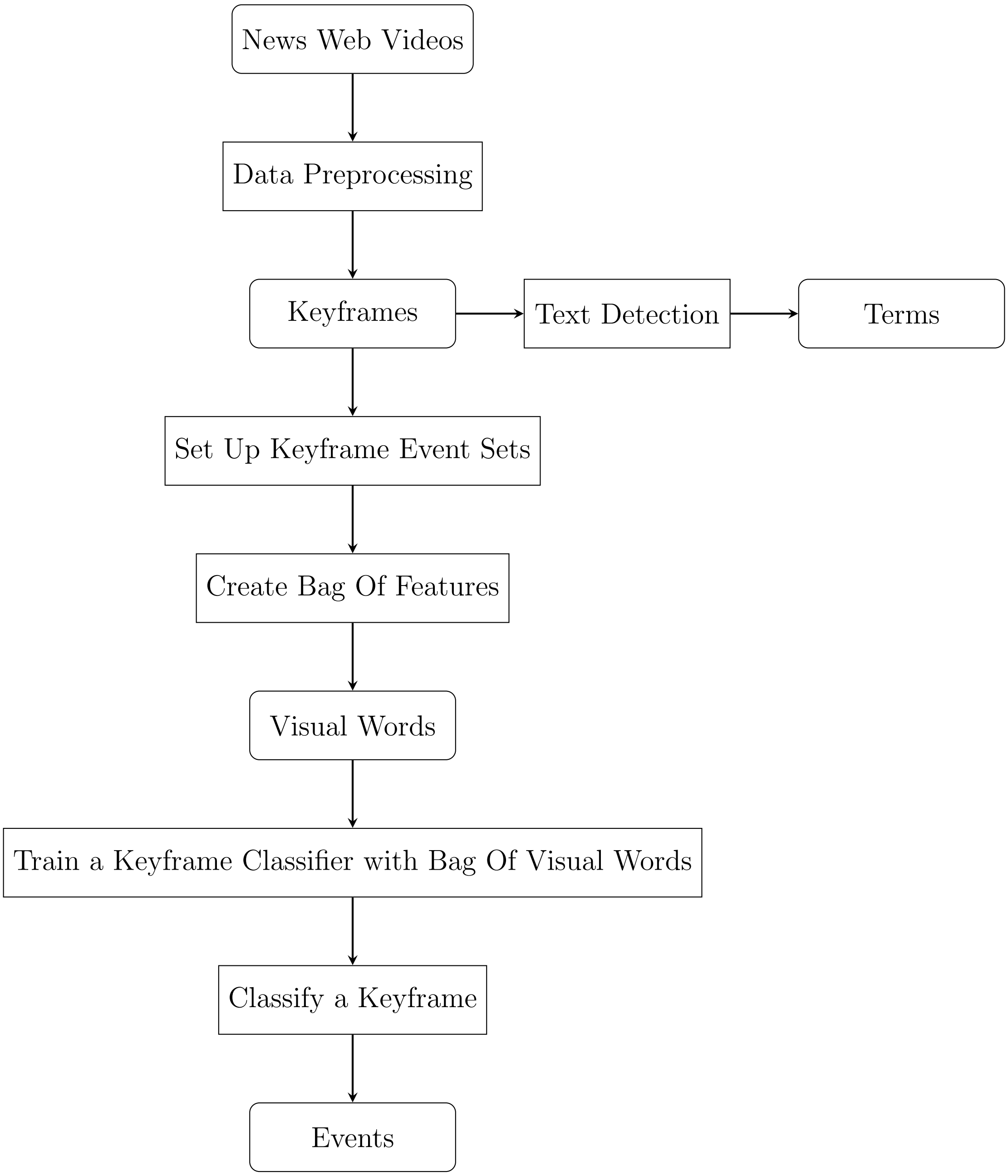


Figure 5.1: Flowchart of proposed system.

Chapter 6

IMPLEMENTATON

The method is implemented on a laptop computer.MATLAB R2016a development toolkit is used to develop this application.Millions of engineers and scientists worldwide use MATLAB to analyze and design the systems and products transforming our world. MATLAB is in automobile active safety systems, interplanetary spacecraft, health monitoring devices, smart power grids, and LTE cellular networks. It is used for machine learning, signal processing, image processing, computer vision, communications, computational finance, control design, robotics, and much more.

The MATLAB platform is optimized for solving engineering and scientific problems. The matrix-based MATLAB language is the worlds most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. A vast library of prebuilt toolboxes lets you get started right away with algorithms essential to your domain. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

MATLAB helps us take our ideas beyond the desktop.We can run our analyses on larger data sets and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling us to deploy algorithms and applications with in web, enterprise, and production systems.

Key Features:

- High-level language for scientific and engineering computing.
- Desktop environment tuned for iterative exploration, design, and problem-solving.

- Graphics for visualizing data and tools for creating custom plots.
- Apps for curve fitting, data classification, signal analysis, and many other domain-specific tasks.
- Add-on toolboxes for a wide range of engineering and scientific applications.
- Tools for building applications with custom user interfaces.
- Interfaces to C/C++, Java, .NET, Python, SQL, Hadoop, and Microsoft Excel
- Royalty-free deployment options for sharing MATLAB programs with end users

MATLAB is the easiest and most productive software for engineers and scientists. Whether we are analyzing data, developing algorithms, or creating models, MATLAB provides an environment that invites exploration and discovery. It combines a high-level language with a desktop environment tuned for iterative engineering and scientific workflows.

The matrix-based MATLAB language is the worlds most natural way to express computational mathematics. MATLAB supports both numeric and symbolic calculations. Linear algebra in MATLAB looks like linear algebra in a textbook; symbolic calculations look like the equations we write on paper. This makes it straightforward to capture the mathematics behind our ideas, which means our code is easier to write, easier to read and understand, and easier to maintain.

We can trust the results of our computations. MATLAB, which has strong roots in the numerical analysis research community, is known for its impeccable numerics. A MathWorks team of 350 engineers continuously verifies quality by running millions of tests on the MATLAB code base every day.

MATLAB does the hard work to ensure our code runs quickly. Math operations are distributed across multiple cores on our computer, library calls are heavily optimized, and all code is just-in-time compiled. We can run our algorithms in parallel by changing for-loops into parallel for-loops or by changing standard arrays into GPU or distributed arrays. Run parallel algorithms in infinitely scalable public or private clouds with no code changes.

The MATLAB language also provides features of traditional programming languages, including flow control, error handling, object-oriented programming, unit testing, and source control integration.

MATLAB provides a desktop environment tuned for iterative engineering and scientific workflows. Integrated tools support simultaneous exploration of data and programs, letting us evaluate more ideas in less time.

- We can interactively preview, select, and preprocess the data you want to import.
- An extensive set of built-in math functions supports our engineering and scientific analysis.
- 2D and 3D plotting functions enable us to visualize and understand our data and communicate results.
- MATLAB apps allow us to perform common engineering tasks without having to program. Visualize how different algorithms work with our data, and iterate until weve got the results we want.
- The integrated editing and debugging tools let us quickly explore multiple options, refine our analysis, and iterate to an optimal solution.
- We can capture our work as sharable, interactive narratives.

Comprehensive, professional documentation written by engineers and scientists is always at our fingertips to keep us productive. Reliable, real-time technical support staff answers our questions quickly. And we can tap into the knowledge and experience of over 100,000 community members and MathWorks engineers on MATLAB Central, an open exchange for MATLAB and Simulink users.

MATLAB and add-on toolboxes are integrated with each other and designed to work together. They offer professionally developed, rigorously tested, field-hardened, and fully documented functionality specifically for scientific and engineering applications.

Major engineering and scientific challenges require broad coordination to take ideas to implementation. Every handoff along the way adds errors and delays.

MATLAB automates the entire path from research through production. We can:

- Build and package custom MATLAB apps and toolboxes to share with other MATLAB users.

- Create standalone executables to share with others who do not have MATLAB.

- Integrate with C/C++, Java, .NET, and Python. Call those languages directly from MATLAB, or package MATLAB algorithms and applications for deployment within web, enterprise, and production systems.

- Convert MATLAB algorithms to C, HDL, and PLC code to run on embedded devices.

- Deploy MATLAB code to run on production Hadoop systems.

MATLAB is also a key part of Model-Based Design, which is used for multidomain simulation, physical and discrete-event simulation, and verification and code generation.

6.1 VIDEO PROCESSING USING MATLAB

MATLAB provides the necessary functionality for basic video processing tasks using a limited number of video formats. Not very long ago, the only video formats supported by built-in MATLAB functions was only the AVI, through functions such as aviread, avifile, movie2avi, and aviinfo. Moreover, the support was OS dependent and limited only to a few video codecs. Then with newer versions of MATLAB, new library functions like mmreader, was introduced to extend video support of video formats such as AVI, MPEG, and WMV (But it was true for only Windows Platform).

MATLAB represent the monochrome and colored video by creating the manipulative, 3 dimensional or 4 dimensional matrix, but provided that the video sequences are short, i.e., only few minutes of it.

When a frame needs to be processed individually, it can be easily converted into an image using the ”frame2im” function, which can then be processed using any of the functions available in the Image Processing Toolbox of MATLAB.

The MATLAB functions associated with reading video files:

- aviread: to read an AVI video file and store the frames into a MATLAB video structure, which may me actually 3D or 4D matrices, containing information about the monochrome or colored frames.
- aviinfo: returns a structure who contains information such as, frame width, frame height, total number of frames, frame rate, file size, etc.
- mmreader: constructs a multimedia [here mm stands for multimedia] object that can read video data from a variety of multimedia file formats.

Processing Video Files In MATLAB: Steps can be put in a loop if the same type of processing is to be applied to all frames of video file:

1. Convert frame to an image using frame2im.
2. Process the image using any function as required by the image processing task.
3. Then Convert the result back into a frame using im2frame function.

Playing Video Files In MATLAB:

- movie: a built-in video player of MATLAB.
- implay: a fully functional built in image and video player with many options.

Writing Video Files in MATLAB:

- avifile: creates a new AVI file that can then be populated with video frames in a variety of ways.
- movie2avi: creates an AVI file, from MATLAB Frame Sequence.

6.2 TOOLBOXES USED

The toolboxes used in this application includes Computer Vision System Toolbox, Image Processing Toolbox and Statistics and Machine Learning Toolbox.

Computer Vision System Toolbox provides algorithms, functions, and apps for designing and simulating computer vision and video processing systems. We can perform feature detection, extraction, and matching; object detection and tracking; motion estimation; and video processing. For 3-D computer vision, the system toolbox supports camera calibration, stereo vision, 3-D reconstruction, and 3-D point cloud processing. With machine learning based frameworks, we can train object detection, object recognition, and image retrieval systems.For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and C-code generation.

Image Processing Toolbox provides a comprehensive set of reference-standard algorithms, functions, and apps for image processing, analysis, visualization, and algorithm development. We can perform image analysis, image segmentation, image enhancement, noise reduction, geometric transformations, and image registration. Many toolbox functions support multicore processors, GPUs, and C-code generation.Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, embedded ICC profile, and tomographic. Visualization functions and apps let us explore images and videos, examine a region of pixels, adjust color and contrast, create contours or histograms, and manipulate regions of interest (ROIs). The toolbox supports workflows for processing, displaying, and navigating large images.

Statistics and Machine Learning Toolbox provides functions and apps to describe, analyze, and model data using statistics and machine learning. We can use descriptive statistics and plots for exploratory data analysis, fit probability distributions to data, generate random numbers for Monte Carlo simulations, and perform hypothesis tests. Regression and classification algorithms let us draw inferences from data and build predictive models. For analyzing multidimensional data, Statistics and Machine Learning Toolbox lets us identify key variables or features that impact our model with sequential feature selection, stepwise regression, principal component analysis, regularization, and other dimensionality reduction methods. The toolbox provides supervised and unsupervised machine learning algorithms, including support vector machines (SVMs), boosted and bagged decision trees, k-nearest neighbor, k-means, k-medoids, hierarchical clustering, Gaussian mixture models, and hidden Markov models.

The first stage is data preprocessing.After shot boundary detection, the middle frame in each video shot is extracted as the keyframe for the shot.Each video can be represented by a series of keyframes and is stored in each folder.Each folder is given the event label.

The algorithm for keyframe extraction is given below:

Algorithm 1 Keyframe Extraction

```
1: Extract frames one by one
2: Histogram difference between two consecutive frames
3: Calculate mean and standard deviation of absolute difference
4: Compute threshold.
5: Compute the difference with threshold (T) and if it is > T, select it as a
   keyframe else go to step 2
6: Continue till end of video
```

The next stage is text detection in keyframes.The algorithm for text detection is given below:

Algorithm 2 Text Detection

```
Detect Candidate Text Regions Using MSER.
Remove Non - Text Regions Based On Basic Geometric Properties.
Remove Non - Text Regions Based On Stroke Width Variation.
Merge Text Regions For Final Detection Result.
Recognize Detected Text Using OCR.
```

Using this algorithm, text in keyframes can be extracted.

The next stages are set up keyframe event sets, create bag of features, train a keyframe classifier with bag of visual words and classify a keyframe. These stages can be accomplished by doing the following steps:

1. Load Keyframe Sets.
   Construct an array of image sets based on the events. ImageSet operates on image file locations. It does not load all the images into memory,so it is safe to use on large image collections.

2. Preparing the Training and Validation Keyframe Sets.
   ImgSets contains an unequal number of keyframes per event. Adjust it to balance the number of keyframes in the training set. Separate the sets into training and validation data, using 30% of images from each set for the training data and the remainder, 70%, for the validation data.

3. Creating a Visual Vocabulary and Training a Keyframe Event Classifier.
   The bag-of-words technique was adapted to computer vision from the world of natural language processing. Since keyframes do not actually contain discrete words, first a vocabulary of visual words is constructed by extracting feature descriptors from representative keyframes of each event.

Feature extraction is a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector.Speeded-up robust features (SURF) detector is used to find interesting points in the images and encode information about the area around the points as a feature vector.Features can be extracted based on a feature detector like SURF, or a regularly spaced grid can be defined to extract feature descriptors. The grid method may lose fine-grained scale information, so it is best used only for images that do not contain distinct features, such as an image containing scenery, like the beach. Using a SURF detector also provides greater scale invariance.

This is accomplished with a single call to the bagOfFeatures function, which:

- Extracts SURF features from all images in all image categories.
- Constructs the visual vocabulary by reducing the number of features through quantization of feature space using K-means clustering.

Additionally, the bagOfFeatures object provides an encode method for counting the visual word occurrences in a keyframe. It produced a histogram that becomes a new and reduced representation of a keyframe.This histogram forms the basis for training a classifier and for the actual keyframe classification. In essence, it encodes a keyframe into a feature vector. Encoded training keyframes from each event are fed into a classifier training process invoked by the trainImageCategoryClassifier function. This function relies on the multiclass linear SVM classifier from Statistics and Machine Learning Toolbox.The function utilizes the encode method of the input bag object to formulate feature vectors representing each keyframe event from the trainingSets array of imageSet objects.

4. Evaluating Classifier Performance.
   Now,the trained classifier can be evaluated. First, test it with the training set, which should produce a near-perfect confusion matrix, that is, with ones on the diagonal.Next, evaluate the classifier on the validationSet, which was not used during the training. By default, the evaluate function returns the confusion matrix, which is a good initial indicator of how well the classifier is performing.

5. Try the Newly Trained Classifier on Test Keyframes.

Now apply the newly trained classifier to classify new keyframes.

Chapter 7

RESULTS AND DISCUSSION

The aim of proposed system is to classify keyframes from videos to events.The input of the proposed framework is the news web videos returned from a user query.Each videos are converted into keyframes using the algorithm for keyframe extraction.Keyframes are stored in each folder labeled with the event name.

The keyframes obtained for the video data Usain Bolt Wins Olympic 100m Gold - London 2012 Olympic Games.mp4 is shown in Figure 7.1.

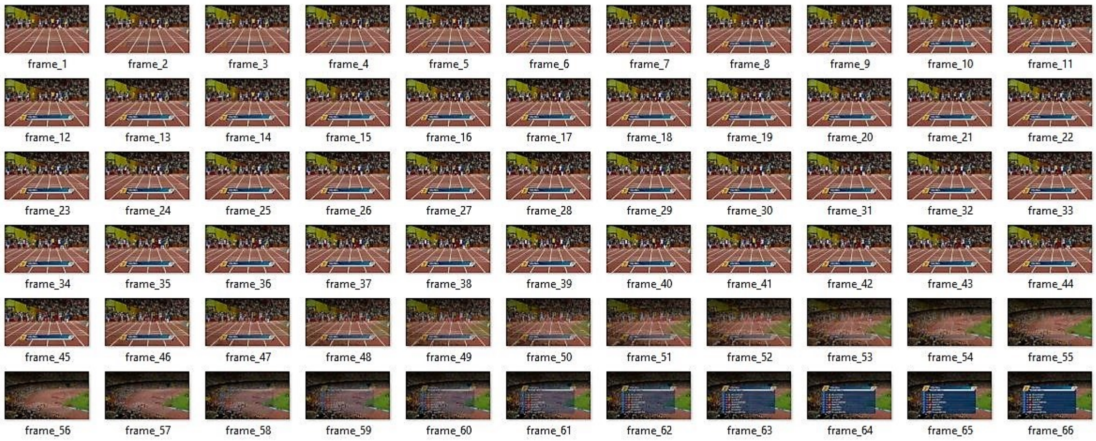


Figure 7.1: Keyframes of data Usain Bolt Wins Olympic 100m Gold - London 2012 Olympic Games.mp4.

The text detection in keyframes can be done by using the algorithm for text extraction from keyframes.The text detected from the keyframe of video data Picking a US president- 5 ways to predict the winner of 2016 election - BBC News.mp4 is shown in Figure 7.2.

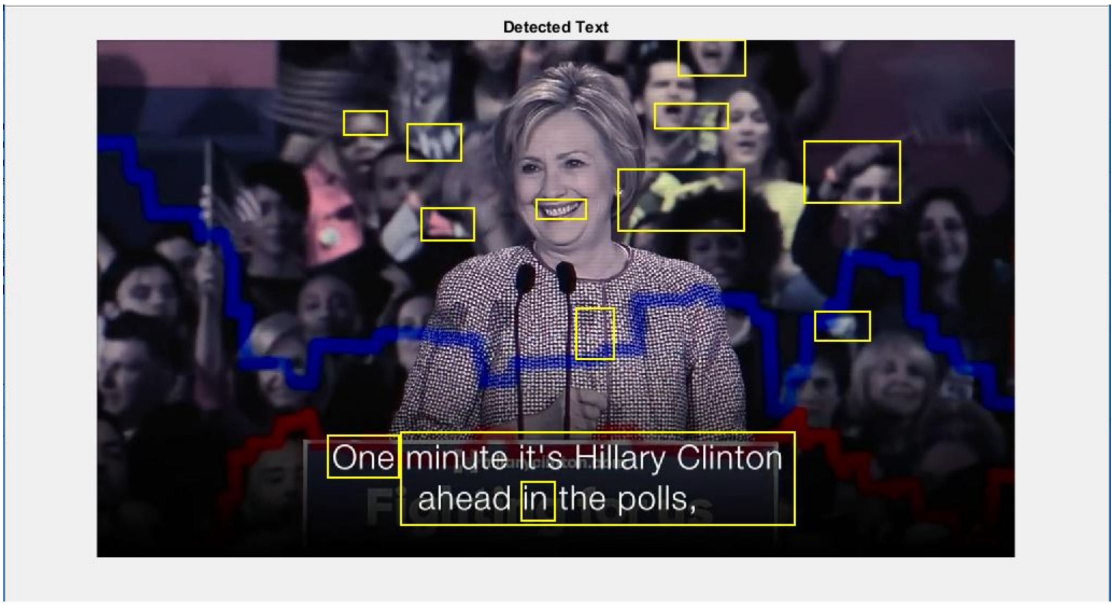
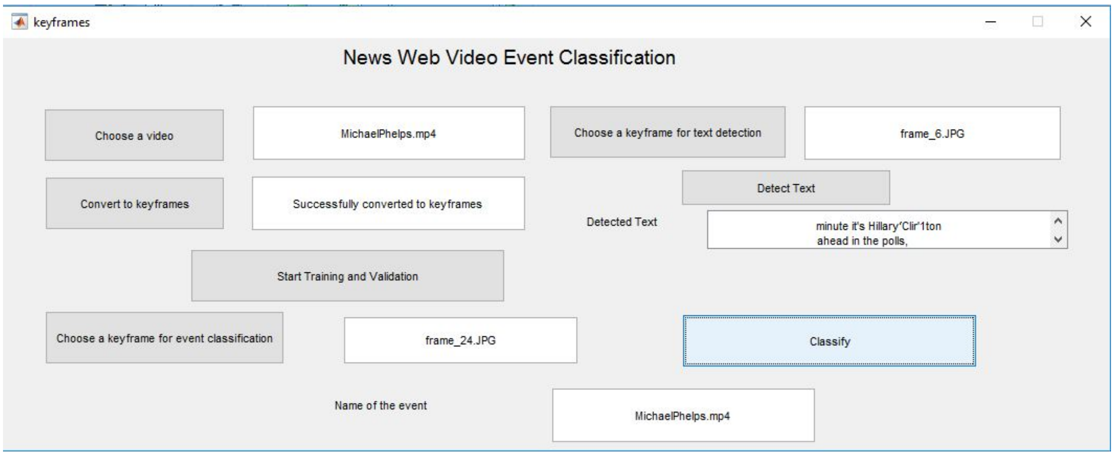


Figure 7.2: Text detected from the keyframe of video data Picking a US president-5 ways to predict the winner of 2016 election - BBC News.mp4.

Then Keyframe event sets are built,bag of features are created, a keyframe classifier with bag of visual words are trained and keyframes are classified.Using this framework, when a keyframe is given, the name of the event to which that keyframe belongs to is returned.This is shown in Figure 7.3.


keyframes
Choose a video
Convert to keyframes
Start Training and Validation
Choose a keyframe for event classification
Choose a keyframe for text detection
Detect Text
Classify

News Web Video Event Classification

MichaelPhelps.mp4

Successfully converted to keyframes

frame_6.JPG

Detected Text
minute it's Hillary'Clir'1ton
ahead in the polls,

frame_24.JPG

Name of the event
MichaelPhelps.mp4

Figure 7.3: Classifying a keyframe into event.

Chapter 8

CONCLUSION AND FUTURE WORK

The Bag of Features representation is notable because of its relative simplicity and strong performance in a number of vision tasks.It is largely unaffected by position and orientation of object in image.It is very successful in classifying images according to the objects they contain. It can significantly improve the event mining performance.

The accuracy of text detection used in the proposed framework is less.The future work of this system is to increase the accuracy of text detection in keyframes and to extract the textual features from the keyframes and use them for training and classifying.

REFERENCES

[1] **Q. He, K. Chang, and E.-P. Lim**(2007) , Analyzing feature trajectories for event detection, in *Proc. 30th ACM Int. Conf. Res. Develop. Inform. Retrieval*, pp. 207-214.

[2] **J. Yao, B. Cui, Y. Huang, and Y. Zhou**(2012) , Bursty event detection from collaborative tags, *World Wide Web*, vol. 15, no. 2, pp. 171-195.

[3] **X. Wu, Y.-J. Lu, Q. Peng, and C.-W. Ngo**(Jan.2011) , Mining event structures from web videos,*IEEE Multimedia* , vol. 18, no. 1, pp. 38-51,

[4] **C. Zhang, X. Wu, M.-L. Shyu, and Q. Peng**(2013), A novel web video event mining framework with the integration of correlation and co-occurrence information, *J. Comput. Sci. Technol.*, vol. 28, no. 5, pp. 788-796.

[5] **Chengde Zhang, Xiao Wu, Mei-Ling Shyu, and Qiang Peng** , Integration of visual temporal information and textual distribution information for news web video event mining

[6] **X. Wu, C.-W. Ngo, and A. G. Hauptmann**(Feb. 2008) , Multimodal news story clustering with pairwise visual near-duplicate constraint, *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 188-199.

[7] **J. Cao, C.-W. Ngo, Y.-D. Zhang, and J.-T. Li**(Dec. 2011), Tracking web video topics: Discovery, visualization, and monitoring, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 12, pp. 1835-1846.

[8] **X. Zhang, C. Xu, J. Cheng, H. Lu, and S. Ma**(Feb. 2009), Effective annotation and search for video blogs with integration of context and content analysis, *IEEE Trans. Multimedia*, vol. 11, no. 2, pp. 272-285.

[9] **C.Xu,Y.-F.Zhang, G.Zhu, Y.Rui, H.Lu, and Q.Huang**(Nov. 2008),Using web-cast text for semantic event detection in broadcast sports video, *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1342-1355 .

[10] **L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen**(2008) , Correlation-based video semantic concept detection using multiple correspondence analysis, *in Proc. 10th IEEE Int. Symp. Multimedia*, pp. 316-321.

[11] **L.Lin, C.Chen, M.-L.Shyu, and S.-C.Chen**(Mar. 2011) ,Weighted subspace ltering and ranking algorithms for video concept retrieval, *IEEE Multimedia*, vol. 18, no. 3, pp. 32-43.

[12] **W.-L. Zhao, X. Wu, and C.-W. Ngo**(Aug. 2010) , On the annotation of web videos by efcient near-duplicate search,*IEEE Trans. Multimedia* , vol. 12, no. 5, pp. 448-461.

[13] **X. Li, C. G. Snoek, and M. Worring**(Nov. 2009), Learning social tag relevance by neighbor voting,*IEEE Trans.Multimedia* ,vol.11,no.7,pp.1310-1322.

[14] **Stephen OHara And Bruce A. Draper**"Introduction to the Bag Of Features paradigm for image classification and retrieval.

LIST OF PUBLICATIONS

Aggie Varghese, Smitha M. Jasmine,(2017),"A Survey Paper on News Web Video Event Mining"*International journal of computer science trends and technology*, Volume 5 Issue 2.