Video Genre Classification Using Support Vector Machine Ensemble

1Ahmed A. M. Hamed, 2Zhang Xiaoming, 3Cheng Xu and 4Renfa Li
1Hunan University, Changsha 410082, China, ahmedhmed@gmail.com
2Hunan University, Changsha 410082, China, zhangxm19712003@yahoo.com.cn
3Hunan University, Changsha 410082, China, cheng_xu@yeah.net
4Hunan University, Changsha 410082, China, lirenfa@vip.sina.com

Abstract
Due to the widening semantic gap of videos, computational tools to classify these videos into
different genre are highly needed to narrow it. Classifying Videos accurately demands good
representation of video data, and efficient and effective model to carry out the classification task. This
paper proposes a new method for enhancing video classification using Support Vector Machine (SVM)
ensemble. The proposed method aims to minimize the time and computation expenses, by running
different independent SVMs trained on subset of a dataset drawn randomly from a complete dataset.
The collective decision of the different SVMs is made by Majority voting vector. Our simulation studies
on datasets representing movies, sports, and news show that our proposed method achieves good
performance in video classification as measured by the Matthew correlation coefficient (MCC), and
Qtotal.

Keywords: Video Classification, SVM, Ensemble, Performance

1. Introduction
Recently video data became a part of our daily life; people have access to huge amount of data
through internet and television. It is difficult for people to find videos of interest among these
tremendous amounts of data and it is not feasible to watch all the videos searching for the one of
interest. To help people finding their target video, videos are categorized in a certain categories or
genre, so people can search for their goal within narrow domain. Because of the large amount of
videos to categorize, automatic video classification is vital for indexing and retrieval tasks, and many
researches has started to work on it. Although many researches have been done on video classification
but it still needs more investigation due to the wide semantic gap of videos. The semantic gap refers to
the gap between video features e.g. (color, texture and audio) and semantic concepts that are
meaningful to human beings such as faces, buildings, fields, etc. Though, there are a lot of methods
have been developed to bridge the semantic gap between the video data and how can the computer
automatically process and extract the meanings of things such as human do. To get good
understanding of video content many different techniques have been developed and different video
features have been identified for better video representation. Many techniques are used for video
classification such as Bayesian, Hidden Markov Model (HMM), Gaussian Mixture Model (GMM),
Neural-Network (NN) and Support Vector Machine (SVM). The most successful used techniques in
video classification are the machine learning based techniques. Hidden Markov Model (HMM) is
popular technique generally used for classifying sequential data and pattern recognition. The basic
idea of HMM is to construct a model with hidden state variables which can be used to explain the
observable variables in sequential data, the model can be applied to other sequential data for...
prediction tasks. Some researchers has applied HMM for video analysis and classification such as Huang et al (1999), Eickeler and Muller (1999), Dimitrova et al (2000), and Gibert et al (2003) [1]. Gaussian Mixture Model (GMM) used to model large number of statistical distribution. With given feature data, a class can be modeled with multidimensional Gaussian distribution [1]. Many researchers as well used GMM for video classification such as Girgensohn and Foote (1999) and Xu and Chang (2008). Support Vector Machine (SVM) is a classification technique that has received considerable attention; it has its root in statistical learning theory and has shown promising empirical results in various different practical applications, ranging from handwritten digit recognition to text categorization [15]. One of the main reasons for SVM to have gained such a wide popularity is due to the fact that it works very well with high-dimensional data and therefore avoids the curse of dimensionality problem. Moreover, SVM has the capability to represent the decision boundary using a subset of the training examples, known as the support vectors. A SVM learning model can be formulated as a convex optimization problem. Also, the SVM can utilize various different algorithms to find the global minimum of the objective function. Other classification methods such as rule-based classifiers and artificial neural networks use a greedy strategy to search the hypothesis space, thus making those solutions only locally optimized. In contrast, the SVM can find a solution that is globally optimized for a given set of data. Another valuable feature in SVM classifier is its ability to handle dummy variables. SVM can be applied to categorical data by introducing dummy variables for each categorical attribute value present in the data. Many researchers used SVM in video classification such as Jing et al (2004), Hogyun Lee et al (2006), Zhang et al (2007), and Mehmet and Hrishikesh (2011), they used cross-video signals to improve classification performance [3, 5, 16]. Although SVM produces a high accuracy and performance but it has a serious problem which is the computational expense for both training and testing [1, 2, 13, 15, 17]. In this paper we propose a video genre classification approach based on SVM ensemble with bagging, in which subsets of the data is taken randomly with replacement to form multiple SVMs learners. The decision of these SVMs is combined with the majority voting method. The features that we have used in our approach are based on information extracted from the visual contents of the videos. Our proposed support vector machine ensemble enhance the performance of video classification, moreover having SVMs trained on subset of the data will reduce the computational time compared to training the SVM with the entire dataset.

The rest of the paper is organized as follows: Section 2 explain the material and methods that needed and used for video classification in this paper. In section 3 experimental results are depicted and section 4 presents the conclusion and discussion of the study.

2. Material and Methods

2.1.1 Features Extraction

Extracting appropriate features is vital for acceptable design of any pattern classifier. In video classification studies the features that used can be categorized as one of three which are text features and audio features and visual features some studies divide the features into two main categories text features and non-text features and the latter divided into low level and semantic video features. Text
features contain text extracted from video and user-generated text feature. Non-text features are features extracted from images, audio and motion, such feature could be referred to as low level features which could be defined as features extracted from video clips and audio track without reference to any external knowledge [1, 18, 19]. Some of researches use features that correspond to cinematic principle for visual features which composite of color, motion and average shot length. One difficulty in using low level feature especially visual features is the huge amount of data and the most used solution for this is using a key frame to represent the shot or by using dimensionality reduction techniques such as Principal Component Analysis (PCA) and wavelet transform application [1, 4, 6]. Discrete transform is widely used for feature extraction and data redundancy reduction. Proportional advantages of deterministic transforms make them an interesting type of feature extraction approaches. One of the important discrete transform is Discrete Cosine Transform (DCT) and special properties of the DCT make it a powerful transform in video processing applications [8, 9, 10, 11]. DCT has ability for data de-correlation. There are fast algorithms for DCT realization. When applying DCT to video, some coefficients are selected and others discarded for dimension reduction of data. DCT coefficient selection is important part of feature extraction process. DCT feature extraction compose of two steps, the first one DCT is applied to the entire Key frame to obtain the DCT coefficients, then in the second step some of coefficients are selected to construct feature vectors. The size of the DCT coefficient matrix is the same as the input frame. DCT by itself does not decrease data dimension, so it compress most signal information in small percent of coefficients [9]. DCT coefficients for M x N frame are given as follows:

$$f(u, v) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \times \cos \left( \frac{(2x+1)\pi u}{2M} \right) \times \cos \left( \frac{(2y+1)\pi v}{2N} \right)$$ \hspace{1cm} (1)$$

Where $$\propto (\omega)$$ is defined by

$$\propto (\omega) = \begin{cases} 
1 & \omega = 0 \\
\sqrt{2} & \omega \neq 0 
\end{cases}$$ \hspace{1cm} (2)$$

$$f(x, y)$$ is the frame intensity function and $$F(u, v)$$ is 2D matrix of DCT coefficients. As mentioned that DCT coefficient matrix is the same as the frame size, and to obtain feasible and compact low dimensional representation of the features PCA could be used [7]. Given N d-dimensional feature vectors $$\{ x_i, i = 1, 2, … N \}$$

The mean vector calculated as

$$\tau = \frac{1}{N} \sum_{i=1}^{N} x_i$$ \hspace{1cm} (3)$$

And covariance matrix is calculated as
Then the Principal Component assume the first P significant eigenvectors of \( c \), i.e. \( \{i, 1, 2, \ldots, P\} \). By constructing the eigenmatrix \( U = [v_1, v_2, \ldots, v_P] \) of \( d \times P \) dimension, an arbitrary \( d \)-dimensional original feature vector \( x \) can be represented as a new low \( P \)-dimensional vector \( y = U^T (x - \tau) \), and \( P < < N \) and \( P < < d \). [7]. In our proposed method we dealt with shot or scene concept, and take a key frame to represent the whole shot or scene. There are different algorithm implemented for extracting key frame, most of these algorithms depends on the motion part to extract the key frames. After extracting the key frames, then we used DCT to obtain the DCT coefficients. PCA is applied on the DCT coefficients to select the most significant DCT coefficients and to improve the performance of the SVM [12].

2.2. Ensemble

Ensemble (classifier combination) is a technique aiming to improve the classification accuracy by aggregating the predictions of multiple classifiers. An ensemble method constructs a set of base classifiers from training data and performs classification by taking a vote of predictions made by each classifier. Ensembles have more flexibility in the functions they can represent, this flexibility in theory can enable them to over-fit the training data more than single model would. There are many types of ensemble techniques such as Bayes-Variance Decomposition, Bagging, Boosting and Random Forests. [15]. The basic idea is constructing multiple classifiers from the original data and then aggregates their predictions when classifying unknown examples. Many ways to construct ensemble classifiers such as, by manipulating training set, in which multiple training sets are created by resampling the original data according to some sampling distribution. The sampling distribution determines how likely it is that an

![Figure1. Logical View of Ensemble Learning Method.](image-url)
example will be selected for training, and it may vary from one trail to another. Multiple classifiers are then built on the training subsets by using a particular learning algorithm. Bagging and Boosting are two examples that manipulate their data sets. Another way to construct ensemble classifier is by manipulating the input features, in this approach a subset of input features is chosen to form each training set. Random Forest is an ensemble method that manipulates its input features and uses decision tree as its base classifiers. A third way to construct ensemble classifier is by manipulating the class labels. In this method the training data is transformed into a binary class problem by randomly partitioning the class labels into two disjoint subsets, A0 and A1. Training examples whose class label belongs to subset A0 are assigned to class 0, while those belong to subset A1 are assigned to class 1, then the relabeled examples are used to train the base classifier. One last method to construct ensemble classifier is by manipulating the learning algorithm, many learning algorithms can be manipulated in such a way that applying the algorithm several times on the same training data may result in different models, such as artificial neural network if it changed its weights or topology or links. The first three methods of constructing ensemble are generic, that applicable to any classifier, but the last one depend on classifier used. The base classifiers for most of these approaches can be generated sequentially or in parallel. Figure 1 shows the logical view of ensemble learning method. The class can be obtained by taking a majority vote on the individual predictions or by weighting each prediction with accuracy of the base classifier. The algorithm below shows the steps needed to build an ensemble classifier in sequential manner.

Let D denote the original training data, k denote the number of base classifiers, and T be the test data

\[
\text{for } i = 1 \text{ to } k \text{ do} \\
\quad \text{create training set, } D_i \text{ from } D. \\
\quad \text{build a base classifier } C_i \text{ from } D_i. \\
\text{end for} \\
\text{for each test record } x \in T \text{ do} \\
\quad C^*(x) = \text{vote}(C_1(x), C_2(x), \ldots, C_k(x)) \\
\text{End for}
\]

2.3. Bagging

Bagging (bootstrap aggregation) is a technique that repeatedly samples (with replacement) from a data set according to uniform probability distribution. Each bootstrap sample has the same size as the original data. On average, a bootstrap sample D_i contain approximately 63% of the original data because each sample has probability 1-(1-1/N)^N of being selected in each D_i. [15] The basic procedure for bagging algorithm is like the follow:

Let k be the number of bootstrap samples.

\[
\text{for } i = 1 \text{ to } k \text{ do} \\
\quad \text{create a bootstrap sample of size } N, D_i. \\
\quad \text{train the base classifier } C_i \text{ on the bootstrap sample } D_i. \\
\text{end for}
\]
\[ C(x) = \arg \max \sum \delta(C_i(x) = y). \]
\[ \{ \delta(\cdot) = 1 \text{ if its argument is true and 0 otherwise} \}. \]

After training the \( k \) classifiers, a test instance is assigned to the class that receives the highest number of votes. Bagging improves generalization error by reducing the variance of the base classifiers. The performance of bagging depends on stability of the base classifier, and since we are using SVM which is stable classifier we claim begging can do the job. [15]

2.4. Video classification using SVM ensemble with bagging

In this paper we used SVM with bagging in order to enhance the accuracy and processing time of video classification. First shot detection is used to extract the key frames from the input videos; shot detection is used because it is more sophisticated method for video summary [10, 14]. Secondly we construct \( X \) and \( Y \) variables, \( X \) variable represents the features of video shot is represented by the DCT_PCA data, and \( Y \) variable which determine that video shot is set manually. Thirdly and before using SVM in our data, feature should be scaled to values between 0 and 1 using the following equation

\[ f(x) = \frac{1}{1+e^{-x}} \quad (5) \]

The advantage of this scaling is to avoid features in greater numeric range dominating those in smaller numeric range. Another advantage is to avoid numerical difficulties during the calculation when large feature value might cause numerical problem. Fourthly the RBF kernel is used because it is reasonable first choice [13]. For this RBF kernel there are two parameters (C and \( \gamma \)) which are not known before hand, to determine their values, grid search is used and the method we followed is V-fold cross validation. In V-fold cross validation, training data first divided into \( v \) subset of equal size, then one subset is tested using the classifier trained on the rest \( v-1 \) subset [13]. Grid search is implemented on C and \( \gamma \) to pick the best cross validation accuracy of C and \( \gamma \). At the end we will choose the values of C and \( \gamma \) that give high accuracy. Lastly Bagging (bootstrap aggregation) technique is used to classify the data as mentioned in figure1 and bagging algorithm above.

2.5. Performance Measures

The quality of the predictions was evaluated using five measures; Matthews correlation coefficient (MCC), QTotal, Predicted Positive Value (PPV), sensitivity, and specificity. \( FP = \) False Positive, \( FN = \) False Negative, \( TP = \) True Positive, \( TN = \) True Negative. Matthews correlation coefficient can be in the range of -1 to 1, where 1 is a perfect correlation and -1 is the perfect anti-correlation. A value of 0 indicates no correlation. It is value can be calculated as follows

\[ \text{MCC} = \frac{(TP\cdot TN - FP\cdot FN)}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}} \quad (6) \]
Q_{total} is the percentage of correctly classified residues, also called the prediction accuracy. It is given as follows

\[ Q_{total} = \frac{(TP+TN)}{TP+TN+FP+FN} \]  

(7)

PPV is the Predicted Positive Value, also called the precision or Q_{pred}. It is given as follows:

\[ Q_{pred} = \frac{TP}{TP+FP} \times 100 \]  

(8)

Sensitivity is also called recall or Q_{obs}, and is the fraction of the total positive examples that are correctly predicted, and it is calculated as follows:

\[ Q_{obs} = \frac{TP}{TP+FN} \times 100 \]  

(9)

Specificity is the fraction of total negative examples that are correctly predicted, it is value can be calculated as follows

\[ Specificity = \frac{TN}{TN+FP} \times 100 \]  

(10)

3-Experimental Results:

The steps mentioned above is tested on group of videos downloaded from youtube and uku websites, containing the three categories, News, Sport and Movies, and the selected frames of those all videos are almost about 10000 frames. The method took 75% randomly of the data for training of different category and rest 25% of the data used for testing. In the training phase 60% of the training data randomly distributed with replacement into multiple SVM base classifiers in parallel, and here we chose 10 SVM classifiers, after that, the prediction made by each classifier is taken then majority vote is applied for those predictions to classify the video shot category. The following table shows the classification performance of the each category against the rest two categories:

<table>
<thead>
<tr>
<th></th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>899</td>
<td>1407</td>
<td>36</td>
<td>38</td>
</tr>
<tr>
<td>Sport</td>
<td>760</td>
<td>1472</td>
<td>51</td>
<td>97</td>
</tr>
<tr>
<td>Movies</td>
<td>583</td>
<td>1681</td>
<td>50</td>
<td>66</td>
</tr>
</tbody>
</table>
Table 2. The Accuracy and MCC for each genre

<table>
<thead>
<tr>
<th>Genre</th>
<th>Accuracy</th>
<th>MCC</th>
<th>Qpred</th>
<th>Qobs</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>97%</td>
<td>93.47%</td>
<td>96%</td>
<td>96%</td>
<td>97.50%</td>
</tr>
<tr>
<td>Sport</td>
<td>93%</td>
<td>86.40%</td>
<td>93.7%</td>
<td>89%</td>
<td>96.65%</td>
</tr>
<tr>
<td>Movies</td>
<td>94%</td>
<td>86.65%</td>
<td>92%</td>
<td>90%</td>
<td>97%</td>
</tr>
</tbody>
</table>

Table 3. Elapsed time for training the whole dataset against 60% of dataset for each genre

<table>
<thead>
<tr>
<th>Genre</th>
<th>Elapsed time (whole data)</th>
<th>Elapsed time (60%) of data</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>18.24</td>
<td>9.23</td>
</tr>
<tr>
<td>Sport</td>
<td>22.79</td>
<td>10.53</td>
</tr>
<tr>
<td>Movies</td>
<td>20.79</td>
<td>10.56</td>
</tr>
</tbody>
</table>

Figure 2. ROC Curve for News Sport and Movie respectively

4. Conclusion and Discussion

This paper presents a new approach for automatic video classification. The paper targets three different video genre which are News Movies and sport. The major contribution of this work is that using SVM with bagging technique to enhance the classification performance and the computation time, since it is well known that SVM is a good classifier but it needs time for doing its job. The elapsed time used to train the SVM with bagging is reduced by 50% when compared with training the SVM with the entire data as depicted in table 3, and the performance is still very high as illustrated in table 2. And with the revolution of the grid computing the use of multiple SVM’s classifier distributed over different computer nodes and with minimizing the training data, for example 60% for each classifier in parallel obviously enhance the computation time and overall performance.
5. References