

Video Forgery Detection using RGB Color Correlation and Multiclass SVM Classifiers

Ms. Anshida K K
Electronics and Communication
KMCT College of Engineering
NIT Campus (PO), Calicut, India

Ms. Sabitha V
Electronics and Communication
KMCT College of Engineering
NIT Campus (PO), Calicut, India

Abstract— The concept of RGB color correlation feature extraction and multiclass SVM Classifiers can be used to locate an input video in a database of videos. The proposed system aims at providing matching video for the query video if present in the database. The proposed system helps in the area of copyright protection of videos and also for reducing the storage redundancy. The system uses the concept of both the SVM (Support Vector Machine) classifiers and RGB components. Firstly, each key frame of the video is split into non overlapping blocks. The order of average intensity of the red, green and blue components of each pixel of the nonoverlapping block and then of the key frame is determined. There are six cases which shows the relation of red, green and blue components which in turn is used to create the feature set of the colour frame and then video. The feature of the database videos are saved and query video is found using RGB intensity variations. The video sequence matching operation is performed using the multiclass SVM classifiers. Multiclass SVM can be used to classify data into multiple classes. In Multiclass SVM, two functions are present. First one is `svmtrain()` which trains the Support Vector Machine using the training data. Second, `svmclassify()` which is used to classify new data using this trained Support Vector Machine. Here `svmtrain()` uses the feature set of the database videos and create model for each of them. `svmclassify()` classifies the query video to any of the database video's class using the trained model. Thus with the help of these two functions the query video could be located in the video database if present. The proposed system provides more accuracy and fastness in determining the matching video. The computational complexity and storage complexity is very low and space complexity is satisfactory for the proposed system.

Keywords—Content Based Technology, Color correlation Support Vector Machine

I. INTRODUCTION

Today there is a large availability of multimedia and internet to common people. With the rapid advancement of multimedia processing and Internet techniques, the amount of digital content available to users has become very large and widespread. Today, it is common to observe many copies of digital media (e.g., digital audio, image, and video) on the Internet, which inevitably leads to a waste of storage resources and problems related to copyright infringement. Copyright infringement is the use of exclusive works under copyright without permission from the copyright holder. The copies of same multimedia contents also cause problems due to memory wastage.

Therefore, it is desirable to have a method of detecting such copies effectively.

Currently, there are two methods for detecting the copies: watermark-based technique and content-based technology. In watermark based techniques, watermarks are introduced into the original copies [2]. This may be some invisible signal or a visible one which helps for the ease of detection of illegal copies. These are done before distribution so this technique is classified as an active technology. But watermark techniques have many limitations. First, watermarks can easily be destroyed by malicious users [3]. The watermarks can be damaged when undergone some media processing such as geometric distortion and severe compression. The most important problem is that if the original image is not watermarked, then it is not possible to detect the other copies. In such cases, the method would fail at detecting copies. In content-based technology no extra information is added to the original copies for detection. Here based on the content of the original copy a unique signature is created. Different copy detection algorithms are used to assign a unique fingerprint using features of the original content. These features could be compared with other ones to detect copies.

Here the proposed system aims at locating copies of an input video from a database of videos [1]. The copy means the transformed version of the original video. The transformations include blurring, geometric distortion, contrast enhancement, noise contamination and reencoding. An important issue for video sequence matching is the robustness of the video feature against the abovementioned operations. Robustness is the amount that the feature of video changes after the content preserving operations.

Many content-based methods have been proposed for image copy detection [4],[5], these image-based methods cannot be extended to detect video copies and sequence matching directly since they usually lead to high storage complexity and demanding computation for feature extraction. Up until now, several effective methods have been proposed for video sequence matching. In [6], each video frame is first partitioned into 3×3 subimages, and then the ordinal measurement [7] of the average block intensity is referred to as a fingerprint for video sequence matching.

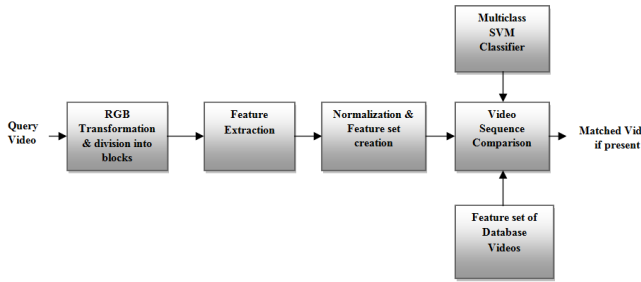


Figure 1: Architecture of video sequence matching based on multiclass SVM classifiers and RGB color correlation

In [8], Kim *et al.* modified the original scheme [6], which improved its performance, especially for letter-box and pillar-box operations. Compared with spatial ordinal measurements, the temporal measurement [9] can achieve better performance regarding the shifting and insertion of pattern. In [10] a principal component analysis (PCA)-based approach was introduced. In [11], Shang *et al.* employed the ordinal relations of 3×3 blocks as a binary feature that performs the near-duplicate web sequence retrieval. Moving Picture Experts Group (MPEG) also developed some video signature descriptors for the fast and robust detection of near-duplicate videos [12]. The robustness of the above-mentioned global features against rotation and flipping operations is still poor. SIFT [13] and CS-LBP [14] are also used to detect sequence copies, although the local features can handle many challenging distortions, their computational and storage complexity is unacceptably high [11]. In addition, some trajectory-based techniques [15], [16] have also been proposed for video sequence matching. However, these methods mainly focus on temporal distortions, such as frame deletion and insertion. Furthermore, these methods are usually expensive for the matching phase, since the trajectories must be aligned first. Methods from video hashing [17] can also be employed for the purpose of video sequence matching.

II. SYSTEM ARCHITECTURE

The system architecture is given in fig. 1. The query video is first divided into N key frames then each key frame is transformed into RGB channels and divided into blocks. For each key frame the intensity of RGB component is found according to six colour relation. This is then normalized and the feature set for a video is created. Video sequence comparison compares the feature set of database video and query video using multiclass SVM Classifier for fast and accurate comparison. The matched video is displayed as output. This matched video is the original video in the data base and hence the forgery is detected. Forgery detection is performed based on video sequence matching.

III. PROPOSED ALGORITHM

The proposed system aims at providing matching video for the query video if present in the database. The system

uses the concept of both the SVM (Support Vector Machine) classifiers and RGB components. This section provides both of them. The concept of RGB component is based on the order of average intensity of the red, green and blue components of each pixel in each of the colour frame [1]. Using this, six cases can be formed which in turn is used to create the feature set of the colour frame and then video. The feature of the database videos are saved and query video is found using RGB intensity variations. The video sequence matching operation is performed using the SVM classifiers. SVM classifiers are learning models with learning algorithms which classify data into one category or the other [18]. SVM classifiers analyze data and recognize patterns so they are used for classification and regression analysis [19]. They perform both linear and non linear classification. Here multiclass SVM classifiers are used. Multiclass SVM can be used to classify data into multiple classes. Multiclass SVM uses `svmtrain` and `svmclassify` functions for training the model and for classifying the new data.

A. Color Correlation in an Image

A digital color image (or video frame) is usually represented as a tuple of numbers, which are typically three or four values of color components [e.g., RGB, cyan, magenta, and yellow (CMY), cyan, magenta, yellow, and black (CMYK), and YCbCr color models]. These color models are employed in different applications. For instance, CMY and CMYK are mainly used for color printing. YCbCr (Y is the brightness component, while Cb and Cr denote the two chrominance components) is widely applied in video compression standards, such as MPEG-2 and H.264/AVC. RGB Model is selected here for the feature generation of video frame. It is because of its easy transformation from and to other color models and its robustness against most content-preserving operations.

Color correlation is defined as the arrangement of red, green, and blue color components in order of intensity. For a color image (or a video frame) of size $w \times h$, we use a tuple of numbers (R_{xy}, G_{xy}, B_{xy}) to represent the intensities of red, green, and blue components of the pixel at the coordinates (x, y) within the input image. The color correlation of the three color components would satisfy one of the following six ($P3 \ 3 = 6$) cases:

$$\text{case 1 : } R_{xy} \geq G_{xy} \geq B_{xy},$$

$$\text{case 2 : } R_{xy} \geq B_{xy} \geq G_{xy}$$

$$\text{case 3 : } G_{xy} \geq R_{xy} \geq B_{xy},$$

$$\text{case 4 : } G_{xy} \geq B_{xy} \geq R_{xy}$$

$$\text{case 5 : } B_{xy} \geq R_{xy} \geq G_{xy},$$

$$\text{case 6 : } B_{xy} \geq G_{xy} \geq R_{xy}$$

where $1 \leq x \leq w$, $1 \leq y \leq h$.

When these six cases are plotted as histogram, it is called as colour histogram. It is one of the most commonly

used method to represent a colour image or a video frame [21]. An image and its corresponding colour histogram is given in fig.2. The colour histogram is very fast to compute and is flexible in terms of storage when compared with other shape based or texture-based features. Features derived from the colour histogram have been used in video retrieval, segmentation, and identification. Since the colour histogram does not use any spatial information, it is expected to provide better robustness against most content preserving operations. The ability of the feature to distinguish different video clips are very promising for the proposed method.

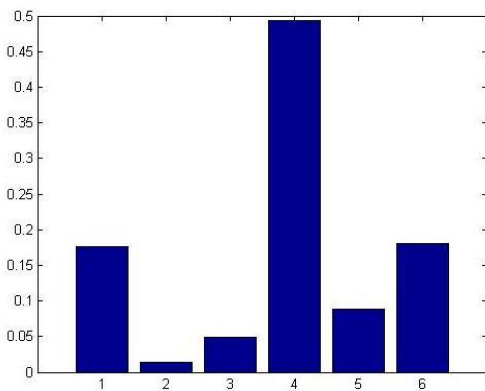


Figure 2: An image and its colour histogram

B. Feature Extraction for a Color Frame

Fig. 3 shows the process of feature extraction for a color frame, which includes the following three steps.

1) RGB Transformation and block splitting:

Each key frame of the input video is first transformed into RGB channels. Then it is divided into non overlapping blocks. Average intensities of RGB components of each pixels in these blocks are calculated to reduce the noise like operations. Let the size of key frame or image is $w \times h$. When it is divided into block of size $b \times b$, an image of size $m \times n$ is obtained where $m = \lceil w/b \rceil$, $n = \lceil h/b \rceil$, and $\lceil x \rceil$ is the nearest integer to x . The size of the block affects the performance of the algorithm.

2) Color Correlation Extraction:

For the resulting image with lower resolution $m \times n$, we extract its color correlation. In order to reduce the effect of special cases, we remove those pixels whose red, green, and blue components have the same value. Finally, we

calculate the percentage of pixels belonging to their corresponding color correlations, and obtain six normalized real values for each image.

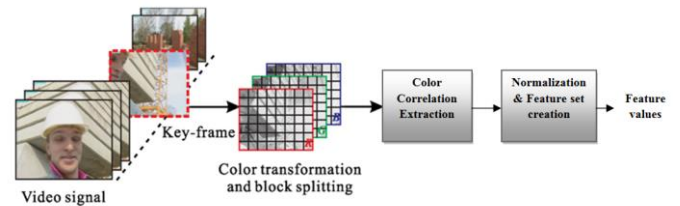


Figure 3: Basic steps of feature extraction for a color frame.

3) Normalization and Feature set creation of Video:

Let P_i denote the set of those pixels whose three color components satisfy the i th case of the six color correlations (1), where \tilde{R}_{xy} , \tilde{G}_{xy} and \tilde{B}_{xy} represent the three color components of the lower resolution image, where $1 \leq x \leq m$, $1 \leq y \leq n$ and $1 \leq i \leq 6$

$$P_i = \{(\tilde{R}_{xy}, \tilde{G}_{xy}, \tilde{B}_{xy}) \mid \tilde{R}_{xy}, \tilde{G}_{xy}, \tilde{B}_{xy} \mid \text{s.t. case } \#i\}. \quad (1)$$

The normalized histogram of the color correlations can then be described as (2), where $|P_i|$ is the cardinality of set P_i

$$H(i) = \frac{|P_i|}{\sum_{i=1}^6 |P_i|} \quad (2)$$

C. Robustness Analysis

Robustness refers to the amount that the color correlation histogram will change after some common content-preserving operations. Some examples of such operations are illustrated in Fig. 4, which shows the operations will preserve the original content of the video frame. In the following, we will analyze the robustness of the proposed feature against several typical operations. Some of the operations are:

1) *Noise-Like Contamination*: These include includes blurring and adding noise, which can change the intensity of RGB Components of individual pixels. However, such effects can be effectively decreased through averaging.

2) *Scaling, Rotation, and Flipping*: Scaling operation will change the spatial resolution of the video, but will not change the feature of original video [1]. For pure rotation, redundant pixels with the same values for the three channels will be removed in the feature extraction. Thus, there is no change for the feature. In flipping (vertical or horizontal), the values of the pixels are not changed. The operations only change the pixel positions. So the feature will be preserved.

3) *Letter-Box and Pillar-Box*: As illustrated in Fig. 4(a) and (b), letter-box and pillar-box operations occur in the video when black bars are placed on the sides of the video. This is a commonly used operation for modifying the aspect ratio of the video. For both of these operations, only the black pixels are added. Since such added pixels have the value of 0 for their red, green, and blue components,

they will be removed in feature extraction, thus the feature will be preserved.

4) *Cropping and Shifting*: As illustrated in Fig. 4(c) and (d), cropping and shifting can be modeled by replacing the original image region with black pixels.

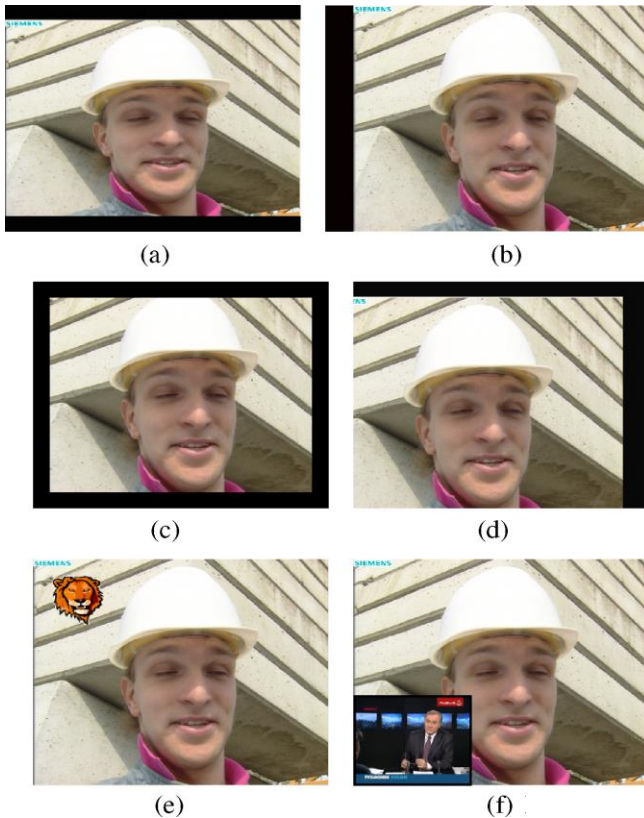


Figure 4: Illustration of six geometric operations. (a) Letter-box. (b) Pillar-box. (c) Cropping. (d) Shifting. (e) Insertion of pattern. (f) Picture in picture.

5) *Insertion of Pattern and Picture in Picture*: As illustrated in Fig. 4(e) and (f) these two operations can be modeled by the replacement of an original image region with a given pattern or picture.

6) *Contrast Enhancement*: Contrast enhancement like histogram equalization and gamma correction, is also a commonly used operation for image manipulation. Unlike grey-scale images, color images contain color information for each pixel.

D. SVM Classifiers

In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. SVMs are a new technique suitable for binary classification tasks.

• Linear SVM

In The linear SVM, the data can be classified into two groups. Here SVM chooses the best hyperplane to separate the two classes. The data point of one class is separated from the other using this hyperplane. When the margin between the two classes is largest, it is the best hyperplane. The data points that are closest to the separating hyperplane

are called support vectors. Maximizing the margin is good because it implies that only support vectors are important i.e. other training examples are ignorable and empirically it works very well.

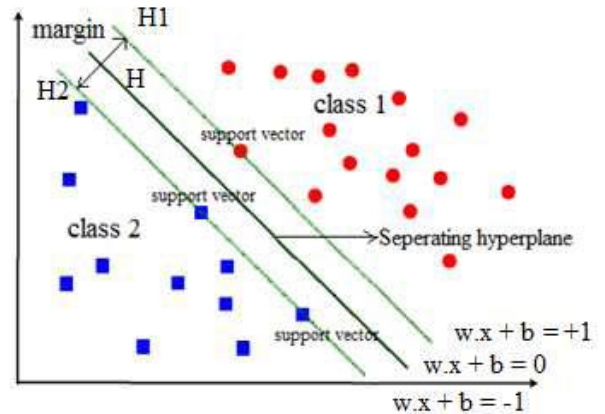


Figure 5: A separating hyperplane H which separates two Classes

The main factor of linear SVM is that the classifier is the hyperplane and the training points are the support vectors. The support vectors define the hyperplane. The fig. 5 illustrates these definitions, with Red colour indicating data points of class1, and Blue colour indicating data points of class 2. All hyperplanes in the region are represented by w , a vector and b , a constant. using the algebraic equation for a hyperplane, The hyperplane H can be expressed as $w.x+b=0$. The linear SVM finds a hyperplane $f(x)=\text{sign}(w.x+b)$, that classify the data correctly. The hyperplane H can be defined such that:

$$X_i \cdot W + b \geq +1 \quad \text{for } y_i = +1$$

$$X_i \cdot W + b \leq -1 \quad \text{for } y_i = -1$$

where Data: $\langle X_i, y_i \rangle, i = 1, \dots, n, X_i \in R^d$ and $y_i \in \{-1, +1\}$

H1 and H2 are the planes and the Support Vectors are the points on the planes H1 and H2

$$H1: X_i \cdot W + b = +1$$

$$H2: X_i \cdot W + b = -1$$

Using the distance formula, the distance between H and H1 is $|w.x+b|/\|w\|=1/\|w\|$ Therefore, the distance between H1 and H2 is: $2/\|w\|$. The main goal of linear SVM is that to maximize the margin, i.e to minimize $\frac{1}{2} w^t w$ with the condition that no datapoints lie between H1 and H2 and also correctly classify all training data $x_i \cdot w + b \geq +1$ when $y_i = +1$; $x_i \cdot w + b \leq -1$ when $y_i = -1$ which can be combined into $y_i(x_i \cdot w) \geq 1$. For that Quadratic Optimization Problem is formulated and solved for w and b . The solution has the form:

$$w = \sum_i \alpha_i y_i x_i \quad \text{and} \quad b = w \cdot x_i - y_i$$

Each non-zero α_i indicates that corresponding x_i is a support vector. Then the classifying function will have the form:

$$f(x) = \sum \alpha_i y_i x_i^T x + b$$

• Soft Margin

If the case is not perfectly separable then the margin is called soft. This means that some error had occurred in the classification of data which should be minimized. Here a non negative slack variable, ξ_i is introduced. To allow misclassification of difficult or noisy examples Slack variables ξ_i can be added. In most cases $\xi_i = 0$, that means data is correctly classified. In the case of a positive ξ_i the data point i is misclassified. The criterion for calculating w and b is that all misclassifications have to be minimized that is, $\Phi(W) = \frac{1}{2} W^T W + C \sum \xi_i$ is minimized and for all $\{(x_i, y_i)\}$

By imposing the constraint that no data point should be within the margin except some classification errors, SVMs require that either

$$X_i \cdot W + b \geq 1 - \xi$$

or

$$X_i \cdot W + b \geq 1 + \xi$$

which can be summarized with:

$$y_i (X_i \cdot W + b) \geq 1 - \xi_i$$

for all $i=1,2,3,\dots,n$

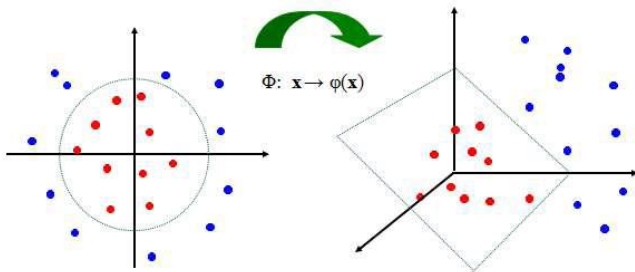


Figure 6: Transformation of an input space to feature space

• Non-linear SVMs

In non-linear SVMs the classification problem is solved by transforming the input space using Φ into a feature space of a higher dimension, where it is easier to find a separating hyperplane. It finds a hyperplane which separates the data's in the feature space and classify data points in that space. It does not need to represent the space explicitly but defines a kernel function. The role of the dot product in the feature space is played by the kernel function. Fig. 6 illustrates the transformation of an input space to a high dimensional feature space by kernel function, classifier depends on dot product between vectors $k(x_i, x_j) = x_i^T x_j$. If every data point is mapped into high-dimensional space with the use of some transformation $\Phi: x \rightarrow \phi(x)$, the dot product will be, $k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$

A kernel function is a function which corresponds to an inner product in a feature space. The non-linear SVM is thus a linear combination, but with new variables, which are derived through a kernel transformation. For nonlinear SVM, the solution is

$$f(x) = \sum \alpha_i y_i k(x_i, x_j) + b$$

• Multiclass SVM

SVM classifiers are mainly used in the case of classification between two classes. They can also be used to handle multiple classes. Then they are called Multiclass Support Vector Machines. Multiclass SVM is used here for video sequence matching. This enables the matching operation to become more fast and accurate. In multiclass SVM, there are two functions called `Svmtrain()` and `Svmclassify()`. `Svmtrain()` trains a Support Vector Machine using the training data. `Svmclassify()` is used to classify new data using this trained Support Vector Machine.

E. Matching Operation

The comparison of two video feature set occurs in this section. The comparison is done to check whether two of them are equal or not. In the matching operation, the feature set of the database video which is already saved is compared with the feature set of the input or query video. The feature set of the video is made using the RGB component relations. The main component in the matching operation is multiclass SVM Classifier.

In multiclass SVM Classifier there are two functions called `Svmtrain()` and `Svmclassify()`. `Svmtrain()` trains a Support Vector Machine using the training data. `Svmclassify()` is used to classify new data using this trained Support Vector Machine. The database videos feature set are used as training data and the query video feature set are used as test data. Group corresponds to number of database videos. The function `svmtrain()` trains the support vector machine with the feature set of database videos. For each of the database video a model is created. The function `svmclassify()` use this model which consist of different data about the database videos. It uses the model information to classify the input video to any of the database video. The classification of the input video to any one of the database video means they are matching. This provides more accuracy to the comparison of two videos and the proposed system is very fast in performance.

IV. CONCLUSION

A very promising feature for video sequence matching based on multiclass SVM Classifiers and the intensity of RGB component is introduced. The feature extracted from the video using the invariance of RGB color correlation is robust against most content preserving operations. It is already proved in the existing system. Multiclass SVM Classifiers does the matching operation between the two videos with greater accuracy and faster than the existing system. It trains the SVM classifier with the feature set of the database video using `svmtrain` function. Thus a model is created for each of the database videos which consist of information about them. Then the query video is classified to one of these models. This system is used for forgery detection of videos. The extensive results showed the effectiveness of the proposed method compared with existing works. The space complexity is also satisfactory.

ACKNOWLEDGMENT

The author express their sincere thanks to HOD, group tutor, guide, staff in Electronics and Communication Department, KMCT College of Engineering and the authors that is used to implement this paper for many fruitful discussions and constructive suggestions during the implementation of this paper.

REFERENCES

- [1] Yanqiang Lei, Weiqi Luo, Yuangen Wang, and Jiwu Huang, "Video Sequence Matching Based on the Invariance of Color Correlation," *IEEE Transactions On Circuits And Systems For Video Technology*, Vol. 22, No. 9, September 2012
- [2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no.7, pp. 1079–1107, Jul. 1999.
- [3] Y. Li and R. H. Deng, "Publicly verifiable ownership protection for relational databases," in *Proc. ACM Symp. Inform. Comput. Commun. Security*, 2006, pp. 78–89.
- [4] C. Kim, "Content-based image copy detection," *Signal Process.: Image Commun.*, vol. 18, no. 3, pp. 169–184, Mar. 2003.
- [5] J.-H. Hsiao, C.-S. Chen, L.-F. Chien, and M.-S. Chen, "A new approach to image copy detection based on extended feature sets," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp.2069–2079, Aug. 2007.
- [6] R. Mohan, "Video sequence matching," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 6. May 1998, pp. 3697–3700.
- [7] D. N. Bhat and S. K. Nayar, "Ordinal measures for image correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 4, pp.415–423, Apr. 1998.
- [8] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 127–132, Jan. 2005.
- [9] L.Chen and F. Stentiford, "Video sequence matching based on temporal ordinal measurement," *Patter Recognit. Lett.*, vol. 29, no. 13, pp. 1824– 1831, Oct. 2008.
- [10] L. Gao, Z. Li, and A. Katsaggelos, "An efficient video indexing and retrieval algorithm using the luminance field trajectory modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no 10, pp. 1566–1570, Oct. 2009.
- [11] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 531–540.
- [12] P. Brasnett, S. Paschalakis, and M. Bober, "Recent developments on standardization of MPEG-7 visual signature tools," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Sep.2010, pp.1347–1352.
- [13] D. G. Lowe, "Distinctive image features from scaleinvariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [14] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, Mar. 2009.
- [15] Y. Caspi, D. Simakov, and M. Irani, "Feature-based sequence-to-sequence matching," *Int. J.Comput. Vision*, vol. 68, no. 1, pp. 53–64, Mar. 2006.
- [16] B. Liu, Z. Li, M. Wang, and A. K. Katsaggelos, "Insequence video duplicate detection with fast point-toline matching," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1037–1040.
- [17] S. Lee and C. D. Yoo, "Robust video fingerprinting for content-based video identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 983–988, Jul. 2008.
- [18] Vojislav Kecman, "Support Vector Machines Basics," *School Of Engineering Report 616, School Of Engineering, The University Of Auckland*, April, 2004
- [19] Laura Auria and Rouslan A. Moro, "Support Vector Machines (SVM) as a Technique for Solvency Analysis," *Berlin*, August 2008
- [20] Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery 2*, 121-167, 1998
- [21] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, Jun. 1991