Content Matching – Detection of Duplicate and Near Duplicate Videos

Aqsa Zahid, Rehana Sharif
Department of Computer Science
Lahore College for Women University Lahore, Pakistan

Abstract—At present, multimedia content is one of the fastest growing data type on internet. As, there are many duplicates on web so while searching users need to spend a lot of time by watching the same clip over and over again. So in order to remove the redundant content from the web and to increase the efficiency and performance of web search there is a straightforward need for the detection of similar and nearly similar videos. Different approaches have been developed in the past and intensive research is still being carried out, at present. Furthermore, there is a straightforward need for the detection of duplicate content. This paper proposes detection of similar and nearly similar videos using texture descriptor. Note that in existing approaches this parameter (Frame rate) is not handled. A local alignment method in bioinformatics is used for the detection of similar regions between sequences. The strength of our approach is that it detects duplicates and nearly duplicate videos having accurate results for similar videos and ~92% accuracy rate for transformed videos.

Index Terms—bio-informatics, DNA matching, Texture matching, Frame rate, Video matching

I. INTRODUCTION

Social websites like yahoo, YouTube, Google, Flicker, your daily media and many more sites are used by hundreds of millions of people all around the world and multiple videos are uploaded in every minute and almost 65,000 videos uploaded per day [1]. Particularly with the popularity of social media, the number of videos on web is increasing day by day [2]. Because there is large number of duplicate content on web and it is very easy to upload a video, duplicate video and near duplicate of that video because there is no such restriction of uploading a duplicate file weather it’s a simple document or any other multimedia content. As, audiovisual material has become one of the fastest growing data type on internet so there is a straightforward need for the detection of duplicate videos. Search quality is being degraded as a result of duplicate and near duplicate content being listed in the search results. Whether Search is related to a single sentence or document level or either it is any multimedia file. The main focus of this research is to reduce laid off content on web, copyright protection. A video is basically a sequence of frames or sequence of still images in motion with respect to time.

A. NEARLY SIMILAR VIDEOS

a. A video is a combination of one or multiple clips
b. A video is a transformed version of original video i.e. change in color brightness, resolution etc.

Near-Duplicate videos are not only the copies of same original content, a transformed version of the same original content. Transformation types includes Transformations include change in color, textures, frequency, intensity, motion, gradient, reflection, rotation, translations, background, foreground, video format, frame rate, lighting effects, resize, shift, crop, gamma, contrast, brightness, saturation, blur, sharpen, many more editing operations etc. [1],[2],[4][3][6]. Same videos are uploaded after inserting logos, text overlay and captions.

As, there are many algorithms which detect duplicate videos but accuracy for matching similarity of videos is not yet 100% so there is still room for improvement in this research and according to my knowledge frame rate is still a big issue which is still not being properly solved. Frame rate as discussed earlier is the number of frames per second and if there is a minor change in frame rate, all sequence of frames will be misaligned. As in many techniques there is a frame by frame comparison of videos like color histogram based Euclidean distance, Self Similarity Matrix, Segment based Methods [1],[3],[6],[2]. These all techniques handle different types of
transformation that is blur, insertion and deletion of logo, brightness etc. but if there is a insertion and deletion of frame at the beginning, middle or end of video all sequence of frames will be misaligned so it becomes difficult to detect these types of variations by using these techniques so Markov stationary feature and co-occurrence matrix extension of color histogram is used to detect the local regions in the video that are similar to the original video. As, matching videos by using color histogram is very efficient technique as well as handle different transformation types but many video could have same color distribution so content matching by histogram is not an accurate technique along with frame by frame comparison is not able to detect duplicate videos of different frame rates.

In this research work, we propose a video matching technique by merging two outwardly unrelated field’s that is bio-informatics and computer vision and it will match sequences of two videos using local alignment method of gene sequencing and these video sequences will be generated by calculating GLCM of video frames. GLCM is one of the most common and well-known texture analysis tool [5][9].

This research will detect duplicate videos by applying local alignment method on protein sequences produced by GLCM. It will detect duplicate video and near duplicate video as well by applying single transformation type that is:

- Change in frame rate
- Insertion and deletion of frames at the beginning and end of video.
- Inserting Watermark

II. METHOD

In the first step, we will import both source and reference video in MATLAB workspace. As there are four different cases to evaluate the algorithm so according to case 1 we will import two same videos for processing then two different videos will be imported to test the algorithm and in third case two same videos having different frame rates will be evaluated and in last case two similar videos having water mark on it are also imported to workspace to assess the algorithm.

A. EXTRACTING FRAMES

As, there are large number of videos frames available so it is not possible to run all those frames at a time because of memory issues in MATLAB so in the second step of frame extraction we will select the number of frames for further processing.

B. GRAY SCALE CONVERSION

In the third step, we will convert a video into gray scale and the reason behind conversion is that each pixel takes less information than other color spaces that is RGB or CMYK. In any gray scale image or video all colors are exclusively shades of gray. Gray scale image only carries intensity information in which the value of each pixel is a single sample instead of three different values in RGB [12]. In Gray scale images gray color is actually the one in which value of Red, Green and Blue components have equal intensities. The gray scale intensity image is stored as an 8-bit integer giving 256 diverse shades of gray from black to white

C. CREATE GLCM

Gray level co-occurrence matrix is also called spatial dependence matrix and it is one of the best texture analysis tool. Co-occurrence matrix can be calculated for intensity values or other color dimensions of the image 24 and because gray scale images take less information and fewer calculations than other color spaces so we will first convert a colored video into gray scale then create co-occurrence matrix from that gray scaled image. GLCM will be calculated for the number of frames selected in step 2 for further processing.

GLCM is calculated as follows:

1. Read Intensity Values
2. In the first step of GLCM, we will read intensity values for each frame.

D. CREATE SCALED IMAGE

As, we are using Gene sequence matching technique that is used in bio-informatics. Now, in order to generate the gene sequences for each frame of video. In BLAST, two types of sequences are used that is a protein sequence and DNA sequence. [7],[8]

Now, in video matching after reading image, we get image intensity values which are much massive to map into letters so there is a need to scale those values in order to map into protein sequence. As, there are four nucleotides used in DNA matching that are A, C, G & T and in protein sequence there are 23 amino acids (i-e A, B, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, X, Y, and Z) available for sequence matching so we will scale our image intensity values up to 20 levels. Then, it will become easy to map scaled values into protein sequence characters. The NumLevels is the parameter of glcm it defines an integer value which is used to specify the number of gray levels for scaling an image which is used for converting integers to protein sequence.

For example, if the Numlevels is 20, gray level co-occurrence matrix scales the image values between 1 and 20. The number of gray levels also defines the dimensions of glcm.

E. CREATE CO-OCCURRENCE MATRICES

Different techniques have been proposed for texture analysis like FFT, semi-varioagram and glcm. The co-occurrence matrix is used to calculate the number of patterns in the scaled image. The intensity variations in an image which depict texture are normally due to some fundamental disparities in the scene. Texture is typically illustrated by 2-d intensity variations in an image. The most common texture analysis tools are 2-d histogram or co-occurrence matrices [5][9].

Spatial Gray level co-occurrence matrix (glcm) is one of the most commonly used texture feature. Gray level co-occurrence matrix is also called a gray-level spatial dependence matrix. For example, Consider an 5x5 image contain three different gray scale values i-e

Now, the three by three glcm of this Image for a displacement vector d= (1,0) is given as follows:

```
0 0 0 0 1
1 1 0 0 1
1 2 1 1 1
1 1 0 0 1
0 1 0 1 0
```
As, there are four pairs of (0, 0) in image matrix so in gray level co-occurrence matrix it contains 4 in the 0th row and 0th column and so on.

In this research, we will create gray-level co-occurrence matrix for each frame of the video. graycomatrix calculates the GLCM by computing how over and over again a pixel with grayscale intensity value i occurs parallel next to a pixel with the value j. Each element (i,j) in glcm specifies the number of times that the pixel with value i occurred horizontally adjacent to a pixel with value j.

graycomatrix calculates the GLCM from a scaled version of the image. The following figure shows how graycomatrix calculates several values in the GLCM of the 4-by-5 image I. Element (1,1) in the GLCM contains the value 1 because there is only one instance in the image where two, horizontally adjacent pixels have the values 1 and 1. Element (1,2) in the GLCM contains the value 2 because there are two instances in the image where two, horizontally adjacent pixels have the values 1 and 2. graycomatrix continues this processing to fill in all the values in the GLCM. If the neighbor pixels are outside the image boundaries, graycomatrix will ignore it.

**Figure 1 Co-occurrence matrix generation for 20 levels by using default offset [0, 1]**

<table>
<thead>
<tr>
<th>Angle</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0 D]</td>
</tr>
<tr>
<td>45</td>
<td>[-D D]</td>
</tr>
<tr>
<td>90</td>
<td>[-D 0]</td>
</tr>
<tr>
<td>135</td>
<td>[-D -D]</td>
</tr>
</tbody>
</table>

After, Co-occurrence matrix is created from scaled image, then in this step we will convert integer values to letter codes using protein sequence conversion table as shown in table 2. Protein sequence have 23 alphabets and we will get a scaled image in the form of an 8x8 matrix then we will map these matrix values according to their corresponding value in the table as shown in table 2. This protein sequence will be created for the number of frames selected in step 2.

**Table 1 Mapping of integer values to Letter Codes**

<table>
<thead>
<tr>
<th>Integer</th>
<th>Code</th>
<th>Integer</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>14</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>R</td>
<td>15</td>
<td>P</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>16</td>
<td>S</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>17</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>18</td>
<td>W</td>
</tr>
<tr>
<td>6</td>
<td>Q</td>
<td>19</td>
<td>Y</td>
</tr>
<tr>
<td>7</td>
<td>E</td>
<td>20</td>
<td>V</td>
</tr>
<tr>
<td>8</td>
<td>G</td>
<td>21</td>
<td>B</td>
</tr>
<tr>
<td>9</td>
<td>H</td>
<td>22</td>
<td>Z</td>
</tr>
<tr>
<td>10</td>
<td>I</td>
<td>23</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>L</td>
<td>24</td>
<td>*</td>
</tr>
<tr>
<td>12</td>
<td>K</td>
<td>25</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>M</td>
<td>0 or ≥26</td>
<td>?</td>
</tr>
</tbody>
</table>

To specify the relationship of pixels there is a two element vector array [row_offset, col_offset]. As the offset is expressed as an angle, the following table lists the offset values that state common angles, given the pixel distance D.

The table illustrates the array: offset = [0 1; -1 1; -1 0; -1 -1]. In this research, offset value is set to be [0, 1] which is default offset value. So, according to the offset value spatial co-occurrence matrix will be created from the scaled image matrix and this scaled image matrix is created from each frame of both videos.

**F. CONVERTING INTO PROTEIN SEQUENCE**

Table 1 Mapping of integer values to Letter Codes
III. As in simple case if we have $2 \times 2$ matrix so its protein sequence according to the conversion table, 2 corresponds to R, 11 corresponds to L, 5 and 6 corresponds to Q & E. As a result the protein sequence of the given matrix is:

$$\{2, 11, 5, 6\}$$

RL
QE

G. RESHAPING PROTEIN SEQUENCE FOR ALIGNMENT

In this step, all protein sequences which are also in an $8 \times 8$ matrix will align horizontally in a single line so these sequences will be further use for alignment. Local alignment Smith Waterman algorithm is used for the calculation of local alignment between two sequences. Local alignment method doesn’t concern the length of videos. Video can be differ in length. Local alignment method is used to detect the similar sub regions from the entire sequence. It is an example of dynamic programming and is useful for dissimilar sequences that are supposed to contain regions of similarity or similar sequence within their larger sequence context.

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

As in simple case if we have $2 \times 2$ matrix so its protein sequence according to the conversion table, 2 corresponds to R, 11 corresponds to L, 5 and 6 corresponds to Q & E. As a result the protein sequence of the given matrix is:

$$\{2, 11, 5, 6\}$$

RL
QE

**Table 2 Offset values**

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Scoring**

The scoring method is used to calculate the alignment matrix that is:

$$H_{ij} = \max \left\{ \begin{array}{c}
H_{i-1,j-1} + s(a_i, b_j) \\
H_{i-1,j} - g \\
H_{i,j-1} - g \\
0
\end{array} \right. $$

**Scoring Metric:**
- Match: $s(a_i, b_j) = 1$
- Mismatch: $s(a_i, b_j) = -1$
- Gap: -2 penalty

**Trace back**
- The trace back step determines the actual alignment
- Trace back starts from the cell with maximum value in the matrix
- Gives alignment in reverse order

**A. Final Trace back**

**Output:**
- Returns the optimal score in bits
- Showing the two sequences, Seq1 and Seq2, in the first and third rows, and symbols representing the optimal local alignment between them in the second row.
- The symbol | indicates Similarity and
- The symbol : indicates dissimilarity.

**B. CREATE DO PLOT**

Dot plot is a sort of recurrence plot. It is a way to visualize correspondence between two sequences. Sequences are represented on x and y axis. One video sequence is represented on x-axis as horizontal direction while other video sequence is represented on y-axis in vertical direction. Seqdotplot (Seq1, Seq2) plots a figure that visualizes the similarity between two sequences. Seq1 and seq2 are the sequences of two videos. Identical pairs are shaded as black and other area will remain white. Dot plot method is used to show the results. The black lines are used to show the identical patterns and white are used to show the dissimilar regions.

**ALGORITHM**

Following is the algorithm of proposed approach.

1. Import two Videos

**BEST ALIGNMENT**

```
T C G
- - -
T C G
```
else 
break
b. specifying the distance between pixel of interest and its neighbor 
offset is set to be [1,0] because its performance is better than other dimension 
offsets0 = [1, 0]
c. Convert frames into gray scale ranges from 0 to 255 gray scale values.
d. Create GLCM
e. Create Scaled Image from GLCM
i. Set Num Levels = 20
Numlevels scales image intensity values up to 20 levels and also set Dimension of GLCM.
f. Mapping of integers into letter codes
if SI value = 1
Replace with A
else if SI value = 2
Replace with R
else if SI value = 3
Replace with N
else if SI value = 4
Replace with D
else if SI value = 5
Replace with C
else if SI value = 6
Replace with Q
else if SI value = 7
Replace with E
else if SI value = 8
Replace with G
else if SI value = 9
Replace with H
else if SI value = 10
Replace with I
else if SI value = 11
Replace with L
else if SI value = 12
Replace with K
else if SI value = 13
Replace with M
else if SI value = 14
Replace with F
else if SI value = 15
Replace with P
else if SI value = 16
Replace with S
else if SI value = 17
Replace with T
else if SI value = 18
Replace with W
else if SI value = 19
Replace with Y
else if SI value = 20
Replace with V
else if SI value = 21
Replace with B
else if SI value = 22
Replace with Z
else if SI value = 23
Replace with X
else if SI value = 24
Replace with *
else if SI value = 25
Replace with -
else (SI value = 0 || S value = 26)
Replace with ?
g. Reshape protein sequence
h. Concatenate all protein Sequences By using vertcat (protienseq{:}) and store in an array.
For instance, B
i. Repeat the steps c to i for the no. of frames selected for processing.

2. Select No. of frames for Sample Video 2
Repeat All steps from c to g and store sequence of 2nd Video in C.
j. Local Alignment of B and C sequences.
i. Initialization
Matrix with X+1 rows and Y+1 columns
First row and first column filled with zero
ii. Scoring Matrix
a. If similar letters then score = 1
b. If Mismatch Score = -1
c. Gap penalty = -2
d. Select maximum value among a, b and c to put in the table.
iii. Trace back starts from the maximum value in the matrix
iv. Trace back values up to zero.
v. Gives Alignment in Reverse order.
k. Create Dot plot for graphical representation of Similar and Dissimilar regions.

III Conclusion

We proposed new algorithm to find visually similar video clips with different temporal durations and spatial variations. Our contribution is as follows. Firstly, we select texture descriptor for content matching then we select number of frames for processing. A GLCM is calculated for each frame of the video based on the texture analysis. Then these integer values are scaled in order to map these values into the protein sequence. These protein sequences are aligned and swalign method is used to detect similar patterns between sequences. As a result, the performance of the proposed algorithms is that is gives accurate results for exactly similar video and ~92% accuracy for nearly similar videos.
REFERENCES


First Author – Aqsa Zahid
Education details

- MS in software engineering-2010-2012. Lahore college for women university, Lahore, Pakistan
- Major Subjects: project management, software engineering, Advance operating system, research methodology, software design, Computer Architecture, Advance Automata Theory.

Rehana sharif

Second Author - Rehana sharif
Education details

- MS in software engineering-2010-2012. Lahore college for women university, Lahore, Pakistan
- Major Subjects: project management, software engineering, Advance operating system, research methodology, software design, Computer Architecture, Advance Automata Theory
- MCS (master computer science) - 2005-2007 with 3.00 GPA. The Islamia University of Bahawalpur, Bahawalpur, Pakistan
- PGD -2005, Degree College for women, Bahawalpur, Pakistan, Major Subjects: Computer science. Experience and skills Job Title: Junior Software Engineer, Micro Integrated Solutions, Lahore, Pakistan, Permanent address :Rana Muhammad Sharif, house number # 485, area sadiwal Kanpur district rahimyarkhan,Pakistan, Ph: +92-3036666087