IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

Role Identification in Movie using ECGM

S. S. Deore

Dept. of Computer Engineering, SRES COE, Kopargaon, SPPU University, Pune, India

Corresponding Author : sandhya.deore@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i4.100104 | Available online at: www.ijcseonline.org

Accepted: 12/Apr/2019, Published: 30/Apr/2019

***Abstract*—** Automatic identification of human face has done a significant research interest and lead to many interesting applications. Identifying faces of human is a challenging problem due to the huge variation in the appearance of each character. In this paper, we are representing the problem of identifying characters in movies or TV serials using video and film script. We are attempting to extract frames from videos and clustering the faces from frames. System will extract names from script using HTML parser. A graph structure of character name and face get generated and matching between names and clustered face tracks under the some circumstances. In complex movie scenes, during the face tracking and face clustering process, noise gets generated and due to this, the performance of the system may get limited. This paper focuses on implementation of graph structure for movie script and movie character, generating a relationship also matching this graph for identification of movie characters role.

***Keywords*—**Character Identification,K-means Clustering, ECGM, graph edit, graph matching.

I. INTRODUCTION

Face detection of characters in movies and TV programs has already peaked significant research interests and led to many challenging applications. For unknown language, identification of characters in movie is difficult task for ordinary people as well as Many TV programs have not seen for long time, so people don't remember names of characters in that program. To get the solution regarding above problems, we proposed a system which is a promising challenge due to the quick changes in the appearance of each character at every frame. Existing system generates accurate result in clear environment but the performance get decreases with increasing number of character due to face tracking and face clustering process.

Face identification and recognition is a key research problem spreading in various fields and disciplines. In Proposed system, we are going to generate graph based approach for movie characters for face recognition which uses the temporal and location based characteristics of the face from the video frames. We are clustering the faces and associate them with the names present in the script.

In movie/ TV serials, the communication between characters generates a relationship network that feels us like a small society. Every character has his/her social identification and maintains a certain relationship with others. In given video frame, the faces are nothing but the characters playing role in movie. The co-occurrence of that character in consecutive frame preserves the relationship of that character with other characters. This way the relationship network among the characters gets generated. Similarly, in given script, the frequency of spoken lines by particular character can build the character name network. Using both networks we can identify the name of character in movie. Recognition and tracking of characters in movie, even though very perceptive to viewers, still have a notable challenge in computing area. Proposed system is going to identify the characters in movie using video shots and film script as input. Most of methods for naming the faces in videos, make the use of the pixel to match both a face and the name extracted from the script of given movie video transcript.

Building Name network is somewhat easy task from give script. For building character network we need to make clusters of faces. We need to make then graph of faces and name matrices. These graphs are match to form association between faces and their name. The graph matching is done by matching the vertices' of two graphs.

Rest of the paper is organized as follows, Section I contains the introduction of Role Identification system, Section II contain the related work of the proposed system, Section III contain the architecture of system, Section IV describes results and discussion of the system, and Section V concludes research work with future directions).

II. RELATED WORK

Literature survey is nothing but the information gathering and information analysis in_software development process. This is the first step in any research work to determine the estimation cost, feasibility and research strength. Once these things are satisfied, then next step is to determine the hardware and software requirements for developing the tool. We have survey some previous systems in detail and found out their working and drawbacks. The above mentioned things have been stated below.

In paper [1], author introduced two methods for matching the faces of characters and names. One method is explained with fixed number of clusters and second method without fixed number of clusters. The final representation can be used to identify characters in news channels, movies. This is also used for character identification from movie.

Viola and Jones in paper [2], implements an algorithm to correctly detect face and increase the processing speed to detect the faces present in the images. This algorithm can be used to detect features of faces accurately. However, in this algorithm images regional facial features which have highest probability of containing the feature is being analysed. Due to regionalization of the detection area, many incorrect pictures are got removed and the processing speed of face detection is increased.

In paper [3], the method introduced by author results in accurate clustering of faces for a dataset that contained inaccurately and repeatedly labelled face images. Each face image has a number of names, automatically extracted from the associated caption. Many of sets contained the correct name. Face images are clustered in appropriate discriminate coordinates. These clustering results are used to break ambiguities in labelling and identify repeatedly labelled faces. Then results are merged to identify variants of names.

III. ARCHITECTURE

Figure1 shows the framework for identification of Characters in videos and teleplays in which inputs are as videos and scripts written for the characters in play.

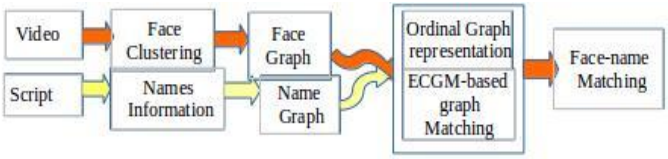


Figure1. Framework for character identification

Figure1 represent the framework of the system. Faces of every video frame are clustered using K-means algorithm, the number of group of faces depends on the number of speakers given in script. Then face clusters and name matrix are used to generate face and name graph. The matching between both graphs is generated by using graph matching method. For face graph and name graph construction, character co-occurrence in rank ordinal level is used, which calculate the bonding between characters and name given in script by ranking the vertices' of both graph. In the traditional global matching an affinity graph is used as an interval measures for relationship between characters. For name and face graph matching, the ECGM algorithm is used which measures the difference between two graphs by edit distance. ECGM contains operation on graph like edit, delete create etc. using the concept of graph. Here it uses the sequence of edit operation to match the face and name graphs.

A. Face Detection and Tracking

The first step of implementation is detection of faces from continuous feature-length film and this is achieved by segmenting videos into frames and then detecting faces. In Face Detection, feature extraction is one of the phases. In feature extraction, a features are get extracted and represent in the feature vector which is stored in database. This will form the training set and used for any query to recognize the image. This algorithm is used for face detection and is implemented using a open-cv framework. Open CV provides different face classifiers. This classifiers use haar-like [4] features that are applied over the image.

After face detection, Face clustering is nothing but clustering the face s which are geometrically aligned. Therefore after face detection, faces are normalized into 64X64 gray scale feature vector. Clustering is performed on all detected faces in LLE space. In this method, the number of cluster is equal to the number of cast list in given script. The first face tracks signature can be represented by taking the centre of cluster and faces similar to this face track with m clusters where m<=k. P={(C_{p1},W_{p1}),...,(C_{pm},W_{pm})} and Q={(Cq_1,Wq_1),...,(Cq_m,Wq_m)}is the signature of the second face track with n(n<=K) clusters. The distance between two face tracks is calculated as follows:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}$$

Where, d_{ij} is the distance between cluster centre of two face tracks. The denominator of the equation is the total weight of the smaller signature represented in normalization factor.

B. Name Entity Recognition

For name entity require to extract names from a movie script which contains a speaker lines. Figure 2 shows the movie script. From this script, through Capital Letters or any special way we can extract the speaker's name. For example WILLIAM, ANNA etc.

Hence, it is require parsing the script and extracting the name with capital or bold or may be in italic form. Then for repeated names we can calculate the number of occurrence of that name in given script.

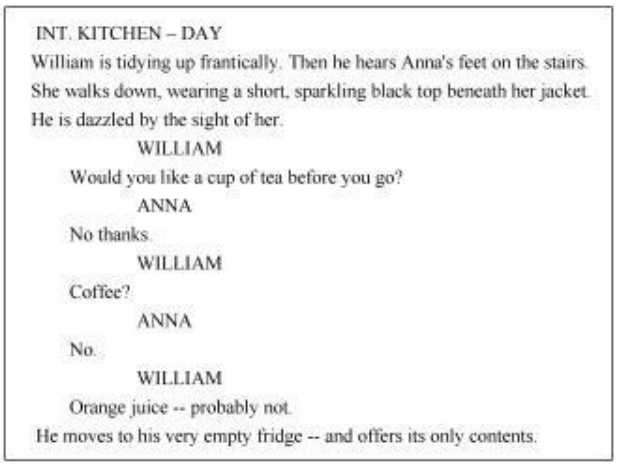

INT. KITCHEN – DAY
William is tidying up frantically. Then he hears Anna's feet on the stairs.
She walks down, wearing a short, sparkling black top beneath her jacket.
He is dazzled by the sight of her.
WILLIAM
Would you like a cup of tea before you go?
ANNA
No thanks.
WILLIAM
Coffee?
ANNA
No.
WILLIAM
Orange juice -- probably not.
He moves to his very empty fridge -- and offers its only contents.

Figure 2. Example of Movie Script

This information we can represent in matrix format as $O_{name}=[O_{ik}]_{mxn.}$ where m is number of names and n is number of scripts. O_{ik} is nothing but the count of i^{th} name in k^{th} script. Using this O_{name} matrix name affinity matrix can be constructed. Name affinity matrix is the co-occurrence of two names in the kth script.

Hence, the entry r_{ij} in name affinity matrix which represents affinity value between name i and j in entire script can be calculated as follows:

$$r_{ij} = \sum_{k=1}^{n} c_{ijk} = \sum_{k=1}^{n} \min(o_{ik}, o_{jk})$$

In Table1, the example of name affinity matrix of names from the script of the film "Notting Hill" is showed. The values are normalized into the [0,1] interval. The diagonal value r_{ii} of matrix represents the occurrence count of the i^{th} name in the entire script.

Table 1. Example of Name Affinity Matrix in Notting Hill

	WIL	SPI	ANN	MAX	BEL	HON
WILLIAM	0.173	0.024	0.129	0.009	0.013	0.008
SPIKE	0.024	0.017	0.007	0.001	0.002	0.003
ANNA	0.129	0.007	0.144	0.000	0.000	0.001
MAX	0.009	0.001	0.000	0.009	0.006	0.004
BELLA	0.013	0.002	0.000	0.006	0.011	0.006
HONEY	0.008	0.003	0.001	0.004	0.006	0.007

C. Face Entity Recognition:

Like name entity recognition, face affinity matrix is required to construct. Face affinity matrix is nothing but occurrences of faces in given video clip. To do this we need to segment video scenes to form occurrences of faces.

Using video scene segmentation same number of scenes can get from the video. Depending on this result, we can construct the face occurrence matrix, which contains the occurrence of every face in all scenes. Finally, the face affinity matrix also called as co-occurrence of faces in scene can be computed.

Table2 shows an example of face affinity matrix of some face clusters calculated from the video. Values of this matrix are normalized into [0,1] interval.

Table 2. . Example of Face Affinity Matrix in Notting Hill

	Face1	Face2	Face3	Face4	Face5	Face6
Face1	0.186	0.011	0.130	0.008	0.014	0.013
Face2	0.011	0.012	0.005	0.000	0.001	0.002
Face3	0.130	0.005	0.157	0.000	0.000	0.001
Face4	0.008	0.000	0.000	0.005	0.004	0.001
Face5	0.014	0.001	0.000	0.004	0.006	0.003
Face6	0.013	0.002	0.001	0.001	0.003	0.006

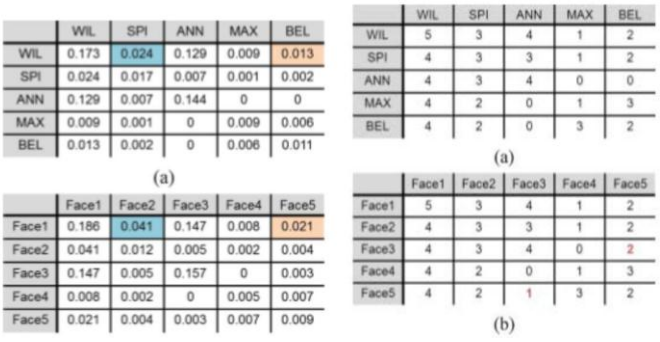
D. Graph Matching

Once name affinity matrix and face affinity matrix is constructed then graphs on both can be constructed. Since there may be noises in face affinity matrix because of incorrect faces detection and tracking. This may lead to incorrect co-occurrence between name and its related movie character. Therefore we represent here a face co-occurrence in rank order [5]. In rank order, we will consider the face affinity matrix and took the diagonal cells (e.g., A co-occurs with A, B co-occurs with B etc.). These diagonal cells values are rank according to ascending order and new face rank ordinal affinity matrix will computed as R.

Where $R = r_{ii} - I_{rii}$

Here I_{rij} = rank index of original diagonal affinity value r_{ii} .

In new matrix zero value cells indicate there is no co-occurrence between raw and column. The value of cells other than zero cell and diagonal cells are the sorted ascending values of original face matrix. Figure 3 shows an example of ordinal affinity matrices that relates to the affinity matrices in Table 1and table 2.



(a) (b)

Figure 3. (a) name affinity matrix and face affinity matrix (b) Name ordinal affinity matrix and Face ordinal affinity matrix

E. ECGM-Based Graph Matching

ECGM is an algorithm to match the similarity between two graphs [5]. It uses the edit operation to match the similarity. In this operation, lower the distortion in two graphs, smaller the cost of errors in graph. This edit graph contains the delete, insert and substitution of vertices' operation. The delete operation in edit graph carried out when particular cell in name ordinal matrix has non-zero value and corresponding cell in face ordinal matrix has zero value. An insertion operation in edit graph is exactly opposite of delete operation.

Using this graph matching method it is expected to map the faces in clusters with same character name in given script using affinity matrices of both name and face. In case if number of face clusters are more than number of names then matching of two graphs may be difficult. Hence before graph matching, a graph partition is used to solve this problem. Graph partition divides the face graph into required number of graphs. For e.g. if there are M faces and N names where M>N then using graph partitioning method face graphs are divided into N disjoint graphs. The result of graph partition is taken as a input for graph matching algorithm.

Figure4 shows the separately performing graph matching and graph partitioning operation.

Figure 4 Simultaneously graph partition and matching method

IV. RESULTS

Basically, it is a stand-alone system which works for video and its script in order to identify characters from video. System is able to identify multiple characters if it is trained. Haar-cascsade of emgu CV library is efficient for face detection from images. Training is done on the 3 scenes of 12 angry men movie which has total 8 characters. In training phase of face classification module, user has to give training images as input then he has to give images for clustering as input i.e. the path of particular folders. Cluster names are shown in one panel in the interface. Figure 3 shows the name ordinal affinity matrix and face ordinal affinity matrix for training video. Figure 5 shows the result for movie "12 angry man" where identified characters are represented in form of clusters and corresponding script names also get extracted. Third table in figure5 shows the matching result between these two graphs.

First table shows the face clusters mapped with second table name associate with that cluster. Since eight clusters means eight faces were detected and mapped them with 8 names as shown in mapped table.

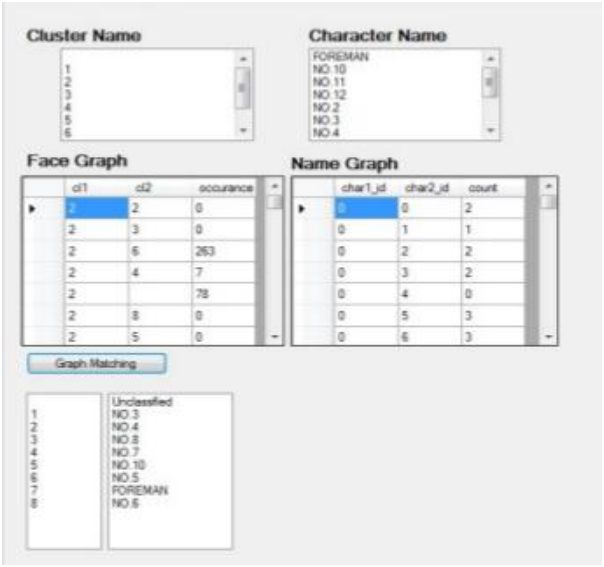


Figure 5. Snapshot of graph matching module

Figure 6. Snapshot of Identification of characters and name

Table 3 shows analysis of system. In analysis accuracy of identification of characters using ECGM is calculated by the ratio. The definition of these measures is given by following formula.

Table 3 Result Analysis

Video	No. of Character in Scene	Characters Identified	Percentage
Sample1.avi	1	1	100
Sample2.avi	4	2	50
Sample3.avi	2	2	100

This analysis is done on 3 scenes of 12 angry men movie. From this we can say system is able to identify and label the identified characters correctly

V. CONCLUSION

We have used clustering technique for identifying the character faces in movie then by building the graph of name

and face matrix we matched both using graph matching algorithm. We have mined the relationship between characters and their role. The system has given promising result for given algorithms and we have provided a platform for character centre browsing for movies.

This system can be extended as, once you identify the characters face can related face with the role he/she had played in other movies in other words history of that character. Also can be used to provide security at confidential area like banking sector.

In future, we would like to extend this work to identify the most favourite dialogues of the movie character and predicting the name of that character.

REFERENCES

[1] Jitao Sang, C. Xu, "*Robust face-name graph matching for movie character identification*", *IEEE* Transaction on Multimedia, Vol. X, NO. X, 2012.
[2] S. Satoh and T. Kanade, "Name-it: Association of face and name in video Pro-ceding's of CVPR", 1997, pp. 368373.
[3] C. Liang, C. Xu, J. Cheng, and H. Lu, "*TV parser: An automatic tv video parsing method*", in CVPR, 2011, pp. 33773384
[4] Paulo Menezes ,Jos´e Carlos Barreto et al.,"*Face Tracking Based On Haar-Like Features And Eigen faces*", ISR-University of Coimbra, Portugal.
[5] J. Yang and A. Hauptmann, "*Multiple instance learning for labelling faces in broad-casting news video*", in ACM International Conference on Multimedia, 2005, pp. 3140.
[6] Y. Zhang, C. Xu, H. Lu, and Y. Huang, "*Character identification in feature-length films using global face-name matching*," IEEE Transaction on Multimedia , vol. 11, no. 7, pp. 1276–1288, November 2009.
[7] M. Everingham, J. Sivic, and A. Zissserman, "*Taking the bite out of automated naming of characters in TV video*", in Journal of Image and Vision Computing, 2009, pp. 545559.

