INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue6)
Available online at www.ijariit.com

Video Based Face Recognition System

S. P Deepika
Student
INFO Institute of Engineering, Anna University, Chennai, Tamil Nadu
deepikaopp@gmail.com

M. Ananthi
Assistant Professor
INFO Institute of Engineering, Anna University, Chennai, Tamil Nadu
mail2ananthiinfo@gmail.com

P. Rajkumar
Assistant Professor
INFO Institute of Engineering, Anna University, Chennai, Tamil Nadu
mail2ananthiinfo@gmail.com

Abstract: The project entitled "Video Based Face Recognition" is used for automatic face identification of characters in movies. Recognition of face in videos has drawn significant research interests and lead to many interesting applications. Automatic face identification of characters in videos is a challenging problem due to the huge variation in the appearance of each character. Although existing methods demonstrate promising results in a clean environment, the performances are limited in complex movie scenes due to the noises generated during the face tracking and face clustering process. In this system, two schemes of the global face-name matching based framework are proposed for robust character identification.

Keywords: Clustering, Graph, Graph Matching, Robust, Kohenen Network, Preprocessing.

1. INTRODUCTION

Video based face recognition system is on annotating characters in the movie, videos, and TVs, which is called movie character identification. The objective is to identify the faces of the characters in the video and label them with the corresponding names in the cast. In a movie, characters are the focus center of interests for the audience. Their occurrences provide lots of clues about the movie structure and content. Automatic character identification is essential for semantic movie index and retrieval, scene segmentation, summarization and other applications. Character identification, though very intuitive to humans, is a tremendously challenging task in computer vision.

1.1 Objective of the project

Weakly supervised textual cues. There is ambiguity problem in establishing the correspondence between names and faces: ambiguity can arise from a reaction shot where the person speaking may not be shown in the frames 1; ambiguity can also arise in partially labeled frames when there are multiple speakers. Face identification in videos is more difficult than that in images. Low resolution, occlusion, no rigid deformations, large motion, complex background and other uncontrolled conditions make the results of face detection and tracking unreliable. In movies, the situation is even worse. This brings inevitable noises to the character identification. The same character appears quite differently during the movie. There may be a huge pose, expression, and illumination variation, wearing, clothing, even makeup and hairstyle changes. Moreover, characters in some movies go through different age stages, e.g., from youth to the old age. Sometimes, there will even be different actors playing different ages of the same character.

1.1.1 Back ground study

The background study is a compilation of sufficient information based on the analysis of your proposed argument or problem and the steps required to arrive at the design and implementation of feasible solutions and the results achieved. Thus background study is the work you did to determine this is a problem, these are the methods required to solve this problem and this and that is the purpose of any method or experiment used. Background study requires research and proper interpretation of the research as well as citation backing the research done. One of the preliminary steps to completing a thesis is the background study for it.

2. EXISTING SYSTEM

Existing paper gives a noise reduction method and an adaptive contrast enhancement for local TM. The local TM algorithm consists of initial luminance compression, image decomposition, noise reduction, local contrast enhancement, image synthesis, and colour reproduction. Finally, the colour of the tone-mapped image is reproduced using an adaptive saturation control parameter. Existing methods demonstrate promising results in a clean environment.

2.1 Drawbacks

The performances are limited in complex movie scenes due to the noises generated during the face tracking and face clustering process. The existing system is taking the too much time to detect the face. Lose lots of high frequency data in bright areas. Exposure to noise is high.

3. PROPOSED SYSTEM

In this Robust Face-Name Graph Matching for Movie Character Identification is used to detect the face of movie characters and the Proposed system is taking the minimum time to detect the face. Beyond existing character identification approaches, we further perform an in-depth sensitivity analysis by introducing two types of simulated noises.

The determination of the number of identical faces is not trivial. Due to the remarkable intra-class variance, the same character name will correspond to faces of huge variant appearances. It will be unreasonable to set the number of identical faces just according to the number of characters in the cast. Our study is motivated by these challenges and aims to find solutions for a robust framework for movie character identification. s

3.1 3-Tier Application

In a typical 3-tier application, the application user's workstation contains the programming that provides the graphical user interface (GUI) and application-specific entry forms or interactive windows. Business logic is located on a local area network (LAN) server or another shared computer. The business logic acts as the server for client requests from workstations. In turn, it determines what data is needed (and where it is located) and acts as a client in relation to the third tier of programming that might be located on a mainframe computer.

The third tier includes the database and a program to manage read and write access to it. While the organization of an application can be more complicated than this, the 3-tier view is a convenient way to think about the parts in a large-scale program. A 3-tier application is an application program that is organized into three major parts, each of which is distributed to a different place or places in a network. In a typical 3-tier application, the application user's workstation contains the programming that provides the graphical user interface (GUI) and application-specific entry forms or interactive windows.

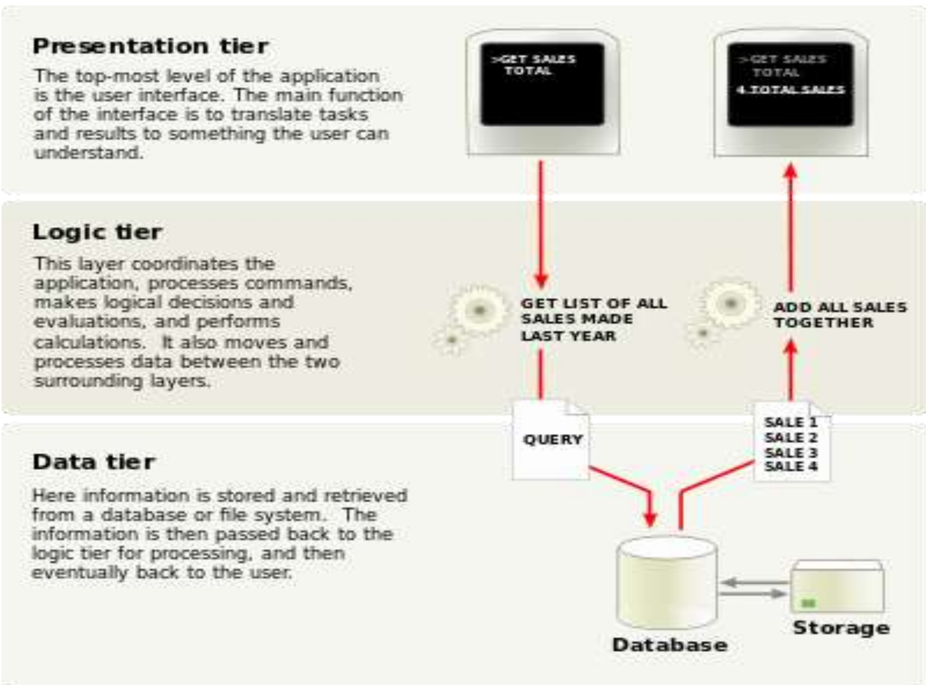

Fig 1:3-Tier Application

3.2 Network Specification

A Wide Area Network (WAN) is a network that covers a broad area (i.e., any telecommunications network that links across metropolitan, regional, or national boundaries) using private or public network transports. Business and government entities utilize WANs to relay data among employees, clients, buyers, and suppliers from various geographical locations.

In essence, this mode of telecommunication allows a business to effectively carry out its daily function regardless of location. The Internet can be considered a WAN as well and is used by businesses, governments, organizations, and individuals for almost any purpose imaginable.

4. ALGORITHMS

The input to the recognition system is acquired by scanning a plain paper containing character either handwritten or printed. The scanned paper is then saved as an image file (.BMP). Ignoring the concept of colored paper or character, the black part of the image is considered as the character and the white part is considered as the paper.

4.1 Feature Extraction and Preprocessing

Feature extraction is the process of extracting essential information contain from the image segment containing a character. It plays a vital role in the whole recognition process. This effectively reduces the number of computation and hence reduce the learning time in the training session of the neural network and faster the recognition process. In this section, each handwritten character is converted to a real valued vector form of 0s and 1s that characterizes the essential information contain the input pattern. The 0s corresponds to white pixels while 1s correspond to black pixels. To achieve this a M×N image is reduced to a $(M/R_1)\times(N/R_2)$ (here 16×16) size image with the reduction factor R1 across height and R2 across width. The cell value of the reduce picture is calculated using $R_1\times R_2$ window and calculating the number of black pixels within each window. If the number of black pixels is more than or equal to 50% of the total number of pixels in that window, the cell value is considered as 1 otherwise the cell value is considered as 0. the algorithm for this conversion is presented in algorithm

```
for I =1 to M/R1
    for j =  1 to N/R2
        BEGIN
            BlackPixelCount = 0.0
            for k= (I-1)* R1 to I*R1
                for L= (j-1)*R2 to j*R2
                    BEGIN
                        if (ReadPixel(k,L)= BlackPixel) then
                            BlackPixelCount= BlackPxilCount+1
                    END
                    if (BlackPixelCount / (R1*R2)>=0.5) then
                        FeatureMat (I,j) = 1
                    else
                        FeatureMat(I,j)= 0
            END
```

Algorithm for converting an M×N image to $M/R_1\times N/R_2$ image.

4.2 Learning and Recognition (using Kohonen Self Organization Map)

The Kohenen network has two layers, an input layer and a Kohonen out layer. The input layer is a size determined by the user and many matches the size of each row (pattern) in the input data file. A typical kohonen network is illustrated in figure-2. A kohonen feature map may be used by itself or as a layer of another neural network. A kohonen layer is composed of neurons that compete with each other. The kohonen SOM use winner takes all strategy. Inputs are feed into each of the neurons in the kohonen layer (from the input layer). Each neuron determines its output according to a weighted sum formula: Output = $\sum w_{ij} x_j$ The weights and the inputs are usually normalized which mean that the magnitude of the weight and input vectors are set equal to one. The neuron with the largest output is a winner. The neuron has a final output of 1. All other neurons in the layer have an output of zero. Different input patterns end up with firing different wining neurons. Similar or identical input patterns classify to the same output neuron. The training law for the kohonen feature map is straightforward.

The change in the weight vector for a given output neuron is a gain constant, alpha, multiplied by the difference between the input vector and the old weight vector:

$$W_{new}=W_{old}+alpha*(Input-W_{old})$$

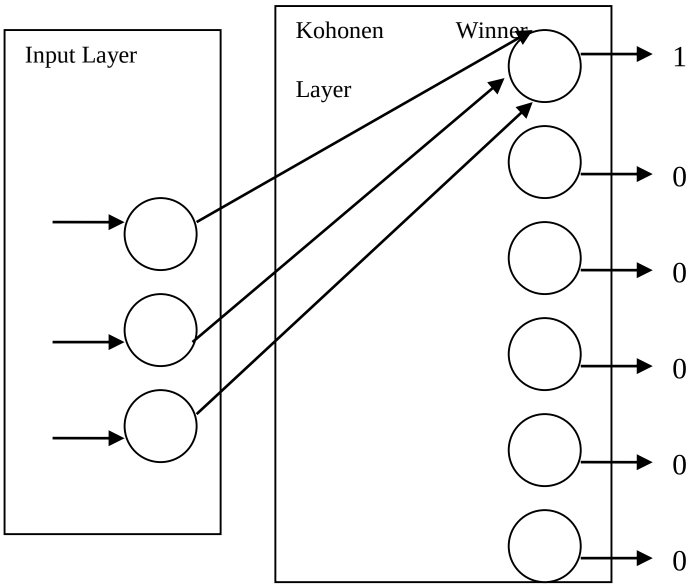

Fig 2: A Kohonen Network

Both the old weight vector and the input vector are normalized to unit length. Alpha is a gain constant between 0 and 1. Let us consider the case of a two dimensional input vector, the effect of the training law is trying to align the weight vector and the input vector. Each pattern to nudge the weight vector closer to a fraction determined by alpha. It is not necessarily ideal to have perfect alignment of the input and weight vectors. We use neural networks for their ability to recognize patterns, but also to generalize the input data sets. By aligning all input vectors to the corresponding winner weight vectors we essentially memorizing the input data set classes. It may be more desirable to come close, so that noisy or incomplete inputs may still trigger the correct classification.

4.3 Initialize network

For each node, I set the initial weight vector W_i (0) to be random.
Set the initial neighborhood N_i (0) to be large.

Present X(t), the input pattern vector at time t ($0<t\leq n$) (where n is the number of iterations defined by the user) to all nodes in the network simultaneously. X may be chosen at random or cyclically from the training data set.

4.4 Calculating winning node

Calculating winning node c based on the maximum activation among all p neurons participating
In a competition $C= \max\sum W_{ij} X_i$

So the neuron with the largest activation is the winner. The neuron has the final output of 1 or this is the firing neuron. All other neurons in the layer have an output of zero.

4.5 Update weights

Update weights for C and nodes within neighborhood Ne (t)

$W_i(t+1) = \quad W_i(t)+\alpha(t)[X(t)-W_i(t)] \qquad$ if i€ $N_e(t)$

W_i (t) $\qquad$ if i€ $N_e(t)$

The term α(t) is a gain tern or learning rate ($0\leq\alpha(t)\leq1$) that decrease in time, so showing the weight adaptation.

4.6 Present next input

Decrease α(t) so that α(t+1)<α(t). and also reduce the neighborhood set. Repeat from step 2 choosing a new unique input vector until all iterations have been made.

In the recognition phase, the network receives the input patters (feature matrix of the characters) and produce the desired the outputs corresponding to each input characters. Then comparing these outputs for finding the closest match with previously stored outputs in the training session of the characters. The test character can be determined either as recognized as unrecognized.

CONCLUSION

The system is developed using.Net as front end tool. The system will be developed after studying the requirements and necessities of the system. The system will be created in a user friendly manner with appropriate message guiding the user; even a person with minicomputer knowledge will be able to use the system. Database of the project is reduced and normalized in order to improve the efficiency and to avoid redundancy.

REFERENCES

1. Sang, C. Liang, C. Xu, and J. Cheng, "Robust movie character identification and the sensitivity analysis," in *ICME*, 2011, pp. 1–6.
2. Y. Zhang, C. Xu, H. Lu, and Y. Huang, "Character identification in feature-length films using global face-name matching," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1276–1288, November 2009.
3. M. Everingham, J. Sivic, and A. Zissserman, "Taking the bite out of automated naming of characters in tv video," in *Jounal of Image and Vision Computing*, 2009, pp. 545–559.
4. C. Liang, C. Xu, J. Cheng, and H. Lu, "Tvparser: An automatic Tv video parsing method," in *CVPR*, 2011, pp. 3377–3384.

