INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Innovation for Description Mapping Database Matching Name Domain in E-Media Using Software Design Concept

Pankaj Agarkar
Ph.D. Research Scholar, JJTU, Department of Computer Engineering,
JJT University, Jhunjhunu, Rajasthan, India
Dr. S. D. Joshi
Ph.D. Research Guide, Department of Computer Engineering,
Bharati Vidyapeeth University, COE, Katraj, Pune, India

Abstract:
Software design is the process of implementing software solutions to one or more set of problems. One of the important parts of software design is the software requirements analysis (SRA). It is a part of the software development process that lists specifications used in software engineering. If the software is "semi-automated" or user centered, software design may involve user experience design yielding a storyboard to help determine those specifications. If the software is completely automated (meaning no user or user interface), a software design may be as simple as a flow chart or text describing a planned sequence of events. There are also semi-standard methods like Unified Modeling Language and Fundamental modeling concepts. In either case, some documentation of the plan is usually the product of the design. Furthermore, a software design may be platform-independent or platform-specific, depending upon the availability of the technology used for the design.
Software design can be considered as creating a solution to a problem in hand with available capabilities. The main difference between software analysis and design is that the output of a software analysis consists of smaller problems to solve. Also, the analysis should not be very different even if it is designed by different team members or groups. The design focuses on the capabilities, and there can be multiple designs for the same problem depending on the environment that solution will be hosted. They can be operations systems, WebPages, mobile or even the new cloud computing paradigm. Sometimes the design depends on the environment that it was developed for, whether it is created from reliable frameworks or implemented with suitable design patterns.

Keywords: *face naming, social network, unconstrained web videos mining, unsupervised, software engineering concept.*

1. Introduction

Face Annotation is a note or description added to the image for better understanding. Also it can help to improve better search due to detailed description. If this annotation technique if used in videos can help in better searching of videos. The goal is to annotate unseen faces in videos with the words that best describe the image. Initially the database containing images and description mapping of that image will be gathered. Later videos that need to be processed will be considered. This videos will be converted to frames. This frame will act as images. These images will be processed with the existing database. If the faces are matched then it will be considered with the matching annotation. The matching results will produce thee matching annotation or null (the images that are not matched). Further training can be provided by the later result. The problem of naming can be traced back to name face association, where the goal is to align the observed faces with a given set of names in videos. It can be further enhanced by considering different parameters like image background and other parameters for providing better description.
Labeling celebrities in Web videos is a challenging problem due to large variations in face appearance. The problem becomes increasingly important due to the massive growth of videos in Internet. According to surveys, 1 about 80% of popular videos are people-related and among the people-related videos, about 75% are about celebrities. To date, most search engines index these videos with user-provided text descriptions (e.g., title, tag), which are often noisy and incomplete. The descriptions are given globally, and hence the correspondences between celebrity names and faces are not explicit. It is not unusual that a mentioned celebrity does not appear in the video, and vice versa, a celebrity actually appearing in a video is not mentioned. For these reasons, searching people-related videos may yield unsatisfactory retrieval performance, either because of low recall or low precision. Ideally, finding the direct correspondences between names and faces could help rectify the potential errors in text descriptions and thus serve as a preprocessing

step for video indexing. Furthermore, user search experience could be improved if the name-face correspondence is visualized, for example, by showing the name of a celebrity when a cursor moves over a face.

The problem of celebrity naming can be traced back to nameface association, where the goal is to align the observed faces with a given set of names. This problem has been attempted in the domains of news videos movies and TV series, capitalizing on the rich set of time-coded information including speech transcripts and subtitles. Nevertheless, these approaches often assume the ideal situation where the text cue is “rich” such that the given name set is free-of-noise and can perfectly match the observed faces. As a consequence, directly extending these approaches to Web video domain is not straightforward. Utilization of rich context information for face naming is also studied in the domain of personal album collection, by using timestamps, geotags, personal contact lists and social networks. Nevertheless, these approaches cannot be directly applied for domain unrestricted videos, because of the absence of context cues and prior knowledge such as family relationships for problem formulation.

2. Literature Survey

A. We investigate the problem of face identification in broadcast programs where people names are obtained from text overlays automatically processed with Optical Character Recognition (OCR) and further linked to the faces throughout the video. To solve the face-name association and propagation, we propose a novel approach that combines the positive effects of two Conditional Random Field (CRF) models: a CRF for person idolization (joint temporal segmentation and association of voices and faces) that benefit from the combination of multiple cues including as main contributions the use of identification sources (OCR appearances) and recurrent local face visual identification of the person clusters that improves identification performance thanks to the background (LFB) playing the role of a named ness feature a second CRF for the joint is of further idolization statistics.[1].

B. We describe a probabilistic method for identifying characters in TV series or movies. We aim at labeling every character appearance and not only those where a face can be detected. Consequently, our basic unit of appearance is a person track (as opposed to a face track). We evaluate our approach on the first 6 episodes of The Big Bang Theory and achieve an absolute improvement of 20% for person identification and 12% for face recognition. [2]

C. In modern face recognition, the conventional pipeline consists of four stages: detect⇒align⇒represent⇒classify. We revisit both the alignment step and the representation step by employing explicit 3D face modeling in order to apply a piecewise affine transformation, and derive a face representation from a nine-layer deep neural network. This deep network involves more than 120 million parameters using several locally connected layers without weight sharing, rather than the standard convolutional layers. Our method reaches an accuracy of 97.35% on the Labeled Faces in the Wild (LFW) dataset, reducing the error of the current state of the art by more than 27%, closely approaching human-level performance. [3]

3. Design Considerations

There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

A. Compatibility - The software is able to operate with other products that are designed for interoperability with another product. For example, a piece of software may be backward-compatible with an older version of itself.

B. Extensibility - New capabilities can be added to the software without major changes to the underlying architecture.

C. Fault-tolerance - The software is resistant to and able to recover from component failure.

D. Maintainability - A measure of how easily bug fixes or functional modifications can be accomplished. High maintainability can be the product of modularity and extensibility.

E. Modularity - the resulting software comprises well defined, independent components which lead to better maintainability. The components could be then implemented and tested in isolation before being integrated to form a desired software system. This allows division of work in a software development project.

F. Reliability - The software is able to perform a required function under stated conditions for a specified period of time.

G. Reusability - parts or all of the software can be used in other projects with no, or only slight, modification.

H. Robustness - The software is able to operate under stress or tolerate unpredictable or invalid input. For example, it can be designed with resilience to low memory conditions.

I. Security - The software is able to withstand hostile acts and influences.

J. Usability - The software user interface must be usable for its target user/audience. Default values for the parameters must be chosen so that they are a good choice for the majority of the users.[5]

K. Performance - The software performs its tasks within a user-acceptable time. The software does not consume too much memory.

L. Portability - The usability of the same software in different environments.

M. Scalability - The software adapts well to increasing data or number of users.

4. Module Description

A. Face sequence extraction: This modules consists of 2 parts
B. Frames extraction: The frames are extracted from video.

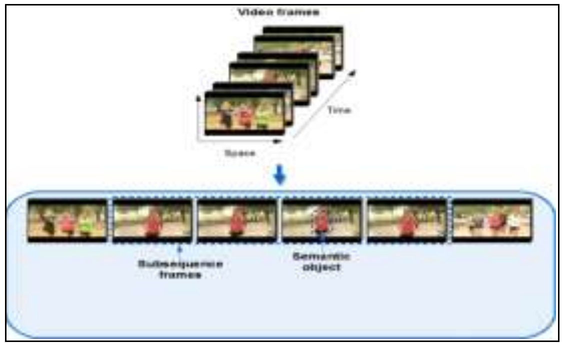

Figure 1: frames Extraction

a. Face sequence association: Frames that contains images are considered.

From that frames faces are extracted. The recognition of human faces is not so much about face recognition at all – it is much more about face detection! It has been proven that the first step in automatic facial recognition – the accurate detection of human faces in arbitrary scenes, is the most important process involved.

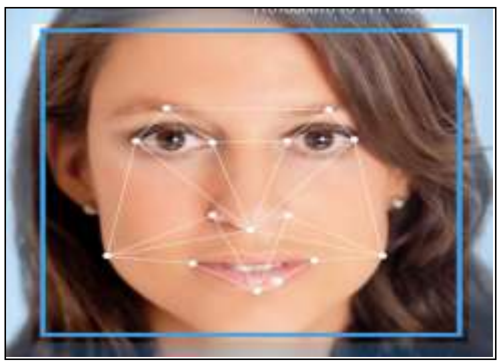

Figure 2: Face sequence association

C. Video caption recognizing: video name is also considered as a part of face identification. Although it does not play an integral part but yet it can help in sometimes.
D. Face name association: This part is the database part in the project. Here all the faces and its corresponding geometry features are stored.

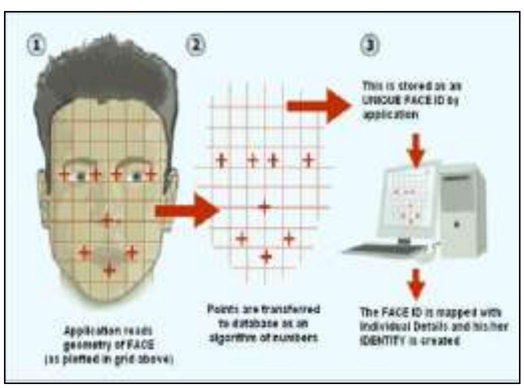

Figure 3: Face name association

E. While undergoing the face- name association, following factors are considered
   a. Lighting compensation: This will adjust contrast of image.
   b. Eigen faces: Face Images are projected into a feature space (“Face Space”) that best encodes the variation among known face images. The face space is defined by “Eigenfaces” which are the eigenvectors of the set of faces.

5. Face to Name Retrieval

1. Skin color extraction: After getting frames skin-tone color is extracted from the input image as the most important information of human face. Further coarse detection, edge extraction and blurring are executed for skin-tone flag map. This emphasizes the edges of skin-tone for face candidate detection.
2. Face judgment: After lines-of-face detection, there may be some remaining noises because the lines-of-face template can only detect skin-tone contour.
3. Template matching: The matched template will be used compares with face name association. And the corresponding name will be considered.

6. Conclusion and Future Work

We have presented the modelling of multiple relationships using CRF for celebrity naming in the Web video domain. In view of the incomplete and noisy metadata, Conditional Random Field (CRF) softly encodes these relationships while allowing null assignments by considering the uncertainty in labelling. Experimental results basically show that these nice properties lead to performance superiority over several existing approaches. The consideration of between-video relationships also results in further performance boost, mostly attributed to the capability of rectifying the errors due to missing names and persons. The price of improvement, nevertheless, also comes along with increase in processing time and the number of false positives. Fortunately, the proposals of leveraging social relation and joint labelling by sequential video processing still make CRF scalable in terms of speed and memory efficiency. While the overall performance of the proposed approach is encouraging, the effectiveness is still limited by facial feature similarity, which is used in the unary energy term and pairwise visual relationship. It can be further enhanced by considering different parameters like image background and other parameters for providing better description.

7. References

i. S. Satoh, Y. Nakamura, and T. Kanade, “Name-It: Naming and detecting faces in news videos,” IEEE Multimedia, vol. 6, no. 1, pp. 22–35, Jan.–Mar. 1999.
ii. Y. F. Zhang, C. S. Xu, H. Q. Lu, and Y. M. Huang, “Character identification in feature-length films using global face-name matching,” IEEE Trans. Multimedia, vol. 11, no. 7, pp. 1276–1288, Nov. 2009.
iii. Z. Stone, T. Zickler, and T. Darrell, “Toward large-scale face recognition using social network context,” Proc. IEEE, vol. 98, no. 8, pp.1408–1415, Aug. 2010.
iv. J. Choi, W. De Neve, K. N. Plataniotis, and Y. M. Ro, “Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks,” IEEE Trans. Multimedia, vol. 13, no. 1, pp. 14–28, Feb. 2011.
v. G. Paul, K. Elie, M. Sylvain, O. Marc, and D. Paul, “A conditional random field approach for face identification in broadcast news using overlaid text,” in Proc. IEEE Int. Conf. Image Process., Oct. 2014, pp. 318–322.
vi. J. S. Yedidia, W. Freeman, and Y. Weiss, “Constructing free-energy approximations and generalized belief propagation algorithms,” IEEE Trans. Inf. Theory, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.
vii. Suryanarayana, Girish (November 2014). Refactoring for Software Design Smells. Morgan Kaufmann. p. 258. ISBN 978-0128013977. Retrieved 31 January 2015.