IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 5, 2018, accepted December 25, 2018, date of publication January 7, 2019, date of current version January 29, 2019.

*Digital Object Identifier 10.1109/ACCESS.2018.2890560*

# DeepStar: Detecting Starring Characters in Movies

**IJAZ UL HAQ, (Student Member, IEEE), KHAN MUHAMMAD, (Member, IEEE), AMIN ULLAH, (Student Member, IEEE), AND SUNG WOOK BAIK, (Member, IEEE)**

Intelligent Media Laboratory, Digital Contents Research Institute, Sejong University, Seoul 143-747, South Korea

Corresponding author: Sung Wook Baik (sbaik@sejong.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) through the Ministry of Education under Grant 2018R1D1A1B07043302.

**ABSTRACT** Recent advances in the film industry have given rise to exponential growth in movie/drama production and adaptation of the Big Data concept. Automatic identification and classification of movie characters have received tremendous attention from researchers due to its applications in video semantic analysis, video summarization, and personalized video retrieval for which several methods have been recently presented. However, these methods cannot detect main characters properly due to their variation in pose and style in different scenes of a movie. To address this problem we present DeepStar, a novel framework for starring character identification based on deep high-level robust features. The proposed framework is threefold: the extraction of shots with clear faces from the input video; face clustering using discriminative deep features; and the occurrence matrix generation, helping in the selection of starring characters. The promising results obtained using representative Hollywood movies demonstrate the effectiveness of our method in detecting starring characters over the state-of-the-art methods.

**INDEX TERMS** Big data, face detection, starring character identification, knowledge discovery, movie analysis, deep learning, clustering.

## I. INTRODUCTION

Developments in multimedia technology have a key role in the exponential growth of video content on the internet and has several applications in various fields including image and video retrieval, human actions and activity recognition and summarization [1]. One of the categories in these videos is movie data [2]. More than 4.7 million titles (i.e. for movies and episodes) are found on a single online database, IMDb (Internet Movie Database) [3]. Besides data, the film industry has a remarkable share in the budgets of developed countries. According to [4], the global film industry revenue was 38.3 billion USD in 2016, and a healthy projection for the year 2020 is 49.3 billion USD. In such an important industry, better media content description, indexing and organization are the key points, so that users can easily browse, skim and retrieve content of interest [5]. Generally, the story of a movie is delivered by the characters through their acting and appearance, which provides meaningful presentation of the video content. Hence, characters are one of the most important contents to index, and character identification becomes a critical step in film semantic analysis [6], [7].

Extensive research efforts based on the identification of movie characters have been investigated in literature, with different applications in movies. For instance, Name-it [8] is the first method designed for news videos to identify anchors. The concept of naming characters using textual cues (script, subtitles and cast list) is used for the alignment of subtitles and indexing scenes or shots in a movie [9]–[14]. In other works [15], [16], character identification is adopted for scene segmentation. Similarly, actor-based movie summarization [6], [17] is mostly used for automatic movie trailer generation.

The available literature for character identification can be classified into two major categories: cast list-based [18]–[26] and matching-based approaches [27]–[32]. Character identification based on textual cues (script, subtitle or closed caption) are matching-based approaches. In such schemes, a text analysis module is also required to extract the names of characters automatically. In cast list-based approaches, faces with similar appearance (single character) are grouped together using clustering techniques and names from the cast list are then given to each cluster manually. Hence, no textual processing is involved for the automatic assigning of a name to a character. Our DeepStar approach belongs to this category.

Unsupervised learning based techniques like clustering have been widely used for different computer vision problems such as rating prediction and objects segmentation [33]–[36].

Similarly, the cast list problem is handled by Zhang *et al*. [5] through clustering and automatic cast list generation in movies. In this method, they utilize graph matching to create an association between the face and name affinity networks which are extracted from the video and script file. Faces are tracked using face track clustering and, for application, relationships between characters are mined using social network analysis. A similar approach is proposed by Yang and Hauptmann [18], but this method is based on speech transcript, which predicts the match between text and face in the current video frame. In another work [19], they formulated the face-name problem as a multi-instance learning (MIL) problem [20]. They focus on partially labelled data to train a model in the anonymity judgement of faces, which results in less user effort to collect. This method is also limited only to news videos. A cast list-based approach proposed by Arandjelovic and Cipolla [9] employs anisotropic manifold space to determine the cast list automatically. They mainly focus on changes of facial appearance caused by extrinsic imaging factors. Ramanan *et al*. [21] suggest a technique in which frontal faces are first grouped together, constructing a colour histogram model for face, hair and torso to track the same face in upcoming frames. The main theme of this work was building large and labelled datasets of faces by leveraging archival videos.

Identifying an individual in videos based on 3D model of the individual face is proposed by Mark and Zisserman [22]. They utilize a tree-structured classifier to detect the individual and estimate the pose over a very wide range scale and poses. Cast2Face [23] is another work presented by Mengdi *et al*. to identify characters using multi-task joint sparse representation. Their work includes retrieving web images of a particular character for training. The real name of the actor is then mapped using the cast list to identify the character name. The cast list-based approaches are simple and easy to implement because they do not have an additional module for textual cues, but manual labelling of clustering is laborious. Matching-based approaches are further divided into local and global matching approaches. The textual cues in local matching-based approaches are subtitles and close caption, while global matching-based schemes rely on script or screenplay. Zhang *et al*. [10] suggest a technique of global matching of name and faces from movie and film script. Faces are grouped into clusters corresponding to character and build a face network according to face co-occurrence relationships. Similarly, a name network is built according to name co-occurrence relationships and, finally, characters are identified after face-name association through hypergraph matching and relationship mining. A similar approach is presented in the earlier version of their work [27]. Cour *et al*. [28] utilize the readily available time-stamped resource, closed captions, which is demonstrated to be more reliable than optical character recognition based subtitles. Their work mainly focuses on overcoming the problem of local alignment between video, screenplay and closed captions.

The above methods show distinctive results and are convincing in the literature on character identification, but they have certain limitations. For instance, text/face matching schemes are highly dependent on two different modalities, which makes the system not implementable in every sort of environment. Similarly, the existing cast list-based systems use low-level features for clustering and can easily be affected by illumination or different scale and pose of the same actor's face, which limits it only to simple scenes. To tackle these problems, we propose a novel deep learning-based framework for character identification in movie data. Our system is not limited only to simple movie scenes but proves valuable and precise even for complex scenes because of its robustness.

In this paper, we use raw videos (without script or subtitles) to identify starring characters in movie scenes. Unlike other methods that use simple face detection for clustering, we present face detection as a part of the problem. Hence, considering only starring characters for clustering, our face detection methodology detects only focused characters. Deep features are extracted using the detected faces followed by clustering. An occurrence matrix is then constructed from the clusters to identify starring characters. The main contributions of this article are summarized as follows:

1. Considering the phenomenon that a starring character in a shot appears on the screen with clear representation (i.e. closeup, centroid and focused), we added three constraints on face detection that help us to select only focused characters. This strategy gives promising results in clustering by removing the unwanted faces.
2. The dominance of deep features over hand-crafted features has been proved in many recent studies. We have investigated it for starring character identification using a pre-trained VGG face recognition model for facial features representation.
3. Clustering approaches which require input to create N number of clusters are not effective for a dynamic environment such as movie data. Therefore, an adoptive clustering approach is used to effectively cluster actor faces to generate an occurrence matrix for starring character identification.

The rest of the paper is organized as follows. In Section II, we present the details of DeepStar for detecting starring characters in movie data. Experimental results and discussion are presented in Section III, followed by the conclusion and future work in Section IV.

## II. THE PROPOSED FRAMEWORK

Our proposed method is divided into three main modules: 1) pre-processing, which includes shots segmentation and face detection; 2) face clustering using deep CNN features; and 3) starring characters' determination based on an occurrence matrix. The overall framework is given in Figure 1 and the details of all three modules are given in the subsequent sections.

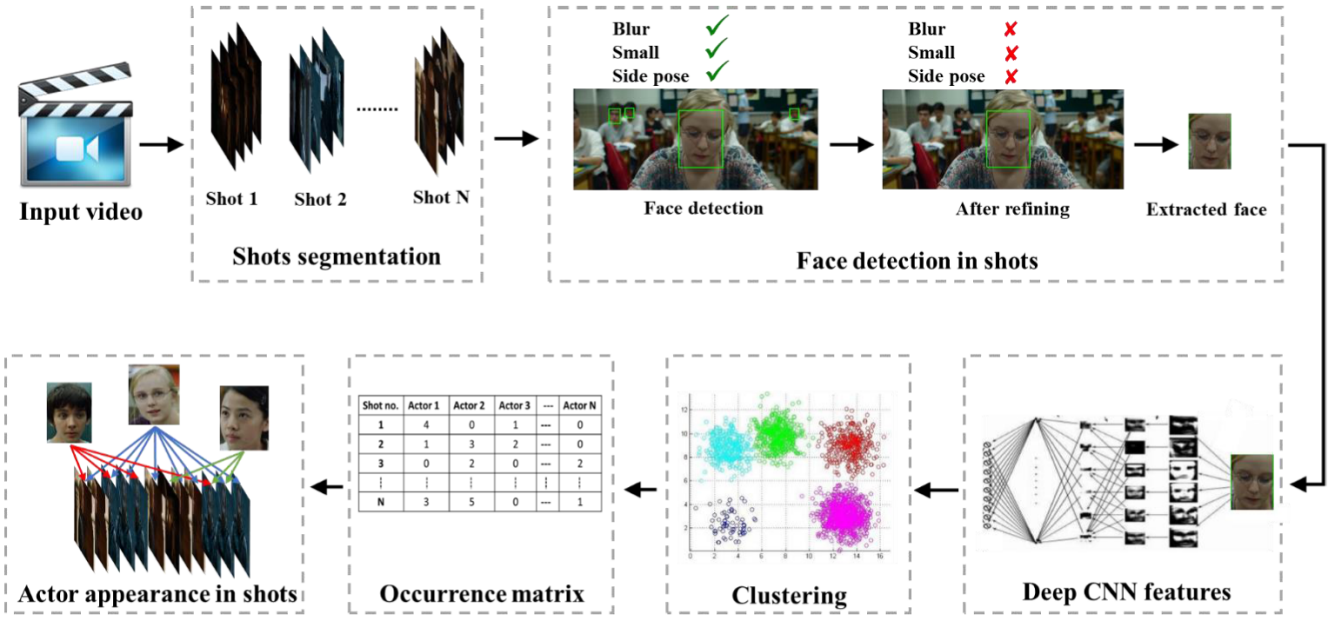


**FIGURE 1.** Overall framework of *DeepStar* for detecting starring characters in a movie.

### A. PREPROCESSING

To get shots with faces only, first we segmented the input video into shots using a histogram-based method [37]. For face detection, we used a multitask cascaded network [38], but this face detector detects almost every face in the frame and our focus is on the starring characters only. To solve this problem, we added three additional constraints to avoid faces with side pose, blur and small size. The side pose face images are simply ignored using the five points detected on the face by the same face detecting method.

To check the blurriness of cropped face images, we convolve the face image with the Laplacian operator and compute the variance as a single floating point value [39]. If the value falls below a predefined threshold, we mark the image as blurry and discard it. We performed experiments on several face images and set the threshold equal to 40 for the blur check. To avoid small detected faces, we calculate the percentage of face portion, such that if it is less than 10% of the frame, we discard it. Comparison of face detection technique used in [38] and our strategy with constraints are given in Figure 2.

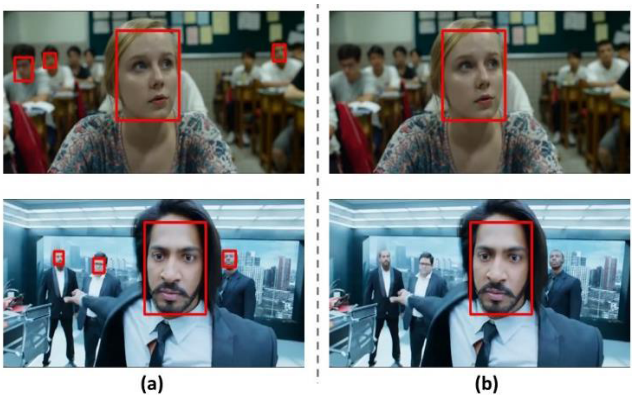


**FIGURE 2.** Face detection problem. (a) Face detection used in [38] (b) Face detection with additional constraints of our strategy.

### B. DEEP FACIAL FEATURES EXTRACTION AND CLUSTERING

Due to the recent intelligent integration of deep learning with high computational power, machines can now recognize objects, motion, activities and speech in real time [40]–[42]. However, high computational devices are expensive and not reliable for startup companies in underdeveloped countries. To solve the problem of expensive devices, we need a lightweight deep learning model which can be run on average computational power devices. Furthermore, the learning mechanism of a new deep model requires a huge amount of data, the parameters of which take many days to learn on expensive machines. Therefore, many recent studies have used a pre-trained deep learning model to solve different problems. Many deep CNN models have been presented for image classification since 2012 and have achieved significant accuracies on large-scale datasets, such as ImageNet. Every deep model has its own merits and demerits. In this study, we are using deep features of a pre-trained Deep Face [43] model for face representation. Deep Face is trained on 2.6 million images of 2,622 celebrities from different regions of the world. This huge dataset is very challenging because of the different views and poses of the actors. We have chosen the Deep Face model because it is trained on face data, unlike other deep models which are trained on the ImageNet dataset, containing generic images with additional objects in addition to faces. Therefore, we argue that Deep Face has the ability to represent faces with

**TABLE 1.** Deep face CNN model configuration details, including layer type, stride, padding, size and dimensions of the filters.

| Layers | Conv1_1 Conv1_2 | Max Pooling | Conv2_1 Conv2_2 | Max Pooling | Conv3_1 Conv3_2 | Conv3_3 | Max Pooling | Conv4_1 Conv4_2 | Conv4_3 | Max Pooling | Conv5_1 Conv5_2 | Conv5_3 | Max Pooling | FC_6 | FC_7 | FC_8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Kernel Size | 3x3 | | 3x3 | | 3x3 | 3x3 | | 3x3 | 1x1 | | 3x3 | 1x1 | | Inner Product | Inner Product | Inner Product |
| Stride, Pad | 1,1 | | 1,1 | | 1,1 | 1,1 | | 1,1 | 1,1 | | 1,1 | 1,1 | | | | |
| Channels | 64 | | 128 | | 256 | 256 | | 512 | 512 | | 512 | 512 | | 4,096 | 4,096 | 2,622 |

learned features. Moreover, it is proved through our experiments that deep features of this model are effective for facial images' representation.

The architecture of the Deep Face CNN model is explained in Table 1. The initial layers are convolved by 3x3 kernels with one padding and one stride. The small kernel size makes the model lightweight by reducing the number of learning parameters, and the small size stride help us to learn all possible patterns in the visual data. It can be seen from Table 1 that in the Deep Face model, not every convolutional layer is followed by a pooling layer, and two or three consecutive convolutional layers allow the model to learn discriminative patterns effectively. The Deep Face model contains three fully connected layers, where the FC_6and FC_7 layers extract 4,096 features each, and the last FC_8 layer extracts 2,622 features. Many recent studies on analysing intermediate layers of CNN models have proved that the initial layers of the CNN model capture the local patterns in images, and final layers are the global representation of visual data [44]–[46]. Therefore, we have used features of the final layer of the Deep Face CNN model to cluster actor faces in movie videos to find starring characters.

The next step is to make clusters of the similar faces based on the extracted deep features from the FC_8 layer of Deep Face CNN model. We used deep adaptive clustering (DAC) [47] for two main reasons.

- Clustering of character faces in movies is a dynamic problem where we cannot specify the total number of characters to be appeared in a movie scene in advance. Thus, traditional clustering algorithms like K-mean and other need a prior knowledge about the number of clusters K, therefore DAC best fits in this scenario and allows us to dynamically select the total number of clusters in a movie.
- DAC is specially designed for high dimensional deep features which performs clustering task using a binary pairwise classification to check whether or not pairs of extracted features belong to the same cluster. In contrast, traditional clustering methods fail to process high dimensional deep features. Further, DAC measures the similarity of features using cosine distance.

Figure 3 shows the robustness of our clustering method—i.e., sample images from a single cluster from the movie *Notting Hill* in which the same characters, *Anna Scott* and *William Thacker,* with different facial expressions and hair styles, are clustered in the same respective clusters.

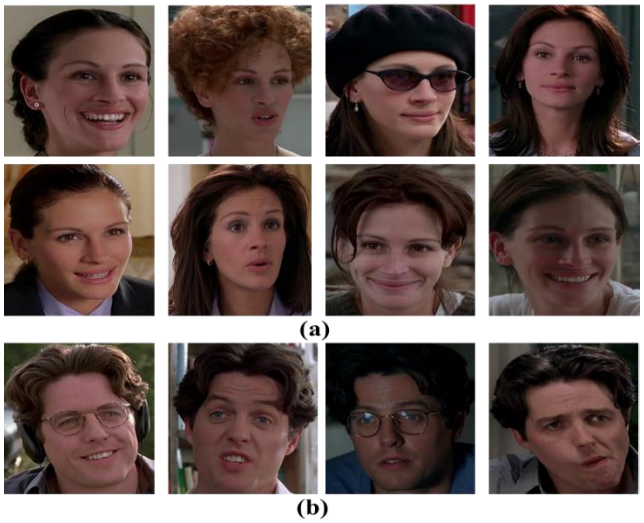


**FIGURE 3.** Sample images from clustered images of characters (a) Anna Scott and (b) William Thacker from the movie *Notting Hill* with various appearances.

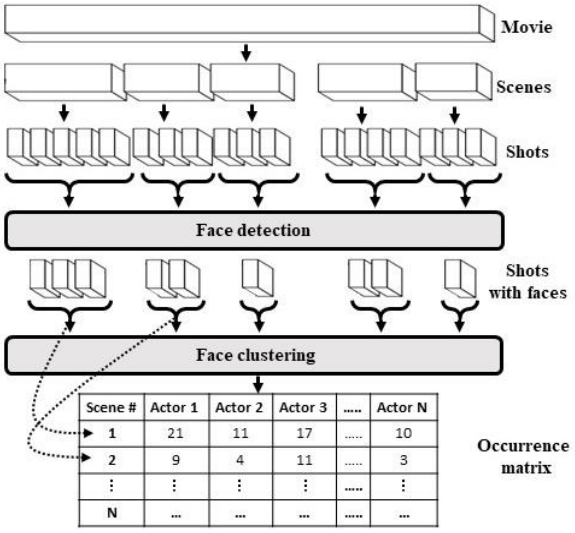


**FIGURE 4.** Flow chart for occurrence matrix generation.

### C. STARRING CHARACTER DETERMINATION

Starring characters have a greater impact on viewers than other characters: therefore, their appearance on the screen is more important than other characters. Based on this observation, we give preference to a cluster with maximum points as a starring character, and the clusters with the second and third most points are considered the next important characters, respectively. Figure 4 represents the flow to obtain the

**TABLE 2.** Information of the evaluation data.

| Movie ID | Movie title | Year | Genre | Resolution | Length | No. of starring characters | % of scenes with faces |
|---|---|---|---|---|---|---|---|
| M1 | **You've Got Mail** | 1998 | Comedy/Romance | 720p | 119 min | 2 | 100% (58/58) |
| M2 | **The Devil Wears Prada** | 2006 | Comedy/Drama | 1080p | 109 min | 1 | 100% (69/69) |
| M3 | **Salt** | 2010 | Action/Crime | 720p | 100 min | 1 | 95% (62/65) |
| M4 | **Notting Hill** | 1999 | Comedy/Romance | 720p | 124 min | 2 | 100% (42/42) |
| M5 | **Gladiator** | 2000 | Action/Adventure | 720p | 155 min | 1 | 97% (58/60) |
| M6 | **The Lake House** | 2006 | Fantasy/Drama | 720p | 105 min | 2 | 98% (75/76) |
| M7 | **My Blueberry Nights** | 2007 | Drama/Romance | 1080p | 90 min | 1 | 89% (51/57) |

**TABLE 3.** Face detection accuracy.

| Scene no. | Length | Face shots | Detected shots | Accuracy |
|---|---|---|---|---|
| S1 | 13 min 21sec | 66 | 62 | 93% |
| S2 | 8 min 19 sec | 41 | 39 | 95% |
| S3 | 11 min 49 sec | 51 | 49 | 96% |

occurrence matrix which we use to determine the starring character.

A single row in the occurrence matrix represents the appearance of all characters in the scene N, while each column represents the appearance of a particular character in all the scenes. The overall appearance of a character over the scenes can be analysed for starring character identification for the movie. Alongside this, our method can be applied to a single scene of a movie to detect the starring character for the specific scene.

## III. RESULTS AND DISCUSSION

We use seven movies of different genres to evaluate the proposed method. A detailed description of the testing data is given in Table 2. For the ground truth, we referred to one of the largest online movie databases, IMDb [3], from where all the cast details about each movie were obtained. We conducted three sets of experiments to evaluate the face detection and clustering accuracy, and finally compared the starring characters detected by our system with the generated ground truth. For easy understanding, we explained the results for the movie *Notting Hill* more broadly, while the rest of results are shown as average graphs and figures.

### A. PERFORMANCE OF FACE DETECTION

The number of detected faces in all frames of a typical movie is up to 100,000, derived from a few hundred shots [27]. Hence we conducted an evaluation experiment on three scenes, S1, S2 and S3, from the movie *Notting Hill*, in which each shot contains a face. Table 3 shows the detail of the selected scenes where 'face shots' and 'detected shots' represent the total number of shots with faces and the number of shots in which a face is detected using the multi-task cascaded network used in [38] with additional constraints. Due to additional constraints, this evaluation is shots-based because we are not detecting every face in the frame. It can be observed from Table 4 that our strategy for detecting faces improves the clustering results.

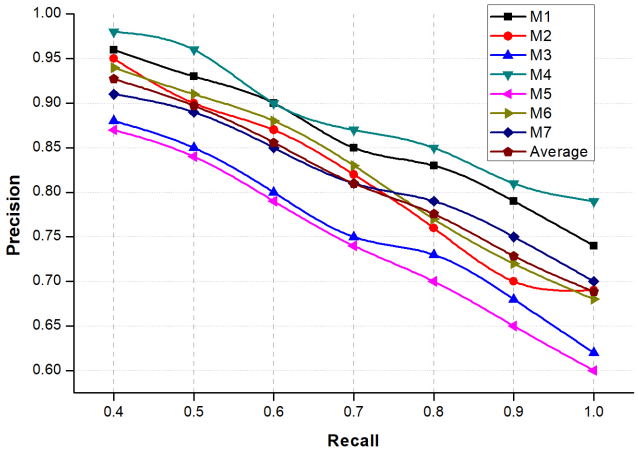


**FIGURE 5.** Precision recall curves for all testing movies (M1–M7) and average precision curve of all movies at different recall levels.

### B. PERFORMANCE OF CLUSTERING

Due to the dynamic nature of movies, we cannot predict the number of characters in a single scene, indicating that the number of clusters is not known in advance. Hence, in our case, adaptive clustering best fits the scenario. To evaluate the performance of clustering, consider $C_1$, $C_2 \ldots C_N$ as clusters for each character, $A_1$, $A_2 \ldots .A_N$, respectively. The precision (P) and recall (R) for each cluster is calculated using Equation (1) and Equation (2).

$$Precison\ (P) = \frac{\#\ of\ faces\ correctly\ clustered\ for\ A_{(K)}}{total\ \#\ of\ \ clustered\ faces\ in\ C_{(K)}} \quad (1)$$

$$Recall\ (R) = \frac{total\ \#\ of\ \ clustered\ faces\ in\ C_{(K)}}{total\ \#\ of\ faces\ in\ A_{(K)}} \quad (2)$$

Here, the term 'precision' means the proportion of correctly clustered faces, and 'recall' is the proportion of total clustered faces [48]. We present detailed clustering results of four scenes from the movie *Notting Hill* in Table 4. Precision at recall 0.8 for individual clusters of each character in a scene where faces are detected using DP2MFD [49], MCCN [38] and using our strategy are given in Table 4. Figure 5 illustrates the average precision of all the testing movies (M1–M7) at different recall. It is observed from our experiments that the clustering accuracy of action movies (M3 and M5) is lower than others due to the fast movement of faces and illumination. Moreover, the main character in M3 is playing the role

**TABLE 4.** Clustering accuracy based on different face detection methods for the movie Notting Hill.

| Scene No. | Length | Detected faces | | | No. of clusters | Clustering results using [49] | | Clustering results using [38] | | Our strategy | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | [49] | [38] | Our strategy | | Individual precision | Average precision | Individual precision | Average precision | Individual precision | Average precision |
| Scene 3 | 6 min 30 sec | 572 | 568 | 404 | Cluster 1 | 0.87 | 0.87 | 0.90 | 0.89 | 0.94 | 0.93 |
| | | | | | Cluster 2 | 0.90 | | 0.92 | | 0.98 | |
| | | | | | Cluster 3 | 0.84 | | 0.84 | | 0.93 | |
| | | | | | Cluster 4 | 0.89 | | 0.91 | | 0.90 | |
| Scene 6 | 2 min 50 sec | 134 | 119 | 58 | Cluster 1 | 0.81 | 0.85 | 0.76 | 0.88 | 0.84 | 0.92 |
| | | | | | Cluster 2 | 0.89 | | 1 | | 1 | |
| Scene 7 | 3 min 48 sec | 211 | 203 | 111 | Cluster 1 | 0.87 | 0.84 | 0.91 | 0.85 | 0.93 | 0.90 |
| | | | | | Cluster 2 | 0.95 | | 0.96 | | 1 | |
| | | | | | Cluster 3 | 0.72 | | 0.70 | | 0.77 | |
| Scene 9 | 1 min 26 sec | 159 | 142 | 32 | Cluster 1 | 0.87 | 0.88 | 0.90 | 0.91 | 1 | 1 |
| | | | | | Cluster 2 | 0.89 | | 0.92 | | 1 | |

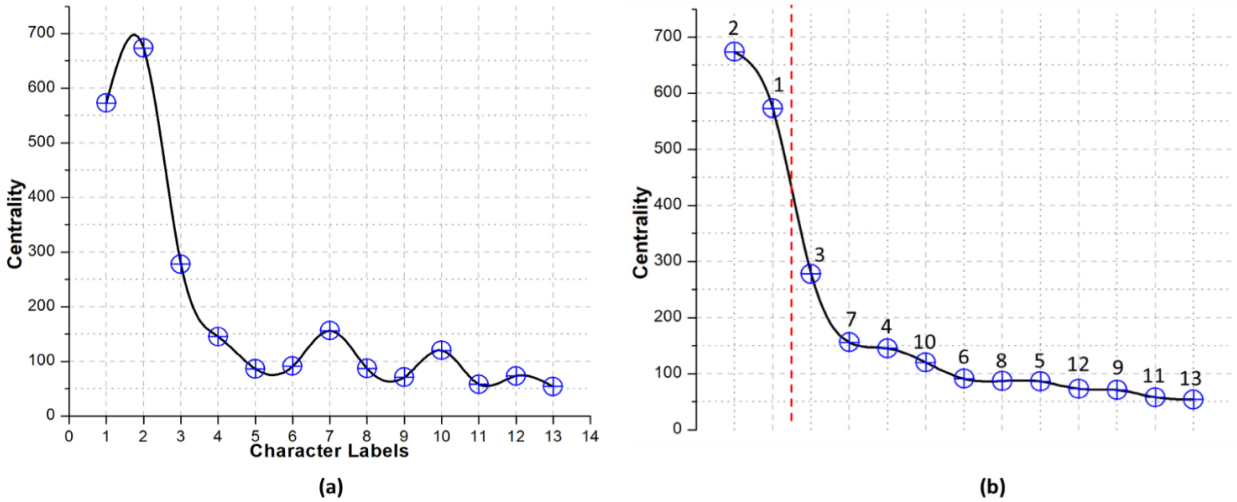


**FIGURE 6.** Character centrality values for the movie *Notting Hill*. (a) Centrality values in detected character order; (b) Descending order of centrality.

of a spy, where she changes her face using different masks throughout the movie, degrading the clustering accuracy.

### C. EVALUATION OF STARRING CHARACTERS' DETERMINATION

In this section we evaluate the determined starring characters for movies given in Table 2. Generally, main characters have greater influence than other characters, so their appearance on the screen is also longer. Based on this, the overall appearance of a character in the movie is determined by adding all the shots in which he/she appears. Such an approach for determining starring character is referred to as centrality value in terms of clustering [50].

We calculate the centrality value $C_{ent}$ of each character $A_i$ in a movie using Equation (3):

$$C_{ent}(A_i) = \sum_j S_j = \begin{cases} 1, & if\ A_i \in S_j \\ 0, & otherwise \end{cases} \quad (3)$$

where $S_j$ is *jth* shot which can be obtained from the occurrence matrix. There may be more than one starring character in a movie: hence, selecting the top appearing character or selecting more than one character based on a threshold is not a suitable approach due to the diverse nature of movies. Therefore, the calculated centrality values for each character are first sorted in descending order, and the difference between every two adjacent centrality values then calculated. The maximum difference value obtained is the boundary between the starring and other characters. We take the movie *Notting Hill* as an example for detailed explanation. Figure 6(a) represents a graph for the movie *Notting Hill* showing the appearance of different characters in the overall shots. In Figure 6(b), it can be observed that characters 1 and 2 have a significant gap from the rest of the characters. Hence, these characters are determined as the starring characters.

To evaluate the overall accuracy [50] of the lead characters for a movie, let's say that $SC = \{lc_1, lc_2, .., lc_l\}$ is a set of starring characters in ground truth and $SC' = \{uc_1, uc_2, .., uc_h\}$ is the set of starring characters determined by the

**TABLE 5. Overall accuracy for starring character determination.**

| ID | Ground truth | Star characters | Overall accuracy |
|---|---|---|---|
| **M1** | 1, 2 | 1, 2 | 100% |
| **M2** | 1 | 1, 2 | 83% |
| **M3** | 1 | 1 | 100% |
| **M4** | 1, 2 | 1, 2 | 100% |
| **M5** | 1 | 1 | 100% |
| **M6** | 1, 2 | 1, 2 | 100% |
| **M7** | 1 | 1, 3 | 87% |

proposed system. Here $l$ and $h$ are the set numbers for *SC* and *SC'*. So, the overall accuracy is calculated using Equation (4).

$$A_{overall} = \frac{\sum_{i=1}^{h} \delta_i C(uc_i)}{\sum_{i=1}^{l} C(lc_i)} \begin{cases} \delta_i = 1 & if\ uc_i \in SC \\ \delta_i = 0 & otherwise \end{cases} \tag{4}$$

Herein, $C(.)$ is the centrality value for each character and $\delta_i$ indicates whether or not $uc_i$ is included in the *SC*. Table 5 represents the performance of starring character determination. The second and third columns of Table 5 show ground truth obtained from IMDb and starring characters identified by the proposed system, respectively. The last column represents the accuracy for each movie using Equation (4). It is clear from Table 5 that the performance of the method is excellent and satisfactory. Except for M2 and M7, all the starring characters are accurately confirmed. We assign the prospective result that starring characters appear in almost all the scenes of the movie.

## IV. CONCLUSION

Starring character identification is a hot area of research in which several hand-engineered and learned representation-based methods are presented. In this article, we presented DeepStar, a threefold scheme for starring characters' identification in movies. Firstly, the pre-processing module of the proposed scheme segments the video into shots and detects faces. Next, it applies some constraints to obtain very clear faces which are input to the next step. The pre-processing step ensures that faces are extracted from each shot of the movie where any character is present, thus preserving the accuracy of presence of each character from each shot. Next, the non-blurry faces are input to the feature extraction module, where face deep features are extracted for face clustering. Based on these features, faces are clustered into several groups depending on the number of characters. Finally, characters are identified through the occurrence matrix computed from the clusters. This work mainly focuses on the main characters' identification, which can be a baseline for movie analysis including summarization, indexing and retrieval.

## REFERENCES

[1] Z. Ma, X. Chang, Z. Xu, N. Sebe, and A. G. Hauptmann, "Joint attributes and event analysis for multimedia event detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2921–2930, Jul. 2018.

[2] K. Muhammad, T. Hussain, and S. W. Baik, "Efficient CNN based summarization of surveillance videos for resource-constrained devices," *Pattern Recognit. Lett.*, to be published, doi: 10.1016/j.patrec.2018.08.003.

[3] C. Needham. *Internet Movie Database (IMDb)*. Accessed: Sep. 3, 2018. [Online]. Available: https://www.imdb.com/pressroom/stats/

[4] F. Schwandt. Statista. Hamburg, Germany. Accessed: Sep. 3, 2018. [Online]. Available: https://www.statista.com/topics/964/film/

[5] Y.-F. Zhang, C. Xu, H. 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.

[6] C.-Y. Weng, W.-T. Chu, and J.-L. Wu, "RoleNet: Movie Analysis from the Perspective of Social Networks," *IEEE Trans. Multimedia*, vol. 11, no. 2, pp. 256–271, Feb. 2009.

[7] J. Zhang, X. Yao, G. Han, and Y. Gui, "A survey of recent technologies and challenges in big data utilizations," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2015, pp. 497–499.

[8] S. I. Satoh and T. Kanade, "Name-it: Association of face and name in video," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 368–373.

[9] O. Arandjelovic and R. Cipolla, "Automatic cast listing in feature-length films with anisotropic manifold space," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1513–1520.

[10] Y.-F. Zhang, C. Xu, J. Cheng, and H. Lu, "Naming faces in films using hypergraph matching," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Jul. 2009, pp. 278–281.

[11] J. Y. Choi, W. De Neve, and Y. M. Ro, "Towards an automatic face indexing system for actor-based video services in an IPTV environment," *IEEE Trans. Consum. Electron.*, vol. 56, no. 1, pp. 147–155, Feb. 2010.

[12] J. Tao and Y.-P. Tan, "Efficient clustering of face sequences with application to character-based movie browsing," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1708–1711.

[13] B.-C. Chen *et al.*, "Scalable face track retrieval in video archives using bag-of-faces sparse representation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 27, pp. 1595–1603, Jul. 2017.

[14] C. Wang, H. Yang, C. Bartz, and C. Meinel, "Image captioning with deep bidirectional LSTMs," in *Proc. ACM Multimedia Conf.*, 2016, pp. 988–997.

[15] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 580–588, Jun. 1999.

[16] C. Liang, C. Xu, J. Cheng, and H. Lu, "TVParser: An automatic TV video parsing method," in *Proc. CVPR*, Jun. 2011, pp. 3377–3384.

[17] J.-Y. Li, L.-W. Kang, C.-M. Tsai, and C.-W. Lin, "Learning-based movie summarization via role-community analysis and feature fusion," in *Proc. IEEE 17th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2015, pp. 1–6.

[18] J. Yang and A. G. Hauptmann, "Naming every individual in news video monologues," in *Proc. 12nd Annual ACM Int. Conf. Multimedia*, 2004, pp. 580–587.

[19] J. Yang, R. Yan, and A. G. Hauptmann, "Multiple instance learning for labeling faces in broadcasting news video," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, Nov. 2005, pp. 31–40.

[20] X. Chang and Y. Yang, "Semisupervised feature analysis by mining correlations among multiple tasks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2294–2305, Oct. 2017.

[21] D. Ramanan, S. Baker, and S. Kakade, "Leveraging archival video for building face datasets," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[22] M. Everingham and A. Zisserman, "Identifying individuals in video by combining 'generative' and discriminative head models," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1103–1110.

[23] M. Xu, X. Yuan, J. Shen, and S. Yan, "Cast2Face: Character identification in movie with actor-character correspondence," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 831–834.

[24] M.-C. Yeh and W.-P. Wu, "Clustering faces in movies using an automatically constructed social network," *IEEE Multimedia*, vol. 21, no. 2, pp. 22–31, Apr./Jun. 2014.

[25] G. Gao, C. H. Liu, M. Chen, S. Guo, and K. K. Leung, "Cloud-based actor identification with batch-orthogonal local-sensitive hashing and sparse representation," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1749–1761, Sep. 2016.

[26] V. Gandhi and R. Ronfard, "Detecting and naming actors in movies using generative appearance models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3706–3713.

[27] J. Sang and C. Xu, "Robust face-name graph matching for movie character identification," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 586–596, Jun. 2012.
[28] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar, "Movie/script: Alignment and parsing of video and text transcription," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 158–171.
[29] D. N. Woo and R. S. Aygun, "Unsupervised speaker identification for TV news," *IEEE Multimedia*, vol. 23, no. 4, pp. 50–58, Oct./Dec. 2016.
[30] Z. Liu and Y. Wang, "Major cast detection in video using both speaker and face information," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 89–101, Jan. 2007.
[31] Y. Zhang, Z. Tang, B. Wu, Q. Ji, and H. Lu, "A coupled hidden conditional random field model for simultaneous face clustering and naming in videos," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5780–5792, Dec. 2016.
[32] M. Wang, R. Hong, X.-T. Yuan, S. Yan, and T.-S. Chua, "Movie2Comics: Towards a lively video content presentation," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 858–870, Jun. 2012.
[33] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "$A^3$NCF: An adaptive aspect attention model for rating prediction," in *Proc. IJCAI*, 2018, pp. 3748–3754.
[34] T. Zhuo, Z. Cheng, P. Zhang, Y. Wong, and M. Kankanhalli. (2018). "Unsupervised online video object segmentation with motion property understanding." [Online]. Available: https://arxiv.org/abs/1810.03783
[35] X. Chang, F. Nie, Z. Ma, Y. Yang, and X. Zhou, "A convex formulation for spectral shrunk clustering," in *Proc. AAAI*, 2015, pp. 2532–2538.
[36] Z. Li, F. Nie, X. Chang, Z. Ma, and Y. Yang, "Balanced clustering via exclusive lasso: A pragmatic approach," in *Proc. AAAI*, 2018, pp. 3596–3603.
[37] N. J. Janwe and K. K. Bhoyar, "Video shot boundary detection based on JND color histogram," in *Proc. IEEE 2nd Int. Conf. Image Inf. Process. (ICIIP)*, Dec. 2013, pp. 476–480.
[38] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[39] R. Bansal, G. Raj, and T. Choudhury, "Blur image detection using Laplacian operator and open-CV," in *Proc. Int. Conf. Syst. Modeling Advancement Res. Trends (SMART)*, 2016, pp. 63–67.
[40] A. M. Badshah *et al.*, "Deep features-based speech emotion recognition for smart affective services," in *Proc. Multimedia Tools Appl.*, 2017, pp. 1–19.
[41] C. Wang, H. Yang, and C. Meinel, "Image captioning with deep bidirectional LSTMs and multi-task learning," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 2S, p. 40, 2018.
[42] I. Sobron, J. Del Ser, I. Eizmendi, and M. Velez, "A deep learning approach to device-free people counting from WiFi signals," in *Proc. Int. Symp. Intell. Distrib. Comput.*, 2018, pp. 275–286.
[43] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC*, 2015, p. 41.1–41.12.
[44] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2017.
[45] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," *Neurocomputing*, vol. 288, pp. 30–42, May 2017.
[46] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2018.2830099.
[47] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5880–5888.
[48] M. R. Everingham, J. Sivic, and A. Zisserman, "Hello! My name is... buffy'–automatic naming of characters in TV video," in *Proc. Brit. Mach. Conf.*, 2006, p. 92-1.
[49] R. Ranjan, V. M. Patel, and R. Chellappa. (2015). "A deep pyramid deformable part model for face detection." [Online]. Available: https://arxiv.org/abs/1508.04389
[50] J. He, Y. Xie, X. Luan, L. Zhang, and X. Zhang, "SRN: The movie character relationship analysis via social network," in *Proc. Int. Conf. Multimedia Modeling*, 2018, pp. 289–301.

**IJAZ UL HAQ** (S'18) received the B.S. degree in computer science from the Islamia College at Peshawar, Peshawar, Pakistan. He is currently pursuing the M.S. degree with the Intelligent Media Laboratory, Sejong University, South Korea. His research interests include video summarization, image and video analysis, image hashing, steganography, and deep learning for multimedia understanding.

**KHAN MUHAMMAD** (S'16–M'18) is currently a Postdoctoral Researcher at IM Lab, Digital Contents Research Institute, Sejong University, South Korea. His research interests include information security, video summarization, computer vision, and video surveillance. He has authored over 40 papers in peer-reviewed international journals such as IEEE TII and IEEE TSMC-Systems, and is a reviewer of over 30 SCI/SCIE journals including IEEE Communications Magazine, IEEE Network, IEEE Internet of Things Journal, TIP, TII, TCYB, and IEEE Access. He is a member of the ACM.

**AMIN ULLAH** (S'17) received the bachelor's degree in computer science from the Islamia College at Peshawar, Peshawar, Pakistan. He is currently pursuing the M.S. degree leading to the Ph.D. degree with the Intelligent Media Laboratory, Sejong University, South Korea. His research interests include human actions and activity recognition, sequence learning, image and video analysis, and deep learning for multimedia understanding.

**SUNG WOOK BAIK** (M'16) received the B.S. degree in computer science from Seoul National University, Seoul, South Korea, in 1987, the M.S. degree in computer science from Northern Illinois University, DeKalb, in 1992, and the Ph.D. degree in information technology engineering from George Mason University, Fairfax, VA, USA, in 1999. He was with Datamat Systems Research, Inc., as a Senior Scientist of the Intelligent Systems Group from 1997 to 2002. In 2002, he joined the Faculty of the College of Electronics and Information Engineering, Sejong University, where he is currently a Full Professor, the Chief of Sejong Industry-Academy Cooperation Foundation, and the Head of the Intelligent Media Laboratory. His research interests include computer vision, multimedia, pattern recognition, machine learning, data mining, virtual reality, and computer games. He served as a Professional Reviewer for several well-reputed journals, such as the *IEEE Communication Magazine*, the IEEE Sensors Journal, *Information Fusion*, *Information Sciences*, the IEEE TIP, MBEC, MTAP, SIVP, and JVCI.