Predicting Salient Face in Multiple-face Videos

Yufan Liu[†], Songyang Zhang[†‡], Mai Xu[†*], and Xuming He[‡]
[†]Beihang University, Beijing, China
[‡]ShanghaiTech University, Shanghai, China

Abstract

Although the recent success of convolutional neural network (CNN) advances state-of-the-art saliency prediction in static images, few work has addressed the problem of predicting attention in videos. On the other hand, we find that the attention of different subjects consistently focuses on a single face in each frame of videos involving multiple faces. Therefore, we propose in this paper a novel deep learning (DL) based method to predict salient face in multiple-face videos, which is capable of learning features and transition of salient faces across video frames. In particular, we first learn a CNN for each frame to locate salient face. Taking CNN features as input, we develop a multiple-stream long short-term memory (M-LSTM) network to predict the temporal transition of salient faces in video sequences. To evaluate our DL-based method, we build a new eye-tracking database of multiple-face videos. The experimental results show that our method outperforms the prior state-of-the-art methods in predicting visual attention on faces in multiple-face videos.

1. Introduction

Saliency prediction [1] models the deployment of attention on visual inputs in biological vision systems, and has potential application in many computer vision tasks, such as object detection [3] and event detection [36]. Particularly, detecting salient objects, such as faces, plays an important role in video analytics, human-computer interface design and event understanding. As a matter of fact, a mass of videos, including movie, interview and variety show, contain multiple faces.

Existing literature on saliency prediction typically focuses on finding salient face in static images [21]. However, few prior work has addressed the problem of predicting saliency in multiple-face videos. While the human subjects generally pay attention to only a single face [21], we find that attention of different subjects consistently transits from one face to another in videos, as shown in Figure 1. Our goal in this work is to capture both static and dynamic properties of the attention on faces in multiple-face videos.

Early work on image saliency prediction uses hand-craft features to predict visual attention for images [2, 10, 20, 26, 35, 42], based on understanding of the human visual system (HVS) [29]. The representative method on predicting image saliency is Itti's model [20], which combines center-surround features of color, intensity and orientation together. In contrast, recent methods [4,14,21,22,24,25,28,32,40, 41, 43] propose a learning-based strategy to predict saliency. For example, Judd *et al.* combined high-level features (e.g., face and text), middle-level features (e.g., gist) and low-level features together, via learning their corresponding weights with the support vector machine (SVM). To predict visual attention in face images, Xu *et al.* [41] proposed to precisely model saliency of face region, via learning the fixation distributions of face and facial features. Besides, Jiang *et al.* [21] explored several face-related features to predict saliency in a scene with multiple faces. Most recently, several deep learning (DL) methods [14, 24, 25, 28, 32] have been proposed to automatically learn features for saliency prediction, instead of relying on handcrafted features. For example, Huang *et al.* [14] proposed saliency in context (SALICON) method to learn features for image saliency prediction by incorporating convolutional neural network (CNN).

For video saliency prediction, earlier methods [6, 8, 12, 17–19] have investigated several dynamic features to model visual attention on videos, in light of the HVS. For example, the Itti's image model was extended in [17] for video saliency prediction, by integrating with two dynamic features: motion and flicker contrast. Later, several advanced video saliency prediction methods have been proposed, which exploits other dynamic features, such as Bayesian surprise in [18] and motion vector in [6]. Recently, learning-based video saliency prediction methods have also emerged [13, 31, 33, 37]. For example, Pang *et al.* [33] proposed a learning-based video saliency prediction method, which explores the top-down information of eye movement patterns, i.e., passive and active states [34],

*Corresponding author: Mai Xu (maixu@buaa.edu.cn).

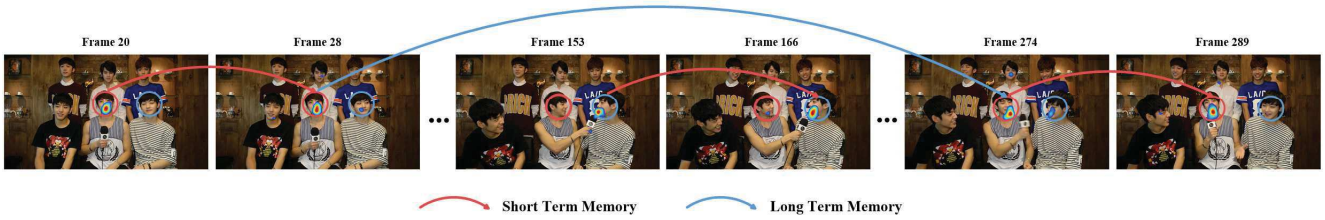


Figure 1. Example of visual attention (viewed by 39 subjects) on a multiple-face video sequence. Each image shows a selected frame with its attention heat map. This figure mainly demonstrates transition of salient faces and characters of long/short-term correlation between salient faces across frames. Note that the video is chosen from our database, to be discussed in Section 2.1.

to model attention on videos. Hua *et al.* [13] proposed to learn middle-level features, i.e., gists of a scene, as the cue in video saliency prediction. Rudoy *et al.* [37] proposed to predict saliency of a given frame according to its high- and low-level features, conditioned on the detected saliency of the previous reference frame. However, to our best knowledge, the existing video saliency prediction methods rely on the handcrafted features, despite CNN being applied to automatically learn features for image saliency prediction in the most recent works of [14, 24, 25, 28, 32]. More importantly, both long- and short- term correlation of salient faces across frames, which is critical in modeling attention transition across frames for multiple-face videos (see Figure 1), is not taken into account in these methods.

In this paper, we propose a DL-based method to predict salient face in multiple-face videos, which learns both image features and saliency transition for modeling attention on multiple faces across frames. To the best of our knowledge, our method is the first aiming at saliency prediction in multiple-face videos. Specifically, we first apply a CNN to automatically extract saliency-related features in each single frame. Built on the long short-term memory (LSTM) of recurrent neural network (RNN), we develop a multiple-stream LSTM (M-LSTM) network for predicting the dynamic transitions of salient faces alongside video frames, taking the extracted CNN features as the input. Finally, saliency maps of multiple-face videos can be generated upon transited face saliency.

To evaluate our method, we create a new eye tracking database of multiple-face videos that consists of two datasets. The first dataset includes fixations of 39 subjects on 65 multiple-face videos, used as a baseline for testing saliency prediction performance. The second dataset is composed of 100 multiple-face videos viewed by 36 subjects, which is utilized for training the saliency prediction model. We provide a detailed analysis on the collected data, which shows that typically only one face (among multiple faces) in a video frame receives attention of viewing subjects, and that attention shifts across frames consistently for different subjects. We test our method on the new database, with comparisons to several state-of-the-art approaches. Our experimental results demonstrate that our method achieves significant improvements on saliency prediction in multiple-face videos.

In summary, the main contributions of our work are three-fold: (1) We introduce an eye tracking database of multiple-face videos, for facilitating the studies on video saliency prediction. (2) We find significant consistency in subjects on viewing multiple-face videos, via analysis on our eye-tracking databases. (3) We propose a DL-based method to predict the salient face with transition across frames, which integrates a CNN and an LSTM-based RNN model.

2. Database establishment and analysis

2.1. Multiple-face Database

This section describes how we conduced the eye tracking experiment to establish our database on MUltiple-Face Videos with Eye Tracking fixations (MUFVET). To the best of our knowledge, our eye tracking database is the first one for multiple-face videos. Note that all videos in MUFVET are with either indoor or outdoor scenes, selected from Youtube and Youku, and they are all encoded by H.264 with duration varying from 10-20 seconds. Besides, MUFVET includes two datasets – MUFVET-I and MUFVET-II. These two datasets are comprised by two non-overlapping groups of videos, each of which is viewed by totally different subjects. In this paper, MUFVET-I is seen as the benchmark for test, while MUFVET-II is used for training. MUFVET is more reasonable than the existing eye tracking databases (e.g., SFU [9] and DIEM [30]), which only contain fixations of videos watched by same subjects. It is because both training and test utilize the fixations of same subjects are not rationale in existing saliency prediction works [1], despite videos being different.

MUFVET-I. Here, 65 multiple-face videos at diverse scenarios (see Table 1 and Figure 2) are included in MUFVET-I. Then, 39 subjects (26 males and 13 females, aging from 20 to 49), with either corrected or uncorrected normal eye-sight, participated in our eye tracking experiment to watch all 65 videos. Among them, two were experts working in the field of saliency prediction. Others did not have any experience on saliency prediction, and meanwhile they were naive to the purpose of our eye tracking experiment. The eye fixations of 39 subjects on viewing each video were recorded by a Tobii X2-60 eye tracker at 60Hz. For the eye tracker, a 23-inch LCD screen was used to display the test videos at their original resolutions. During the eye tracking experiment, all subjects were required to sit on a comfortable chair with the viewing distance being $\sim$60 cm from the LCD screen. Before viewing videos, each

Table 1. Video categories in MUFVET-I and MUFVET-II.

Category	TV play/movie	interview	video conference	TV show	music/talk show	group discussion	overall
Number of videos (I)	12	20	6	7	10	10	65
Number of videos (II)	21	13	5	35	18	8	100

Figure 2. One example for each category of videos in MUFVET-I and MUFVET-II. From left to right, the videos belong to TV play/movie, interview, video conference, TV show, music/talk show, and group discussion.

subject was required to perform a 9-point calibration for the eye tracker. Afterwards, the subjects were asked to free-view videos displayed at random order. In order to avoid eye fatigue, the 65 test videos were divided into 3 sessions, and there was a 5-minute rest after viewing each session. Moreover, a 10-second blank period with black screen was inserted between two successive videos for a short rest. Finally, 1,252,822 fixations of all 39 subjects on 65 videos were obtained.

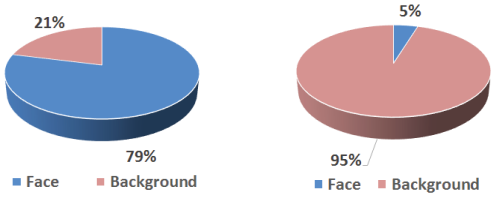
It is worth mentioning that our dataset includes the salient objects other than faces. Among 65 videos in MUFVET-I, for instance, 24 videos have salient objects other than faces, among which 3 videos have new objects appearing in the scenes. The ratio of frames containing salient objects other than faces is 37.6%. Besides, the average number of faces per frame is 3.66.

MUFVET-II. For this dataset, 100 multiple-face videos, which are totally different from MUFVET-I, were used for the eye-tracking experiment. For more details about these videos, refer to Table 1 and Figure 2. The overall experiment procedure for MUFVET-II is the same as that for MUFVET-I. The difference is that other 36 subjects (20 males and 16 females, aging from 20 to 55) were asked to view all 100 videos in MUFVET-II. Besides, the Tobii TX300 eye tracker was used to record fixations. During the experiment, 100 videos were equally divided into 2 sessions to avoid eye fatigue. At last, there were in total 1,737,826 fixations acquired from all 36 subjects in this dataset, which is used as the training set for learning attention model of multiple-face videos. For facilitating the future research, MUFVET is available online[1].

2.2. Data Analysis

In this section, we thoroughly analyze the collected eye tracking data of MUFVET, in order to further learn the visual attention model on multiple-face videos. According to the analysis, two findings are investigated as follows.

[1]https://github.com/yufanLIU/salient-face-in-MUVFET/tree/master/MUVFET.



(a) Fixation proportion (b) Pixel proportion

Figure 3. Proportions of fixations and pixels in face and background over all videos of MUFVET.

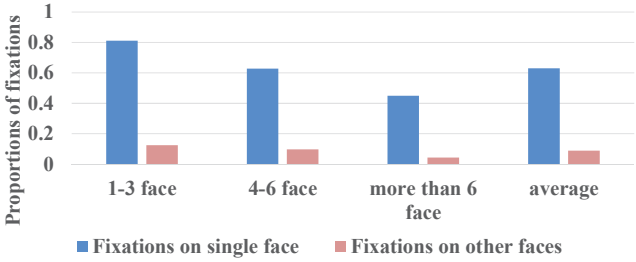


Figure 4. Proportions of fixations falling into one face and other faces.

Finding 1: In multiple-face videos, faces draw a significant amount of attention. At each video frame, attention of different subjects consistently focuses on one face among all faces.

Figure 3 shows the proportions of fixations and pixels belonging to face and background, in MUFVET. We can see from this figure that despite taking up only 5% pixels, faces receive 79% fixations. This verifies that faces attract almost all visual attention in multiple-face videos. Figure 4 further plots the proportions of fixations falling into one face to those into other faces. We can find from this figure that visual attention of different subjects is generally consistent in being attracted by one face among all faces. Beside, the subjective examples of Figure 1 also imply that faces, normally one face, draw most attention in a video.

Finding 2: Humans probably fixate on the face that is close to the video center, among all faces at a video frame.

The center-bias [1] is an obvious cue to predict human fixations on generic videos. It is also intuitive that people are likely to pay their attention on the face which is close to the video center. We hence investigate the correlation of attention on a face with Euclidean distance of this face to the video center. To quantify such correlation, we measure the averaged Spearman rank correlation coefficient [15] ($\rho = -0.22$) . This negative value of ρ indicates that humans probably fixate on the face that is close to the video center. The small value of ρ also implies that other features need to be learned for predicting salient face.

3. Proposed Method

In this section, we introduce our DL-based method for saliency prediction in multiple-face videos, which integrates CNN and LSTM in a uniform framework. The overall pipeline of our method is summarized in Figure 5. First, we detect faces in each frame and feed them into CNNs, detailed in Section 3.1. Second, we design a CNN to learn the features related to salient face at each static video frame, which is discussed in Section 3.2. Section 3.3 presents M-LSTM that learns to predict salient face, by taking into consideration saliency-related features of CNN and the temporal transition of salient faces across video frames. In the end, we adopt a post-processing step to generate saliency maps of multiple-face videos, discussed in Section 3.4.

3.1. Face Candidate Generation

Base on *Finding 1*, we first extract faces as our candidate regions for visual attention prediction in a multiple-face video. To this end, we leverage the latest face detection method, the funnel-structured cascade (FuSt) detection model [39], to extract candidate faces from an input video. Moreover, in order to handle challenging cases, such as partial occlusion and poor light conditions, we explore temporal information to improve face detection performance in multiple-face videos.

More specifically, we first match the faces across frames, by searching the face with nearest Euclidean distance. We then identify the nearest faces of two consecutive frames as the matched face of a same person, provided their distance is less than a threshold:

$$th_E = \gamma \times \sqrt{w^2 + h^2}, \tag{1}$$

where w and h are width and height of the detected face. Otherwise, we regard them as non-matching faces, belonging to different persons. In (1), γ is a parameter to control the sensitivity of face matching, and it is simply set to 0.5 in this paper. On one hand, a smooth filter is leveraged to improve precision rate, via eliminating some false alarms of wrongly detected faces. On the other hand, we apply a linear interpolation to extend face detections to neighboring frames within a sliding window, such that the missing faces can be recovered. In this paper, the length of sliding window is empirically chosen to be 17, to achieve sufficiently high recall rate on the face detection results.

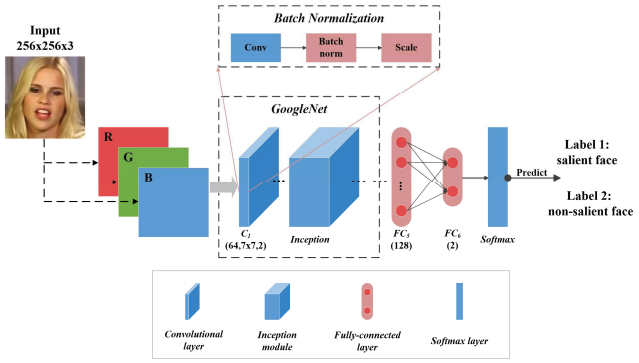


Figure 6. Architecture of our CNN for the task of predicting salient face.

3.2. CNN for Feature Extraction

We now design a CNN to automatically learn features from the detected faces, for the task of predicting whether the detected face is salient. The detected face regions are resized to be 256×256 before being sent to the CNN. Our CNN is based on the *GoogleNet* [38], with an additional batch normalization layer [16] after each convolution layer to avoid over-fitting. We also use the pre-trained *GoogleNet* and then fine-tune the network using MUFVET-II. Figure 6 shows the architecture of our CNN. After the convolutional feature extraction in *GoogleNet*, we use two fully connected (FC) layers, with *softmax* activation function, to decide whether the face is salient or not. The first FC layer has 128 units, whose outputs are used as the features for predicting the salient face. The second FC layer has 2 units, indicating the salient or non-salient face.

For training CNN, we automatically label each detected face to be salient or non-salient, according to the fixations falling into the face region. Our *Finding 1* indicates that the salient face in each video frame averagely draw more than 60% fixations of all faces. Hence, the faces that take up above 60% fixations are annotated as salient faces, and other faces are seen as non-salient ones. We then train our CNN by the backpropagation (BP) algorithm using MUFVET-II of our eye tracking database as the training data. Given the trained CNN, 128-dimension features of the first FC layer can be extracted from each detected face, which are fed into our recurrent network as input.

3.3. M-LSTM for Salient Face Prediction

The CNN defined above mainly extract spatial information of each face at a single frame. To model temporal dynamics of attention transition in videos, we now develop a novel M-LSTM to predict salient face in the video setting.

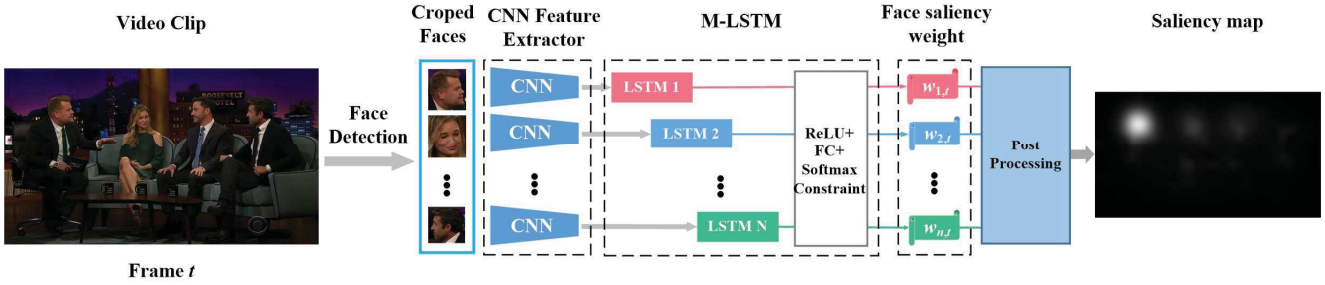


Figure 5. Overview pipeline for our DL-based method.

We formulate the multiple-face saliency prediction as a regression problem, and build an M-LSTM network to generate the continuous saliency weights of multiple faces. This differentiates our approach from the conventional LSTM [11] for classification. Formally, we aim to predict saliency weight of each face defined by $w_{n,t}$, which is the ground truth (GT) attention proportion of the n-th face to all faces in frame t. For such prediction, M-LSTM network generates an estimated saliency weight $\hat{w}_{n,t}$, which can be further regarded as the optimization formulation:

$$\begin{aligned} &\min \textstyle\sum_{t=1}^{T}\sum_{n=1}^{N} ||\hat{w}_{n,t} - w_{n,t}||_2^2 \\ &s.t. \textstyle\sum_{n=1}^{N} \hat{w}_{n,t} = \sum_{n=1}^{N} w_{n,t} = 1, t = 1, 2, ..., T, \end{aligned} \quad (2)$$

Our M-LSTM takes the CNN features $\{\mathbf{f}_{n,t}\}_{n=1,t=1}^{N,T}$ as input, where $\mathbf{f}_{n,t}$ stand for feature vector of the n-th face at frame t. We assume an upper limit of N faces per video. When fewer faces ($< N$) are detected in a video frame, the corresponding input to M-LSTM is zero vectors. Note that once the face of a person disappears after a few frames in a video sequence, the corresponding feature vector $\mathbf{f}_{n,t}$ is set to zero vector. On the other hand, if a new face appears after a few frames of a video sequence for one more person, its extracted input features $\mathbf{f}_{n,t}$ replace the zero vector. Given $\mathbf{f}_{n,t}$, a single LSTM chunk is applied to obtain hidden variable vector $\mathbf{h}_{n,t}$, as follows,

$$\mathbf{h}_{n,t} = LSTM(\mathbf{f}_{n,t}, \mathbf{h}_{n,t-1}), \quad (3)$$

where $LSTM(\cdot)$ represents an LSTM chunk. For the LSTM chunk, we use the standard LSTM which includes input, forget and output gates. It is worth mentioning that the LSTM chunk is capable of learning long/short-term dependency of salient face transition as well as overcoming the problem of vanishing gradient.

The hidden feature $\mathbf{h}_{n,t}$ is then passed through a FC layer followed by Rectified Linear Units (ReLU) as follows,

$$s_{n,t} = \max(0, \mathbf{V} \cdot \mathbf{h}_{n,t}), \quad (4)$$

where $\mathbf{V}$ is the parameter matrix for the FC layer. To capture the correlation between faces in one video frame, we build a second FC layer that takes $\{s_{n,t}\}_{n=1}^{N}$ of different faces at the t-th frame as input, and then generates N outputs. These outputs are then passed through a *softmax* layer to produce the final saliency weight predictions:

$$\hat{w}_{n,t} = \frac{\exp\{\theta_n \cdot \sum_{n=1}^{N}(U_{n,t} \cdot s_{n,t} + b_{n,t})\}}{\sum_{n=1}^{N} \exp\{\theta_n \cdot \sum_{n=1}^{N}(U_{n,t} \cdot s_{n,t} + b_{n,t})\}}, \quad (5)$$

where $U_{n,t}$ and $b_{n,t}$ are parameters of the FC layer, while θ_n is parameter of *softmax* layer.

Finally, $\hat{w}_{n,t}$ can be obtained by our M-LSTM denoted as $ML(\mathbb{P}, \mathbf{f}_{n,t})$, where $\mathbb{P}$ is the parameter set of M-LSTM to be learned. For $\mathbb{P}$, beyond parameter sharing across time in one conventional LSTM , our multiple LSTMs in one frame also share parameters for different faces. It is because the saliency changing mode in different faces is similar. As such, parameters from different LSTMs are updated at the same time. To learn all parameters $\mathbb{P}$, the loss function of our M-LSTM derived from (2) is

$$Loss = \sum_{t=1}^{T}\sum_{n=1}^{N} ||ML(\mathbb{P}, \mathbf{f}_{n,t}) - w_{n,t}||_2^2. \quad (6)$$

When training our M-LSTM with loss function (6), back propagation through time (BPTT) is utilized to learn parameters $\mathbb{P}$ with adaptive moment estimation (Adam) gradient descent optimization algorithm [23]. After training M-LSTM, $\hat{w}_{n,t}$ can be achieved for predicting salient face.

3.4. Postprocessing

After obtaining saliency weight of each face from M-LSTM, postprocessing is required to generate final saliency map. More specifically, we first make use of predicted saliency weights $\{\hat{w}_{n,t}\}_{n=1}^{N}$ to generate conspicuity map[2] of face channel, denoted as $\mathbf{M}_t^F$. It can be computed by

$$\mathbf{M}_t^F = \sum_{n=1}^{N} \hat{w}_{n,t} \cdot c_{n,t} \cdot \mathbf{M}_t^{F_n}, \quad (7)$$

where $\mathbf{M}_t^{F_n}$ denotes the conspicuity of the n-th face, and $c_{n,t}$ is the center-bias weight of each face. In our method, $\mathbf{M}_t^{F_n}$ is calculated by the latest work [41], which models the conspicuity map of a face with the Gaussian mixture model (GMM). In addition, *Finding 2* has revealed that visual attention is also correlated with the center-bias feature

[2]Note that saliency produced by the channel of single feature is defined as the conspicuity map, in order to make difference from the saliency map which is generated by all channels.

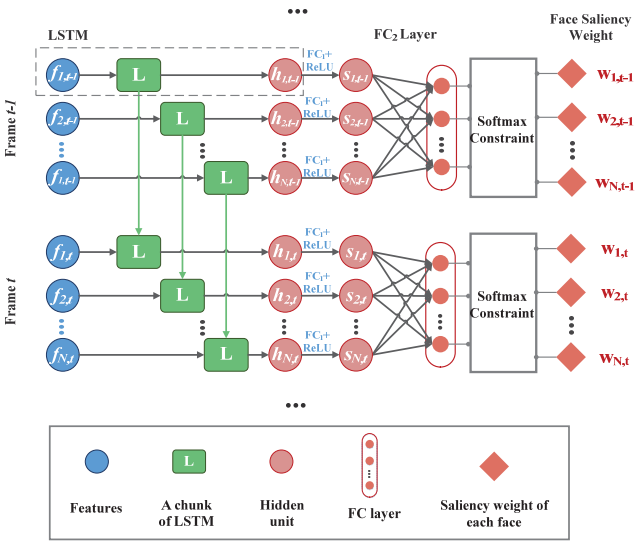


Figure 7. Structure of M-LSTM.

of faces in multiple-face videos. Therefore, we apply the following way for taking into account the face center-bias by weighting Gaussian model $c_{n,t}$ in (7). Assuming that $d_{n,t}$ is the Euclidean distance of the n-th face to the video center at the t-th video frame, $c_{n,t}$ of (7) can be calculated by the following Gaussian model:

$$c_{n,t} = \exp\left(-\frac{(d_{n,t} - \min_n d_{n,t})^2}{\sigma^2}\right). \quad (8)$$

In (8), σ is the standard deviation of the Gaussian model, which encodes the impact of center-bias on face saliency. Obviously, we have $c_{n,t} = 1$ for the face that is closest to the video center, and other faces have smaller $c_{n,t}$. Note that Gaussian center-bias weights of (8) are only imposed on conspicuity of each face in our method, rather than all pixels of the conventional center-bias in [5].

In order to consider background in saliency prediction, our method combines face consipicuity map $\mathbf{M}_t^F$ with consipicuity maps of three saliency-related feature channels of GBVS [10] (i.e., $\mathbf{M}_t^I$ for intensity, $\mathbf{M}_t^C$ for color and $\mathbf{M}_t^O$ for orientation). Let $\mathbf{S}_t$ be the final saliency map of the t-th video frame. It can be computed by the linear combination:

$$\mathbf{S}_t = \beta_1 \cdot \mathbf{M}_t^F + \beta_2 \cdot \mathbf{M}_t^I + \beta_3 \cdot \mathbf{M}_t^C + \beta_4 \cdot \mathbf{M}_t^O, \quad (9)$$

where $\{\beta_k\}_{k=1}^4$ are the channel weights of the k-th conspicuity map.

Next, we can compute (9) to predict saliency maps of multiple-face videos, once the values of $\{\beta_k\}_{k=1}^4$ are known. In fact, channel weights of β_k can be learnt from training data via solving the following optimization formulation:

$$\underset{\{\beta_k\}_{k=1}^4}{\operatorname{argmin}} \sum_{l=1}^{L} || \sum_{k=1}^{4} \beta_k \mathbf{M}_l^k - \mathbf{S}_l^* ||_2, s.t. \sum_{k=1}^{4} \beta_k = 1, \beta_k > 0, \quad (10)$$

where $\{\mathbf{M}_l^k\}_{l=1}^L$ and $\{\mathbf{S}_l^*\}_{l=1}^L$ are the conspicuity maps and GT fixation maps, for all L training video frames. In this paper, we apply the disciplined convex programming (CVX) [7] to solve the above optimization formulation. Finally, saliency map $\mathbf{S}_t$ of each multiple-face video frame can be yielded via postprocessing on the prediction of M-LSTM.

4. Experiment

4.1. Settings

In our experiments, we tested all 65 videos from MUFVET-I (mentioned in Section 2.1). In this paper, the saliency prediction results are reported by averaging over those 65 videos. For training set, all 100 videos from MUFVET-II are selected and segmented into 3,443 4-second-clips. Note that overlap is applied in the clip segmentation for the purpose of data augmentation. For tuning hyperparameters, 5-fold cross validation is implemented on the training set. As a result, 32-dimension is applied for all hidden states $\{\mathbf{h}_{n,t}\}_{n=1,t=1}^{N,T}$ of LSTM. Besides, the batch size is set to be 128. Learning rate is 0.0001, and it is reduced by a factor of 0.01 every 500 iterations with Adam [23].

For postprocessing, a 2-dimension Gaussian filter, with the cut-off frequency being 6 dB, is applied to smooth the fixations of face regions in the training frames. Then, $\{\mathbf{S}_l^*\}_{l=1}^L$ can be obtained for calculating the weight of each feature channel by (10). Moveover, σ of (8) is set to $10^{-0.2}$ for imposing center-bias on saliency of each face, in order to make saliency prediction appropriate. The impact of different σ on saliency prediction results is to be discussed in Section 4.3.

4.2. Evaluation on saliency prediction

In this section, we compare our method with 8 conventional saliency prediction methods[3], including Xu *et al.* [41], Salicon [14], Jiang *et al.* [21], GBVS [10], Rudoy *et al.* [37], PQFT [8], Surprise [18] and OBDL [12]. Among them, [37], [8], [18] and [12] are the latest video saliency prediction methods. Besides, [41], [14], [21] and [10] are recent image saliency prediction methods. To be more specific, [41] and [21] work on saliency prediction of single-face and multiple-face images, respectively. We compare our method to these two top-down methods, as there is no multiple-face saliency prediction method for videos. Note that we use our multiple-face detection technique of Section 3.1 to detect faces for [41], as its face detection only handles the single-face scenario. On the contrary, [10] is a bottom-up method, which provides background saliency for our method. Therefore, [10] is also included in our comparison. In addition, [14] is another latest DL-based method

[3]In our experiments, we run the codes provided by the authors with default parameters, to obtain saliency prediction results.

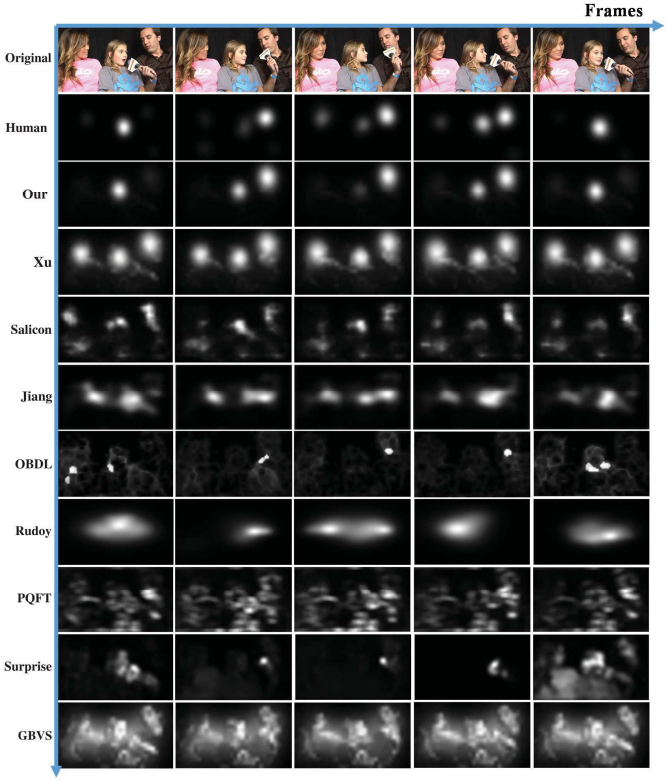


Figure 8. Saliency maps for different frames of a video sequence selected from MUFVET-I. These maps are generated by GT human fixations, our method, Xu *et al.* [41], Salicon [14], Jiang *et al.* [21], OBDL [12], Rudoy *et al.* [37], PQFT [8], Surprise [18] and GBVS [10].

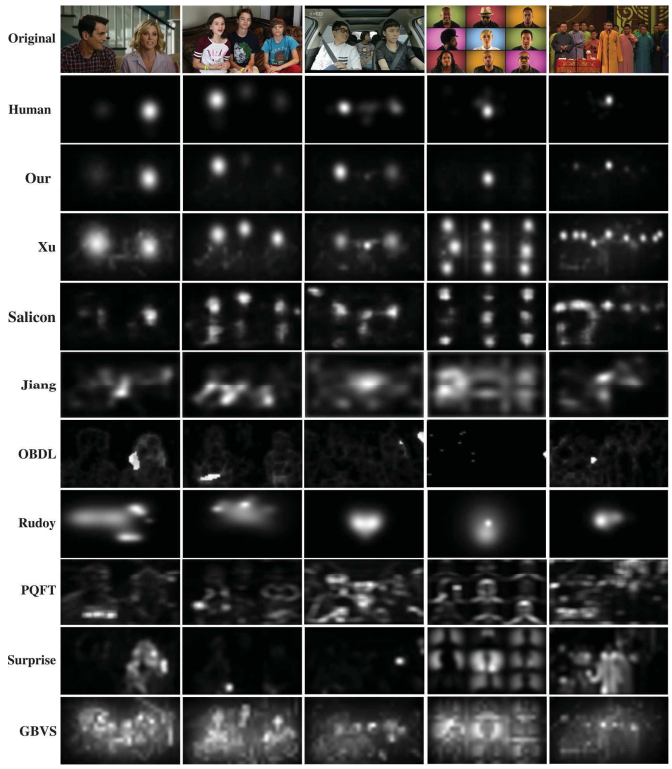


Figure 9. Saliency maps for several frames selected from different video sequences of MUFVET-I. These maps are generated by GT human fixations, our method, Xu *et al.* [41], Salicon [14], Jiang *et al.* [21], OBDL [12], Rudoy *et al.* [37], PQFT [8], Surprise [18] and GBVS [10].

for saliency detection, which is also compared with our DL-based method.

The most recent work of [27] has reported that normalized scanpath saliency (NSS) and correlation coefficient (CC) perform the best among all metrics in evaluating saliency prediction accuracy[4]. Thereby, we compare our method with other 8 methods in terms of NSS and CC. Table 2 reports the comparison results of saliency prediction, averaged over all 65 test videos of MUFVET-I. We can see from this table that our method is much better than all other methods in predicting saliency of multiple-face videos. Specifically, our method has 0.98 NSS and 0.13 CC improvement over [41], the performance of which ranks second. Such an improvement is mainly due to the following reason: Saliency of all faces is with equal importance in [41], while the consideration of temporal transition enables our method to accurately predict salient face across frames. Moreover, it is worth pointing out that both our method and [41] are superior to [21] which imposes unequal importance on different faces in an image. The main reason is that the utilization of only static features in [21] may predict wrong salient face in a video. On the other hand, long short-term temporal transition of our method is really effective in finding the salient face in multiple-face videos.

Next, we move to the comparison of subjective results. We show in Figure 8 the saliency maps of several frames in a video, generated by our and other 8 methods. From this figure, one may observe that our method is capable of finding the salient face. As a result, the saliency maps of our method are more accurate than those of other methods. For example, we can see from Figure 8 that the salient face is changed from the girl to the man and then back to the girl, which is extremely consistent with our prediction. On the contrary, [41] finds all three faces as salient ones, and [21] misses the salient face of the speaking man. In addition, Figure 9 provides the saliency maps of the frames selected from 5 videos. Again, this figure verifies that our method is able to precisely locate salient face by considering temporal saliency transition in M-LSTM.

4.3. Performance analysis of saliency prediction

Since our M-LSTM presented in Section 3.3 aims at predicting saliency weights of faces across video frames, it is worth evaluating the prediction error of M-LSTM. To this end, Figure 10 plots saliency weights of faces by CNN, M-LSTM and GT, for the video sequence of Figure 8. In this figure, the curves of CNN refer to the output of CNN (either 0 or 1), and the curves of M-LSTM are obtained upon

[4] [27] has also shown that area under ROC (AUC) is the worst metric in measuring accuracy of saliency prediction.

Table 2. Accuracy of saliency prediction by our and other 8 methods, averaged over all test videos of MUFVET-I.

	Our	GT	Xu *et al.* [41]	Salicon [14]	Jiang *et al.* [21]	GBVS [10]	Rudoy *et al.* [37]	PQFT [8]	Surprise [18]	OBDL [12]
NSS	**4.12**	4.21	3.14	2.96	0.97	1.23	1.42	0.88	0.88	1.62
CC	**0.74**	0.77	0.61	0.52	0.29	0.33	0.36	0.22	0.21	0.30

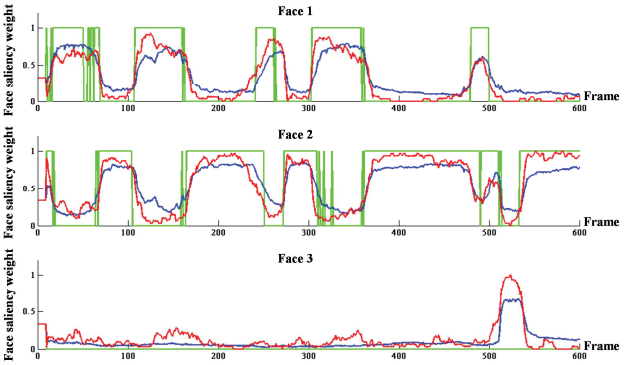
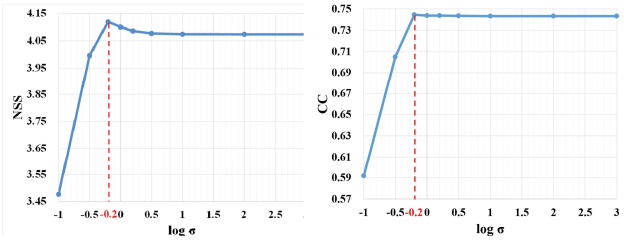


Figure 10. Saliency weights of faces along with processed frames for the video in Figure 10, predicted by our CNN (green line), M-LSTM (blue line) and GT (red line). Note that the GT in this curve is plotted with proportion of human fixations in each face to those in all faces. In this figure, the mean squared error (MSE) between M-LSTM and GT averaged over 3 faces is 0.0081.

the predicted face saliency weights output by M-LSTM. Besides, the curves for GT are the target output of M-LSTM. We can see from Figure 10 that the predicted face saliency weights of M-LSTM approach to the target, with significant improvement and smooth over the curves of CNN. More importantly, similar results can be found for other videos in our database. Here, we calculate quantify mean squared error (MSE) of face saliency weights between M-LSTM and GT, averaged over all faces in MUVFET-I. The averaged MSE is 0.0081, the same as the result of the video sequence in Figure 10. This also implies the small gap of M-LSTM in predicting saliency weights of faces.

Next, it is interesting to see how the gap between our predicted and GT face saliency weights influences saliency prediction performance. To this end, we use GT face saliency weights $\{w_{n,t}\}_{n=1,t=1}^{N,T}$ as the input to (7) for generalizing the final saliency maps of multiple-face videos. The averaged results are reported in the second column of Table 2. It can be found that saliency prediction performance of using estimated (M-LSTM) and target (GT) saliency weights of faces is close, implying that our method is approaching to the "upper bound" performance.

At last, it is necessary to investigate the effectiveness of face center-bias introduced in our method. To this end, standard deviation σ in (8) is traversed, imposing different impact of face center-bias on saliency prediction. Figure 11 plots the NSS and CC results at different σ, averaged over all videos. It is obvious that the best performance is achieved once $\sigma = 10^{-0.2}$, and thus σ was set to $10^{-0.2}$ in our above experiments. Note that center-bias is not the most important factor influencing performance improvement of our approach. We test the baseline that places all the weight



(a) NSS (b) CC

Figure 11. Saliency prediction performance versus different center-bias parameter σ of (8).

to the face closest to the center, and the average NSS of this baseline is 2.57, which is much lower than 4.12 of our approach. In addition, we conduct an experiment for the baseline relying on [41] with the center prior. The results show an improvement of 0.03 in NSS and 0.007 in CC over [41], which is still much inferior to our method.

5. Conclusion

Interestingly, we found that when viewing multiple-face videos, humans are consistently attracted by one face in each single frame. Such a finding was verified by the statistical analysis on the eye tracking database of MUFVET established in this paper, in which MUFVET-I is set for test and MUFVET-II is for training. To predict the salient face in multiple-face videos, we proposed in this paper a DL-based method, in which both CNN and RNN are combined in a framework and then trained over MUFVET-II. Specifically, CNN, fined-tuned on Google Net, was adopted in our DL-based method, for automatically learning the features relevant to locating the salient face. After observing CNN features in each video frame, M-LSTM, as a deep RNN proposed in this paper, was utilized to take into account the transition of face saliency from previous frames, either in short-term or long-term. As a result, saliency maps of multiple-face videos can be generated upon the predicted salient face. Finally, the experimental results illustrated that our method is able to significantly advance state-of-the-art saliency prediction on multiple-face videos.

Acknowledgement. We would like to thank KingFar International Inc to provide the eye tracker and its technical support. Also, we thank all participants in the eye-tracking experiment. This work was supported by the NSFC projects under Grants 61573037 and 61202139, and Fok Ying-Tong education foundation under Grant 151061.

References

[1] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, Jan. 2013.

[2] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *Advances in neural information processing systems (NIPS)*, 2005.

[3] N. J. Butko and J. R. Movellan. Optimal scanning for faster object detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2751–2758, 2009.

[4] M. Cerf, J. Harel, W. Einhäuser, and C. Koch. Predicting human gaze using low-level saliency combined with face detection. In *Advances in neural information processing systems (NIPS)*, 2008.

[5] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu. Visual saliency detection by spatially weighted dissimilarity. In *Computer Vision and Pattern Recognition (CVPR)*, pages 473–480, 2011.

[6] Y. Fang, W. Lin, Z. Chen, C.-M. Tsai, and C.-W. Lin. Video saliency detection in the compressed domain. In *ACM international conference on Multimedia (ACM MM)*, pages 697–700, 2012.

[7] M. Grant, B. Stephen, and Y. Ye. Cvx: Matlab software for disciplined convex programming. *cvxr.com*, 2008.

[8] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, 19(1):185–198, Jan. 2010.

[9] H. Hadizadeh, M. J. Enriquez, and I. V. Bajić. Eye-tracking database for a set of standard video sequences. *IEEE Transactions on Image Processing*, 21(2):898–903, 2012.

[10] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems (NIPS)*, pages 545–552, 2006.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] S. Hossein Khatoonabadi, N. Vasconcelos, I. V. Bajic, and Y. Shan. How many bits does it take for a stimulus to be salient? In *Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2015.

[13] Y. Hua, Z. Zhao, H. Tian, X. Guo, and A. Cai. A probabilistic saliency model with memory-guided top-down cues for free-viewing. In *International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2013.

[14] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *International Conference on Computer Vision (ICCV)*, pages 262–270, 2015.

[15] R. L. Iman and W.-J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics-Simulation and Computation*, 11(3):311–334, 1982.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[17] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304 – 1318, Dec 2004.

[18] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306, Jun. 2009.

[19] L. Itti, N. Dhavale, and F. Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. *Optical Science and Technology*, 64:64–78, Jan. 2004.

[20] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov. 1998.

[21] M. Jiang, J. Xu, and Q. Zhao. Saliency in crowd. In *European Conference on Computer Vision (ECCV)*, pages 17–32. Springer, 2014.

[22] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Proc. ICCV*, pages 2106–2113, 2009.

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] S. S. Kruthiventi, K. Ayush, and R. V. Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *arXiv preprint arXiv:1510.02927*, 2015.

[25] M. Kummerer, T. S. Wallis, and M. Bethge. Deepgaze ii: Reading fixations from deep features trained on object recognition. *arXiv*, 1610.01563, 2016.

[26] J. Li, M. D. Levine, X. An, X. Xu, and H. He. Visual saliency based on scale-space analysis in the frequency domain. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2013.

[27] J. Li, C. Xia, Y. Song, S. Fang, and X. Chen. A data-driven metric for comprehensive evaluation of saliency models. In *International Conference on Computer Vision (ICCV)*, 2015.

[28] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 362–370, 2015.

[29] E. Matin. Saccadic suppression: a review and an analysis. *Psychological bulletin*, 81(12):899–917, Dec. 1974.

[30] P. K. Mital, T. J. Smith, R. L. Hill, and J. M. Henderson. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive Computation*, 3(1):5–24, Mar. 2011.

[31] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, and S. Yan. Static saliency vs. dynamic saliency: a comparative study. In *ACM international conference on Multimedia (ACM MM)*, pages 987–996, 2013.

[32] J. Pan, K. McGuinness, E. Sayrol, N. O'Connor, and X. Giro-i Nieto. Shallow and deep convolutional networks for saliency prediction. *arXiv preprint arXiv:1603.00845*, 2016.

[33] D. Pang, A. Kimura, T. Takeuchi, J. Yamato, and K. Kashino. A stochastic model of selective visual attention with a dynamic bayesian network. In *International Conference on Multimedia and Expo (ICME)*, pages 1073–1076, 2008.

[34] R. J. Peters and L. Itti. Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[35] U. Rajashekar, I. Van Der Linde, A. C. Bovik, and L. K. Cormack. Gaffe: A gaze-attentive fixation finding engine. *Image Processing, IEEE Transactions on*, 2008.

[36] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] D. Rudoy, D. B. Goldman, E. Shechtman, and L. Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1147–1154, 2013.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[39] S. Wu, M. Kan, Z. He, S. Shan, and X. Chen. Funnel-structured cascade for multi-view face detection with alignment-awareness. *Neurocomputing*, 2016.

[40] M. Xu, L. Jiang, Z. Ye, and Z. Wang. Bottom-up saliency detection with sparse representation of learnt texture atoms. *Pattern Recognition*, 2016.

[41] M. Xu, Y. Ren, and Z. Wang. Learning to predict saliency on face images. In *International Conference on Computer Vision (ICCV)*, 2015.

[42] J. Zhang and S. Sclaroff. Saliency detection: A boolean map approach. In *International Conference on Computer Vision (ICCV)*, pages 153–160, 2013.

[43] Q. Zhao and C. Koch. Learning a saliency map using fixated locations in natural scenes. *Journal of vision*, 2011.