Temporal Recurrences for Video Saliency Prediction

Panagiotis Linardos[1]
linardos.akis@gmail.com

Eva Mohedano[1]
eva.mohedano@insight-centre.org

Juan Jose Nieto[1]
juanjo.3ns@gmail.com

Noel E. O'Connor[1]
noel.oconnor@insight-centre.org

Xavier Giro-i-Nieto[2]
xavier.giro@upc.edu

Kevin McGuinness[1]
kevin.mcguinness@insight-centre.org

[1] Insight Center for Data Analytics
Dublin City University
Dublin, Ireland

[2] IDEAI-GPI
Universitat Politecnica de Catalunya
Barcelona, Catalonia/Spain

Abstract

This paper investigates modifying an existing neural network architecture for static saliency prediction using two types of recurrences that integrate information from the temporal domain. The first modification is the addition of a ConvLSTM within the architecture, while the second is a conceptually simple exponential moving average of an internal convolutional state. We use weights pre-trained on the SALICON dataset and fine-tune our model on DHF1K. Our results show that both modifications achieve state-of-the-art results and produce similar saliency maps.

1 Introduction

Visual saliency pertains to how an object or any piece of information may stand out from its surroundings. Detecting saliency is an integral part of how sentient organisms process information. We live in a world where the visual data we receive on a daily basis is immense and cluttered with noise; therefore, the brain has evolved in such a way that allows living organisms to focus their attention on the most relevant information, so as to function efficiently. Efforts in the computer vision community have been ongoing for many years to simulate this biological process artificially leading to the development of large-scale static gaze datasets, (e.g. SALICON [10]) and, more recently, dynamic gaze datasets (e.g. DHF1K [24]). Based on these datasets, model-driven approaches tackle the task of saliency prediction by estimating heatmaps of probabilities, where every probability corresponds to how likely it is that the corresponding pixel will attract human attention. Thanks to the availability of large-scale datasets, deep learning architectures have managed to significantly improve the accuracy achievable in this task (e.g. [6, 10, 13, 16, 24]).

Most scientific interest has so far been focused on image-based saliency models, with video saliency prediction gaining more traction in recent years with the introduction of large-scale video saliency datasets ([13, 24]). When it comes to extracting visual information from the temporal domain, ConvLSTMs have become increasingly popular, achieving state-of-the-art results in various computer vision tasks (e.g. [24, 25, 26]). In this work we augment a state-of-the-art architecture for image saliency [16] by adding a ConvLSTM module within its internal structure, similar to [6, 24]. More interestingly, we also test a much simpler method for temporal stability. We wrap a convolutional layer with a temporal exponential moving average (EMA) [17] operation. Using this recurrence, the output will always be a smoothed average of its previous states. This method is already used in gradient descent with momentum [21] to speed up convergence, replacing the current gradient with the exponential moving average of current and past gradients, derived from mini-batches of the data. To the best of our knowledge, this is the first time that this method has been applied within the architecture of a neural network.

Ablation studies are commonly used to better understand the performance impact of added components. Whilst this has merit, we propose that simple functions should also be used to investigate the necessity of complex modifications. To this end, in this work we consider both an elaborate ConvLSTM recurrence and a much simpler weighted average recurrence, and show that the simpler approach competes with the ConvLSTM on the task of video saliency.

2   Related Work

Video saliency prediction with deep neural networks has basically adapted to this task the architectures proposed for video action recognition. A first popular option are two-stream networks [19], in which the motion information is encoded by a pre-computation of the optical flow and adding it in a separate tower from the RGB channels. This is the approach adopted by STSConvNet [1]. This solution presents two important limitations: the computation overhead that is necessary to compute the optical flow, and the lack of temporal perspective further than the pairs of consecutive frames typically considered when computing optical flow. These shortcomings are partially addressed with the neural architectures where the temporal relation across frames is computed by a recurrent neural network (RNN) [4]. RNN-based deep models for saliency prediction have already been explored [2, 6, 9, 24] and are the core of the state of the art solutions. Similarly to [14] for activity detection, RMDN [2] combined the short-term memory encoded by C3D spatio-temporal convolutions [22] with a long short-term memory encoded by a plain LSTM. However, most current works have adopted a ConvLSTM layer as temporal recurrence, so that the recurrent layer would have a notion of space at a local scale. The OM-CNN model proposed in [9] fuses the RGB and optical flow from two-stream architecture with two ConvLSTMs. The authors of the largest dataset for video saliency prediction, the DHF1K (Dynamic Human Fixation 1K) dataset[24], trained a deep neural model based on ConvLSTM layers with attention (ACLNet). The authors of [6] exploit an existing model pre-trained for static saliency prediction, but with a more complex architecture composed of four branches fused with a ConvLSTM.

Our model outperforms the presented state of the art with a simple architecture that only considers RGB frames as input. As in some of the referred works, we exploit a model pre-trained with static images and study its enhancement with two types a temporal recurrence.
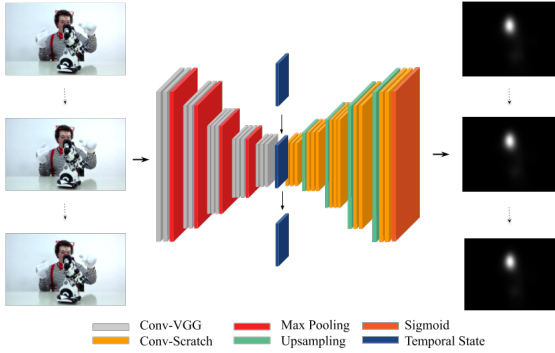
Figure 1: Architecture of the our model. A frame is input to the model at each time step. Information encoded from the past frames persists via our recurrence that is located deeper in the network. The output is a per-frame saliency map.

3 Architecture

The adopted neural architecture follows an encoder-decoder scheme that processes the temporal recurrence in the bottleneck. The topology of both encoder-decoder is adopted from SalGAN [16], the current top performing static saliency model on the DHF1K saliency benchmark. SalGAN encoder corresponded to the popular VGG-16 convolutional network [18] designed and trained to solve an image classification task. At the decoder side, SalGAN used the same layers as in the encoder in reverse order, and interspersed by upsampling instead of pooling operations. The original SalGAN model was trained using a combination of adversarial and binary cross entropy (BCE) loss. Here, for simplicity, we use only BCE and term the resulting architecture *SalBCE*.

We introduce a temporally aware component into the SalBCE network. This is either the addition of a ConvLSTM layer or an exponential moving average (EMA) applied on a pre-existing convolutional layer. Figure 1 presents a schematic of our architecture.

3.1 ConvLSTM

An LSTM is an autoregressive architecture that controls the flow of information in the network using 3 gates: update, forget, and output (Figure 2, *left*). In ConvLSTMs [25], the operations at each gate are convolutions. Temporal information is preserved in the cell state C_t upon which gated element-wise operations are performed by the update and forget gate. The hidden state H_t is concatenated with the input at each step and propagated through linear and non-linear operations at the gates. At each gate the current state S_t of the model is passed through the ConvLSTM gates and the cell state C_t and hidden state H_t are updated. In the following equations '∘' represents the element-wise product, '∗' a convolution operation, 'σ' the sigmoid logistic function and 'tanh' the hyperbolic tangent. The **update**, **forget**, and
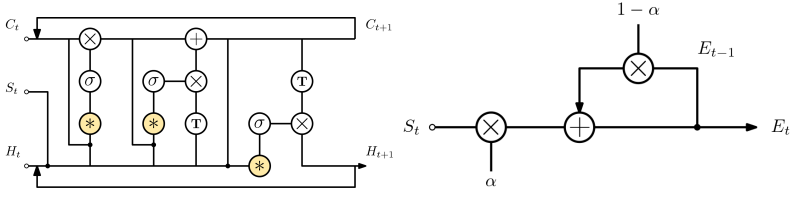
Figure 2: *(Left)* LSTM recurrence. Parametric operations are highlighted in yellow. *(Right)* EMA recurrence

output gates can be written as:

$$u_t = \sigma(W_u^S * S_t + W_u^H * H_{t-1} + W_u^C \circ C_{t-1} + b_u) \tag{1}$$

$$f_t = \sigma(W_f^S * S_t + W_f^H * H_{t-1} + W_f^C \circ C_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma(W_o^S * S_t + W_o^H * H_{t-1} + W_o^C \circ C_{t-1} + b_o) \tag{3}$$

and the new cell state C_t and hidden state H_t are then given by:

$$C_t = f_t \circ C_{t-1} + u_t \circ \tanh(W_C^S * S_t + W_C^H * H_{t-1} + b_C) \tag{4}$$

$$H_t = o_t \circ \tanh(C_t) \tag{5}$$

where W_*^* and b_* are the model parameters.

We added the ConvLSTM architecture at the bottleneck of our model, so that the input to the ConvLSTM is an encoded representation of the frame at time t. The output cell state C_t is fed to the decoder for further processing that results in a saliency map. To obtain the saliency map, a 1×1 convolution is used at the final layer of the decoder, so as to filter out all channels but one. We sequentially pass video frames to the model as input and get a sequence of time-correlated saliency maps in the output. The ConvLSTM component learns to leverage the temporal features during training. The name we gave to this type of model is *SalCLSTM*.

3.2 Exponential Moving Average

As an alternative approach, the exponential moving average (EMA) recurrence [7] is added on a specified layer so that at time t the convolutional state of this layer will be a decaying weighted average of the current and all previous states (Figure 2, *right*). At time t the convolutional layer S_t outputs a state that is fed to the exponential weighted average. The output E_t is then propagated further in the model. Note that there is a hyperparameter α that affects the impact of previous states on the current time step (the lower the value the higher the impact).

$$E_t = \alpha S_t + (1 - \alpha)E_{t-1} \tag{6}$$

This recurrence is straightforward to implement, especially compared to the ConvLSTM. We experimented with the placement of the EMA function at several different layers with $\alpha = 0.1$. We name our model *SalEMA*. On the initial step, where there is no past information, the model runs like a static saliency map predictor.

	tuned on DHF1K	AUC-J	s-AUC	NSS	CC	SIM
SalBCE (Baseline)	✗	0.874	0.724	2.047	0.382	0.268
SalBCE	✓	0.880	0.632	2.285	0.420	0.339
SalEMA	✗	0.883	**0.734**	2.144	0.400	0.276
SalEMA	✓	0.883	0.685	**2.402**	**0.435**	**0.349**
SalCLSTM	✓	**0.887**	0.693	2.364	**0.435**	0.322

Table 1: Performance results on the DHF1K validation set.

4 Training

The parameters of SalCLSTM and SalEMA were estimated by backpropagating a pixel-wise content loss that compared the value of each pixel in the predicted saliency map with its corresponding pixel in the ground truth map. The total binary cross entropy loss was computed as the average of the individual binary cross entropies (BCE) over all pixels:

$$L_{BCE} = -\frac{1}{N} \sum_{n=1}^{N} P_n \log(Q_n) + (1 - P_n) \log(1 - Q_n) \tag{7}$$

where P represents the predicted saliency map and Q the ground truth saliency map.

SalCLSTM and SalEMA were not trained from scratch though, as the parameters of the encoder-decoder convolutional layers were adopted from SalBCE. SalBCE was trained for 27 epochs over the SALICON [10] dataset of still images using only the same BCE loss. We also utilized data augmentation techniques (mirroring and rotation of frames) which resulted in improved performance.

Our next step was adding recurrence that uses the intrinsic temporal information of video datasets and train it with the DHF1K dataset [24]. The DHF1K dataset [24] contains 700 annotated videos at 640×360 resolution. We extracted frames at their original 30 fps rate, and resized them to 192×256 resolution. We loaded them using a batch size of 10 frames from a single video at a time. By backpropagating the loss through time up to a maximum of 10 frames, we avoid exceeding memory capacity and potential vanishing or exploding gradients. We found it was necessary to initialize the ConvLSTM recurrence with the Xavier initialization method [5], otherwise this model would converge to black images rather than saliency maps. This was likely due to oversaturation of the sigmoid activation layer. We trained all our models for 7 epochs, where we observed the loss reaching a plateau on our baseline. We used the Adam optimizer [11] with a learning rate of 10^{-7}.

5 Evaluation

The effect of temporal recurrences proposed for SalEMA and SalCLSTM was assessed with five different visual saliency metrics: Normalized Scanpath Saliency (NSS), Similarity Metric (SIM), Linear Correlation Coefficient (CC), AUC-Judd (AUC-J), and shuffled AUC (s-AUC). In all cases, a higher value corresponds to a better performance. The reader is referred to [3] for a detailed description of these metrics. The reported figures correspond to an average per video, that is, we first compute the metric on each frame, then average across all frames of each video, and we finally average across all videos.

We train and evaluate our models on three video saliency datasets, namely DHF1K [24], Hollywood-2 and UCF-sports [13]. DHF1K is a large scale dataset with a high diversity of contents and variable length (from 400 frames to 1200 frames at 30fps). It includes 1000 videos, out of which 700 are publicly annotated, and 300 are withheld for testing purposes. In contrast to DHF1K, Hollywood-2 [12] and UCF-sports [20] are limited to human actions and can be categorized as task-driven, given that the observers were explicitly asked to identify actions and scene context. These datasets were originally formed for the task of action recognition and were later adopted as a video saliency benchmark. Furthermore, both datasets have been divided into separate shots, so that no scene change occurs in the sequences that are fed into the models. Hollywood-2 is split into a training set of 3100 clips and a test set of 3559 clips, while UCF-sports has been split to a training set of 104 clips and a test set of 48 clips. These shots are much smaller in size than a DHF1K video sample, ranging from 40 frames to just a single frame per shot. We also use SALICON [10], a large-scale image saliency database, to set a baseline. DHF1K is used for experimenting with variations over the proposed models, as well as for comparison with the state of the art together with Hollywood-2 and UCF-sports.

The results in Table 1 indicate that the simple addition of EMA even without extra training does almost as well as a sophisticated ConvLSTM recurrence, and even improves it after being fine-tuned with the DHF1K training partition. EMA essentially performs a smoothing over the frames of the video by averaging. A possible explanation for why this boosts performance in video saliency is that saliency tends to be relatively consistent across frames, with the exception of rapid movements.

Encouraged by the positive results of our EMA modification at the bottleneck (layer 30), we explored more possible locations of the EMA function. In particular we tested its placement on: output (layer 61), decoder (layer 56), encoder (layer 7). We also implemented a variation that integrates EMA at two separate layers simultaneously, one in the encoder (7) and one in the decoder (56). In that case we set α to 0.3 at each location so as to not have an over-smoothing effect that would result in a significant lag at adapting to changes in the scene. Furthermore, in a video there can be spontaneous scene changes. In such instances, it would be optimal to have the EMA reset and forget all the previous states. However, EMA is not adaptive in this way, so we experimented with a skip connection that allows information to bypass this layer instead [7]. We also applied a second type of regularization, the dropout technique [8], at the convolutional layer right before the EMA layer. Dropout essentially turns off neurons with a preassigned probability (0.5) at each training step. This mitigates co-adaptation of neurons during training, allowing for clusters of neurons to learn independently. This way, at test time, we get the average from an ensemble of layers at location 30. The average of this ensemble pertains to spatial information, but since we are also using EMA, we get the moving average across the temporal dimension as well. The results reported in Table 2 do not show a clear winning configuration across the five metrics metrics but, as NSS and CC are considered as the most appropriate ones to capture viewing behavior [3], we adopted SalEMA30 with dropout as our best configuration.

Furthermore, we evaluated our two models on Hollywood-2 and UCF-sports [13]. We compare our models to the current state-of-the-art as evaluated on the test split of the corresponding datasets by Wang *et al*. [24]. Like ACLNet [24], our models were trained first for DHF1K in all cases, and later fine-tuned for the specific Hollywood-2 or UCF-Sports dataset. Table 3 shows how, for DHF1K, SalEMA achieves the best performance compared to other models in the current benchmark across all metrics but s-AUC. On the other hand, SalCLSTM obtains the best results on all metrics for UCF-Sports and leads the performance

Model	tuned on DHF1K	AUC-J	s-AUC	NSS	CC	SIM
SalEMA30	✗	0.883	0.734	2.144	0.400	0.276
SalEMA30	✓	0.883	0.685	2.402	0.435	0.349
SalEMA30 (dropout)	✓	0.886	0.690	**2.495**	**0.450**	**0.360**
SalEMA30 (residual)	✓	0.875	0.670	2.274	0.415	0.339
SalEMA61	✗	0.884	**0.737**	2.133	0.399	0.270
SalEMA61	✓	**0.888**	0.681	2.394	0.438	0.354
SalEMA54	✗	0.883	0.734	2.149	0.401	0.276
SalEMA7	✗	0.872	0.656	2.217	0.409	0.318
SalEMA7&54	✓	0.828	0.561	1.403	0.366	0.344

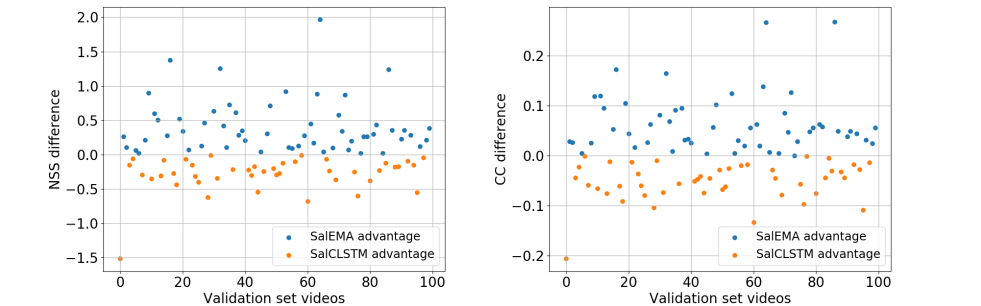Table 2: Performance of SalEMA variants on DHF1K.

Figure 3: Per-video comparison between SalEMA and SalCLSTM using the NSS and CC metric on the DHF1K validation set. The values represent the margin by which a model's performance differs from the other.

on AUC-J, NSS and CC for Hollywood-2.

A more detailed analysis between SalEMA and SalCLSTM was obtained by plotting the difference in their NSS and CC performance per video in the DHF1K validation set (100 videos). Concretely, we subtracted the metric value achieved by the SalCLSTM from that of SalEMA in each video and display the results in Figure 3. This way, we can assess whether the two configurations end up producing similar results. In this case we would expect the variance to be low and the NSS difference to be close to zero most of the time. However, the results are sparse and diverge from video to video. This observation serves as evidence that the function approximated by the ConvLSTM is differs from that of an exponential moving average, despite its similar overall effectiveness.

We also delved deeper into the Hollywood-2 dataset for potential clues that would explain the difference in performance. This dataset consists of very small shots, including even single-frame shots. In these cases we found that the ConvLSTM does much better than the EMA (by a margin of around 4 NSS points). We also noticed, however, that in these cases the ground truths for the saliency maps correspond to a central Gaussian, despite the fact that other salient objects are present in other locations of the frame. Figure 5 shows two examples

Dataset	Model	AUC-J	s-AUC	NSS	CC	SIM
DHF1K	SalEMA	**0.890**	0.667	**2.573**	**0.449**	**0.465**
	SalCLSTM	0.887	0.693	2.364	0.435	0.322
	ACLnet [24]	**0.890**	0.601	2.354	0.434	0.315
	SalGAN [16]	0.866	**0.709**	2.043	0.370	0.262
	DVA [23]	0.860	0.595	2.013	0.358	0.262
Hollywood-2	SalEMA	0.919	0.708	3.186	0.613	0.487
	SalCLSTM	**0.933**	0.715	**3.499**	**0.672**	0.530
	ACLnet [24]	0.913	**0.757**	3.086	0.623	**0.542**
	OM-CNN [9]	0.887	0.693	2.313	0.446	0.356
	DVA [23]	0.860	0.727	2.459	0.482	0.372
UCF-sports	SalEMA	0.906	0.740	2.638	0.544	0.431
	SalCLSTM	**0.914**	**0.782**	**3.063**	**0.611**	**0.477**
	ACLnet [24]	0.897	0.744	2.567	0.51	0.406
	DVA [23]	0.872	0.725	2.311	0.439	0.339
	OM-CNN [9]	0.870	0.691	2.089	0.405	0.321

Table 3: Comparison of SalEMA and SalCLSTM with the state of the art on DHF1K, Hollywood-2, and UCF-sports test sets.

α	AUC-J	s-AUC	NSS	CC	SIM
0.05	**0.886**	0.687	2.470	0.448	0.358
0.1	**0.886**	**0.690**	**2.495**	**0.450**	**0.360**
0.2	0.885	0.688	2.476	0.446	0.358
0.3	0.884	0.685	2.451	0.442	0.356

Table 4: Sensitivity of SalEMA30 to α on DHF1K validation.

in which the provided ground truth focuses in the center, although different faces appear in the image. In these examples, SalEMA captures these salient objects better, while SalCLSTM seems to focus on the center.

Finally, we experimented with the α hyperparameter by varying its value and also by making it trainable. Table 4 shows relatively stable performance despite the variations on the value. We also had our model learn alpha on its own by introducing a trainable parameter p. To ensure that the resulting update equation represents a convex combination of the current features and previous state, p is passed through a sigmoid so that the final value is constrained to $[0,1]$. The resulting recurrence is:

$$E_t = \sigma(p)S_t + (1 - \sigma(p))E_{t-1} \qquad (8)$$

Whereas all other parameters of the model are set to a learning rate of 10^{-7}, the learning rate of alpha was set to 0.1 and was trained separately for 3 epochs on SalEMA pretrained with $\alpha = 0.1$. We set $\sigma(p)$ to 0.5 at the start of this tuning and by the end, it converges to 0.1477. The final performance was found to be approximately the same as the best model in Table 4.
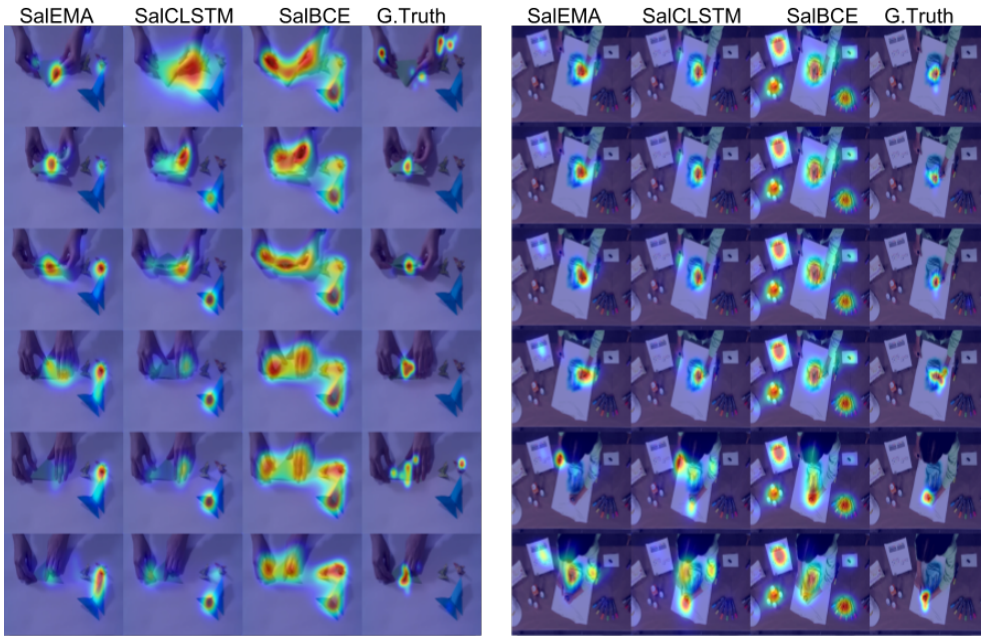
Figure 4: We picked two samples that showed high divergence in performance between the two methods and visualized the predictions. SalEMA did much better than SalCLSTM on the sample displayed on the left side, while it was the opposite for the other sample. The images displayed correspond to intervals of 100 frames in the video.

6 Conclusions

This work has presented SalEMA and SalCLSTM, two variations of a convolutional neural network for video saliency prediction. Their main difference is how temporal recurrence is modelled, whether with a simple yet effective exponential moving average with a single parameter, or a convolutional LSTM that despite being adopted for many video sequence processing tasks, seems needlessly complex for this specific task of video saliency prediction. This indicates that, in some cases, components of more sophisticated models may just learn to approximate much simpler functions. It is likely that similar methods can be conceived of in other types of tasks as well.

On another note, ablation studies are a common practice for evaluating the contribution that an added component has on a model's performance. We argue that there should be a more detailed effort in analyzing the behavior of deep architectures. Using predefined functions like the one presented in this work may shed more light on the necessity of a complex architecture.
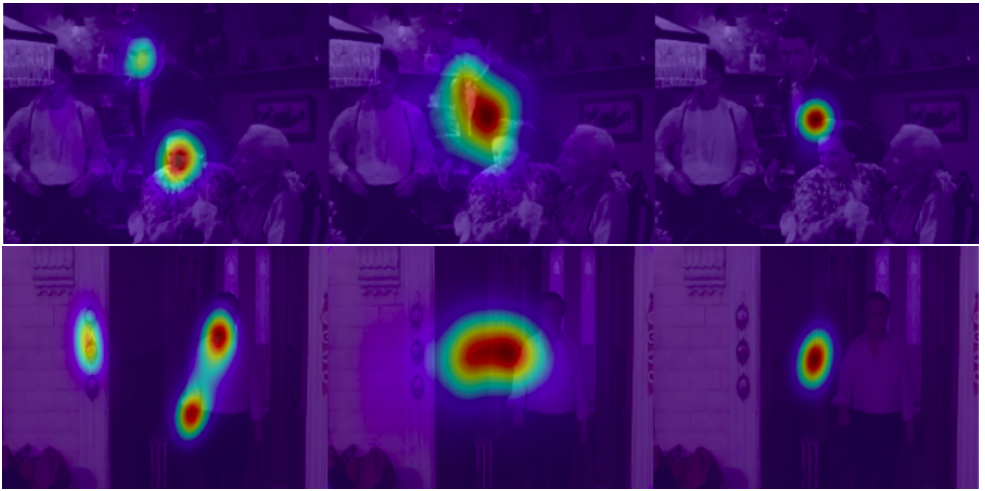
Figure 5: Predictions from two Hollywood outliers where SalEMA performed particularly bad. The order corresponds to: SalEMA (left image), SalCLSTM (middle image), ground truth (right image). The ground truth appears aberrant, as it completely ignores human faces that are well-known to be salient objects.

References

[1] Cagdas Bak, Aysun Kocak, Erkut Erdem, and Aykut Erdem. Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, 2017.

[2] Loris Bazzani, Hugo Larochelle, and Lorenzo Torresani. Recurrent mixture density network for spatiotemporal visual attention. In *International Conference on Learning Representations (ICLR)*, 2017.

[3] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757, 2019.

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[6] Siavash Gorji and James J Clark. Going from image to video saliency: Augmenting image salience with dynamic attentional push. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7501–7511, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning

for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[9] Lai Jiang, Mai Xu, and Zulin Wang. Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm. *arXiv preprint arXiv:1709.06316*, 2017.

[10] M. Jiang, S. Huang, J. Duan, and Q. Zhao. Salicon: Saliency in context. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1072–1080, June 2015. doi: 10.1109/CVPR.2015.7298710.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition*, pages 2929–2936. IEEE Computer Society, 2009.

[13] Stefan Mathe and Cristian Sminchisescu. Actions in the eye: Dynamic gaze datasets and learnt saliency models for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1408–1424, 2015.

[14] Alberto Montes, Amaia Salvador, Santiago Pascual, and Xavier Giro-i Nieto. Temporal activity detection in untrimmed videos with recurrent neural networks. *arXiv preprint arXiv:1608.08128*, 2016.

[15] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, 2016.

[16] Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[17] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[20] Khurram Soomro and Amir R Zamir. Action recognition in realistic sports videos. In *Computer vision in sports*, pages 181–208. Springer, 2014.

[21] Ilya Sutskever, James Martens, George E Dahl, and Geoffrey E Hinton. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28(1139-1147):5, 2013.

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[23] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378, 2018.

[24] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4903, 2018.

[25] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[26] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018.