Long Spatio-Temporal Saliency Map Prediction

Halil İbrahim Öztürk
Hacettepe University Computer Science Department
halil_ozturk@hacettepe.edu.tr

Abstract

In this work, we propose Long Spatio-Temporal Saliency (LSTS) models which estimate saliency map from input frame or optic flow sequence. Intuition behind the models is that longer sequences provides more meaningful features to estimate visual saliency map. Our models accept features extracted from I3D networks which holds best scores in action recognition benchmarks. We trained our models on DHF1K dataset which is largest and diverse eye tracking dataset. Scores of LSTS models are slightly worse than state-of-the-art methods. Our results show that optic flow is more important than only RGB inputs in visual saliency map estimation.

1. Introduction

When a human looks a scene, he/she focuses to salient parts of the scene. Ignoring irrelevant parts in a scene provides less effort to understand a complex scene. This process is named as visual attention. Computational models which imitate human visual system gained attention in last decades. There are two different groups of the computational models. First of them is saliency map estimation from static scene viewing, second of them is saliency map prediction from dynamic scene viewing. Saliency detection models are used in video captioning, object detection, question answering, action recognition, video summarization.

Dynamic saliency map prediction is difficult than static saliency map prediction. Static saliency models utilize only spatial features, dynamic saliency estimation requires utilizing temporal features with spatial features. In this respect, dynamic saliency estimation shares same challenge with action recognition problem. In recent years, action recognition models have been proposed [2] [16] [15] [17]. Action recognition models usually use 3D convolutions which firstly used in action recognition in C3D [16]. C3D network accepts 16 consecutive RGB frames. But as reported in [15], activations of networks which consist 3D convolutions and accept RGB sequences are fired on background context pixels. But activations of networks which accept optic flow sequences are fired on movements.

Features extracted from mid or high level layer of action recognition models are used to training new models for different tasks [14] [9]. I3D[2] holds best scores on action recognition datasets. Our models use features which extracted from I3D networks. We propose LSTS-RGB, LSTS-Flow and LSTS-RGB&Flow. We evaluated our models on DHF1K [21] dataset.

2. Related Works

Spatial Saliency Network (SSNet), Temporal Saliency Network (TSNet), Spatio-Temporal Max Fusion Network (STSMaxNet), Spatio-Temporal Convolution Fusion Network (STSConvNet) networks have been proposed in [1]. First two networks use only RGB or flow information, others use RGB and flow information together. STSMaxNet fuses RGB and flow stream by max fusion, STSConvNet fuses RGB and flow stream by convolution after concatenation of two streams.

SUSiNet [8] employs 3D convolutions in a deep network to predict saliency map. The proposed network is multi-task spatio-temporal network. There are three output branches. First of the output branch is designed for video summarization, the branch makes binary classification. Second task is action classification, it predicts class of action from 51 different classes. Last one generates saliency estimation. Skip connections are used to avoid gradient vanishing problem.

Wang Et. Al. [19] introduces a new benchmark dataset DHF1K for saliency estimation from video. They propose an architecture to predict saliency map. The method uses CNN-LSTM architecture, the features are extracted from VGG-16 [13] network and fed to LSTM unit.

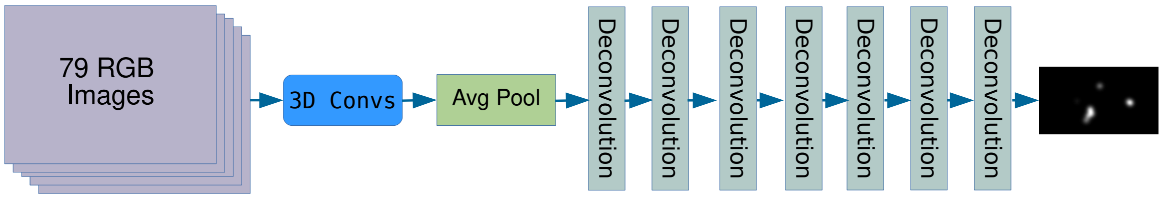


Figure 1: Long Spatio-Temporal Saliency RGB Network architecture

3. Method

I3D [2] network which we use as feature extractor consists two networks. First network (I3D-RGB) takes RGB frame sequence as input, second network (I3D-Flow) takes optic flow sequence extracted from the RGB frame sequence as input. Classification outputs of two networks are summed and normalized to predict action class.

Longer input sequence contains more information than short sequence for salient pixels estimation. For this reason, we suggest three models which accept extracted features from I3D networks.

3.1. Long Spatio-Temporal Saliency RGB (LSTS-RGB)

This model accepts 79 RGB frames sequence as input. 3D convolution filters are applied to input sequence, the weights of filters are taken from I3D-RGB network. Since I3D network contains a lot of parameter, training of I3D requires large-scale dataset such as Sports-1m [7] and computation power. We did not train 3D convolution filters to decrease training time. Following layer of last 3D convolution layer is average pooling layer. Average pooling layer shrinks input size of transposed convolution (deconvolution) layer, therefore inference time and training time decrease. Batch normalization [4] layers follow after each transposed convolution layers except last one.

Our network consists 7 transposed convolution layer as in figure 1, we used more layers with smaller kernels instead of less layers with larger kernels. Since nonlinearities are increased by more layers, representation capability of the network increases. Also VGG network [13] works in same principle, the authors mentions that multiple smaller filters with additional nonlinearities in between approximate the effect of a filter with big kernel.

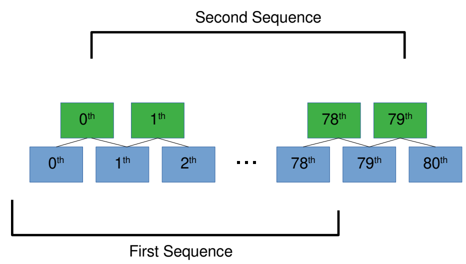


Figure 3: Input sequence of LSTS-RGB

Saliency map estimation for n^{th} frame of a video requires preceding 78 frames. For this reason, first estimation can be made for 78^{th} frame. Following sequences are created by 1 frame stride in temporal space.

3.2. Long Spatio-Temporal Saliency Flow (LSTS-Flow)

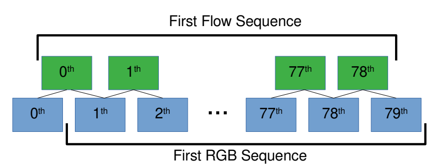


Figure 4

LSTS-Flow network architecture is same with LSTS-RGB network architecture, except input size is 244x244x2 instead of 224x224x3. Also weights of 3D convolution layers are taken from I3D-Flow instead of I3D-RGB.

Optic flow is calculated from consecutive two frames. First optic flow is extracted when 1^{th} frame is captured, not 0^{th} frame. Because of that, first optic flow sequence is ready when 79^{th} frame is captured. Next sequence is ready when 80^{th} frame is captured. Next sequences are ready after each frame is captured.

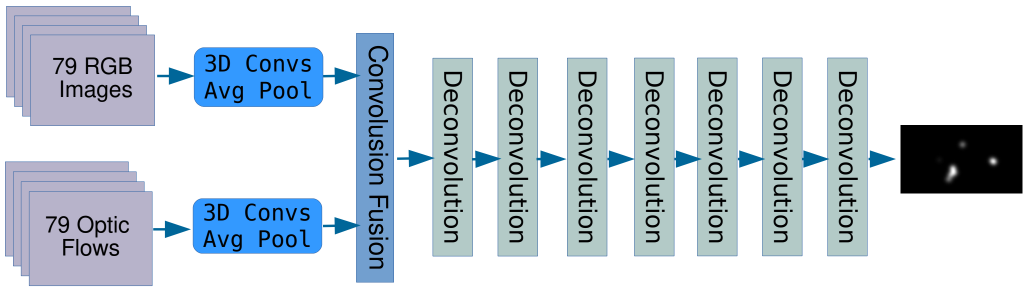


Figure 2: Long Spatio-Temporal Saliency RGB and Flow Network architecture

3.3. Long Spatio-Temporal Saliency RGB & Flow (LSTS-RGB&Flow)

LSTS-RGB&Flow network accepts RGB and optic flow sequence of same frames. We do not include first frame to RGB sequence to overlap all of the input frames, first RGB and optic flow sequences are visualized in figure 4.

Until Convolution Fusion layer RGB and flow inputs follows RGB and flow branches. Weights of 3D convolutions in the branches are taken from I3D networks. The branches output same size activation maps. The outputs are concatenated and fed to 3D convolution layer as in STSConvNet[1]. Fused output is fed to 7 consecutive transposed convolution layers in LSTS-RGB&Flow network, as in figure 2.

4. Implementation Details

4.1. Network Architectures

LSTS-RGB network accepts 79 RGB image sequence as input. The input sequence is passed through I3D layer until mixed5c layer. Average pooling is applied to mixed5c activations. Output size of mixed5c layer is decreased from 1024x7x7x7 to 1024x5x5x5 after average pooling layer. Last 3D convolution layer generates output which has 512x1x5x5 shape.

D1(K(f1,f2),P(p1,p2),S(s1,s2)) represents first transposed convolution layer with f1xf2 kernel applied to the input with p1xp2 padding and s1xs2 stride. Operations after last 3D convolution are D1(K(3,3), P(1,1), S(2,2)) $->$ D2(K(4,3),P(1,1),S(3,2)) $->$ D3(K(1,3), P(0,1), S(1,2)) $->$ D4(K(4,3), P(1,1), S(3,2)) $->$ D5(K(1,3), P(0,1), S(1,2)) $->$ D6(K(3,3) P(1,1), S(2,2)) $->$ D7(K(3,3), P(1,1), S(2,2)). ReLU [10] activation layer follows transposed convolution and 3D convolution layers except last transposed convolution layer. In order to generate saliency map values between 0 and 1 sigmoid activation layer follows last transposed convolution layer. Output size of the networks is 640x360 where groundtruth maps of DHF1K [19] dataset is 640x360.

4.2. Data Preprocessing

During training of I3D networks, videos are resized to 256x256 and input images are randomly cropped to 224x224. Salient objects can be positioned everywhere in image. For this reason, we do not crop image during training or evaluation.

We resize inputs to 112x112 and upscale to 224x224 during optic flow calculation. Because, optic flow calculation by TV-L1 algorithm is costly. Optic flows are truncated to [-20,20] range, then rescaled to [-1,1]. Also RGB frames are rescaled to [-1,1] range before passing to our networks.

4.3. Training

In order to decrease training time, we extracted features from I3D networks and stored them at disk. Stored features were shrunk by average pooling, thus storage size was decreased. We trained network which consists 7 transposed convolution layers and one 3D convolution layer. The network takes extracted features from I3D network as input. We employed RMSProp optimizer and Mean Squared Error objective function. There are 249497 clips in training set, we set batch size to 128. LSTS-RGB network has been trained 10 epochs where learning rate is 0.0001, weight decay is 0.00005 and momentum is 0. We trained LSTS-Flow and LSTS-RGB&Flow networks 2 and 1 epochs, respectively. Learning rate is halfed after each 50 iterations. Loss value is calculated from estimated saliency map and groundtruth saliency map.

We made experiments on the computer which has Nvidia GeForce GTX TITAN X and 2x Intel Xeon CPU E5-2640 v4 @ 3.4GHz. We calculated optic flows and stored them on disk. Calculating optic flows for first 700 videos takes 3 days. TV-L1 optic flow algorithm [20] does not work in parallel. Feature extraction from I3D networks takes between 1-2 days. Evaluation of the metrics which we mention in

	sAUC	AUC-Judd	NSS	CC
SUSiNet (1-task) [8]	0.6991	0.8843	2.5908	0.4676
Deep-Net [11]	0.6432	0.8421	1.5804	0.2969
DVA [18]	0.6572	0.8609	2.0644	0.3593
SAM [3]	0.6562	0.8680	2.1180	0.3684
ACLNet [19]	0.6523	0.8883	2.2962	0.4167
DeepVS [5]	0.6405	0.8561	1.9680	0.3500
LSTS-RGB (Ours)	0.6415	0.8280	1.441	0.3868
LSTS-Flow (Ours)	0.6983	0.8616	1.2561	0.3370
LSTS-RGB & Flow (Ours)	0.6218	0.8306	1.38033	0.3715

Table 1: Quantitative results on DHF1K dataset

5.1 is made on CPU. There is not CUDA implementations of the metrics. Testing takes about 1 day. We used mini-validation set to decrease validation time.

5. Experiments

5.1. Evaluation Metrics

We employed Area Under Curve Judd (AUC-Judd) [6], shuffled AUC (sAUC) [22], Normalized Scanpath Saliency (NSS) [12] and Pearson's Correlation Coefficient (CC). We used fixations as groundtruth during evaluation.

In a center-biased dataset, a center prior baseline will achieve a high AUC score. sAUC specifically penalizes models that include the center bias. sAUC accepts estimated saliency map, groundtruth fixation map and random fixation map from dataset. We randomly selected second fixation map from minibatch during validation. In test step, second fixation map is selected randomly from all of the test set. NSS is a simple correspondence measure between saliency maps and ground truth, computed as the average normalized saliency at fixated locations. NSS is sensitive to false positives, relative differences in saliency across the image. CC is a statistical method used generally in the sciences for measuring how correlated or dependent two variables are.

5.2. Experiments on DHF1K dataset

DHF1K [19] contains 1000 videos with different length and content. The eye tracking is made by 17 observers. Size of the videos in tha dataset is 4GB. The dataset is split into training validation and test parts. First 600 videos are in training split, following 100 videos in validation split and last 300 videos in test split. Groundtruths of training and validations splits are published, but groundtruths of test split are not published. For this reason, we used validation set as test set. We split training set to new training set and validation set which we used in validation.

5.3. Fair Evaluation

As we mentioned before, our models do not generate output for first 78 frames or first 79 frames. In test step, we copy output for 79th frame previous 78 times. We assumed the copied saliency maps as generated saliency maps in test step. We did not use same method in validation step.

5.4. Results

We compared our results with 6 state-of-the-art methods. Since groundtruths of test set of DHF1K dataset are not published, the quantitative results were computed on validation set of the dataset. Scores of other methods are taken from [8] paper. LSTS-Flow is more successful than LSTS-RGB and LSTS-RGB&Flow networks in sAUC and AUC-Judd metrics. Susinet which trained for only saliency prediction task outperforms other methods. Our best method LSTS-Flow is slightly worse than state-of-the-art in all metrics.

We did not train LSTS-Flow and LSTS-RGB&Flow enough epochs because of the time constraint. We think the models would be more successful, if we had more time to train them.

Qualitative results in figure 5 shows our networks outputs similar results. We checked our code several times to avoid possible mistakes, but the outputs are true. Quantitative results are different but qualitative results are similar.

6. Conclusion

We have proposed saliency estimation networks which consist 3D convolutions with weights taken from I3D networks. We could not train our models enough because of the time constraints. However the scores of our methods are slightly worse than state-of-the-art. This shows weights of action recognition models can be used to initialize saliency estimation network. Skipping first 78 frames and training cost are drawbacks our methods. We have not measured inference time, but most probably it is higher than state-of-the-art methods. We suggest a shallow action recognition network which takes long input sequence can be used instead of I3D as future work.

References

[1] Cagdas Bak, Aysun Kocak, Erkut Erdem, and Aykut Erdem. Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, 20(7):1688–1698, 2018.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. Predicting human eye fixations via an lstm-based saliency attentive model. *IEEE Transactions on Image Processing*, 27(10):5142–5154, 2018.

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

[5] Lai Jiang, Mai Xu, Tie Liu, Minglang Qiao, and Zulin Wang. Deepvs: A deep learning based video saliency prediction approach. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–617, 2018.

[6] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *2009 IEEE 12th international conference on computer vision*, pages 2106–2113. IEEE, 2009.

[7] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[8] Petros Koutras and Petros Maragos. Susinet: See, understand and summarize it. *arXiv preprint arXiv:1812.00722*, 2018.

[9] Federico Landi, Cees G. M. Snoek, and Rita Cucchiara. Anomaly locality in video surveillance. *CoRR*, abs/1901.10364, 2019.

[10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[11] 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.

[12] Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of bottom-up gaze allocation in natural images. *Vision research*, 45(18):2397–2416, 2005.

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

[14] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018.

[15] An Tran and Loong-Fah Cheong. Two-stream flow-guided convolutional attention networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3110–3119, 2017.

[16] 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.

[17] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

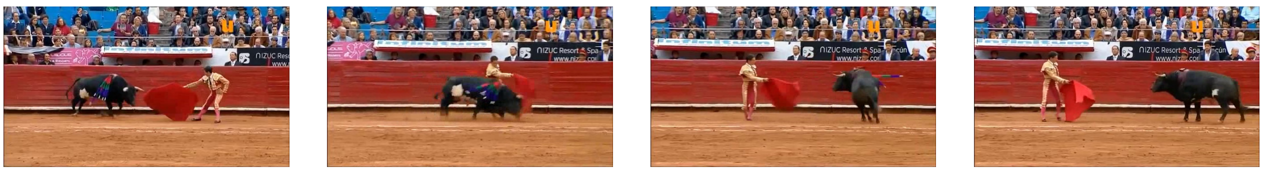
[18] Wenguan Wang and Jianbing Shen. Deep visual attention prediction. *IEEE Transactions on Image Processing*, 27(5):2368–2378, 2017.

[19] 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.

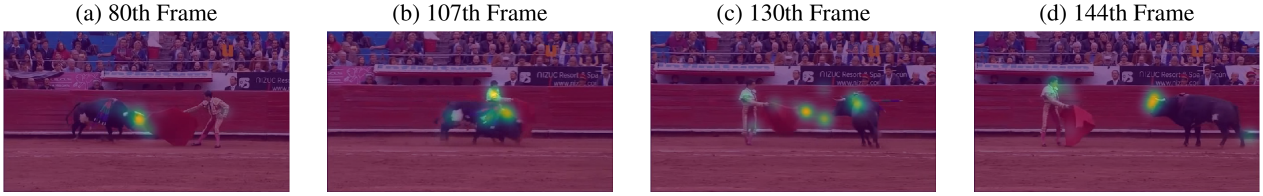
[20] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007.

[21] Kao Zhang and Zhenzhong Chen. Video saliency prediction based on spatial-temporal two-stream network. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

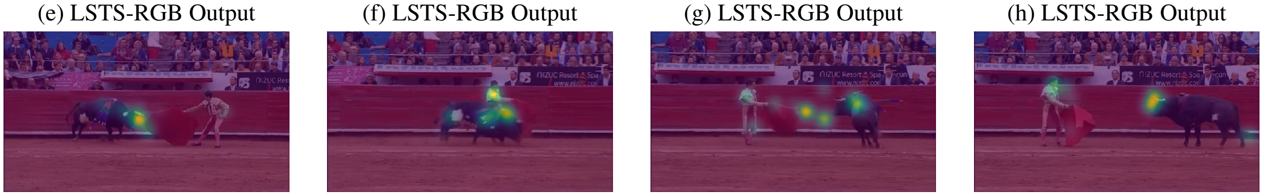
[22] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32–32, 2008.



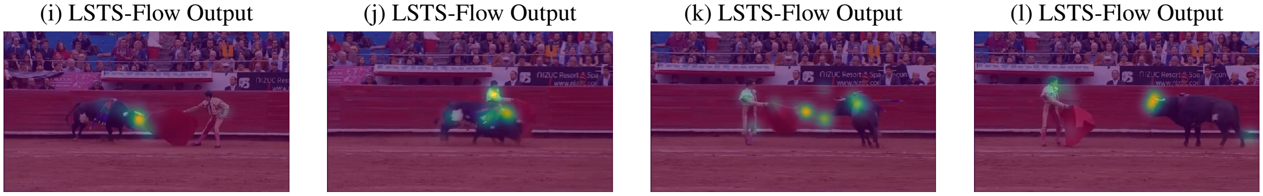
(a) 80th Frame (b) 107th Frame (c) 130th Frame (d) 144th Frame



(e) LSTS-RGB Output (f) LSTS-RGB Output (g) LSTS-RGB Output (h) LSTS-RGB Output



(i) LSTS-Flow Output (j) LSTS-Flow Output (k) LSTS-Flow Output (l) LSTS-Flow Output



(m) LSTS-RGB&Flow Output (n) LSTS-RGB&Flow Output (o) LSTS-RGB&Flow Output (p) LSTS-RGB&Flow Output

Figure 5: Inputs and outputs for 607^{th} video of DHF1K dataset