Spatio-Temporal Saliency Modeling for Graphical User Interfaces

Bharath Bangalore Somashekar
University of Stuttgart
Stuttgart, Germany
st164288@stud.uni-stuttgart.de

Keerthana Jaganathan
University of Stuttgart
Stuttgart, Germany
st164482@stud.uni-stuttgart.de

ABSTRACT

Modeling visual attention helps in determining the behavior of users while they observe the most important regions in a visual scene. Existing methods of visual attention modeling mostly require an eye-tracking device. A computational model to predict users' Spatio-temporal visual attention on WIMP-style (windows, icons, menus, pointer) graphical user interfaces solely using information available to the interface, such as users' mouse, keyboard input and the UI components was proposed and successfully implemented in [8]. Individual feature extraction processes for spatial and temporal attention models were performed that requires considerable amount of time and resources. We propose to use a deep learning LSTM(Long-Short Term Memory) model for joint Spatio-temporal attention prediction that is capable of handling huge time series data. Our LSTM model predicts attention maps more accurately and with lesser latency compared to our re-implementation of [8].

KEYWORDS

spatio-temporal saliency, visual attention, graphical user interfaces, LSTM

ACM Reference Format:
Bharath Bangalore Somashekar and Keerthana Jaganathan. 2020. Spatio-Temporal Saliency Modeling for Graphical User Interfaces. In *Proceedings of Fachpraktikum Interaktive Systeme (FIS'20).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION AND RELATED WORK

The tremendous growth in computer technology over the last few decades created a demand for efficient and intuitive ways for humans to interact with computers other than the methods like the mouse, keyboard, and printers. The human gaze was one such input modality introduced in the field of human-computer interaction. Gaze, therefore, has been used as an input modality for tasks ranging from desktop control, eye typing and target selection to password entry and notification display [8]. Gaze can not only be used as an explicit input but can also be used as an implicit input for applications such as detecting human activity or even to detect the visual attention of the user. Hence, the gaze is considered as one of the important modalities that can be used in a multitude of user applications.

The challenge lies in estimating the human gaze as it requires expensive eye trackers and complicated usage procedures. Even more importantly, eye trackers themselves only provide users' current and past gaze locations. They do not provide information on which locations or components the user will likely attend to or interact with in the future [8]. To overcome this, there is a need to get the human fixation information without the usage of an eye tracker.

One solution is to use a computational visual attention prediction model. Visual attention models mimic the capacity of a primate's visual system to focus on particular places in a visual scene [7]. Specifically, they produce the so-called saliency maps from the visual data where a high saliency score at an image location indicates that the point is more likely to be fixated [1].

Itti et al. in [5] describe in detail, the two models for visual attention prediction, one is the bottom-up model which considers low-level image features such as color, contrast, etc. while predicting saliency maps. The other is the top-down model that considers high-level task-related features like object classes to obtain saliency maps [1]. The recent trend has been to incorporate machine learning and deep learning approaches to predict the attention in not only static images(static models) but also in videos(dynamic models).

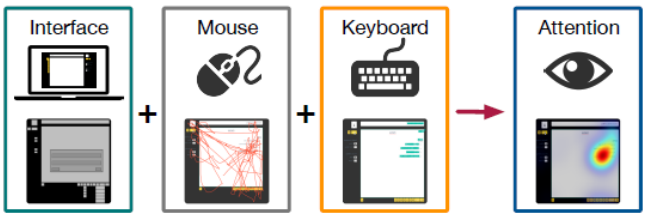


Figure 1: A computational model to predict users' spatio-temporal visual attention on a graphical user interface using interface and users' mouse and keyboard input [8].

1.1 Related work

Our project is based on [8] in which the authors argue that existing attention models are limited because they mainly rely on visual components, and they capture neither user input, such as from mouse and keyboard, nor interface information, such as components, nor do they take the history of user interactions with the interface into account. [8] proposes a computational model to predict users' Spatio-temporal visual attention prediction on WIMP style (windows, icons, menus, pointer) graphical user interfaces.

FIS'20, January 2020, Stuttgart, Germany

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

The spatio-temporal visual attention model in [8]:

(1) does not require any eye-tracking equipment.
(2) predicts attention solely using information available to the interface, specifically users' past mouse and keyboard input actions and the UI components they interacted with.
(3) introduces synthesized user interface layouts functionally equivalent to real-world interfaces, such as from Gmail, Facebook, or GitHub.
(4) quantitatively analyzes attention allocation and its correlation with user input and UI components using ground-truth gaze, mouse, and keyboard data of 18 participants performing a text editing task.

Further the spatio-temporal visual attention model [8] predicts attention maps more accurately than state-of-the-art methods like SALICON [6], graph-based visual saliency (GBVS) [4] and Covsal [3].

To extend the research in the attention prediction for graphical user interfaces we first re-implemented the model in [8] and further extended the model by incorporating an LSTM approach to predict the Spatio-temporal visual attention. The proposed LSTM model predicts attention maps more accurately with lesser latency than our re-implemented version of the base model.

We believe the proposed LSTM model has a significant potential in Spatio-temporal attention prediction on WIMP style graphical user interfaces using mouse and cursor as input.

2 DATA ANALYSIS

Dataset used in [8] was provided by the department consists of data obtained from a user study with 11 participants on a display with a resolution of 1920×1200 px.

Mouse input is the recorded mouse movement which is the mouse *(x,y)* position. For keyboard input, we used the *(x,y)* position of the cursor was used which was only visible in the text input area. Human eye fixations are obtained from the Tobii TX300 eye tracker as fixation *(x,y)*. The position and timestamps of mouse and cursor per user, per task, are available.

In total, gaze and interaction data for 245 text editing tasks are available and we have 34,695 fixations, 3,884 mouse clicks, and 43,158 key-presses(not cursor movements). The average completion time for one task was 98.4 s [8].

2.1 Correlations

Table 1: Pearson product-moment correlation coefficient of eye fixations with mouse and cursor [8].

Signal	*x*	*lag x (ms)*	*y*	*lag y (ms)*
Mouse	0.35	114	0.51	193
Cursor	0.44	2	0.36	9
Combined	0.60	-	0.56	-

Table 1 shows positive correlation coefficients from [8] indicating that the eye fixation is correlated to the mouse and cursor movements.

2.2 AAUC patterns

3 frequent patterns of the interaction between attention (eye), action (mouse and keyboard) and user interface in the coordinate space (AAUC) were observed [8],

(1) mouse following the eye,
(2) eye focusing on cursor during text editing, and
(3) mouse remaining stationary while the eye inspected text content or UI components.

3 BASE PAPER RE-IMPLEMENTATION

The models in the base paper[8] was not available as off-the-shelf model so we re-implemented the visual attention prediction model. User attention is divided into static and dynamic depending on the time-factor. Explanation about the three AAUC pattern [8].

To extend the research in the attention prediction for Graphical user interface we first re-implemented the models mentioned in the earlier work. Two models are considered, Static attention prediction model - where the prediction is used to find the attention for static images - and Dynamic attention Prediction - where the prediction is used to find the attention for dynamic images.

3.1 DATA PREPROCESSING

The bounding box was created manually from the interfaces and the coordinates for each UI component in each interface was noted.

3.1.1 Static Saliency Prediction: The position co-ordinates from the data set was collected for mouse, cursor and fixations. The bounding box values corresponding to the mouse position is read from the bounding box image. The mouse, cursor and the bounding box values corresponding to a particular pixel position are taken as input. The fixation value for that particular pixel is taken as output.

3.1.2 Dynamic Saliency Prediction: For each user and each interface, the position coordinates from the data set were collected for mouse, cursor and fixation positions along with their respective timestamps. The positions of the mouse, cursor, and fixation at an intermediate time t are considered as the last observed position, this can be achieved by interpolation the data with respect to time for all the mentioned modalities, this helps in getting the respective mouse, cursor and fixation positions for a common timestamp. The timestamp was normalized to have a range from 0 to 1. Instead of collecting the data from all the pixel positions in an interface of size 1900x1200, around 200 data samples were selected in the range, further 200 samples were collected from random pixel positions in the interface.

The attention distribution with respect to mouse, cursor, UI and the corresponding fixation distributions are visualized in the figure 3

3.2 TRAINING

Linear Regression: Pixel wise linear regression models are used for the prediction for both static and dynamic prediction. The training is done on the selected pixel values for all the interfaces and half the users in the data-set. The rest half of the data-set is used for validation and testing.

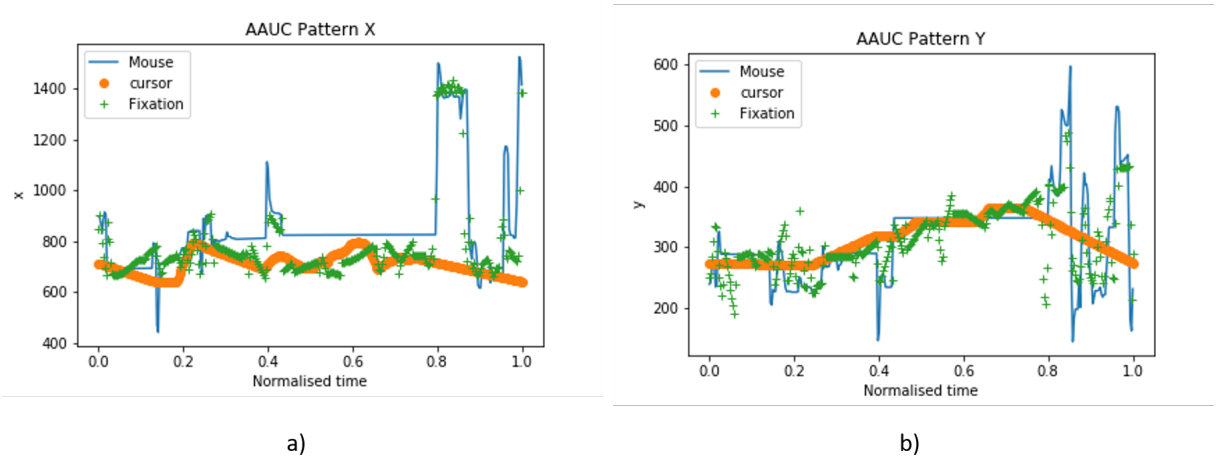


Figure 2: Three frequently occurring patterns(AAUC)[8]

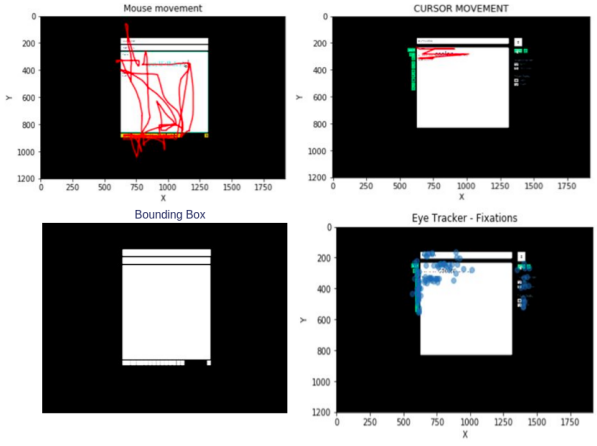


Figure 3: The sample plot of the mouse, cursor, bounding box and fixation points for a particular task and particular user

3.2.1 STATIC ATTENTION PREDICTION. The model predicts the static attention distribution of the user over the interface irrespective of the time. For each user and each interface, the values are aggregated over time and the input and output values are taken as distributions. The mouse, cursor and the bounding box input are blurred using a Gaussian filter of -6dB cut off frequency [8]. We set the standard deviation of Gaussian filter to 64 which corresponds to -6dB cut off frequency. The model is trained on the fixation distribution that is blurred with the same Gaussian filter.

The static prediction model focuses on the static attention prediction, while the dynamic prediction model focuses on the localization of attention points over time. There are two scenarios in this,

- one is the offline dynamic predictor - which predicts the dynamic attention offline
- the other is the online analysis - which predicts the dynamic attention in real-time, i.e when the user is using the GUI.

In this paper, we focus only on the offline dynamic prediction model.

3.2.2 DYNAMIC ATTENTION PREDICTION. In this model, the same linear regression, as station attention predictor, is used but this time instead of input and output distributions, the values are spread out in time. For each time t, the features extracted would be from the window [t-d to t+d], i.e there are 2d+1 input features maps. The output is the fixation value at time t. The input and the output values are also blurred so as to get a better correlation between input values and the fixation.

3.3 TESTING

3.3.1 STATIC ATTENTION PREDICTION. The trained model was tested on the testing data set - few users and few interfaces - using the mouse, cursor and the bounding box values of the interfaces the whole 1900x1200 images were constructed for the interface.

Figure 4 illustrates the static attention maps for a sample data. The ground truth attention distribution and the prediction attention distribution is visualized

3.3.2 DYNAMIC ATTENTION PREDICTION. Similar to the testing done for the static prediction model but as one additional step, the region of interest is considered to optimize the testing time.

3.3.3 Region of Interest. The testing time - linear regression over 1900x1200 pixels for each interface - is more, compared to the actual data contained in the input and the output. Although the input and the output data are localized in very few points on a 1900x1200 image, the prediction was done for all the pixel positions in the image. Most of the predictions done are on the redundant data and hence it would be efficient to reduce the redundancy and focus mainly on the region where the data lies. To solve the performance issue, a square window of size 100x100 is placed around each input feature - mouse and cursor positions. Both the square windows are merged to form a rectangular window whose co-ordinates start at the lowest co-ordinate and ends at the highest coordinate values of the two square windows combined. This becomes the region of

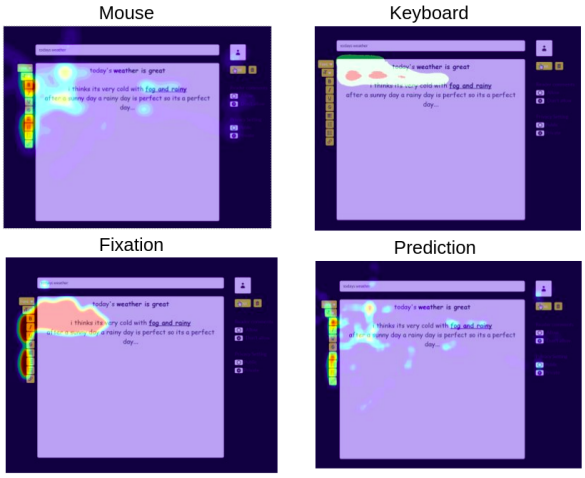


Figure 4: Static attention map of the mouse (Top left), the cursor (Top right), ground truth (Bottom left) and static prediction (Bottom right)

interest where the pixel-wise linear regression can be performed efficiently with lower redundancy and latency.

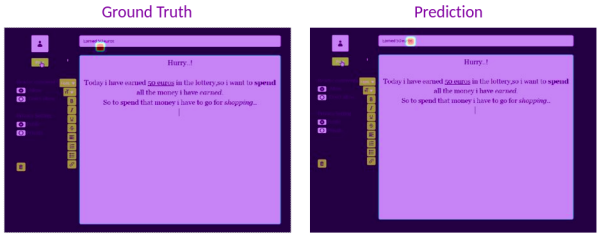


Figure 5: Dynamic attention prediction for the blogger task. The ground truth and the predictions are illustrated. The colored square on the UI highlights the attention over the UI

Figure 5 illustrates the dynamic attention maps for sample data. The ground truth attention points and the prediction attention points are visualized.

3.4 RESULTS OF RE-IMPLEMENTATION

Some of the standard evaluation metric for saliency prediction are as follows[2][8]:

(1) Normalized Scan-Path Saliency (NSS): This calculates the mean value of the normalized predicted saliency map s at n ground truth fixation locations: $NSS = \frac{1}{n}\sum_{i=1}^{n}\frac{s(x_i,y_i)\mu_s}{\sigma_s}$.
(2) Area Under ROC curve (AUC): The predicted saliency map is treated as a binary classifier. The values above the threshold are considered fixated and the values below are not non-fixated. A plot of true positives and false positives are made and the area under this curve is calculated. Many variants exist such as AUC Judd, AUC Borji, etc.
(3) Correlation Coefficient (CC): This is of measure of the correlation between predicted attention and user attention map. Uncorrelated maps have CC = 0. $CC(s,h) = \frac{cov(s,h}{\sigma_s\sigma_h}$.
(4) Similarity: This indicates a measure of the similarity between two distributions. The maps are first normalised to usm to 1, then $Similarity = \sum_i min(s_i, h_i)$.

According to [2] evaluation metrics for saliency are categorized into **location-based** and **distribution-based** depending on whether the ground-truth is represented as discrete fixation locations or a continuous fixation map respectively.The metrics are distinguished as follows: NSS and AUC are location-based, Similarity and CC are distribution-based.

For the static model, we consider the NSS, AUC Jud, AUC Borji, CC and Similarity metrics. The evaluation results along with the comparison to the base model are depicted in Table 2. For the dynamic prediction model since we have only one fixation point over time, we use a discrete fixation map for the evaluation of accuracy. So we consider only NSS and AUC Judd which are the location-based metrics. Evaluation results are depicted in the Table 2

Table 2: Evaluation for the base re-implemented static prediction model [8].

Method	*NSS*	*AUC-Judd*	*AUC-Borji*	*CC*	*Similarity*
Base Model	3.43	0.96	0.89	0.86	0.73
Re-Implementation	3.7	0.85	0.77	0.53	0.86

Table 3: Evaluation for the base re-implemented dynamic attention prediction model [8].

Method	*NSS*	*AUC-Judd*
Base Model	6.26	0.98
Re-Implementation	3.161	0.831

4 DEEP LEARNING PIPELINE

This section discusses the deep learning architecture, LSTM model which is the final working implementation for this project.

4.1 LSTM

Recurrent Neural Networks(RNNs) are a generalization of feedforward neural network that has internal memory. Unlike traditional neural networks in RNN, the outputs of some layers are fed back into the inputs of a previous layer. Long Short Term Memory networks (LSTMs) are a special kind of Recurrent Neural Networks which generally perform better in time series prediction problems. Therefore, the LSTM model is chosen for the Spatio-temporal attention prediction task.

The core idea of LSTM is the cell state which runs along with the entire LSTM unit and is capable of adding or removing information via three gates - input gate, output gate and forget gate.

Figure 6 illustrates an LSTM memory cell with sigmoid and hyperbolic tangent functions forming the gates.

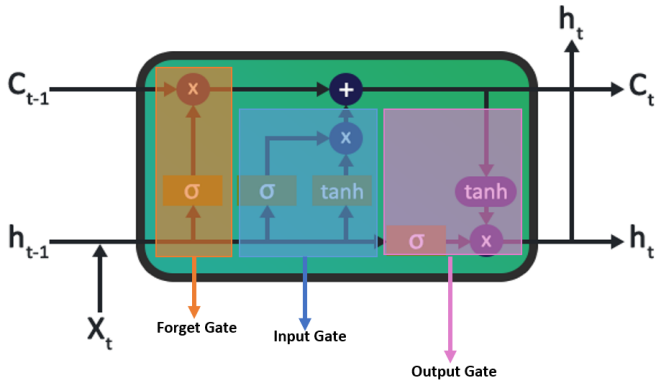


Figure 6: LSTM cell with gates

4.2 DATA PREPROCESSING

The following transforms are performed on the dataset prior to fitting the LSTM model and making predictions.

(1) The period for each user end each task of the user keeps varying. So all the time span is normalized in the range [0,1] as mentioned in [8].
(2) We need a uniform number of samples in mouse, cursor and fixation co-ordinates. But the data available is not uniform(large data points for mouse and sparse data points for the cursor, fixations). We linearly interpolate mouse, cursor and fixation positions and evenly sample 400 time points for each feature.
(3) It is important to feed data as a sequence of time points to the LSTM model. The time window is chosen as *[t+d]* . The window size *d* is treated as a hyperparameter and will be discussed in the next section.
(4) Transform the (x,y) coordinates to have values between 0 and 1. This is done by dividing x and y with the width(1920) and height(1200) of the GUI display.

4.3 Model Architecture

Given the stream of mouse and cursor observations at the previous *d* time steps, the LSTM model should predict the human gaze fixation at the current time step. Inputs to the network are the (x,y) coordinates from the mouse and cursor positions. The output of the network is the (x,y) coordinates of the eye gaze fixation.

The network consists of two LSTM layers with a dropout for the first layer acting as the regularizer. Finally, a dense layer outputs the coordinates x and y. The *mean square error(mse)* loss is used as it reduces the euclidean distance between ground truth and prediction coordinates. The commonly used activation functions (hyperbolic tangent and hard sigmoid for recurrent activation) are used in layers 1 and 2.

4.3.1 Hyperparameter Tuning. We use Bayesian hyperparameter optimization in order to obtain the most promising hyperparamter values for our model. The following are the hyperparameters optimized -

(1) number of neurons in each hidden layer (LSTM cells)
(2) dropout for first layer
(3) window size *d*

Table 4: Hyperparameters from Bayesian Optimization

Hidden layer 1	Hidden layer 2	Dropout	Window size
158	50	0.2085	60

Table 4 shows values of hyperparameters after Bayesian optimization. The window size obtained was 63, it has been approximated to 60 for computational convenience.

Figure 7 shows the complete model architecture along with input, output shape, loss and accuracy plots.

4.4 LSTM Model Results

Our LSTM model was trained on first half of the users and tested on the next half of users. Each user task has 400 evenly sampled data points after transforming to fit LSTM model.

4.4.1 ***Visual saliency Metrics.*** To find the accuracy of the LSTM model the visual saliency metrics discussed in section 3.4 are employed.

Table 5: Static attention prediction evaluation

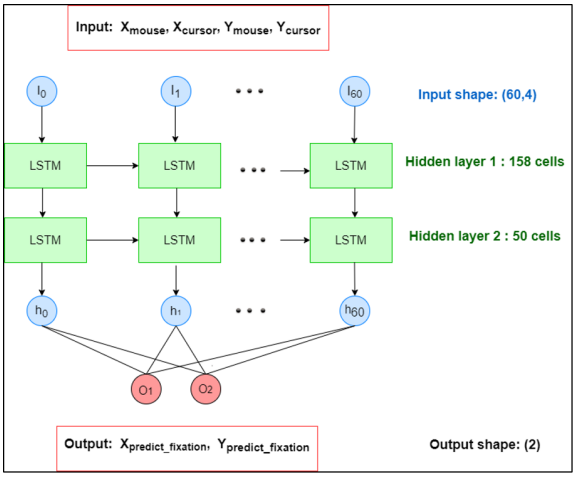
Method	*NSS*	*AUC-Judd*	*AUC-Borji*	*CC*	*Similarity*
GT	6.71	0.98	0.94	1.00	1.00
Mouse	2.75	0.88	0.86	0.48	0.84
Re-Implementation	3.7	0.85	0.77	0.53	0.86
Ours(LSTM)	**5.12**	**0.94**	**0.92**	**0.82**	**0.91**

4.4.2 ***Static Attention Prediction.*** We compare the LSTM model with base paper re-implementation.Table 2 and Table 5 provides the prediction scores of these models. The scores for GT (ground truth:human fixations) are considered as upper bound for all the scores.

It can be seen in Table 5 that our LSTM model performance is close to the human fixations than our re-implemented version of [8]. Our model has NSS of 5.12 (GT:6.71), AUC-Judd of 0.94 (GT:0.98) and AUC-Borji of 0.92 (GT:0.94). The CC (0.82) and Similarity (0.91) are slightly less than ground truth (1.00).

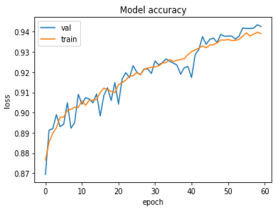
The LSTM model has the best performance for static attention prediction without considering the UI components(bounding boxes) as a feature. We believe this is due to the persistence of previous (sequence) inputs from the user interaction.

4.4.3 ***Dynamic Attention Prediction.*** The dynamic attention prediction model has a sequence of fixation points (x,y) over time as output. For evaluating the accuracy we consider the continuous map of a single predicted point i.e. convolution of discrete prediction point with Gaussian filter of standard deviation 64 and discrete location of the ground truth fixation. As discussed in section 3.4 this corresponds to location-based accuracy evaluation. So the metrics NSS and AUC-Judd are chosen to compare the performance. From

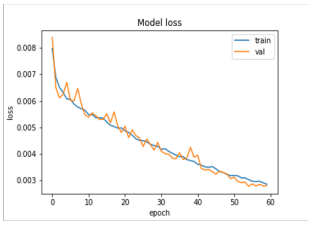


a) LSTM Architecture

Accuracy (%)		Loss (mse)	
Training set	93.91	Training set	0.0029
Validation set	**94.26**	Validation set	**0.0028**



b) Accuracy plot



c) Loss plot

Figure 7: LSTM model architecture

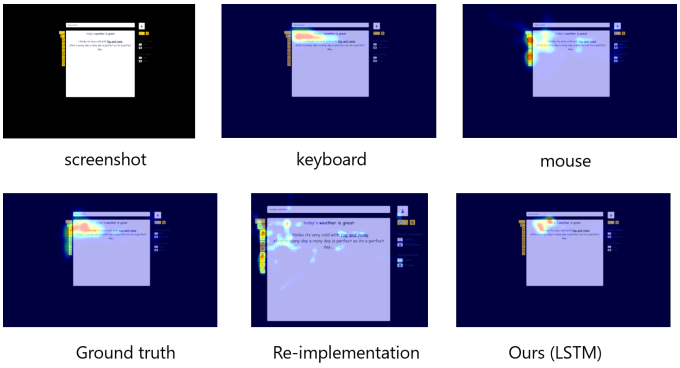


Figure 8: Re-implementation attention map resembles mouse attention map, LSTM predicts attention considering mouse and cursor movements.

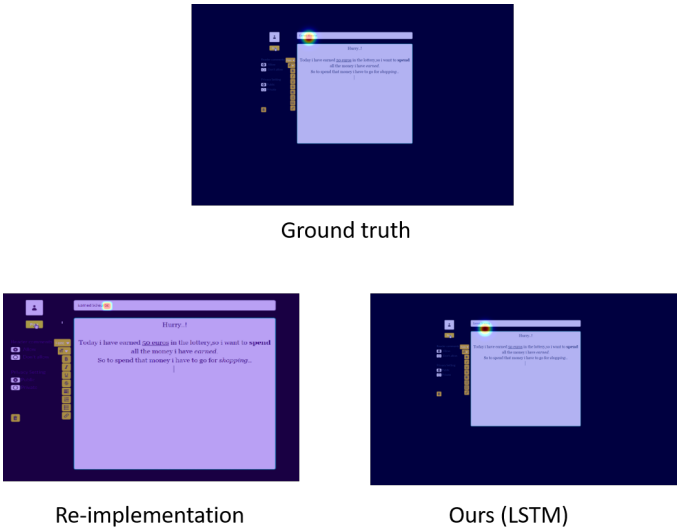


Figure 9: Re-implementation attention map with required region og interest'. LSTM attention prediction is close to human (ground truth)

Table 6: Dynamic attention prediction evaluation

Method	*NSS*	*AUC-Judd*	*AUC-Borji*	*CC*	*Similarity*
Mouse	3.11	0.69	0.68	0.19	0.10
Re-Implementation	3.16	0.83	0.82	0.0004	0.01
Ours(LSTM)	**7.05**	**0.84**	**0.83**	**0.004**	**0.23**

Table 6 it can be seen that the LSTM model performs best with NSS of 7.05 and AUC-Judd of 0.84.

We analyze the predictions for the frequent patterns of the interaction between attention (eye), action (mouse and keyboard) and user interface in the coordinate space (AAUC) [8] mentioned in section 2.2. [8] states that a prediction model should have the capacity to capture the three AAUC patterns. As illustrated in Figure 10, our LSTM model successfully achieves this goal by learning from previous observations.

5 DISCUSSION

In this work, we have proposed and implemented a deep learning LSTM model for Spatio-temporal attention prediction in an environment that consists of a graphical user interface and hence eliminated the usage of an eye tracker device. The initial models were inspired from the earlier work [8] where the static attention prediction and the dynamic attention prediction models were re-implemented and evaluated. The performance speed of the dynamic attention prediction model in the re-implementation was also improved by using the region of interest approach.

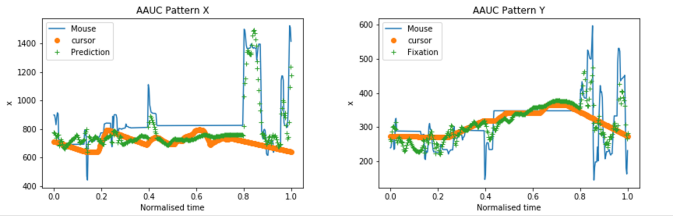


Figure 10: Predictions from LSTM model: 1) mouse follows the eye, 2) eye focuses on cursor during text editing, and 3) the mouse remains stationary while the eye inspects text or UI components.

We enhanced the attention modeling method from [8]by employing a deep learning LSTM model capable of handling time series data. We also transformed the dataset to fit into the LSTM model and hence we extended the temporal prediction task done in [8].

The LSTM model also performed more accurately for the static attention prediction even though the UI components(bounding boxes) were not fed as input. The model can be extended further by using the UI components as one of the input features and online prediction for dynamic attention by means of a user study.

The evaluation results for the extended model was promising compared to the base re-implementation version and also we believe it is a performance-enhancing approach for Spatio-temporal saliency prediction.

6 CONCLUSION

In this paper, we re-implemented the computational models from the earlier work that focused on Spatio-temporal visual attention on graphical user interface [8]. In order to reduce the computational complexity, we also introduced a method that identifies the region of interest for computation to be performed. We also enhanced the visual attention modeling of [8] by using an LSTM network. The model yielded the best accuracy in terms of evaluation metrics and faster results compared to the earlier re-implementation. We strongly believe that our work shall provide insights into Spatio-temporal saliency prediction for graphical user interfaces.

7 ACKNOWLEDGMENTS

REFERENCES

[1] Cagdas Bak, Aysun Kocak, Erkut Erdem, and Aykut Erdem. 2016. Spatio-Temporal Saliency Networks for Dynamic Saliency Prediction. *IEEE Transactions on Multimedia* 20 (2016), 1688–1698.

[2] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. 2016. What do different evaluation metrics tell us about saliency models? *CoRR* abs/1604.03605 (2016). arXiv:1604.03605 http://arxiv.org/abs/1604.03605

[3] Erkut Erdem and Aykut Erdem. 2013. Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of vision* 13, 4 (2013), 11–11.

[4] Jonathan Harel, Christof Koch, and Pietro Perona. 2007. Graph-based visual saliency. In *Advances in neural information processing systems*. 545–552.

[5] Laurent Itti. 2000. *Models of bottom-up and top-down visual attention.* Ph.D. Dissertation. California Institute of Technology.

[6] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. 2015. Salicon: Saliency in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1072–1080.

[7] A. Rahman, D. Houzet, D. Pellerin, S. Marat, and N. Guyader. 2011. Parallel Implementation of a Spatio-Temporal Visual Saliency Model. *J. Real-Time Image Process.* 6, 1 (March 2011), 3–14. https://doi.org/10.1007/s11554-010-0164-7

[8] Pingmei Xu, Yusuke Sugano, and Andreas Bulling. 2016. Spatio-Temporal Modeling and Prediction of Visual Attention in Graphical User Interfaces *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3299–3310. https://doi.org/10.1145/2858036.2858479

A DATASET

We used the dataset provided by the department. The dataset and all relevant data(images,files,etc.)used in this project are available in **gpuws2.perceptualui.org**:

- Main dataset in the path: **'/home/kkjk/Dataset'**

For privacy reasons, we have not shared the dataset or any data connected to the evaluation of results in the github repository.