Deep Scanpath: Predicting Human Sequences of Eye-Fixations using Recurrent Neural Networks

Mengmi Zhang†,3, Xavier Boix†,2, Keng Teck Ma3, Ming Jiang§,1, Joo Hwee Lim¶,3, and Qi Zhao∥1,4

1National University of Singapore, Singapore
2Massachusetts Institute of Technology, USA
3Institute for Infocomm Research, Astar, Singapore
4University of Minnesota, USA

Abstract

Recent computational models of visual attention are based on deep neural networks to predict saliency maps. These models adapt the neural networks for object recognition to saliency prediction, as they exploit the strong link between object recognition and visual attention. In this paper, we show that deep neural networks can make predictions beyond saliency maps, as these networks can also estimate the sequence of eye-fixations across time, i.e. the visual scanpath. We introduce a novel recurrent neural network, Deep Scanpath Neural Network (DSNN), which integrates the information from all the past eye fixations to predict the next fixation location. We evaluate DSNN in three challenging benchmark datasets. DSNN demonstrates an unprecedented scanpath prediction accuracy, while it obtains a competitive predictive accuracy of the saliency map with state-of-the-art models. Our analysis of the learnt model reveals that the recurrent connections in DSNN are effective to improve the predictive visual scanpath accuracy, and it also shows the emergence of a temporally changing spatial bias during the scanpath prediction.

1. Introduction

Humans cannot analyse an entire scene in details at once. Instead, we selectively attend on regions of the scene to allocate the processing resources, c.f. [1]. Computational models of visual attention are useful for predicting properties of human visual attention, engineering applications, e.g. in advertisement [33], and also other computer vision tasks, e.g. object tracking [47] and image compression [13].

Computational saliency models were pioneered by Koch and Ullman [24] and Itti et al. [18], and they originate from feature-integration theory [43]. Since then, a rich set of models have been introduced to predict saliency maps, e.g. [48, 14, 4, 51]. A remarkable feat was the use of object detectors for saliency prediction [10], which boosted the accuracy of the saliency maps due to the tight association between saliency and object recognition [11]. This
further leads to the use of machine learning techniques to combine multiple detectors of different objects and features [22, 53] for better saliency prediction. However, these models can only fit to datasets of images in very controlled settings due to the limited number of object detectors.

For the past few years, the advent of deep learning has opened a new avenue for computational models of visual attention by replacing handcrafted features with the features learnt from raw images [45]. Current state-of-the-art methods of saliency prediction are built upon deep neural networks for object recognition, and they have substantially improved saliency prediction accuracy [16, 31, 27].

Bylinskii et al. analyzed the limitations of neural network models for saliency prediction [9] and addressed the need to extend beyond saliency. A promising strand of research is based on enriching the computational models’ capabilities by capturing more sophisticated properties of visual attention other than saliency maps. One recent work in that direction extends saliency prediction on static images to videos [2]. Volokitin et al. also proposes an interesting problem of estimating the consistency of the eye fixations among different subjects in a given image [46].

Following this strand of research, in this paper, we show that deep neural networks can also be adapted to predict the temporal sequence of eye-fixations across time, i.e. the visual scanpath. We present the first recurrent neural network (RNN) for scanpath prediction, Deep Scanpath Neural Network (DSNN). DSNN learns to integrate the information from all the past eye fixations to predict the next fixation location.

Our results in three challenging benchmark datasets show that DSNN successfully extends the capacity of current state-of-the-art saliency models by substantially outperforming state-of-the-arts in scanpath prediction meanwhile obtaining satisfying saliency prediction accuracy on par with state-of-the-art saliency models. In order to decipher which components of the neural network contribute to the improvement of the accuracy, we analyze DSNN with visualization and ablation study. The analysis shows that the recurrent connections are effective for scanpath prediction. In particular, they learn the emergence of a temporally changing spatial bias via the dynamics of the neural network.

2. Related Work on Scanpath Prediction

In this section, we review the literature on visual scanpath prediction on still images. To the best of our knowledge, there are no deep neural networks for visual scanpath prediction. Some early works applied inhibition of return on a saliency map in order to generate a sequence of eye-fixations [23, 48, 14, 4, 51]. Yet, inhibition of return alone does not capture many characteristics of the temporal sequence of eye fixations. To better model the scanpath, there was an early work built upon the hypothesis that the scanpath is generated based on the principle of information maximization [30]. More recently, scanpath prediction models have adopted statistical approaches. Sun et al. introduces the super-Gaussian component (SGC) approach [41] and Liu et al. improves the accuracy by exploiting semantic information and learning the transition between fixations [32]. Moreover, several current methods exploit reinforcement learning on scanpath prediction [34, 19] where a very limited set of weights are assigned to hand-crafted features in order to predict fixation locations at each fixation stage of the visual scanpath. However, their predictive scanpath accuracy is severely limited.

3. Deep Scanpath Neural Network

In this section, we introduce the formulation of our model, Deep Scanpath Neural Network (DSNN), for scanpath prediction problem.

Visual memory plays a crucial role in scanpath prediction, as it allows fusing information of different fixations over time, c.f. [39]. Recurrent Neural Network (RNN) are one of the promising neural network models that effectively learns to process information across time using memory-based mechanisms. Compared to hidden Markov model [37] which can also model dynamic states, the memory-based mechanisms of RNN are learnt automatically from raw images directly. This characteristics of RNN alleviates the strong dependence on the handcrafting process and provides us more flexibility to build more sophisticated models compared with hidden Markov models.

The architecture of DSNN is built upon RNN. Though
there are also RNN-based visual attention models, e.g. [35, 50], none of these address human visual scanpath prediction problem as they are used for applications such as object recognition. In contrast, DSNN learns to mimic the human visual scanpath behaviors.

In the following, we introduce an overview of the architecture of DSNN, and the different modules in which it is divided. We then describe the learning procedure and the implementation details.

3.1. Overview

We define the fixation stages as the order in the sequence of fixations. The fixation stage of a scanpath discards the duration of the fixation and the saccade, and only takes into account the location and the order of the fixations. We use fixation stages rather than time to describe the scanpath. The aim of our paper is to analyse the temporal sequence of fixations rather than the duration of the fixation. We use $t \in \mathbb{N}$ to index the different fixation stages, and we define $T$ as the total number of fixation stages in which the scanpath is divided.

We formulate the scanpath prediction problem as an iterative process learnable by DSNN. At the fixation stage $t$, DSNN predicts the fixation $(x_t, y_t)$ given the image $I$ and the fixation location $(x_{t-1}, y_{t-1})$ that has been predicted at the fixation stage $t - 1$.

Note that humans subjects may have different visual scanpaths while looking at one static image. In order to handle the inconsistency among human visual scanpaths, we align the fixations of the human subjects using the fixation stages. In other words, we align the fixation stage of the visual scanpath of two human subjects using the order of the fixation even though the fixations may happen at a different time. Rather than directly predicting the fixation location, DSNN predicts a temporal saliency map that captures the scanpath variability among humans, i.e. the probabilit-
tic map of the fixation locations across time. This temporal saliency map is used to predict a representative fixation location at each stage by using the spatial coordinate corresponding to the maximum of the temporal saliency map at stage $t$. This process is summarized in Figure 2.

We define the standard 2D convolution operation as $\text{conv}$, the fully connected operation as $\text{fc}$ and the linear rectifier function as $\text{relu}$ in deep learning. We use 0-padding in all convolutions in order to maintain the spatial resolution. Refer to [29] for the details of these operations.

DSNN is built on RNN as shown in Figure 3. DSNN comprises of two parts: GazeModule, and RecurrentModule. GazeModule is based on the deep Convolutional Neural Network (CNN) architectures for object recognition as it has been shown to be effective to predict saliency maps. The GazeModule also uses a mechanism to mimic inhibition of return that discourages DSNN to explore the already visited locations [23]. We implement inhibition of return at time $t$, which is of size $H \times W$. $S_t$ is defined as function $g$ dependent on the previous predicted fixation $(x_{t-1}, y_{t-1})$. Specifically, we choose $g$ to be an inverted gaussian mask centered at $(x_{t-1}, y_{t-1})$ with variance $\sigma$ and normalized to $[0, 1]$. The low intensity values on $S_t$ near to $(x_{t-1}, y_{t-1})$ indicate the low probability for DSNN to explore in that location. Let $F^2_t$ be the feature maps after applying the inhibition of return on $F^1_t$. $F^2_t$ is defined as

$$F^2_t = F^1 \odot g(x_{t-1}, y_{t-1}) \quad (1)$$

where $\odot$ represents element-wise product over each feature map of $F^1_t$, and hence, $F^2_t$ has the same dimensions as $F^1_t$.

We can see by analyzing Eq. (1), that the $H \times W$ feature maps in $F^1_t$, which encode the semantic content among the image, are multiplied by values close to 0 in the location of the previous fixation. As a result, the feature maps of $F^2_t$ encode less salient content in the previous fixation location.

### 3.3. Recurrent Module

The RecurrentModule is attached after GazeModule for modeling the dynamics of the scanpath. RecurrentModule has two recurrent layers, recurrent convolution layer (RC) and recurrent fully connect layer (RF). Let $M^1_{t-1}$ be the hidden state of RC at fixation stage $t$, and let $M^2_t$ be the hidden state in RF at fixation stage $t$. Between the two recurrent layers, there are other conv and fc layers that we introduce in the following.

First, $F^2_t$ is the same size as $F^1_t$. The recurrent layer RC then integrates the past feature maps stored in the memory, $M^1_{t-1}$, with the feature maps $F^2_t$ at fixation stage $t$. Let $F^3_t$ be the output of RC. $M^1_{t-1}$ and $F^3_t$ are of the same size as $F^2_t$. RC integrates the feature map $F^2_t$ with the memory $M^1_{t-1}$ using the element-wise addition $\oplus$. Thus, we define $F^3_t$ as

$$F^3_t = \text{relu}(F^2_t \oplus M^1_{t-1}). \quad (2)$$

Instead of memorizing all the output feature maps $F^3_t$, we use conv to learn how to selectively store these features in the memory $M^1_t$. A small weight in the convolution filter indicates that the corresponding feature map in $F^3_t$ is easier to forget. Thus, the memory $M^1_t$ at fixation stage $t$ is updated as

$$M^1_t = \text{conv}(F^3_t). \quad (3)$$

A conv operation is applied to $F^3_t$ to obtain the feature map $F^4_t$, which is of size $H \times W$ with the number of feature maps to be 1. We then use fc to transform the feature information $F^4_t$ into the latent representation of the following layer denoted as $F^5_t$. This latent representation in $F^5_t$ is a vector of length $D$ where $D = H \cdot W^2$.

Similar to RC, RF uses an element-wise addition to integrate $M^2_{t-1}$ with $F^5_t$, and obtains the output of RF, denoted as $F^6_t$. $M^2_{t-1}$ and $F^5_t$ have the same dimension as $F^5_t$, i.e. $D = H \cdot W$. Thus, $F^6_t$ is defined as

$$F^6_t = \text{relu}(F^5_t \oplus M^2_{t-1}). \quad (4)$$

$^2(\cdot)$ is the scalar multiplication.
Also, instead of storing $F_t^6$ directly in the memory, we use $fc$ to tune its latent representation, i.e.

$$M_t^2 = fc(F_t^6).$$

In Section 4, we show that this $fc$ in $RF$, in fact learns a changing spatial bias across fixation stages.

Finally, the integrated latent representation $F_t^6$ is decoded into $F_t^7$ using $fc$. $F_t^7$ is again of dimension $D = H \cdot W$. Since this is equal to the spatial domain, $F_t^7$ can be used as the temporal saliency map before normalization. The spatial coordinate with the maximum probability from the temporal saliency map, i.e. $F_t^7$, is taken as the predicted fixation location $(x_t, y_t)$ at fixation stage $t$. In the next iteration, i.e. fixation stage $t+1$, DSNN feeds back $(x_t, y_t)$ as input together with image $I$ to predict the subsequent fixation location $(x_{t+1}, y_{t+1})$. It is a sequential process. Hence, DSNN predicts the scanpath by generating a sequence of fixations across time.

### 3.4. Training

We train our network using end-to-end back-propagation in a fully supervised manner, i.e. all the parameters in our network are trained jointly.

We generate the ground truth data to learn the model by aligning all the fixations from all human subjects with the fixation stages. The aligned eye fixations from all human subjects produce a sparse fixation map, we put gaussian mask with variance $\sigma$ over these maps to generate temporal fixation maps.

Let $P_t$ be the temporal fixation map (the ground truth) and $Q_t$ be the estimated temporal saliency map by DSNN at fixation stage $t$. The goal of the learning is to minimize a loss function between these two probability distributions across time. We use Kullback-Leibler divergence (KLD) loss function which has been shown to be one of the most effective loss functions for achieving the best saliency prediction [16]. In our case, we average it across the fixation stages, i.e. the loss function is

$$\text{KLD}(P, Q) = \sum_{i} \sum_{t} P_t(i) \log \left( \frac{P_t(i)}{Q_t(i)} \right)$$

where $i$ refers to the $i$th pixel on the maps $P_t$ and $Q_t$.

We use stochastic gradient descent with learning rate fixed at 0.001 and batch size 1 to avoid being trapped in the local optimum. We stop the training at the turning point where we achieve the best scanpath prediction performance in the validation set, before there is over-fitting. Within each epoch, we randomize the sequence of inputs to the network. We train the network in a single NVIDIA Titan GPU with 12 GB memory.

### 3.5. Parameters of the Model

DSNN can predict sequences of eye-fixations for any number of fixation stages $T$. In our implementation and all the following experiments, we fix the number of stages to be $T = 6$. This choice produces an approximate correspondence between a fixation stage and the mean duration of an eye fixation (300ms), as the eye fixation recordings in the datasets used in the experiments are of duration between 1.5 to 2.5 seconds per image (i.e. 2s/6stages = 333ms/stage). This choice is also made in accordance with previous works, e.g. [32, 19].

The parameters of the first 30 layers in GazeModule are preloaded from the first 30 layers in VGG16 [40]. These parameters are fine-tuned to scanpath prediction during learning.

The input image size is 300 $\times$ 400 with RGB channels. All the input images are normalized into $[0, 1]$. $F^0$ is denoted as the output from the 30th layer of VGG16, thus, $F^0$ has 512 feature maps with each feature map of size 19 $\times$ 25, i.e. we set $H_0 \times W_0 = 19 \times 25$ and the number of features $K_0 = 512$. After one upsampling layer to scale up the size of each feature map from $F^0$ and one conv layer to increase the pool of feature maps $F^0$, $F^1$ has 1024 feature maps with each feature map of size 38 $\times$ 50. That is, we set $H \times W = 38 \times 50$ and the number of features $K = 1024$ to maximize the rich representations of features extracted from VGG16 while maintaining a proper spatial resolution.

In the spatial prior maps for inhibition of return, and the temporal fixation maps, we empirically fix $\sigma$ to be 5 with respect to the size of the temporal saliency map 38 $\times$ 50, which is of the same size as the feature maps extracted from VGG16 after one upsampling layer and one conv layer. We set the width and the height of all the conv filters to be $1 \times 1$ in the last conv layer in GazeModule as well as all the conv layers in RecurrentModule. This is to assign a probability indicating how salient the response for each coordinate on the feature maps is. Detailed architecture of DSNN is provided in Supplementary Material.

### 4. Experiments

In this section, we evaluate DSNN for scanpath and saliency prediction, and finally, we analyze properties of the learnt representation by the neural network.

### Datasets

All the datasets are collected from [7], which include CROWD500 [20], MIT1003 [7], MIT2000 [7], FIGRIM2787 [6], KTH101 [26], LeMeur27 [28], VI-U800 [25], OSIE700 [49], NUSEF760 [38] and Toronto120 [5]. In these datasets, the number of subjects per image vary from 7 to 104 depending on the datasets, and the subjects look at the images under the free-viewing conditions.
We use 3 different testing sets: 501 randomly chosen images from MIT1003, and 350 randomly chosen images from OSIE700 and all images from NUSEF760. For training, we use the training sets of all the aforementioned datasets, excluding all the testing images. We have about 9000 images in total for training.

We learn two different models of DSNN. In order to check for any dataset bias, the first model is tested in two testing sets, which are the ones from MIT1003 and NUSEF760. The second model is tested in OSIE700. For validation sets, the first model uses the test set of OSIE700, and the second model uses the 501 images randomly selected from MIT1003.

### 4.1. Scanpath Prediction

We now evaluate DSNN based on the standard evaluation metric SS in scanpath prediction.

**Evaluation Metric** Sequence score (SS) is proposed by Borji et al. [3], and it has been used to evaluate the accuracy of scanpath in the literature. We use the implementation by Jiang et al. [19]. SS computation is summarized: a mean-shift clustering for all human fixations is computed, and a unique character is assigned to each cluster center and corresponding fixations. The Needleman-Wunsch string match algorithm [36] is implemented to evaluate the similarity between human scanpath and the predicted scanpath.

**Comparative Methods** For evaluation purposes, we provide a few comparative methods as below:

- **MeanHuman**: is the mean SS among pairs of sequences of human scanpath for all images.
- **BestHuman**: is obtained by taking the averaged SS of all the best subjects for all images. The best subject for each image is defined to have the maximum averaged SS across all fixation stages among all human pairs. This is the “gold standard” for scanpath prediction.
- **Winner-take-all from Saliency Maps**: It generate scanpath from saliency maps with inhibition of return [18, 48]. During testing, we include the following saliency models: Graph-based Visual Saliency (GBVS) [14], Saliency Using Natural Statistics (SUN) [52], Adaptive Whitening Saliency (AWS) [12], Attention based on Information Maximization (AIM) [4], Itti’s Model (Itti) [17], Image Signature Saliency (ImSig) [15], and SALICON [16].
- **Previous Scanpath Models**: We also compare our results with the previous methods for scanpath prediction: Least Squares Policy Iteration (LSPI) [19], Support Vector Machine (SVM) to combine the features at each fixation stage as in [19] ³, Hidden Markov Model from Liu (LiuICCV) [32] and Super Gaussian Component (SGC) [41].

These models have been reviewed in Section 2. We used the implementation of all these models from [19].

- **Previous Scanpath Models With Deep Features**: For a fair comparison, we re-implement previous models and augment them with the deep learning features. We use LSPI (DeepLSPI) and SVM (DeepSVM) algorithms and extract the last convolution layer of SALICON as feature inputs to these algorithms. For DeepLSPI and DeepSVM, the number of superpixels is set to be 300. The rest of parameters remain the same as [19].

- **Center Bias**: To explore the effects of spatial biases, we create artificial fixation sequence with each fixation always in the center.

**Results** We show the SS scores for the comparative analysis on MIT1003 in Figure 4a, OSIE700 in Figure 4b and NUSEF760 in Figure 4c. DSNN generalizes well across all three datasets. It outperforms state-of-the-art models, substantially reducing the gaps between machine and human. In particular, DSNN prediction surpasses MeanHuman on NUSEF760 (note that this is not surprising as NUSEF760 includes provocative and controversial images and the consistency of the eye fixations among subjects might be low).

To quantify how much DSNN has improved the state-of-the-art results, we report the mean difference between the SS score of DSNN and the second best algorithm, in percentage with respect to DSNN, i.e.,

\[
A(SS^r) = \frac{1}{T} \sum_t \frac{SS_t^{DSNN} - SS_t^r}{SS_t^{DSNN}},
\]

where \(SS_t^{DSNN}\) is the SS score of DSNN, and \(SS_t^r\) is the SS score for the second best algorithm (LSPI on MIT1003, LSPI on OSIE700 and Center Bias on NUSEF760 respectively). They are 10.5% on MIT1003, 3.6% on OSIE700, and 21.4% on NUSEF760.

We observe that using deep learning features boosts the performance of all algorithms. This is because hand-crafted semantic features may not be sufficient to cover the wide range of salient objects. It is also shown that the algorithms which model the temporal information perform better than conventional saliency prediction methods with inhibition of return. For example, we observe that SALICON, which is based on deep neural networks, takes advantage of abundant semantic features, it cannot perform well for the first few fixations as it does not model the eye fixation temporal dynamics.

Finally, one example of our predicted scanpath is shown in Figure 1. More examples on predicted scanpath are provided in Supplementary Materials.

### 4.2. Saliency Prediction

We have shown that DSNN surpasses state-of-the-art algorithms on scanpath prediction. In order to show that D-
SNN in fact extends the predictive power of current state-of-the-art saliency predictors, we now show that DSNN can recover the accuracy of the most accurate saliency predictors in the literature. To convert the scanpath obtained with DSNN to a saliency map, we simply average the predicted temporal saliency maps.

Evaluation Metrics We use several common evaluation metrics. The fixation map is based on all human subjects, and we evaluate across all images in the testing datasets.

- Area Under The Curve (AUC) [3]: It is the area under a curve of true positive versus false positive rates under various discrimination threshold values on saliency maps.
- Shuffled-AUC (sAUC) [42]: It compensates the center bias problem of AUC. Instead of uniformly sampling at random, it gets negative samples from other images.
- Normalized Scanpath Saliency (NSS) [42]: It computes the normalized saliency at fixation locations. It is sensitive to false positives and relative difference in saliency. Given a normalized saliency map \( P \), and a binary fixation map \( Q \) with \( N \) fixations located at \( i \)th pixel (\( i \) is from 1 to \( N \)), NSS is defined as

\[
NSS(P, Q) = \frac{1}{N} \sum_i P(i) \times Q(i).
\]

-Correlation Coefficient (CC) [21]: treats saliency map \( P \) and a binary fixation map \( Q \) as random variables. It computes the linear relationship between them. This is useful in the context of scanpath analysis where relative saliency values at different image regions are concerned [8].

\[
CC(P, Q) = \frac{\text{cov}(P, Q)}{\text{cov}(P, P) \times \text{cov}(Q, Q)}
\]

where \( \text{cov}(\cdot) \) is the covariance of two random variables

Comparative Methods We compare the saliency map with the state-of-the-art saliency prediction algorithms as introduced in Section 4.1. We use the source codes from the authors, and follow the same parameter settings for saliency map prediction as [19]. Besides these saliency prediction algorithms, we also include the center bias.

Results Table 1 shows that the accuracy of the saliency map by DSNN is comparable with the state-of-the-art models. Also, DSNN is best in AUC, NSS and CC, while it is almost as good as SALICON in sAUC over the three datasets. One possible reason for lower sAUC is that DSNN predicts temporal saliency maps which have strong center bias for the first few fixations, but sAUC gives more credit to off-center information [8].

4.3. Model Analysis

In this subsection, we provide an analysis of DSNN via the ablation tests and the visualization of the hidden states in the recurrent module.

Ablation study We report the performance in Table 2 after removing various components in DSNN to study their effects in SS on scanpath prediction. DSNN is relevant for each case in the ablation study. These ablated models are:
Table 1. Quantitative Results in Saliency Prediction. We compare our model with all 7 saliency prediction algorithms and center bias using standard metrics Area Under the Curve (AUC), Normalized Scanpath Saliency (NSS), Correlation Coefficient (CC), shuffled AUC (sAUC) on MIT1003, OSIE700, NUSEF760 datasets respectively. All the algorithms are introduced in Subsection 4.1. Number denoting in bold is the best.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>MIT1003 Dataset</th>
<th>OSIE700 Dataset</th>
<th>NUSEF700 Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>NSS</td>
<td>CC</td>
</tr>
<tr>
<td>DSNN(ours)</td>
<td>0.82</td>
<td>1.90</td>
<td>0.46</td>
</tr>
<tr>
<td>SALICON [16]</td>
<td>0.80</td>
<td>1.51</td>
<td>0.36</td>
</tr>
<tr>
<td>CenterBias</td>
<td>0.78</td>
<td>1.05</td>
<td>0.27</td>
</tr>
<tr>
<td>GBVS [14]</td>
<td>0.77</td>
<td>1.21</td>
<td>0.29</td>
</tr>
<tr>
<td>AWS [12]</td>
<td>0.74</td>
<td>1.07</td>
<td>0.26</td>
</tr>
<tr>
<td>AIM [4]</td>
<td>0.72</td>
<td>0.86</td>
<td>0.21</td>
</tr>
<tr>
<td>SUN [52]</td>
<td>0.69</td>
<td>0.80</td>
<td>0.19</td>
</tr>
<tr>
<td>Itti [17]</td>
<td>0.63</td>
<td>0.55</td>
<td>0.13</td>
</tr>
<tr>
<td>ImSigLab [15]</td>
<td>0.55</td>
<td>0.55</td>
<td>0.13</td>
</tr>
</tbody>
</table>

1) R(ReFC): DSNN with the recurrent fully connected layer (RF) removed; 2) R(ReConv): DSNN with the recurrent convolution layer (RC) removed; 3) R(ReConv): DSNN with two recurrent layers replaced with the convolution layer and the fully connected layer respectively. Also, we include other variants based on the saliency map of SALICON with different spatial bias: 4) SAL(1stCB): the predicted visual scanpath of SALICON with the first fixation in the center; 5) SAL(SP): the spatial prior map pre-computed from human scanpaths at each fixation stage. The predicted saliency map from SALICON multiplied with the spatial prior map at each fixation stage, and then, inhibition of return is applied on these maps.

In Table 2, we show results for the test set from MIT1003. We also report the relative performance compared to DSNN (Row 1 in Table 2) as defined in Eq. (7). We find that removing any of the recurrent connections (R(ReFC), R(ReConv), and R(ReAll) in Table 2) reduces the accuracy of DSNN. This demonstrates that recurrent connections in DSNN are essential in learning the dynamics of the eye fixations.

The experiments with SALICON with different spatial bias (SAL(1stCB) and SAL(SP) in Table 2), obtain lower accuracy than DSNN. This shows that DSNN learns representations that are more useful than a spatial bias for deep learning features.

Visualization of Hidden States In order to better understand the role of recurrent modules in DSNN, we provide a visualization method of the hidden state in RF by converting it to the spatial domain. T-Distributed Stochastic Neighbor Embedding (t-SNE) [44] is used for dimension reduction and clustering. We visualize the latent representations of the hidden state in RF over the first 6 fixation stages \((t = 1, ..., 6)\) from 501 images in MIT1003, i.e. 3006 latent representations of the hidden state.

In Figure 5, we show the visualization result of the hidden states in RF. We use a different color to denote the different stages \(t\), i.e. dots with the same color are from different images at the same fixation stage. We observe that the hidden states at the first and second fixation stages form cluster A and B respectively, while cluster C contains the hidden states at the remaining 4 fixation stages. By analysing the pattern of these hidden states, we find that there exists a strong center bias for the first fixation. The latent representations in the hidden state shows higher activation to the surroundings as the fixation stages increase. At the 6th fixation stage, the spatial prior becomes more spread-out. This suggests that DSNN can emulate human visual scanpath behaviors by focusing attention on the salient objects nearest to the center at initial stages, and moves on to surrounding salient objects at later stages.
5. Conclusion

In this paper, we go beyond the current deep neural network-based saliency map prediction and extend it to visual scanpath prediction. We introduced DSNN, the first RNN on scanpath prediction. It integrates the sequence of fixations to estimate the temporal saliency maps, and it makes decisions on where the human subjects may look next. In addition to substantial improvements on scanpath prediction compared with the state-of-the-arts, DSNN also obtains a competitive predictive accuracy of the saliency map with state-of-the-art models. Our analysis on the learnt model demonstrates the utility of recurrent connections in the predictive scanpath accuracy and the emergence of a temporally changing spatial bias during the scanpath prediction.

References

[27] Y. Lin, S. Kong, D. Wang, and Y. Zhuang. Saliency detection with a deep convolutional architecture. 2
[46] A. Volokitin, M. Gygli, and X. Boix. Predicting when saliency maps are accurate and eye fixations consistent. 2
[50] K. Xu. Show, attend and tell: Neural image caption generation with visual attention. 3