Attentional Push: A Deep Convolutional Network for Augmenting Image Salience with Shared Attention Modeling in Social Scenes

Siavash Gorji James J. Clark
Centre for Intelligent Machines, Department of Electrical and Computer Engineering, McGill University
Montreal, Quebec, Canada
siagorji@cim.mcgill.ca clark@cim.mcgill.ca

Abstract

We present a novel visual attention tracking technique based on Shared Attention modeling. By considering the viewer as a participant in the activity occurring in the scene, our model learns the loci of attention of the scene actors and use it to augment image salience. We go beyond image salience and instead of only computing the power of image regions to pull attention, we also consider the strength with which the scene actors push attention to the region in question, thus the term Attentional Push. We present a convolutional neural network (CNN) which augments standard saliency models with Attentional Push. Our model contains two pathways: an Attentional Push pathway which learns the gaze location of the scene actors and a saliency pathway. These are followed by a shallow augmented saliency CNN which combines them and generates the augmented saliency. For training, we use transfer learning to initialize and train the Attentional Push CNN by minimizing the classification error of following the actors' gaze location on a 2-D grid using a large-scale gaze-following dataset. The Attentional Push CNN is then fine-tuned along with the augmented saliency CNN to minimize the Euclidean distance between the augmented saliency and ground truth fixations using an eye-tracking dataset, annotated with the head and the gaze location of the scene actors. We evaluate our model on three challenging eye fixation datasets, SALICON, iSUN and CAT2000, and illustrate significant improvements in predicting viewers' fixations in social scenes.

1. Introduction

Modeling visual attention has attracted much interest recently and there are several frameworks and computational approaches available. The current state-of-the-art of attention prediction techniques are based on computing image salience maps, which provide, for each pixel, its probability of attracting viewers' attention and have often been characterized by how well they predict eye movements. Almost all attention models are directly or indirectly inspired by cognitive findings and traditionally, they are based on hand-crafted features emerging from neuroscience studies. The basis of many attention models dates back to Treisman and Gelade's feature integration theory [34], Koch and Ullman's [18] feed-forward neural model and Clark and Ferrier's [9] computational demonstration of the link between image salience and eye movements. The first complete implementation of the Koch and Ullman model was proposed in the pioneering work of Itti et al. [13] which inspired many later models and has been the standard benchmark for comparison.

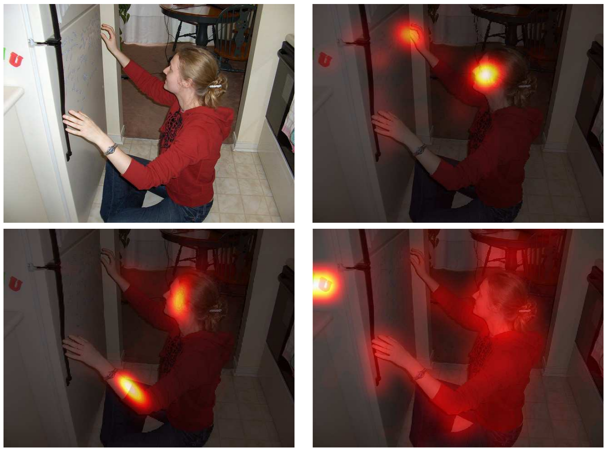

Figure 1: Input Image (top left) and ground-truth fixation heat map (top right), eDN saliency [35] (bottom left) and BMS saliency [37] (bottom right).

The recent publication of large-scale fixation datasets has motivated many saliency models based on convolutional neural networks (convnets), which have made quite significant improvements over traditional saliency models which are mostly designed using hand-crafted features. This trending increase in performance improvement of convnet-based saliency models seems to have saturated the prediction performance to the extent that further improvements require new and deeper insights into the concept of saliency-based attention tracking [7]. One of the shortcomings of the standard saliency-based attention tracking approaches is that, for the most part, they concentrate on analyzing regions of the image for their power to attract attention. However, in many instances, a region of the image may have low salience, but nonetheless still have attention allocated to it. Clearly, in such cases there are no salient features that attract viewers' attention to these regions other than the manipulating effect of higher-level concepts in the image. This suggests that in building an attention model we should go beyond image salience and instead of only computing the power of an image region to *pull* attention to it, we should also consider the strength with which other regions of the image *push* attention to the region in question.

Our proposed method models the viewer as a passive participant in the activity occurring in the scene. While viewers cannot affect what is going on in the scene, their attentional state can nonetheless be influenced by the scene actors. We will treat every image viewing situation as one of *Shared Attention*, which is the process by which multiple agents mutually direct and follow each others attentional state [17]. The goal of an agent in a shared attention setting is to coordinate its attention with other agents. While shared attention usually requires both agents to be able to manipulate and understand the attentional state of the other agent, our particular situation is a restricted asymmetric form of shared attention, in that the viewer has no control over the attentional state of the actors in the imagery. However, the actors in the image are assumed to have some control over the attentional state of the other actors in the image, as well as that of the viewer. Our working assumption will be that the attentional locus, i.e. the gaze location of each actor in a scene compels the viewers to direct their attention to that region, even if it has low salience by the scene actor's gaze. Figure 1 demonstrates an example in which the viewers' attention is pushed to an image region with low saliency. The figure compare the performance of two of the best-performing saliency models (according to the MIT saliency benchmark), eDN [35] (neural network-based) and BMS [37] (non-neural network), with the ground truth fixation heat maps. Clearly both methods perform poorly in predicting the viewers' attention. To improve on the performance of these methods, we need to also track the attention of the scene actors, and use this to augment the saliency. We use the term *Attentional Push* [30] to refer to the power of the scene actors to direct and manipulate the attention allocation of the viewer. Although there are other reported Attentional Push cues in the literature (see Section 6), in this work, we focus on the most prominent of these, i.e. the actors' gaze.

We propose a model that learns to follow the gaze location of the scene actors and augments saliency models with the Attentional Push effect of the actors' gaze in social scenes (everyday scenes depicting human activities). Instead of designing a saliency model from scratch, we purposefully use pre-built saliency models to illustrate that even the state-of-the-art in saliency models, either built from hand-crafted features or complete data-driven models based on convnets, can still benefit from the manipulating effect of Attentional Push. We present a deep convolutional neural network which augments saliency models with Attentional Push. Our network contains two pathways: a saliency pathway, which embeds saliency methods, and an Attentional Push pathway, containing a deep convnet which learns to estimate the gaze location of the scene actors. These are followed by a shallow augmented saliency convnet that combines them and generates the augmented saliency. While the saliency pathway is fed with the whole input image to compute the saliency map, we only provide a 2-D grid location of the head of the scene actors and a cropped image region around them to the Attentional Push convnet. We use transfer learning to initialize the Attentional Push convnet and train it to minimize the error of classifying the actors' gaze location on a 2-D grid. For training and validating the Attentional Push convnet, we use a large-scale gaze following dataset, the GazeFollow dataset [26], containing more than 120000 social images annotated with the center of the eyes and the gaze location of the scene actors. We use a soft-max layer in the output of the network to compute a multinomial logistic loss during training, and also for computing the Attentional Push map, i.e. a 2-D distribution of the actor's gaze location over all possible image regions. The Attentional Push convnet is then fine-tuned along with the augmented saliency convnet to minimize the Euclidean distance between the augmented saliency with ground truth fixations. We use more than 4000 social images from the SALICON [15] dataset and annotate them with the center of the eyes and the gaze location of the scene actors. To evaluate the performance of the proposed network, we provide evaluation metrics for social images from three challenging eye fixation datasets: SALICON, iSUN [36] and CAT2000 [2] dataset.

The rest of this paper is organized as follows. Section 2 presents related work on attention tracking and saliency models that have employed gaze following as a subcomponent and data-driven saliency methods using convnets. We explain the structure and the training scheme for the Attentional Push CNN and the augmented saliency CNN in Section 3 and 4, respectively. Section 5 outlines the experiments and presents the prediction performance of our model. Future works and concluding remarks are given in Section 6.

2. Related work

The manipulating effect of the gaze direction of scene actors has been studied in the visual attention literature. Ricciardelli et al. [27] showed that perceived gaze enhances attention if it is in agreement with the task direction, and inhibits it otherwise. They showed that in spite of top-down knowledge of its lack of usefulness, the perceived gaze automatically acts as an attentional cue and directs the viewer's attention. Similarly, Kuhn and Kingstone [21] showed that even in task-driven viewing of a scene, the actors' eye gaze cannot be ignored by the viewer and causes voluntary saccades even if it is counterpredictive to the visual task. Birmingham et al. [1] assessed the ability of the Itti et al. [13] saliency map in predicting eye fixations in social scenes and showed that its performance is near chance levels. They concluded that the viewers' eye movements are not largely affected by early visual features and are instead manipulated by their interest to social information and cues of the scenes. Castelhano et al. [8] showed that while the actor's face is highly likely to be fixated, the viewer's next saccade is more likely to be toward the object that is fixated by the actor, compared to any other direction. In [32], by inspecting viewers' eye movements in social scene, Subramanian et al. showed that the viewers' fixations follow the attention patterns of the scene actors. Borji et al. [3] investigated the effect of the gaze direction of the scene actors and showed that on average, the ratio of the viewers' saccades that start from the head and end inside the gazed-at object to that of the ignored object is more than 3. Recently, Bylinskii et al. [7] analyzed the prediction performance of the best-performing saliency models (according to the MIT saliency benchmark [5] and [16]) in social scenes. The study shows that on average, around 50 percent of the under-predicted regions in the images could be predicted using the gaze location of the scene actors.

To the authors knowledge, the first attention tracking model benefiting from the actors' gaze was proposed by Parks et al. [25]. This model uses a two-state Markov chain, describing the transition probabilities between head region and non-head region states, which are used to predict whether the next fixation is gaze related or being saliency driven. Our proposed method differs from the Parks et al. model in that this method requires the sequence of the viewers' eye movements to predict the next fixation point, whereas our method is based the image information only. Recasens et al. [26] proposed a convnet to learn the likely gazed-at object in a scene. Their proposed architecture approach is similar to our model in that it learns to combine information about the head orientation and head location with the scene content. However, their combination scheme is based on a simple element-wise multiplication of a predicted gaze map with a saliency map, whereas our model employs a convnet to effectively merge the complementary information given by the saliency map and the Attentional Push map. In addition, their model outputs an estimation of the location of the gazed-at object (classification formulation), whereas our model solves the challenging regression problem of estimating viewers' fixation with the augmented saliency map.

The recent publication of large-scale fixation datasets has motivated many saliency models based on convolutional neural networks. Perhaps the first attempt of predicting image salience using convnets was the eDN model [35] which uses three optimally-chosen convnets that are used as feature extractors which are followed by a linear SVM classifier. Similarly, Liu et al. [23] proposed a multi-resolution convnet in which three different convnets, each trained on a different scale, are followed by two fully connected layers. Other models usually benefit from transfer learning in their convnets. Kummerer et al. [22] adopted the pre-trained AlexNet network [19] in their DeepGaze model and used the output of the convolutional layers to create and train a linear model to compute image salience. Similarly, the SALICON model [12] benefits from two pre-trained convnets, each on a different image scale, that are concatenated to produce the saliency map. The DeepFix model [20] uses the pre-trained VGG network [29] and extracts feature across different scales by employing multiple inception layers from GoogLeNet [33]. The ML-Net model [10] also uses the convolutional layers of the VGG network and instead of using the feature maps of the final layers, it computes the saliency map by combining feature maps extracted from different levels of the VGG network. Pan et al. [24] proposed a shallow and a deep convnet. The shallow convnet uses three convolutional and two fully connected layer, which are all randomly initialized. The deep convnet, i.e. the SalNet model, contains ten convolutional layers with the first three initialized using the VGG network.

3. Attentional Push CNN

While viewing a scene, a viewer infers the gaze location of the scene actors by first looking at their eyes, or if the eyes are not visible, by looking at their head pose. After perceiving the gaze direction, the viewer would look for possible gaze-at objects in the actors' field of view, i.e. the Attentional Push effect. This process is inherently ambiguous, as there are many situations in which the viewer might be unable to perceive the correct gaze direction [3], in addition to the uncertainty of following the actor's gaze direction to the attended image region, in cases that there are multiple salient regions in the actor's field of view. Since our goal is to track the viewers' attention, we need to consider all the ambiguities arising during this. Therefore, instead of directly solving the problem of finding the gazed-at object, we learn a probability distribution of the actor's gaze location over all possible locations in the image. This way,

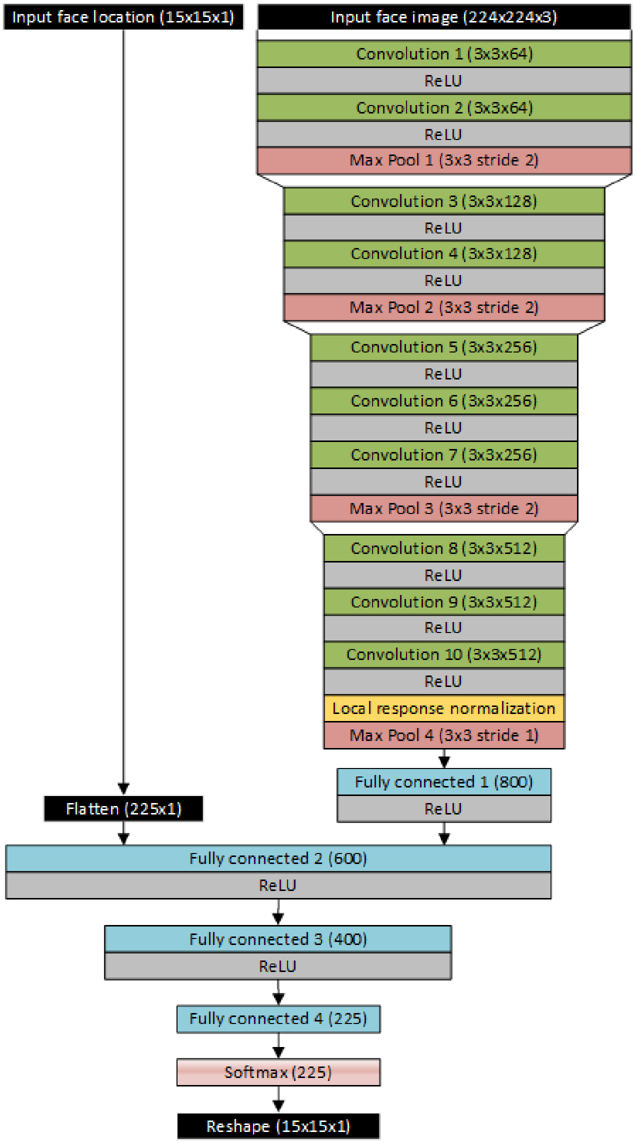

Figure 2: Architecture of the Attentional Push network. Data conditioning layers are depicted in black.

we obtain information solely coming from Attentional Push effect of the actors' gaze and we let the augmented saliency convnet to merge them with the information coming from the saliency pathway. To achieve this, we restrict the input of the Attentional Push network to a cropped image region, centered around the viewer's head and the spatial coordinates of the actor's head in the image and we train it to estimated the actor's gaze location solely based on them. In the following sections, we provide the architecture and the training procedure for the Attentional Push convnet.

3.1. Architecture

As illustrated in Figure 2, the network consists of fourteen weight layers, including ten convolutional and four fully connected layers. Considering the amount of available training instances, we followed the structure of the VGG-16 net [29] and restricted the convolutional kernels to 3×3 in size. All convolutional layers are followed by a Rectified Linear Unit(ReLU) activation to introduce non-linearity to the network.

We formulate the gaze location learning as a classification problem, i.e. classifying the gazed-at location to one of a pre-defined set of possible locations. Assuming an $M \times M$ spatial grid, the number of output classes would be M^2. The network takes two inputs: a close-up, cropped image region around the actor's head and the location of the head within the $M \times M$ spatial grid. Given an annotated RGB input image $I \in \mathbb{R}^{W \times H \times 3}$, we use the location of the center of the eyes of the scene actor, denoted by (x_h, y_h), and center a region-of-interest (ROI) around it, $F = I(x_h - sW : x_h + sW, y_h - sH : y_h + sH, :)$, where $:$ denotes the slicing operator and s is the scale factor (in our experiments, we set s to 0.25). Note that if the ROI exceeds the image boundaries, the remaining pixels of F are set to zero. Finally, F is then resized to 224×224 pixels and is fed through the convolutional layers. To create the face location input, we create a zero-initialized image $L \in \mathbb{R}^{M \times M \times 1}$, and set $L(\frac{x_h}{W}M, \frac{y_h}{H}M) = 1$. We set M to 15 in our experiments.

The model contains four max-pool layers, three of them having strides of two, which effectively halves the size of the following feature maps in the network. Therefore, after the last convolutional layer, the feature maps are of the size of $(28 \times 28 \times 512)$. Before concatenating the above with the face location input, we use a fully connected layer to encode the feature maps into a more compressed representation. The size of the fully connected layer is set low to prevent over-fitting. This continues in the remaining layer and the last fully connected layer generates the network estimation of the gaze location in a flattened $M^2 \times 1$ representation. We use a soft-max layer in the output of the network to compute a multinomial logistic loss during training, and also for computing the 2-D probability distribution of the actor's gaze location over all possible image regions; i.e. the Attentional Push map.

3.2. Training

We implemented the network using Caffe [14]. Transfer learning is used to initialize the parameters of the convolutional layers from the VGG-16 net [29]. The weights of the fully connected layers are randomly initialized using the Xavier method [11] and the bias set to 0. For training, we use 119125 images from the GazeFollow dataset [26] train set, and 3018 images from the GazeFollow dataset test set for validation (see Section 5.1). To zero center the pixel intensities, we subtract the mean pixel value of the training images from all of the training and validation images. We

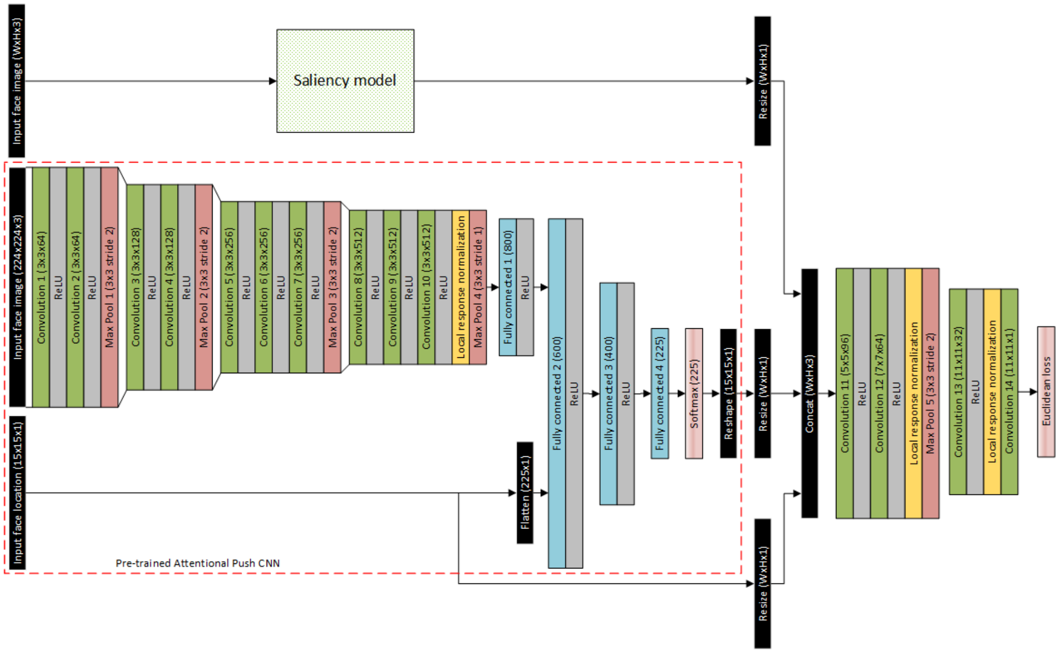

Figure 3: Architecture of the augmented saliency network. Data conditioning layers are depicted in black. The attentional push network is indicated by the red dashed line.

also use random horizontal flips during training to increase the number of training examples. The network is trained by back-propagating the multinomial logistic regression loss between the soft-max output and the ground truth gaze location, using mini-batch stochastic gradient descent with a mini-batch size of 2, a momentum of 0.9 and a weight decay of 0.0005. We train the network with a base learning rate of 1×10^{-5} for all of the fully connected layers and the last three convolutional layers. The learning rate of the first two convolutional layers are set to 0, while the learning rate of the rest of the convolutional layers is set to 1×10^{-7}. We used drop-out and batch normalization after each of the fully connected layers to speedup convergence. The network is validated against the validation set every 1000 iterations and the learning rates are scaled down by a factor of 0.5, if the performance saturates on the validation set, to prevent over-fitting.

4. Augmented Saliency CNN

We combine the complementary information given by the saliency and the Attentional Push maps using a shallow convnet. The augmented saliency convnet takes the saliency map, the Attentional Push map, and the actors' head locations as inputs and generates the augmented saliency map. The reason that we feed the actors' head locations to the augmented saliency convnet is that the augmentation process should vary as a function of both the actors' head and gaze location. For instance, if an image contains multiple faces, not all of them are equally important and the Attentional Push effect would change as a function of the actors' location inside the image. Another example would be cases in which an actor located near the image boundaries is looking outside of the image boundaries, which are not as strong to push the viewers' attention. In addition, since we are augmenting pre-trained saliency models, the augmentation should vary depending on the employed saliency model. Therefore, we train the network once for each saliency model. In the following sections, we provide the architecture and the training procedure for the augmented saliency convnet.

4.1. Architecture

As illustrated in Figure 3, the network takes three inputs: the saliency map, the Attentional Push map, and the location of the head of the scene actors; all resized to the same size as the input image. In addition to the Attentional Push layers, the network consists of four convolutional layers, three of them followed by a ReLU layer to introduce non-linearity to the network. These added layers are responsible for combining the provided input information and to compute the augmented saliency. The architecture of the network is designed and restricted to convolutional layers in order to keep the number of parameters small, considering the amount of available training instances. The output of the last convolutional layer is used as the augmented saliency map and is fed to the loss layer during training. We use a Euclidean loss layer to minimize the Euclidean distance between the augmented saliency with ground truth fixations during training.

Table 1: Summaries of the used datasets.

Dataset	Annotations	Viewers	Added annotations	Train	Validation	Test
GazeFollow [26]	Head and gaze location	Crowd	-	119125	3018	-
SALICON [15]	Mouse tracking based eye-movement	Crowd	Head & gaze location	3246	603	200
CAT2000 [2]	Eyetracker	28	Head location	-	-	200
iSUN [36]	Web-cam eyetracker	Crowd	Head location	-	-	200

To illustrate the effectiveness of Attentional Push in augmenting image saliency, we employ five saliency models and train and evaluate the network in Figure 3 once for each of them. We used the MIT saliency benchmark [5] in selecting the best-performing saliency models which have available implementations. We used three neural network-based saliency models: ML-Net [10], SalNet [24] and eDN [35]; and two best-performing non neural network models: BMS [37] and RARE [28].

4.2. Training

We implemented the network using Caffe [14]. We use transfer learning to initialize the parameters of the Attentional Push layers from the pre-trained Attentional Push convnet and fine-tune them along with the added convolutional layers to minimize the Euclidean loss. The weights of the added convolutional layers are randomly initialized using the Xavier method [11] with the bias set to 0. We use 3246 social images from the SALICON [15] dataset (see Section 5.1) for training and 603 images for validating the network. We subtract the mean pixel value of the training images from all of the training and validation images. We also normalize the fixation heatmaps between 0 and 1 prior to training. We also use random horizontal flips of the training images and fixation heatmaps during training.

The network is trained by back-propagating the Euclidean distance between the augmented saliency and the fixation heatmaps, using mini-batch stochastic gradient descent with a mini-batch size of 2, a momentum of 0.9 and a weight decay of 0.0005. We train the network with a base learning rate of 1×10^{-8} for the randomly initialized layers. The learning rate of the pre-trained Attentional Push convnet is set 1×10^{-10}. The network is validated against the validation set every 100 iterations and the learning rates are scaled down by a factor of 0.5, if the performance saturates on the validation set to prevent over-fitting.

5. Evaluation and Comparison

5.1. Datasets

We use the following four datasets to train, validate and test the performance of proposed methodology. Table 1 summarizes the employed datasets.

The GazeFollow dataset [26] is a large-scale dataset of social scenes, annotated with the location of the head and the location of where the scene actors are looking. The dataset annotations are obtained using an Amazon Mechanical Turk setup. As suggested in [26], we use 119125 images for training and the rest are used for validating the network. The SALICON dataset [15] is obtained using mouse-contingent-tracking as a replacement for eye-contingent-tracking. The dataset contains fixation locations and fixation heatmaps for 10000 training and 5000 validation images. We selected 3246 social images from the training and 803 social images from the validation set respectively (all images contain at least one actor), and added annotation for the location of the actors' head and the actors' gaze location. The CAT2000 dataset [2] contains 2000 images from 20 different categories. Images in this dataset are annotated with eye-movement data from 28 viewers. We use 200 images from the Action and the Social categories of this dataset during evaluation. We provide annotations for the head location of the scene actors during testing. The iSUN dataset [36] contains 6926 images from natural scenes. The images are annotated with eye-tracking data, obtained from viewer gaze-tracking using web-cams on an Amazon Mechanical Turk setup. We use 200 social images from this dataset for evaluation. We provide annotations for the head location of the scene actors during testing.

5.2. Evaluation protocol

We employ three neural network-based saliency models and two best-performing non neural network saliency models and train the full network in Figure 3 for each. All networks are trained and validated using the training and the validation subset of the annotated SALICON images. We evaluate the performance of the networks on three test sets: the test subset of the annotated SALICON, the action and social categories of CAT2000 and the social images from the iSUN dataset. Note that although our network requires the actors' head location during evaluation, we do not use a face detector and instead, use human annotations for the actors' head location. The reason is that in many of the images, the actors are looking sideways or even looking away from the camera, which makes it a challenging task for even the best-performing face detectors such as [38]. Since our goal is to illustrate the effectiveness of Attentional Push, we assume to have head location annotations in both training and testing.

Attention models have commonly been validated against

Table 2: Average evaluation scores for the augmented saliency vs. saliency models on the SALICON, CAT2000 and iSUN test sets.

	SALICON			CAT2000			iSUN		
	AUC	NSS	CC	AUC	NSS	CC	AUC	NSS	CC
ML-Net [10]	0.66	0.36	0.13	0.73	0.62	0.26	0.68	0.52	0.27
augmented ML-Net	0.82	2.84	0.71	0.81	1.65	0.61	0.73	0.98	0.50
SalNet [24]	0.66	1.41	0.26	0.66	0.96	0.31	0.60	0.57	0.22
augmented SalNet	0.83	3.04	0.74	0.80	1.68	0.63	0.74	1.12	0.52
eDN [35]	0.78	1.03	0.34	0.78	1.06	0.43	0.72	0.85	0.44
augmented eDN	0.85	2.90	0.74	0.83	1.71	0.66	0.77	1.25	0.59
BMS [37]	0.77	1.28	0.38	0.79	1.18	0.46	0.68	0.72	0.35
augmented BMS	0.86	2.94	0.74	0.83	1.68	0.65	0.75	1.15	0.55
RARE [28]	0.77	1.33	0.39	0.79	1.25	0.49	0.68	0.75	0.36
augmented RARE	0.85	2.98	0.75	0.83	1.72	0.66	0.75	1.19	0.55
Average improvements	0.11	1.86	0.44	0.07	0.67	0.25	0.08	0.44	0.21

the eye movements of human observers based on various evaluation metrics in the literature (e.g. [4, 5]). Since the performance of a model may change remarkably while using different metrics, we use three popular evaluation metrics: the Area Under the ROC Curve (AUC), the Normalized Scan-path Saliency (NSS), and the Correlation Coefficient (CC) to ensure that the main qualitative conclusions are independent of the choice of metric. We use MATLAB's implementation of the evaluation scores from [6].

Table 2 compares the prediction performance of the Attentional Push-based augmented saliency with the standard saliency methods on the SALICON, CAT2000 and the iSUN test sets respectively. The results show that the augmented saliency consistently improves upon the standard saliency methods. The results indicate that all the employed saliency models, both neural network and non-neural network based models, can benefit from Attentional Push to improve the prediction accuracy. Interestingly, the BMS and the augmented BMS models outperform the ML-Net and SalNet and their augmented version in many cases. Comparing the average improvements over the three datasets, it is clear that the proposed methodology is able to perform well across different eye-tracking datasets, even though it only uses the SALICON train/validation sets during the training procedure. We present qualitative results for comparing the augmented saliency and the saliency methods in Figure 4. The figure compares the ground-truth fixation heatmaps with different components of our model, i.e. the saliency map, the Attentional Push map, the input head location and the augmented saliency map. As seen in the figure, augmented saliency maps clearly benefit from the all of them to provide an improved prediction of the ground-truth fixations.

Table 3: Performance analysis with some components disabled. The results are based on BMS saliency and the SALICON test set.

	AUC	NSS	CC
augmented saliency	0.86	2.94	0.74
No Attentional Push	0.83	2.30	0.56
No head location	0.82	2.13	0.54
Another saliency	0.81	2.79	0.70
No saliency	0.80	2.30	0.50
Saliency	0.77	1.28	0.38

To investigate the relative significance of each component in the augmented saliency, Table 3 reports the prediction performance with each component disabled at a time. 3 shows the results for the augmented BMS network, with the BMS saliency, the Attentional Push map and the actors' head location disabled. We also included the results for using the augmented BMS network, fed with SalNet saliency during testing to illustrate the performance with sub-optimal information fusion. The results show that all three components contribute to the augmented saliency. The performance of the model without the saliency input suggests that while viewing social scenes, the viewers tend to focus on social cues instead of irrelevant salient regions.

6. Conclusion and Future work

We presented an attention modeling scheme which combines Attentional Push cues, i.e. the power of image regions to direct and manipulate the attention allocation of the viewer, with standard saliency models, which generally concentrate on analyzing image regions for their power to pull attention. We presented a deep convolutional convnet which learns to follow the gaze location of the scene actors and augments saliency models with Attentional Push. Based on evaluation using three eye-tracking datasets, our methodology significantly outperforms saliency methods in predicting the viewers' fixations. Our results showed that by employing Attentional Push cues, the augmented saliency maps can improve upon the state of the art in saliency mod-

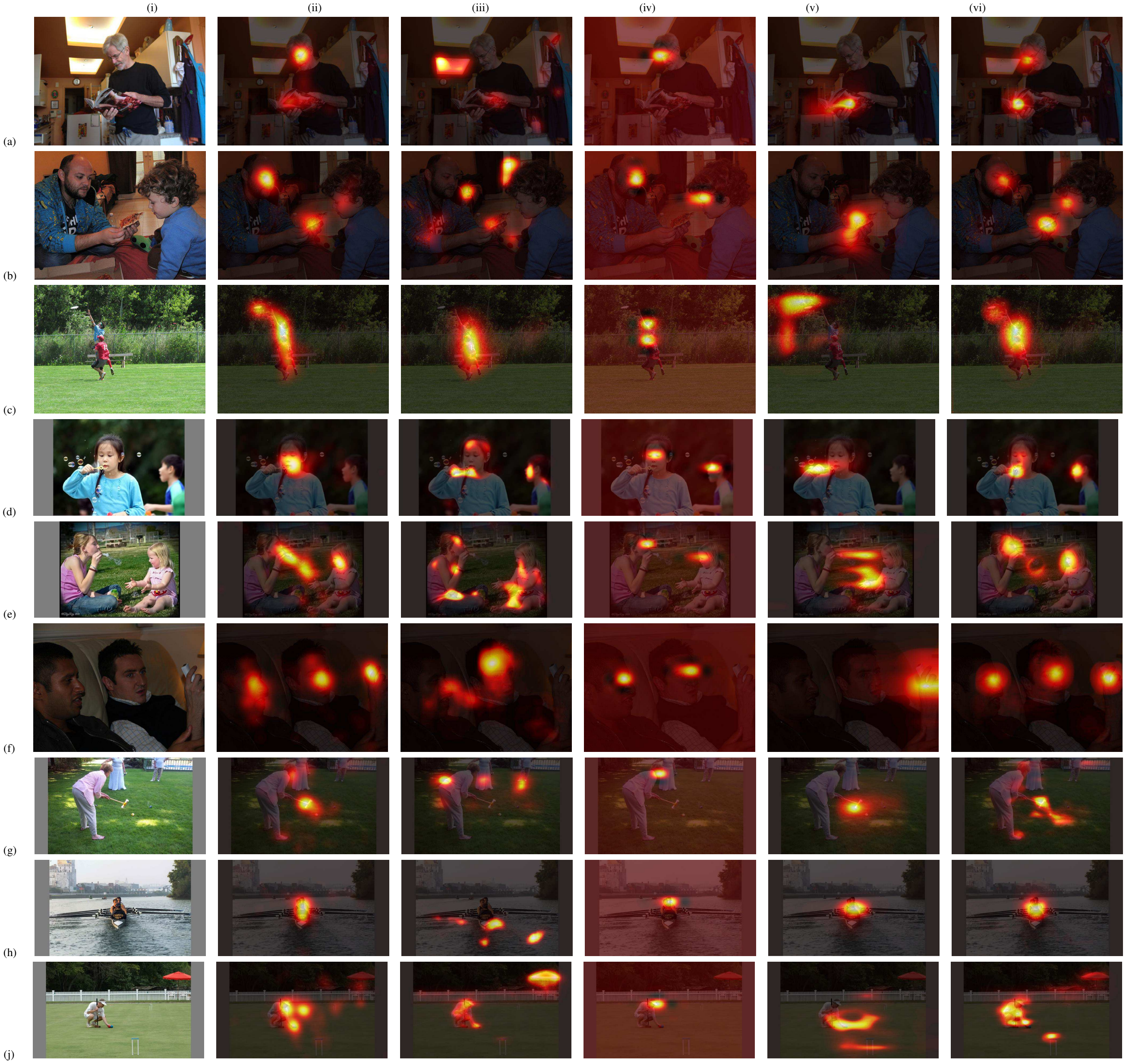

Figure 4: Qualitative results. (i) Input image, (ii) ground-truth fixation heatmap, (iii) saliency map, (iv) face location input, (v) Attentional Push map, and (vi) augmented saliency map. The input images are from the SALICON and the CAT2000 test sets. The employed saliency models are: (a) BMS,(b) BMS,(c) eDN,(d) eDN,(e) eDN,(f) ML-Net,(g) ML-Net,(h) ML-Net, and (j) Rare.

els. In this work, we limited the Attentional Push effect to the scene actors' gaze. However, there are other Attentional Push cues reported in the literature of attention tracking. One of the most frequently cited Attentional Push cues in the literature is the center bias. We can treat the center-bias effect in the shared attention setting by considering the photographer as an actor in the shared attention setting, which tries to put the semantically interesting and therefore, salient elements in the center of the frame. In addition, Attentional Push cues can also arise from dynamic events. For example, Smith [31] showed that sudden movements of the heads of actors are a very strong cue for attention. Smith [31] also notes the "bounce" in the attention of a movie viewer back to the center of the movie screen when tracking an object which moves off the screen to one side. Similarly, abrupt scene changes are the contribution of the center bias in predicting viewer's attention while watching dynamic stimuli. We believe that the introduction of attention tracking techniques based on treating the viewer as a participant in a shared attention situation, either in static or in dynamic scenes, will open new avenues for research in the attention field.

References

[1] E. Birmingham, W. F. Bischof, and A. Kingstone. Saliency does not account for fixations to eyes within social scenes. *Vision Research*, 49(24):2992 – 3000, 2009.

[2] A. Borji and L. Itti. CAT2000: A large scale fixation dataset for boosting saliency research. *CVPR 2015 workshop on "Future of Datasets"*, 2015.

[3] A. Borji, D. Parks, and L. Itti. Complementary effects of gaze direction and early saliency in guiding fixations during free viewing. *Journal of Vision*, 14(13):1–32, 2014.

[4] A. Borji, D. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Trans. Image Processing*, 22(1), 2013.

[5] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. Mit saliency benchmark. http://saliency.mit.edu.

[6] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *arXiv preprint*, arXiv:1604.03605, 2016.

[7] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand. Where should saliency models look next? In *Computer Vision – ECCV 2016: 14th European Conference*, 2016.

[8] M. S. Castelhano, M. Wieth, and J. M. Henderson. I see what you see: Eye movements in real-world scenes are affected by perceived direction of gaze. In *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, pages 251–262. Springer Berlin Heidelberg, 2007.

[9] J. J. Clark and N. J. Ferrier. Modal control of an attentive vision system. In *Computer Vision., Second International Conference on*, pages 514–523, Dec 1988.

[10] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. A deep multi-level network for saliency prediction. In *23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[11] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

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

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

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[15] M. Jiang, S. Huang, J. Duan, and Q. Zhao. Salicon: Saliency in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[16] T. Judd, F. Durand, and A. Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012.

[17] F. Kaplan and V. V. Hafner. The challenges of joint attention. *Interaction Studies*, 7(2):135–169, 2006.

[18] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–227, 1985.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

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

[21] G. Kuhn and A. Kingstone. Look away! eyes and arrows engage oculomotor responses automatically. *Attention, Perception and Psychophysics*, 71:314–327, 2009.

[22] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze I: boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint*, arXiv/1411.1045, 2014.

[23] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu. Predicting eye fixations using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[24] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[25] D. Parks, A. Borji, and L. Itti. Augmented saliency model using automatic 3D head pose detection and learned gaze following in natural scenes. *Vision Research*, 116, Part B:113 – 126, 2015.

[26] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[27] P. Ricciardelli, E. Bricolo, S. M. Aglioti, and L. Chelazzi. My eyes want to look where your eyes are looking: Exploring the tendency to imitate another individuals gaze. *Neuroreport*, 13(17):2259–2264, 2002.

[28] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, and T. Dutoit. Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Processing: Image Communication*, 28(6):642–658, 2013.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556, 2014.

[30] K. Smith, S. Ba, J. Odobez, and D. Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1212–1229, July 2008.

[31] T. J. Smith. The attentional theory of cinematic continuity. *Projections*, 6(1):1–27, 2012.

[32] R. Subramanian, V. Yanulevskaya, and N. Sebe. Can computers learn from humans to see better?: Inferring scene semantics from viewers' eye movements. In *Proceedings of the*

19th ACM International Conference on Multimedia, pages 33–42. ACM, 2011.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint*, arXiv:1409.4842, 2014.

[34] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[35] E. Vig, M. Dorr, and D. Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2798–2805, June 2014.

[36] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint*, arXiv:1504.06755, 2015.

[37] J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: A boolean map approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

[38] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012.