# Predicting the Eye Fixations with Prior Knowledge: A Bayesian Learning Architecture

Lauren Arnett and Chengzhi Mao
Columbia University in the City of New York
{lba2138,cm3797}@columia.edu
Link to GitHub Repository

## Abstract

*Applications of tracking eye fixation location span from neuroscience and the study of human vision to advertising and human computer interaction. By creating a model to generate a prediction of where the eye may be most attracted to, industries can circumvent the expense of running studies on eye-tracking with actual human subjects. We look to improve upon existing models of saliency by using Bayesian methods with deep learning techniques.*

## 1. Introduction

It is estimated that 80% of all external sensory input processed by the human brain is processed by the visual pathway [1]. As such, optimizing image layout for processing by the human brain allows for better information retrieval and retention across the image. Studying how humans' eyes move across images is thus relevant for fields from neuroscience to advertising and art. Being able to predict where humans are most likely to look provides a guideline as to whether the image has an effective layout, what humans are attracted to in viewing art, or how an image should be cropped to feature the subject. Using an existing dataset of eye movements, we build a predictive model to generate the most likely fixation locations on a new image.

Our main contribution in this paper is combining Bayesian methods with deep learning to perform the eye-fixation prediction task. We also explore a dataset of fixation locations that has not yet been used for the fixation prediction task. Additionally, we introduce a method to use an auxiliary dataset to learn image priors based on the area of the image that encodes the most semantic meaning.

## 2. Related Work

Traditionally, studies on eye movements have been carried out such that viewers look at images on a monitor while an eye tracker records the eye-fixations that stay within a threshold angle of movement. This procedure is very costly, and necessitated formulating a method to predict where users will look. Thus, models of saliency—the likelihood of a location to attract the visual attention of a human—developed that are modeled mathematically using biologically plausible linear filters. For example, linear combinations of filters for low-level features such as color, intensity, and orientation filters can be used to compute a total saliency map for an image, providing a bottom-up understanding of the image [2].

These models do not account for particular tasks that the viewer may have in looking at the image, and often do not align with the ground truth fixation locations. Judd *et al*. [3] propose using deep learning for this task rather than deriving mathematical models and show that training from a large database of eye-tracking data outperforms existing models. Kümmerer *et al*. [4] also employ deep learning for predicting fixation locations, using the AlexNet architecture in *DeepGaze I* and building upon the VGG-19 network in *DeepGaze II*.

Developing a dataset for use by saliency models is also a field of exploration. Judd *et al*. [3] create a dataset of 1003 images with the fixation locations from 15 viewers each and make it publicly available. Jiang *et al*. [5] relies on an assumption of eye-mouse coordination—they simulate recording eye-tracking data by instead recording mouse-tracking data using Amazon Mechanical Turk. This provides a less expensive and training-intensive method of developing a simulated saliency map. Finally, the MIT300 dataset [6] looks to provide a performance benchmark for new predictive models for saliency, with performance statistics for over 80 models at the time of writing.

## 3. Dataset

We make use of two datasets: ImageNet [7] for the train set for our prior and an eye-fixation dataset for our classifier.

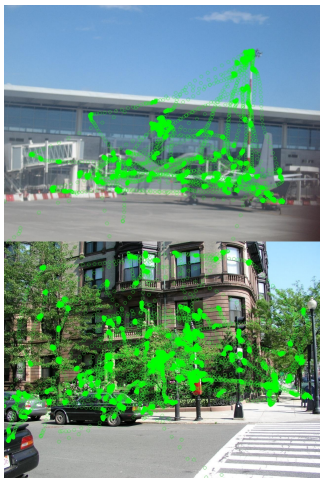

Figure 1. Eye-tracking locations on sample images from dataset.

### 3.1. Use of ImageNet for Training a Prior

We use the complete training set with the original image and the label information of ImageNet for training our prior. We select the best prior model based on the performance on the validation set.

### 3.2. Source for Eye-Tracking Data

We use an open-source dataset developed at Osnabrück University and the University Medical Center in Hamburg-Eppendorf [8, 9]. This dataset comprises images of many categories, including urban and rural settings, fractals, faces, and websites. For our purposes, we use the images of urban and rural settings, which have been taken from the LabelMe dataset [10]. Each image in this category is shown to the viewer for eight seconds, and this category has the x- and y-coordinates of 70,026 fixation locations for seven observers over 600 images. Figure 1 shows examples of fixation locations across the image. We use a 500/100 train/test split.

### 3.3. Preprocessing

We preprocess the fixation data by adding weight to those pixels in the image that have fixations to produce ground truth labels. We apply a Gaussian blur to also add weight to neighboring pixels to account for the $0.5°$ angle of error due to the eye-tracking machine. Figure 2 shows the fixation locations and corresponding ground truth labels.

## 4. Learning a Model

### 4.1. Baseline Model Setting

We conduct a binary classification task for each output pixel in our baseline eye fixation model. We propose a U-Net architecture, using a fixed pretrained VGG model. We

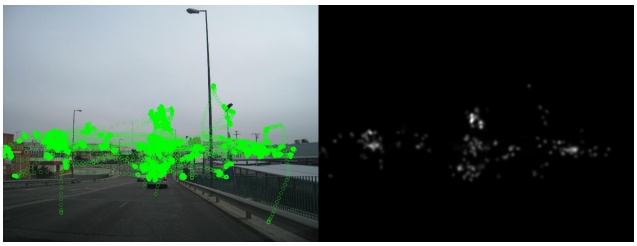

Figure 2. Preprocessing of the fixation locations.

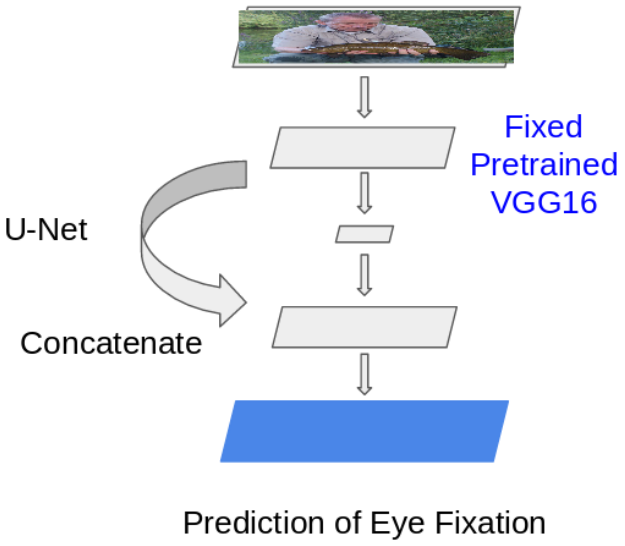


Figure 3. U-Net Architecture for our baseline model.

take the features of layer numbers 3, 8, 15, and 22 in the VGG model and feed them into our fixation prediction network. We first learn a upsampling of the high-level, low-resolution features of VGG, and then concatenate it with the low-level, high-resolution features to produce a final prediction.

### 4.2. Overcoming the Checkerboard Artifacts of Upsampling

Building the upsampling using the deconvolution operation introduces checkerboard artifacts, as shown in Figure 4. This is partly due to the overlap of the deconvolution, according to Odena *et al*. [11]. We overcome this by first sequencing a nearest-neighbor interpolation along with a normal convolution operation.

### 4.3. Learning Prior from the ImageNet

Due to the high cost of collecting eye movement data, which requires subjects to sit in front of the computer with an expensive eye-tracking machine, the number of samples that could be acquired in a training set may be limited.

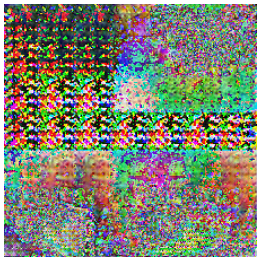

Figure 4. Example of checkerboard artifacts.

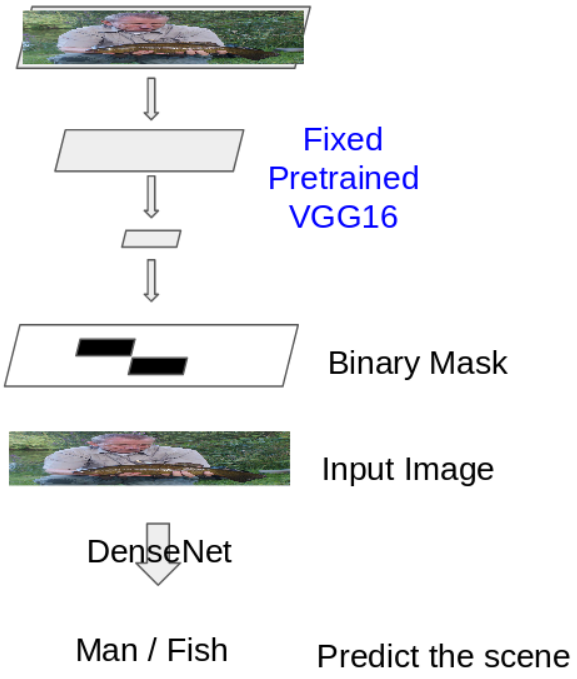


Figure 5. Architecture for learning the prior

We make an attempt to address these challenges by utilizing some prior knowledge of eye fixation from an auxiliary dataset.

We interpret eye fixation points as the places which encode the most semantic information for the image. With this prior, we construct a model which predicts a binary mask with 0 and 1 to select the appropriate locations of the image. Applying this binary mask to the image, we then use a neural network to predict the semantic information of the masked image.

The elements with value 1 in the predicted binary mask mimic saliency map, where the selection of these points should maximize the semantic information in the resulting image after training.

We denote the semantic label of a given image $x_i$ as $y_i$, the predicted mask as $M$, and the loss function as $L$,

$$M_i = F_1(x_i, k)$$

$$L(X, Y) = \sum_{x_i \in X} logP(y_i | F_2(M_i * x_i))$$

where * denotes element-wise multiplication and $k$ denotes the number of elements we use in the binary mask. Here, $F_1$ denotes the mask-prediction network, and $F_2$ denotes the pretrained classification network.

After training, the $M_i$ can be interpreted as a learned prior knowledge, which is fixed during the Bayesian learning procedure.

### 4.4. Incorporating Prior Improves Prediction

The main contribution of our work is a Bayesian inference procedure that builds upon a prior knowledge learned with our auxiliary dataset. We denote the eye-fixation ground truth as $t$,

$$L = logP(M, t|x) = log(P(M|x)) + log(t|M, x)$$

The prior $M$ gives a good estimation of the eye fixation in advance, which enhances the optimization achieved by this loss function.

### 4.5. Training

We train our baseline model using SGD, with a learning rate of $1e{-}5$ and weight decay of $1e{-}6$. We train 50 epoch before we stop.

We use ImageNet data for our prior training, where the mask prediction network is based on a pretrained VGG–16 network, and the network to predict semantic meaning from the masked image is a DenseNet, which we believe to have more accurate gradient information. We train this mask prediction network using SGD with a learning rate of $1e{-}5$.

### 4.6. Evaluating the Prior

We visualize our binary mask prediction using both ImageNet and our fixation dataset. In Figure 6, we show the output of applying the mask to an ImageNet sample. This output is in line with expectations of eye-fixations on the bird. We then extrapolate our learned prior model for usage with our eye-fixation dataset, as seen in Figure 7. Through visualizing the application of the mask across images in our eye-fixation dataset, we see that the prior, after training on ImageNet, is able to generalize to the new dataset.

### 4.7. Evaluation Metric

In accordance with Judd *et al*. [3], we use the true positive rate $\frac{TP}{TP+FN}$ as the evaluation metric, where $TP$ is number of true positives, and $FN$ is the number of false

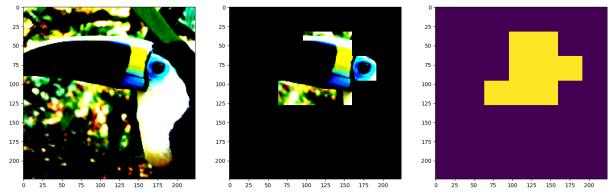

Figure 6. Example of mask learned on ImageNet

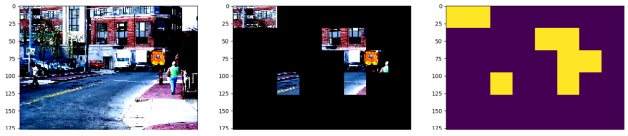

Figure 7. Example of extrapolating the prior learned on ImageNet to the eye fixation dataset

negatives. This is evaluated using the probability prediction output of how salient each pixel is. We threshold this saliency map at the top $k$ percent probability, where $k = 1, 3, 5, 10, 15, 20, 25$ and 30 percent of the image saliency map.

### 4.8. Performance

Figure 8 shows our model's performance. We draw the ROC curve for the true positive rate. We can see that the Bayesian model outperforms the baseline model by a large margin. Specifically, at the 10% threshold for the saliency map, the performance of the baseline model achieves 94.1% performance of the Bayesian model, and at the 30% threshold, the baseline model achieves 97% performance of the Bayesian model.

## 5. Conclusion

Our study looks at using new methods and a new dataset for the task of predicting eye-fixation locations. We use a subset of a large dataset of fixation locations to account for the difficulty of obtaining such data. By combining a neural network classification model with a learned image prior, we achieve a higher performance than that of our classifier alone.

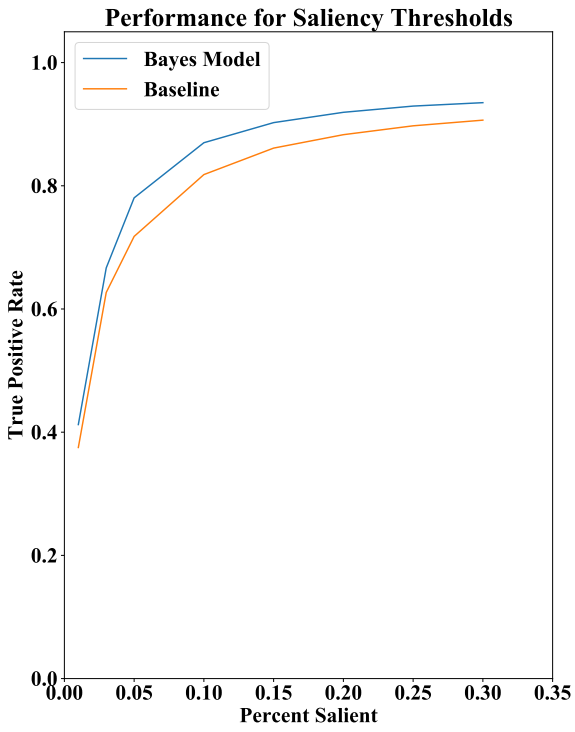


Figure 8. Example of a short caption, which should be centered.

## References


[1] R. Jerath, M. W. Crawford, V. A. Barnes. Functional representation of vision within the mind: A visual consciousness model based in 3D default space. *Journal of Medical Hypotheses and Ideas*, 9(1):45-56, 2015.

[2] L. Itti and C. Koch. A Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention. *Vision Research*, 40(10-12):1489–1506, 2000.

[3] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2106–2113, 2009.

[4] M. Kümmerer, T. S. A. Wallis, L. A. Gatys, M. Bethge. Understanding Low- and High-Level Contributions to Fixation Prediction. *The IEEE Int. Conf. Comput. Vis.*, pp. 4789-4798, 2017.

[5] M. Jiang, S. Huang, J. Duan, Q. Zhao SALICON: Saliency in Context. *CVPR*, pp. 1072-1080, 2015.

[6] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba. MIT Saliency Benchmark.

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *CVPR*, 2009.

[8] N. Wilming, S. Onat, J. Ossandón, A. Acik, T. C. Kietzmann, K. Kaspar, R. R. Gamiero, A. Vormberg, P. König. Data from: An extensive dataset of eye movements during viewing of complex images. `https://doi.org/10.5061/dryad.9pf75`. *Dryad Digital Repository*, 2017.

[9] N. Wilming, S. Onat, J. Ossandón, A. Acik, T. C. Kietzmann, K. Kaspar, R. R. Gamiero, A. Vormberg, P. König. An extensive dataset of eye movements during viewing of complex images.

https://doi.org/10.1038/sdata.2016.126. *Nature Scientific Data*, 4(1):160126, 2017.

[10] B. Russell, A. Torralba, K. Murphy, W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *The IEEE Int. Conf. Comput. Vis*, pp. 157-173.

[11] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and Checkerboard Artifacts *Distill*, 2016.