International Journal of Multimedia Data Engineering and Management
Volume 12 • Issue 1 • January-March 2021

Automated Filtering of Eye Movements Using Dynamic AOI in Multiple Granularity Levels

Gavindya Jayawardena, Old Dominion University, USA
Sampath Jayarathna, Old Dominion University, USA

ABSTRACT

Eye-tracking experiments involve areas of interest (AOIs) for the analysis of eye gaze data. While there are tools to delineate AOIs to extract eye movement data, they may require users to manually draw boundaries of AOIs on eye tracking stimuli or use markers to define AOIs. This paper introduces two novel techniques to dynamically filter eye movement data from AOIs for the analysis of eye metrics from multiple levels of granularity. The authors incorporate pre-trained object detectors and object instance segmentation models for offline detection of dynamic AOIs in video streams. This research presents the implementation and evaluation of object detectors and object instance segmentation models to find the best model to be integrated in a real-time eye movement analysis pipeline. The authors filter gaze data that falls within the polygonal boundaries of detected dynamic AOIs and apply object detector to find bounding-boxes in a public dataset. The results indicate that the dynamic AOIs generated by object detectors capture 60% of eye movements & object instance segmentation models capture 30% of eye movements.

KEYWORDS

Area-of-Interest, Eye Tracking, Filtering

1. INTRODUCTION

Eye tracking can reveal objective and quantifiable information about the quality, predictability, and consistency of the underlying covert process of the human brain when carrying out cognitively demanding tasks (McCarley and Kramer 2008; Radach et al. 2003; Van der Stigchel et al. 2007). According to the eye-mind hypothesis (Just and Carpenter 1980), observers attend where their eyes are fixating. Thus, eye tracking measurements enable us to investigate cognitive behavior when visually exploring a stimulus. With the advancement of eye tracking technology, gaze tracking measurements have become reliable and accurate.

Eye gaze measurements include various metrics relevant to oculomotor control (Komogortsev et al. 2013) such as saccadic trajectories, fixations, and other relevant measures including velocity, duration, amplitude, pupil dilation (Krejtz et al. 2018). Studies have shown that the size of the pupil diameter correlates with the task complexity (Kosch et al. 2018) enabling the use of pupillary behavior as biomarkers of mental workload when completing a task. Several studies (Gehrer et al. 2018; Jayawardena et al. 2020) have incorporated eye tracking to obtain insights into underlying covert processes. As a standard practice in the community, upon successful completion of the study, performance of users is measured, traditional positional gaze metrics and advanced gaze metrics are

DOI: 10.4018/IJMDEM.2021010104

calculated, and statistical significance of computed numerous metrics are evaluated (Gehrer et al. 2018; Jayawardena et al. 2020).

Eye tracking experiments utilize area of interests (AOIs) to the aid the analysis process by extracting eye gaze metrics within a predefined AOIs. An AOI is a region of stimuli that is used to study the eye gaze metrics and link eye movement measures to the part of the area of the stimuli (Hessels et al. 2016). Studies in visual attention and eye movements (Noton and Stark 1971; Privitera and Stark 2000) have shown that humans only attend to a few AOIs in a given stimulus. Analysis of eye gaze metrics within AOIs can provide important cumulative clues to the underlying physiological functions supporting the allocation of visual attention resources. For instance, in the context of user interface interaction, the number of fixations within an AOI (e.g., a user interface component) indicates the efficiency of finding that component among others, whereas the maximum and average fixation duration within that AOI indicates the informativeness of that component (Goldberg and Kotval 1999). In addition, the fixation frequency and blink frequency within AOIs can indicate cognitive workload when interacting with the particular component of the user interface (Van Orden et al. 2001).

The analysis of eye movements in dynamic AOIs, such as in video sequences is not new (Marchant et al. 2009; Goldstein et al. 2007; Crossland et al. 2002; Timberlake et al. 2005). Studies with the primary focus on detecting fixation sequences within identified AOI have used clustering techniques to group gaze locations to determine the AOI (Nguyen et al. 2004), and various image processing algorithms (Privitera and Stark 2000) to automatically identify the AOI. For the analysis of cognitive workload and allocation of visual attention resources, many studies (Santella et al. 2006; Shanmuga Vadivel et al. 2015; Khosravan et al. 2016; Wang et al. 2019) have utilized saliency models of viewers in conjunction with visual information from video frames. In addition, computer vision techniques (Weibel et al. 2012) have been applied to generate AOI-mapped gaze coordinates by using a template of the desired object derived from a single frame of the eye tracking stimuli video. But it only works for pre-recorded eye tracking stimuli using the manual specification of the AOI template generated beforehand.

To overcome these challenges, we propose a computer vision-based deep neural network approach to identify the AOIs in video streams in real-time to filter gaze locations that fall into the identified AOIs for the analysis of both positional and advanced eye gaze metrics. We focus on two filtering techniques to dynamically generate AOI-mapped gaze locations on video streams:

- Pre-trained object detectors to identify bounding boxes of dynamic AOI in video sequences, and;
- Object instance segmentation models for offline detection of dynamic AOI via precise pixel-wise masks.

Since computer vision techniques can be adapted to detect a wide range of objects, we apply computer vision techniques to extract dynamic AOI-mapped eye movement data. We use transfer learning to remodel existing image classifiers for dynamic AOI detection. Upon detection of dynamic AOIs, we extract eye movement data which falls within the detected dynamic AOIs by checking if the gaze coordinate falls within any dynamic AOIs' polygonal boundaries. Our approach represents two levels of granularity in AOIs, (1) a lower level of granularity (i.e. bounding box), and (2) a higher level of granularity (i.e. pixel-wise mask). We evaluate our filtering methodology in terms of the percentage of the eye movements captured from each filtering technique proposed. We utilize our prior work on Real-Time Advanced Eye Movements Analysis Pipeline (RAEMAP) (Jayawardena 2020) designed to analyze traditional positional gaze measurements as well as advanced eye gaze measurements.

In contrast to the existing studies on analyzing eye movements within static AOIs (often pre-determined), which only works for pre-recorded eye tracking stimuli, our proposed methodology could extract eye movement data from dynamically detected AOIs in real time. In this study, we present pre-trained object detectors and object instance segmentation models for the detection of dynamic AOIs as the initial step towards extracting eye movement data from dynamically detected AOIs in

real time. We evaluate the object detectors and object instance segmentation models to be integrated into our RAEMAP eye movement analysis pipeline.

We are particularly interested in using these dynamic AOIs in real-time to label individual faces to allow estimation of gaze switching within faces between the eyes, mouth, and nose, as well as between faces. We are most interested in extending the results of this work over static face stimuli by capturing gaze attention during live social interactions. In particular, we want to record gaze transitions within and among AOIs, i.e., on faces as well as on specific objects during social interactions. From the application point-of-view, dynamic AOI-based filtering can be applied to screen-magnifiers for low-vision users using automatic zooming of AOI of the context across frames (Aydin et al. 2020), and applications on neurodiverse (e.g. Autism Spectrum Disorder and Attention-Deficit/Hyperactivity Disorder) population during social interactions.

We begin by outlining existing studies that incorporate AOIs for the analysis of AOI-mapped gaze data. Next, we discuss existing methodologies to generate the AOI-mapped gaze location and present our implementation and evaluation for the extraction of dynamic AOI-mapped eye movement data.

2. RELATED WORK

Eye tracking experiments usually involve AOIs for the analysis of eye gaze data as they could reveal potential cognitive load and attentional patterns of the participants. Static AOIs are widely used to capture eye gaze metrics for detecting neurocognitive indices of Attention-Deficit / Hyperactivity Disorder (ADHD) symptomatology (G. Jayawardena et al. 2019), including various gaze features within AOIs to predict a diagnosis of ADHD with 86% accuracy. Similarly, (Gehrer et al. 2018) explored eye gaze patterns and statistically compared gaze transitions between static AOIs in a group of antisocial violent offenders through an emotion recognition task. Analysis of gaze patterns was based on four predefined AOIs, i.e., left eye, right eye, nose, and mouth. The eye gaze metrics were processed in various static AOIs of the face (such as eyes, mouth, and nose) to reveal insights into the underlying categorization process of emotions.

The analysis of eye movements using dynamic AOIs, such as in videos, has recently gained traction (Kurzhals et al. 2014; Burch et al. 2013; Zhang et al. 2018; Fichtel et al. 2019). This includes visually and statistically analyzed viewers' experience using eye movement data on video feeds (Marchant et al. 2009), and eye movements of 20 normal visioned subjects as each watched six movie clips, to examine the similarities in their viewing behaviors (Goldstein et al. 2007). The centers of interest in movie scenes were calculated using the areas of the best-fit bi-variate contour ellipses (Crossland et al. 2002; Timberlake et al. 2005) obtained from the gaze points of subjects. In terms of potential applications, the dynamically controlled magnification around these centers of interest can aid people with visual impairments.

Shot-based, spatio-temporal clustering (Kurzhals and Weiskopf 2013) of data has also been used to find potential AOIs in a time sequence to identify the objects that received more attention. The visual analytics which provides multiple coordinated views for analyzing various spatio-temporal aspects of gaze data on dynamic stimuli focused on identifying trends in the general viewing behavior, including objects with strong attentional focus. Similarly, (Tien et al. 2012) has measured the gaze path overlaps of task videos between the expert surgeon and third-party observers comparing gaze data files by calculating the Euclidean distance between the gaze points in pixels and by comparing with the target separation. For the analysis of cognitive workload and allocation of visual attention resources, studies (Shanmuga Vadivel et al. 2015; Khosravan et al. 2016) have utilized saliency models of viewers in conjunction with visual information from video frames. These algorithms extract salient objects which attract visual attention from videos and perform video segmentation utilizing eye tracking to obtain favorable object extraction.

The existing tools with AOIs to extract eye movement data for the analysis of gaze measurements, require users to draw boundaries of AOIs on eye tracking stimuli manually or use markers to define

AOIs in the space to generate AOI-mapped gaze locations. For instance, Tobii Pro Studio eye tracking software allows researchers to export both the raw eye tracking data and the AOI-mapped gaze locations for further processing and visualization. But it requires to draw boundaries of AOIs on static stimuli or use infrared (IR) markers to define AOIs in space to generate AOI-mapped gaze locations. Similarly (Lessing and Linge 2002; Stellmach et al. 2010) has introduced tools for defining AOIs and for extraction of AOI-mapped gaze locations including annotations for gaze data in dynamic eye tracking stimuli. These tools allow users to visualize the dynamic changes of AOIs and to explore the eye tracking data of multiple participants over time. In addition, (Weibel et al. 2012) introduced an approach that applies computer vision techniques to map their gaze coordinates to objects of interest using a template of the desired object derived from a selected single frame of the eye tracking stimuli video. If an AOI is detected in a frame, the tool can check whether the raw eye gaze coordinates for that video frame fall within the bounds of the AOI. This approach only works for pre-recorded eye tracking stimuli using the manual specification of the AOI template generated beforehand.

Though IR markers and tools provide the capability to manually define AOIs to extract AOI-mapped gaze locations, there are challenges when using them. For instance, when placing the IR markers in the field of view of the subject, there might be irregular surfaces or motion of the surface. Furthermore, manually annotating the AOIs frame-by-frame takes time and effort for large video sequences, which demands costly labor. To overcome these challenges, we propose a dynamic AOI-mapped gaze extraction workflow that uses deep neural networks for object detection. Since improvements in the field of computer vision have enabled the successful identification of objects and regions of possible interest, we incorporate computer vision techniques to detect dynamic AOIs in eye-tracking stimuli.

3. METHODOLOGY

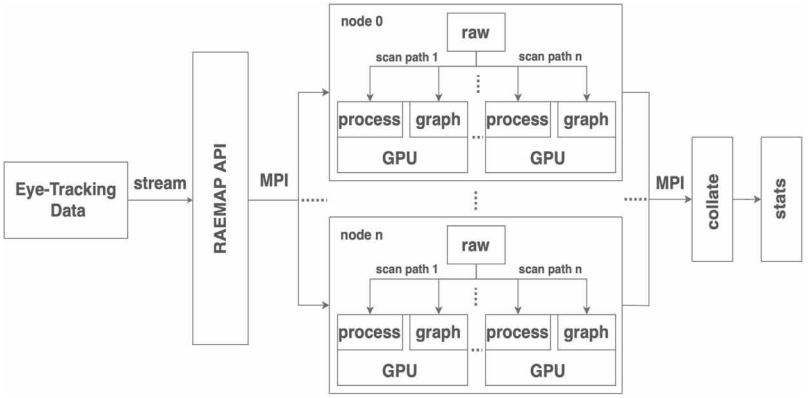
We utilize the extraction of dynamic AOI-mapped eye movement data using our eye movement processing framework RAEMAP (Jayawardena 2020), designed to analyze traditional positional gaze measurements as well as advanced eye gaze measurements in real time. The advanced gaze measurements include gaze transition entropy (Krejtz et al. 2015), and complex pupillometry measurements such as the index of pupillary activity (IPA) (Duchowski et al. 2018; Duchowski et al. 2020). The architecture of this pipeline is shown in Figure 1.

We select four convolutional neural networks based on real-time object detectors that were pre-trained on MS COCO (Lin et al. 2014) images dataset as the baseline models for dynamic AOI-mapped gaze extraction. The selected object detectors represent two categories of object detection, (1) one-stage object detection using dense prediction, and (2) two-stage object detection using sparse prediction. One-stage object detectors densely cover the space of possible image boxes using a fixed sampling grid, whereas two-stage object detectors classify image boxes at any position, scale, and aspect ratio. We use the YOLOv3 (Redmon and Farhadi 2018) method to represent a one-stage object detector and faster region-based convolutional neural networks (faster R-CNN) (Ren et al. 2015) to represent two-stage object detectors. Table 1 provides a summary of each object detector used.

Table 1. Object detectors

Method	Backbone	Head
Faster R-CNN	ResNet-50-FPN	Two-stage
Faster R-CNN	ResNet-101-FPN	Two-stage
Faster R-CNN	ResNet-50-DC5	Two-stage
YOLOv3	Darknet-53-FPN	One-stage

Figure 1. The Architecture of the RAEMAP (Jayawardena 2020). The API distributes tasks among the nodes using Message Passing Interface (MPI). Each node hosts an instance of the RAEMAP providing the functionality to extract raw gaze data, along with parallel processing of process and graph steps. The Process step calculate fixations, fixations in AOIs, saccade amplitudes, saccade duration, and IPA, whereas graph step generate visualizations. MPI gather function facilitates the aggregation of calculated eye gaze metrics in collate step, which provides data for statistical analysis in stats step.



We select three models for object instance segmentation to achieve a higher level of granularity in AOI using a variant of the faster R-CNN (Ren et al. 2015) called Mask R-CNN (He et al. 2017). Mask R-CNN extends Faster R-CNN by adding a branch for object mask prediction in parallel with the existing bounding box recognition process. Table 2 provides a summary of each object instance segmentation model used.

3.1 Faster R-CNN

We used three faster R-CNN (Ren et al. 2015) object detectors with a backbone of depth 50 and 101 ResNets (He et al. 2016). Among the three faster R-CNN object detectors used, two had a Feature Pyramid Network (FPN) (Lin et al. 2017) constructed on top, whereas one used a ResNet conv5 backbone with dilation in conv5 i.e. Dilated-C5 (DC5) (Dai et al. 2017). All the faster R-CNN models were trained on the COCO images dataset using an image scale of 600 pixels with the 3x schedule (37 COCO epochs).

Faster R-CNN object detectors have the capability of classifying image boxes at any position, scale, and aspect ratio. The Faster R-CNN is implemented with an $(n \times n)$ conv layer followed by two (1×1) conv layers. ReLUs are applied to the output of the $(n \times n)$ conv layer. It uses regression to achieve the bounding-box. We apply faster R-CNN in two stages, (1) Region Proposal Network (RPN) to generate candidate object bounding boxes, and (2) feature extraction using RoIPool (Ren et al. 2015) from each candidate box to performs classification and regression. We minimize loss function:

Table 2. Object instance segmentation models

Method	Backbone	Head
Mask R-CNN	ResNet-50-FPN	Two-stage
Mask R-CNN	ResNet-101-FPN	Two-stage
Mask R-CNN	ResNet-50-DC5	Two-stage

$$L\left(p_i, t_i\right) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, p_i^*\right) + \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \quad (1)$$

where, i is the index of an anchor, p_i is the predicted probability of anchor i being an object, p_i^* ground-truth label is 1 if anchor is positive and 0 otherwise, t_i is a vector representing the coordinates of the predicted bounding box, t_i^* is a vector representing the coordinates of the ground-truth box of a positive anchor, L_{cls} is the classification log loss over two classes (object vs. not object), $L_{reg}\left(t_i, t_i^*\right)$ is the regression loss, and $p_i^* L_{reg}$ is regression loss activated only for positive anchors and disabled otherwise. The classification log loss and regression loss are normalized by N_{cls} and N_{reg} and weighted by λ, a balancing parameter.

3.2 YOLOv3

We used the YOLOv3 object detector (Redmon and Farhadi 2018) with DarkNet-53 backbone. Darknet-53 uses successive (3×3) and (1×1) convolutional layers with shortcut connections and 53 convolutional layers. YOLOv3 was trained on the COCO image dataset. It passes an ($n \times n$) image once in a fully convolutional neural network (FCNN) for object detection. YOLOv3 selects the entire frame to apply a neural network to predict bounding boxes of detected objects and their probabilities. YOLOv3 splits the image into ($m \times m$) grids and generates boundaries around each detected object and their class probabilities. It uses a logistic regression with a threshold to calculate the class label of an object and a binary cross-entropy loss for each label for the classification loss.

3.3 Mask R-CNN

We utilize Mask R-CNN object instance segmentation models with a backbone of depth 50 and 101 ResNets with an FPN constructed on top, or ResNet conv5 backbone (i.e. DC5). For Mask R-CNN, we apply the same two-stage procedures of faster R-CNN in parallel to predict the class and box offset. Mask R-CNN also generates a binary mask for each AOI. During the training of Mask R-CNN, a multi-task loss on each sampled RoI is defined using:

$$L = L_{cls} + L_{box} + L_{mask} \quad (2)$$

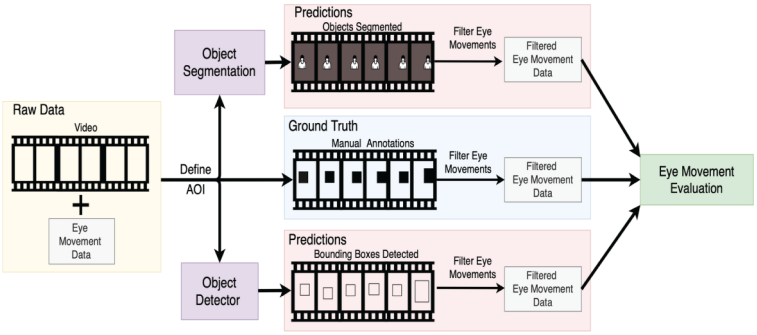
where L_{cls} is the classification loss and L_{box} is the bounding-box loss. L_{mask} is the average binary cross-entropy loss, and it allows the network to generate masks for every class without having a competition among classes.

3.4 Video Frame Object Detection

We first load the pre-trained object detectors and COCO object names (class labels) using OpenCV and Detectron (Lin et al. 2017). Next, we configure our RAEMAP eye movement processing pipeline to use these object detectors to dynamically detect AOIs in each frame. For each frame, the models output the COCO class label and location of detected objects in that frame in the form of bounding box coordinates. The goal here is to provide the capability of defining an object of interest in the eye tracking stimuli such that the RAEMAP can process the eye tracking stimuli to dynamically detect the corresponding AOIs.

Next, we define the object of interest in the eye tracking stimuli prior to the processing of eye movement data. Based on the defined object of interest (no restriction on the object of interest by default), the RAEMAP processes the eye tracking stimuli offline to detect corresponding dynamic

Figure 2. The Workflow of Object Detection and Segmentation for Dynamic AOI Filtering an Eye Movement Processing. Raw video sequence is given to the object detector and object instance segmentation model. Upon defining object(s) of interest, object detector outputs bounding box coordinates of each object detected, whereas, object instance segmentation model outputs identified pixel-wise masks of AOI detected in each frame. These dynamic bounding boxes and pixel-wise masks are considered as dynamic AOIs. Raw eye tracking data is filtered if they fall inside the boundaries of dynamic AOIs. For the evaluation, raw video sequences are manually annotated using BeaverDam (Shen 2016) software to create the ground truth of dynamic AOIs. Raw eye tracking data is then filtered if they fall inside the boundaries of manually annotated dynamic AOIs. Finally, detected dynamic AOIs (bounding boxes) are evaluated using intersection over union (IoU) and mean average precision (MAP), and filtered eye movements are evaluated using precision, recall, accuracy.



AOIs using the object detectors (see Figure 2). Note that when extracting dynamic AOIs from the selected models the default coordinate representation of the bounding boxes returned by the faster R-CNN and YOLOv3 models are different. The YOLOv3 object detector returns the bounding boxes in the form of (*x_center, y_center, width, height*), where *x_center* and *y_center* represent coordinates of the center of the bounding box, while *width* and *height* represent its width and height. In contrast, faster R-CNN object detectors returns the bounding boxes in the form of (*x_top_left, y_top_left, x_bottom_right, y_bottom_right*), where *x_top_left* and *y_top_left* represent the top-left coordinate of the bounding box, while *x_bottom_right* and *y_bottom_right* represent its bottom-right coordinate. Therefore, we reconfigured the RAEMAP to transform all bounding box coordinates into (*x_top_left, y_top_left, x_bottom_right, y_bottom_right*) form. For each video sequence, RAEMAP first identifies the bounding box coordinates of AOIs detected in each frame and writes them into a file. Next, RAEMAP extracts the raw eye gaze data which falls within the detected dynamic AOIs by checking if the gaze coordinate falls within the bounding boxes. The advantage of this approach is that it does not require prior annotation of the AOI or physical equipment to mark the boundaries of the AOI in dynamic eye tracking stimuli, thus eliminating the need for manual annotation of AOIs.

Similar to object detection, for each video sequence, RAEMAP identifies pixel-wise masks of AOI detected in each frame using object instance segmentation model. Next, we filter and extract the raw eye gaze data within the detected dynamic AOI by the gaze coordinate within the pixel-wise masks.

4. EVALUATION

4.1 Dataset

We evaluate our method using a publicly available eye tracking dataset (Hadizadeh et al. 2011) from 15 participants while watching 12 video sequences. Participants (2 F and 13 M) were aged between 18 and 30 (Hadizadeh et al. 2011) and had normal or corrected-to-normal vision. The eye movement data were collected using Locarna "Pt-Mini" head-mounted eye tracker with two 30 fps cameras, (1) the eye camera, and (2) the scene camera. Participants were presented with the twelve video segments sequentially in uncompressed at (352×288) resolution and 30fps. Each row in the original data files corresponds to a particular frame in the video sequence, and the columns provide the (*x, y*)

gaze coordinates of all participants at that frame. All coordinates were measured from the bottom left corner of current the video sequence. The dataset also provided a binary flag matrix indicating the accuracy of gaze locations. Gaze locations are considered as incorrect if, (1) the gaze location is out of frame boundaries, (2) the gaze location is at the frame boundaries or within 5 pixels of the frame boundaries, or (3) the gaze location remains constant for 30 consecutive frames. We pre-processed gaze data of each participant separately for each video and filtered out the incorrect gaze locations from the gaze data based on the binary flag matrix. The goal here was to separate the gaze locations of each participant to pass into the RAEMAP, since the calculations of eye gaze metrics of each participant require a separate processing of the raw data.

4.2 Dynamic AOI Detection

We selected four video sequences (Foreman, Bus, Mother and Daughter, and Hall Monitor) out of the twelve sequences available to evaluate our method. These videos were selected as they had dominant objects to draw boundaries for AOIs, which was already a class label in the COCO names list. We identified one dominant object from each video sequence and defined it as the AOI label for that video sequence.

For each video sequence, we applied an object detector, and dynamically detect the AOI at each frame. We repeated this step for each object detector and obtained a prediction for the bounding box coordinates at each frame of the video sequences. Similarly, we applied object instance segmentation models, and obtain a prediction for the pixel-wise mask for the dynamic AOI at each frame. Next, we utilized the RAEMAP to filter and extract eye gaze data within the bounding boxes or pixel-wise masks of dynamic AOI.

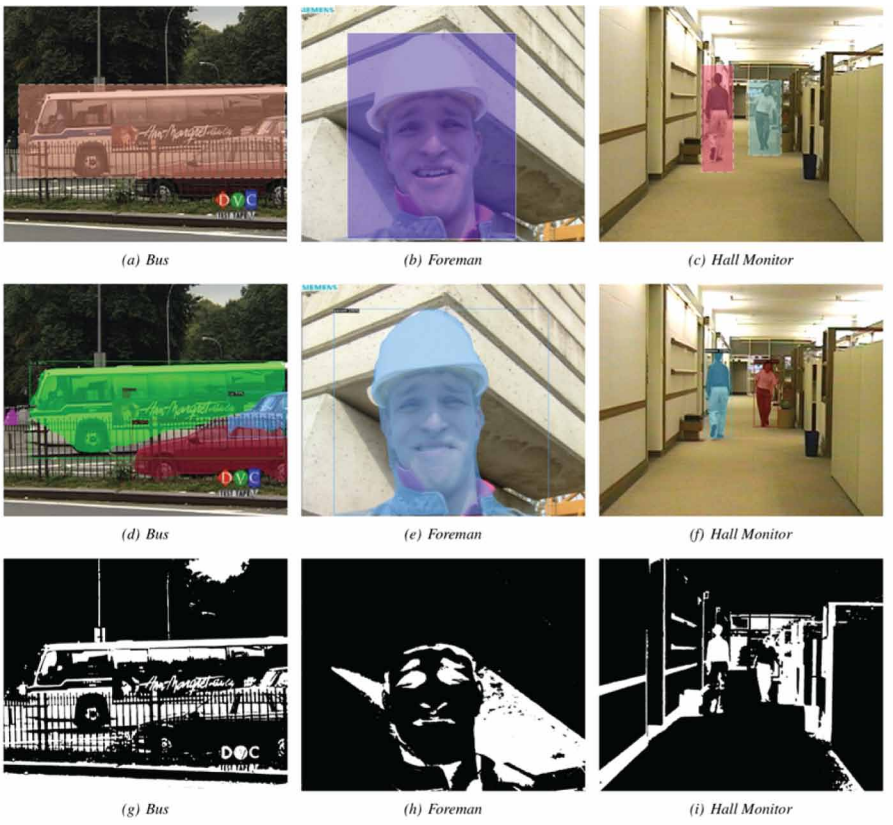
When evaluating the dynamic AOIs detected, we created a ground truth dataset by manually annotating each video sequence with the expected AOI in the form of bounding boxes using BeaverDam (Shen 2016) video annotation tool. BeaverDam is designed for drawing bounding boxes on video frames and annotating them with class labels. It also allows arbitrary annotation of frames in the video sequence as it provides a parameter indicating whether linear interpolation should be continued for each AOI annotated arbitrarily. Video annotations made in BeaverDam can be exported in JSON file format. Exported annotations consist of bounding box coordinates at each marked frame along with the linear interpolation parameter. We generate four JSON objects corresponding to each video file, and linearly interpolate the bounding boxes between the start and end frames to obtain a continuous annotation. We use this interpolated result as the ground truth for evaluating each object detector and subsequently extracted the gaze data using them.

Next, we use intersection over union (IoU) and mean average precision (MAP) as the evaluation metrics of dynamic AOIs generated by object detectors. IoU is a measurement of the overlap between two boundaries (see equation (3)), whereas the MAP is a metric used to evaluate object detectors:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{3}$$

We use IoU to calculate how much of the boundaries predicted using each object detector overlaps with the ground truth bounding boxes. We empirically define IoU threshold to be 0.5 to classify the predicted bounding boxes. The predicted bounding box is classified as true positive (TP) if IoU $\geq$ 0.5 and false positive (FP) otherwise. The precision and recall are calculated based on the classification of the predicted bounding boxes. Finally, we calculate MAP in both COCO style and Pascal VOC2008 (Everingham et al.) style using Average Precision (AP). In Pascal VOC2008, an average for the 11-point interpolated AP is calculated, whereas, in COCO, an average for the 101-point interpolated AP is calculated.

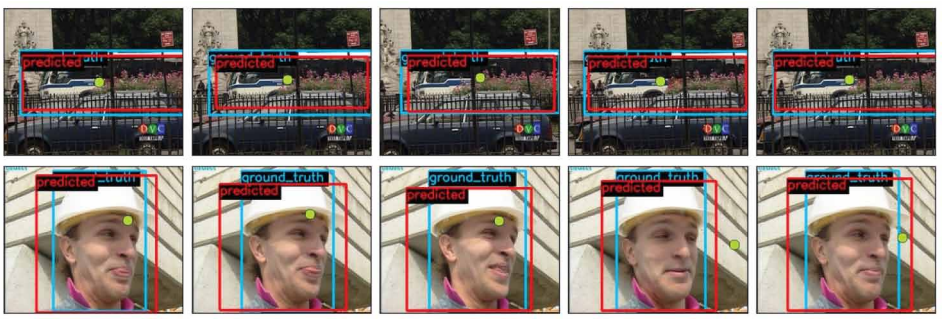
Figure 3. AOI in Video Sequences. (a,b,c) Manually annotated AOIs using BeaverDam annotation tool. (d,e,f) Bounding boxes representing a lower level of granularity, and pixel-wise masks (shaded area) representing a higher level of granularity of dynamic AOI. (g,h,i) Salience of objects revealed by OpenCV's static saliency spectral residual detector.



4.3 Eye-Movement Extraction

After detecting dynamic AOIs on video sequences, we pass the raw eye tracking data to the RAEMAP and extracts eye gaze data that fall within the bounding boxes of dynamic AOIs as generated by each object detector and eye gaze data that falls within the pixel-wise masks of dynamic AOIs as generated

Figure 4. Dynamic AOI Detection. Visualization of manually annotated AOIs (blue), AOIs detected by faster R-CNN object detector (red), the gaze positions of a participant (green) in five consecutive video frames in Bus and Foreman video sequences.



by each object instance segmentation model. We also compute traditional positional gaze metrics such as fixation count and fixation duration using the extracted gaze data.

Next, we pass the ground truth bounding boxes of AOIs to the RAEMAP and extract eye gaze data within in the form of (x_i, y_i, t_i) where x_i and y_i represent the coordinates of the gaze position at time t_i. To evaluate the dynamic AOI-mapped eye movements, we classify the eye movements according to the confusion matrix in Table 3. We use standard information retrieval domain evaluation metrics such as precision, recall, and accuracy for the evaluation of filtered eye movements.

Table 3. Confusion matrix for eye movements evaluation

		Ground Truth AOI	
		Falls Within	**Falls Outside**
Predicted AOI	**Falls Within**	TP	FP
	Falls Outside	FN	TN

For the comparison of the dynamic AOI-mapped eye movements filtered from bounding boxes and pixel-wise masks, we calculate the captured percentage of eye movements when using each filtering technique. We expect an overall percentage reduction of eye movement data filtered from the baseline bounding boxes, object detectors, and object instance segmentation models as the level of granularity changes. We are interested in investigating which AOI method would give us the highest percentage of eye movements filtered from these models.

5. RESULTS

5.1 Dynamic AOI Detection

Figures 3(a), 3(b), and 3(c) show manually annotated objects in a single frame of *Bus*, *Foreman*, and *Hall Monitor* video sequences. In comparison, Figures 3(d), 3(e), and 3(f) show detected objects in a single frame of *Bus*, *Foreman*, and *Hall Monitor* video sequences using an object detector and an object instance segmentation model.

Table 4 shows the MAP values in both COCO style and Pascal VOC2008 style for each object detector. AP corresponds to the average AP for IoU from 0.5 to 0.95 with a step size of 0.05, AP@.50 corresponds to the average AP for IoU = 0.5, and AP@.75 corresponds to the average AP for IoU = 0.75. Faster R-CNN object detector with ResNet-101-FPN backbone has the highest AP in both COCO style and PASCAL style (see Table 4) with the $AP \geq 0.19$.

Table 4. Comparison of bounding box AP of Object Detectors

Method	**Backbone**	**COCO Style**			**Pascal Style**		
		AP	**AP@.50**	**AP@.75**	**AP**	**AP@.50**	**AP@.75**
Faster R-CNN	ResNet-50-FPN	0.1812	0.3707	0.1768	0.1918	0.3926	0.1989
Faster R-CNN	ResNet-101-FPN	**0.1998**	0.3877	0.1985	**0.2180**	0.4000	0.2136
Faster R-CNN	ResNet-50-DC5	0.1406	0.3290	0.1111	0.1566	0.3238	0.1238
YOLOv3	Darknet-53-FPN	0.1269	0.3083	0.1123	0.1430	0.3123	0.1388

5.2 Eye Movement Extraction

Figure 4 illustrates the dynamic AOIs detected from a faster R-CNN object detector with ResNet-101-FPN backbone in comparison with the manually annotated ground truth AOIs. Red color bounding boxes indicate the predicted bounding boxes, whereas blue color bounding boxes indicate the ground truth. The green color circle indicates the gaze position of that frame.

Table 5 shows the precision, recall, and accuracy of eye movements extracted using dynamic AOIs generated by each object detector. The Faster R-CNN object detector with ResNet-101-FPN backbone filters eye movement data with the highest accuracy of 64.5%.

Table 5. Comparison of Filtered Eye Movements of Object Detectors

Method	Backbone	Precision	Recall	Accuracy
Faster R-CNN	ResNet-50-FPN	0.648	0.717	0.644
Faster R-CNN	ResNet-101-FPN	0.646	0.713	**0.645**
Faster R-CNN	ResNet-50-DC5	0.647	0.697	0.639
YOLOv3	Darknet-53-FPN	0.637	0.717	0.641

Table 6 shows the comparison of the dynamic AOI-mapped eye movements filtered from bounding boxes and pixel-wise masks, in terms of the percentage of eye movements captured when using each filtering technique.

Table 6. Percentage of eye movements filtered when using bounding boxes and pixel-wise masks as dynamic AOIs

Method	Backbone	Eye Movements%
Object Detection	ResNet-50-FPN ResNet-101-FPN ResNet-50-DC5	60.643 60.224 58.989
Segmentation	ResNet-50-FPN ResNet-101-FPN ResNet-50-DC5	30.574 30.088 30.531
Baseline Boxes	-	56.183

6. DISCUSSION

Our evaluation of dynamic AOIs generated by object detectors indicate that a faster R-CNN object detector with ResNet-101FPN backbone achieves the highest AP in both COCO style and PASCAL style (see Table 4). Though faster R-CNN object detector with ResNet-101-FPN backbone achieves the highest AP rate, we observe it to be slower compared to one-stage detector YOLOv3, supporting the literature that two-stage object detectors are typically slower (Soviany and Ionescu 2018). Two-stage object detectors are slow because they generate regions of interest in the first stage and classify objects and find bounding-boxes by regression in the second stage. On the other hand, one-stage object detectors treat object detection as a simple regression problem by learning the class probabilities and bounding box coordinates, thus reaching lower AP rates, but performing much faster than two-stage

object detectors. Our results indicate that two-stage object detectors, despite being slow, perform the best in classifying objects and finding bounding-boxes as dynamic AOIs. However, further evaluation is required with a larger representation of one-stage object detectors since we only used a single one-stage object detector, YOLOv3 for the evaluation.

Figure 4 illustrates dynamic AOIs detected from faster R-CNN (ResNet-101-FPN) object detector in comparison with the manually annotated ground truth AOIs. We observe that some eye movements that fall within the ground truth bounding box may not fall within the predicted bounding box, and they are classified as FN. This happens when either there is no dynamic AOI detected in that frame, or the detected object is surrounded by a tighter bounding box compared to the ground truth. Also, eye movements that do not fall within the ground truth bounding box may fall within the predicted bounding box, since the predicted bounding box is relaxed compared to the ground truth bounding box (see the last image in the second row of Figure 4). Those eye movements are classified as FP. Eye movements extracted by all four object detectors, do not differ much in terms of evaluation precision, recall, or accuracy. As shown in Table 5, a faster R-CNN object detector with ResNet-101-FPN backbone filters eye movements data with the highest accuracy of 64.5%. Since eye movement classification is highly dependent on both manually defined bounding boxes and bounding boxes found by the object detector, we believe it is essential to retrain the object detectors to find better bounding boxes instead of using pre-trained object detectors.

The comparison of the percentage of eye movements when using each filtering technique (Table 6) shows that dynamic AOI generated by object detectors had the least reduction of eye movements compared to the baseline. We expected an overall percentage reduction of eye movement data as the level of granularity increased. Though the difference of the percentage reduction of eye movements filtered by dynamic AOI generated by object detectors and the baseline is around 3%. The difference is caused by tighter bounds of the manually annotated bounding boxes. Among all three methods, dynamically generated AOI using pixel-wise masks yield the highest percentage of eye movements reduction, since it represents the polygonal shape of the object with a tighter bound compared to the bounding boxes. Figures 3(d), 3(e), and 3(f) show the pixel-wise masks (shaded area) representing a higher level of granularity of dynamic AOIs generated by object instance segmentation models. Figure 3(g), 3(h), and 3(i) illustrate salience revealed by OpenCV's static saliency spectral residual detector. Though salience revealed by the eye movements of viewers could be utilized to define an AOI, they can introduce extra bias, and additional steps before generating the AOI. In contrast, we incorporate pre-trained object detectors, and object instance segmentation models, generalizing the overall process of detecting dynamic AOI.

Interestingly, we observe in both evaluation criteria, the Faster R-CNN object detector with ResNet-101-FPN backbone scored the highest with one of the lowest percentage reduction of eye movements around 3% compared to the baseline. Based on the performance in both evaluation criteria, we choose the Faster R-CNN object detector with ResNet-101-FPN backbone as the object detector in the RAEMAP. Apart from the RAEMAP integration, the proposed pipeline with faster R-CNN (ResNet-101-FPN) as the object detector could be used for offline extraction of eye gaze metrics from dynamic AOIs.

7. CONCLUSION

In this study, we present pre-trained object detectors and object instance segmentation models for the detection of dynamic AOI using bounding boxes and precise masks in video streams representing two levels of granularity in AOI. We evaluate the object detectors, and object instance segmentation models integrated into the RAEMAP eye movement analysis pipeline to filter eye movement data within the polygonal boundaries of detected dynamic AOI. Our results in terms of dynamic AOI detection and eye movement extraction indicate that faster R-CNN with ResNet-101-FPN backbone

object detector performs the best in classifying objects and finding bounding-boxes as dynamic AOIs, and it suits the best for the RAEMAP integration.

We compare the two levels of granularity in dynamic AOI in terms of the percentage captured in filtered eye movements compared to a baseline model. Our results indicate that the dynamic AOI-mapped eye movements generated by object detectors could capture a maximum of 60% of eye movements, whereas object instance segmentation models captured only about 30% of eye movements. Since we observe multiple levels of reduction in filtered eye movements as the granularity of the dynamic AOIs increases, we anticipate this would contribute towards a layered analysis of both positional and advanced eye gaze metrics in the future.

Our proposed methodology for detecting dynamic AOIs is the initial step towards extracting eye movement data from dynamically detected AOIs in real time. The proposed work will help develop dynamic AOI-mapped eye movement filtering and transition workflow. To identify the AOIs, we will apply our proposed dynamic deep learning and computer vision based, real-time AOI detection architecture. Since computer vision techniques can be adapted to accurately detect a wide range of objects, we will apply them to extract dynamic AOI-mapped gaze data. Specifically, in the future we will use deep transfer learning to remodel existing image classifiers for dynamic AOI detection in real time.

REFERENCES

Aydin, A. S., Feiz, S., Ashok, V., & Ramakrishnan, I. (2020). Towards making videos accessible for low vision screen magnifier users. *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 10–21. doi:10.1145/3377325.3377494

Burch, M., Kull, A., & Weiskopf, D. (2013). Aoi rivers for visualizing dynamic eye gaze frequencies. *Computer Graphics Forum, 32,* 281–290. doi:10.1111/cgf.12115

Crossland, M. D., & Rubin, G. S. (2002). The use of an infrared eye-tracker to measure fixation stability. *Optometry and Vision Science*, *79*(11), 735–739. doi:10.1097/00006324-200211000-00011 PMID:12462542

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. *Proceedings of the IEEE international conference on computer vision*, 764–773.

Duchowski, A. T., Krejtz, K., Gehrer, N. A., Bafna, T., & Bækgaard, P. (2020). The low/high index of pupillary activity. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12.

Duchowski, A. T., Krejtz, K., Krejtz, I., Biele, C., Niedzielska, A., Kiefer, P., Raubal, M., & Giannopoulos, I. (2018). The index of pupillary activity: measuring cognitive load vis-à-vis task difficulty with pupil oscillation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. doi:10.1145/3173574.3173856

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2008). *The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results.* http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html

Fichtel, E., Lau, N., Park, J., Parker, S. H., Ponnala, S., Fitzgibbons, S., & Safford, S. D. (2019). Eye tracking in surgical education: Gaze-based dynamic area of interest can discriminate adverse events and expertise. *Surgical Endoscopy*, *33*(7), 2249–2256. doi:10.1007/s00464-018-6513-5 PMID:30341656

Gehrer, N. A., Schönenberg, M., Duchowski, A. T., & Krejtz, K. (2018). Implementing innovative gaze analytic methods in clinical psychology: A study on eye movements in antisocial violent offenders. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research Applications*. Association for Computing Machinery. doi:10.1145/3204493.3204543

Goldberg, J. H., & Kotval, X. P. (1999). Computer interface evaluation using eye movements: Methods and constructs. *International Journal of Industrial Ergonomics*, *24*(6), 631–645. doi:10.1016/S0169-8141(98)00068-7

Goldstein, R. B., Woods, R. L., & Peli, E. (2007). Where people look when watching movies: Do all viewers look at the same place? *Computers in Biology and Medicine*, *37*(7), 957–964. doi:10.1016/j.compbiomed.2006.08.018 PMID:17010963

Hadizadeh, H., Enriquez, M. J., & Bajic, I. V. (2011). Eye-tracking database for a set of standard video sequences. *IEEE Transactions on Image Processing*, *21*(2), 898–903. doi:10.1109/TIP.2011.2165292 PMID:21859619

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hessels, R. S., Kemner, C., Van Den Boomen, C., & Hooge, I. T. (2016). The area-of-interest problem in eyetracking research: A noise-robust solution for face and sparse stimuli. *Behavior Research Methods*, *48*(4), 1694–1712. doi:10.3758/s13428-015-0676-y PMID:26563395

Jayawardena, G. (2020). Raemap: Real-time advanced eye movements analysis pipeline. In *Symposium on Eye Tracking Research and Applications 2020*. ACM. doi:10.1145/3379157.3391992

Jayawardena, G., Michalek, A., & Jayarathna, S. (2019). Eye tracking area of interest in the context of working memory capacity tasks. *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, 208–215. doi:10.1109/IRI.2019.00042

Jayawardena, G., & Michalek, A. M. P. (2020). Pilot study of audiovisual speech-in-noise (sin) performance of young adults with adhd. In *Proceedings of the 2020 ACM Symposium on Eye Tracking Research Applications*. Association for Computing Machinery. doi:10.1145/3379156.3391373

Just, M. A. & Carpenter, P. A. (1980). A theory of reading: From eye fixations to comprehension. *Psychological Review, 87*(4), 329.

Khosravan, N., Celik, H., Turkbey, B., Cheng, R., Mccreedy, E., Mcauliffe, M., Bednarova, S., Jones, E., Chen, X., & Choyke, P. (2016). Gaze2segment: a pilot study for integrating eye-tracking technology into medical image segmentation. In Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging. Springer.

Komogortsev, O., Holland, C., Jayarathna, S., & Karpov, A. (2013). 2d linear oculomotor plant mathematical model: Verification and biometric applications. *ACM Transactions on Applied Perception (TAP), 10*(4), 27.

Kosch, T., Hassib, M., Buschek, D., & Schmidt, A. (2018). Look into my eyes: using pupil dilation to estimate mental workload for task complexity adaptation. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery. doi:10.1145/3170427.3188643

Krejtz, K., Duchowski, A., Szmidt, T., Krejtz, I., González Perilli, F., Pires, A., Vilaro, A., & Villalobos, N. (2015). Gaze transition entropy. *ACM Transactions on Applied Perception (TAP), 13*(1), 4.

Krejtz, K., Duchowski, A. T., Niedzielska, A., Biele, C., & Krejtz, I. (2018). Eye tracking cognitive load using pupil diameter and microsaccades with fixed gaze. *PloS One, 13*(9).

Kurzhals, K., Bopp, C. F., Bässler, J., Ebinger, F., & Weiskopf, D. (2014). Benchmark data for evaluating visualization and analysis techniques for eye tracking for video stimuli. *Proceedings of the fifth workshop on beyond time and errors: novel evaluation methods for visualization*, 54–60. doi:10.1145/2669557.2669558

Kurzhals, K., & Weiskopf, D. (2013). Space-time visual analytics of eye-tracking data for dynamic stimuli. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2129–2138. doi:10.1109/TVCG.2013.194 PMID:24051779

Lessing, S. & Linge, L. (2002). Iicap? a new environment for eye tracking data analysis. Academic Press.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer.

Marchant, P., Raybould, D., Renshaw, T., & Stevens, R. (2009). Are you seeing what i'm seeing? an eye-tracking evaluation of dynamic scenes. *Digital Creativity*, *20*(3), 153–163. doi:10.1080/14626260903083611

Mccarley, J. S., & Kramer, A. F. (2008). Eye movements as a window on perception and cognition. *Neuroergonomics*, *3*, 95.

Nguyen, A., Chandran, V., & Sridharan, S. (2004). Visual attention based roi maps from gaze tracking data. *2004 International Conference on Image Processing, 2004. ICIP'04*, 3495–3498. doi:10.1109/ICIP.2004.1421869

Noton, D., & Stark, L. (1971). Eye movements and visual perception. *Scientific American*, *224*(6), 34–43. PMID:5087474

Privitera, C. M., & Stark, L. W. (2000). Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(9), 970–982. doi:10.1109/34.877520

Radach, R., Hyona, J., & Deubel, H. (2003). *The mind's eye: Cognitive and applied aspects of eye movement research*. Elsevier.

Redmon, J., & Farhadi, A. (2018). *Yolov3: An incremental improvement.* arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 91–99.

Santella, A., Agrawala, M., Decarlo, D., Salesin, D., & Cohen, M. (2006). Gaze-based interaction for semiautomatic photo cropping. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 771–780.

Shanmuga Vadivel, K., Ngo, T., Eckstein, M., & Manjunath, B. (2015). Eye tracking assisted extraction of attentionally important objects from videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3241–3250.

Shen, A. (2016). *Beaverdam: Video annotation tool for computer vision training labels* (M.S. thesis). EECS Department, University of California, Berkeley.

Soviany, P., & Ionescu, R. T. (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE. doi:10.1109/SYNASC.2018.00041

Stellmach, S., Nacke, L., & Dachselt, R. (2010). Advanced gaze visualizations for three-dimensional virtual environments. *Proceedings of the 2010 symposium on eye-tracking research & Applications,* 109–112. doi:10.1145/1743666.1743693

Tien, G., Atkins, M. S., & Zheng, B. (2012). Measuring gaze overlap on videos between multiple observers. *Proceedings of the symposium on eye tracking research and applications*, 309–312. doi:10.1145/2168556.2168623

Timberlake, G. T., Sharma, M. K., Grose, S. A., Gobert, D. V., Gauch, J. M., & Maino, J. H. (2005). Retinal location of the preferred retinal locus relative to the fovea in scanning laser ophthalmoscope images. *Optometry and Vision Science*, *82*(3), E177–E187. doi:10.1097/01.OPX.0000156311.49058.C8 PMID:15767869

Van Der Stigchel, S., Rommelse, N., Deijen, J., Geldof, C., Witlox, J., Oosterlaan, J., Sergeant, J., & Theeuwes, J. (2007). Oculomotor capture in adhd. *Cognitive Neuropsychology*, *24*(5), 535–549. doi:10.1080/02643290701523546 PMID:18416506

Van Orden, K. F., Limbert, W., Makeig, S., & Jung, T.-P. (2001). Eye activity correlates of workload during a visuospatial memory task. *Human Factors*, *43*(1), 111–121. doi:10.1518/001872001775992570 PMID:11474756

Wang, W., Song, H., Zhao, S., Shen, J., Zhao, S., Hoi, S. C., & Ling, H. (2019). Learning unsupervised video object segmentation through visual attention. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3064–3074. doi:10.1109/CVPR.2019.00318

Weibel, N., Fouse, A., Emmenegger, C., Kimmich, S., & Hutchins, E. (2012). Let's look at the cockpit: exploring mobile eye-tracking for observational research on the flight deck. *Proceedings of the Symposium on Eye Tracking Research and Applications*, 107–114. doi:10.1145/2168556.2168573

Zhang, X., Yuan, S.-M., Chen, M.-D., & Liu, X. (2018). A complete system for analysis of video lecture based on eye tracking. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 49056–49066. doi:10.1109/ACCESS.2018.2865754

Gavindya Jayawardena is a PhD student at Old Dominion University, where she is associated with Web Science and Digital Libraries (WS-DL) research group. Her research interest includes eye tracking, machine learning, and data science. She received the best student poster paper award at ACM/IEEE JCDL 2020 and the best paper award at IEEE MerCon 2019. She is currently researching on real-time eye movement analysis methodologies. Gavindya is a student member of ACM and Computer Science Graduate Society at Old Dominion University.

Sampath Jayarathna (PhD) is an Assistant Professor of Computer Science at Old Dominion University, where he is associated with Web Science and Digital Libraries (WS-DL) research group. Prior to joining ODU in 2018, he was an Assistant Professor at California State Polytechnic University Pomona. Dr. Jayarathna earned a PhD in Computer science from the Texas A&M University College Station in 2016. His research interest includes machine learning, information retrieval, data science, eye tracking, and brain computer interfacing. Dr. Jayarathna has published in venues such as ACM CSUR, CIKM, CHIIR, JCDL, ETRA, IRI, Big Data, ACM TAP, and IEEE TBE. He received the best student paper award at IEEE IRI 2019, best paper award at MerCon 2019, and best student paper nomination at ACM/IEEE JCDL 2015. His service includes, teaching coding to incarcerated men at Norfolk City Jail, chair/co-chair of organizing committees at ACM CHIIR, JCDL, IRI conferences. He currently teaches python coding at the NET Academy for students meeting Title I criteria as neglected and delinquent youth. Dr. Jayarathna is a member of ACM, IEEE, and Sigma XI.