Automated Filtering of Eye Gaze Metrics from Dynamic Areas of Interest

Gavindya Jayawardena
Computer Science
Old Dominion University
Norfolk, VA 23529
gavindya@cs.odu.edu

Sampath Jayarathna
Computer Science
Old Dominion University
Norfolk, VA 23529
sampath@cs.odu.edu

***Abstract*—Eye-tracking experiments usually involves areas of interests (AOIs) for the analysis of eye gaze data as they could reveal potential cognitive load, and attentional patterns yielding interesting results about participants. While there are tools to define AOIs to extract eye movement data for the analysis of gaze measurements, they may 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. In this paper, we introduce a novel method to dynamically filter eye movement data from AOIs for the analysis of advanced eye gaze metrics. We incorporate pre-trained object detectors for offline detection of dynamic AOIs in dynamic eye-tracking stimuli such as video streams. We present our implementation and evaluation of object detectors to find the best object detector to be integrated in a real-time eye movement analysis pipeline to filter eye movement data that falls within the polygonal boundaries of detected dynamic AOIs. Our results indicate the utility of our method by applying it to a publicly available dataset.**

***Keywords*- Eye Movements, Dynamic AOIs, Computer Vision**

I. Introduction

Eye-tracking can reveal objective and quantifiable information about the quality, predictability, and consistency of underlying covert process of the human brain when carrying out cognitively demanding tasks [1], [2], [3]. According to the eye-mind hypothesis [4], observers attend where their eyes are fixating. Thus, eye-tracking measurements enable us to investigate the cognitive behavior when visually exploring a stimulus. With the advancement of eye-tracking technology, gaze tracking measurements have become reliable and accurate.

Eye gaze measurement includes a number of metrics relevant to oculomotor control [5] such as saccadic trajectories, fixations, and other relevant measures including velocity, duration, amplitude, pupil dilation [6]. Studies have shown that the size of the pupil diameter correlates to the task complexity [7] enabling the use of pupillary behavior as bio-markers of mental workload when completing a task. Several studies [8], [9] have incorporated eye-tracking to obtain insights into underlying covert processes and exploration 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 calculated, and statistical significance of computed numerous metrics are evaluated [8], [9].

Eye-tracking experiments add AOIs to the analysis process to extract eye gaze metrics. An AOI is a region of a stimuli that is used to study the eye gaze metrics and link eye movement measures to the part of the area of the stimuli [10]. Studies in visual attention and eye movements [11], [12] 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 (a user interface component in this example) 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 [13]. In addition, the fixation frequency and blink frequency indicates cognitive workload [14].

Visual attention allocation may differ from subject to subject, thus enabling grouping of gaze locations with k-means clustering to determine the AOIs [15], and using different image processing algorithms [12] along with clustering to automatically identify AOIs. These methods are used when the primary focus is on detecting fixations sequences within identified AOIs [15], [12]. While there are methodologies to define AOIs to extract eye movement data for the analysis of gaze measurements, they require users to draw boundaries of AOIs on eye tracking stimuli manually or use markers to define AOIs in the space or post process the gaze locations to determine AOIs using clustering to generate AOI-mapped gaze locations. In contrast, we propose a computer vision with deep neural network approach to identify the AOIs in video streams to filter gaze locations that fall into the identified AOIs for the analysis of both positional and advanced eye gaze metrics. From the application point-of-view, dynamic AOI-based filtering can be applied in screen-magnifiers for low-vision users using automatic zooming of AOI of the context across frames [16].

We begin by outlining existing studies that incorporate AOIs for the analysis of AOI-mapped gaze data. Upon doing so, we discuss existing methodologies to generate AOI-mapped gaze location. Then we present our implementation for the

978-1-7281-1054-7/20/$31.00 ©2020 IEEE
DOI 10.1109/IRI49571.2020.00018

extraction of dynamic AOI-mapped eye movement data. This work is based on our Real-Time Advanced Eye Movements Analysis Pipeline (RAEMAP) [17], designed to analyze traditional positional gaze measurements as well as advanced eye gaze measurements.

II. Background

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 [18], including various gaze features within AOIs to predict a diagnosis of ADHD with 86% accuracy. Similarly, [8] 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 has been based on four predefined AOIs, i.e., left eye, right eye, nose, and mouth. Participants have been asked to label the emotion of the given image and antisocial, violent offenders, and participants of control group have shown similarities in eye gaze metrics in some AOIs (i.e. eyes). 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.

Though a majority of past studies have analyzed eye-movements using static AOIs, the analysis of eye-movements using dynamic AOIs, such as in videos, has recently gained traction. This includes visually and statistically analysed viewers' experience using eye movement data on video feeds [19], and eye movements of 20 normal visioned subjects as each watched six movie clips, to examine the similarities in their viewing behaviors [20]. The centers of interest in movie scenes were calculated using the areas of the best-fit bi-variate contour ellipses [21], [22] obtained from the gaze points of subjects. In terms of potential applications, the dynamic controlled magnification around these centers of interest can aid people with visual impairments.

Shot-based, spatio-temporal clustering [23] 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, [24] 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.

The existing tools which are capable of defining 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[1] 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 researchers 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 [25], [26] have introduced tools for defining AOIs and for extraction of AOI-mapped gaze locations including annotations for gaze data in dynamic eye tracking stimuli. This tools allow users to visualize the dynamic changes of AOIs and to explore eye tracking data of multiple participants over time.

In addition, [27] introduced an approach which applies computer vision techniques to map their gaze coordinates to objects of interest using template of a 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 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 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 successful identification of objects and regions of possible interest, we incorporate computer vision techniques to detect dynamic AOIs in eye-tracking stimuli.

III. Methodology

We base our implementation of extraction of dynamic AOI-mapped eye movement data using our work of RAEMAP [17], 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 [28], and complex pupillometry measurements such as index of pupillary activity (IPA) [29] which indicate cognitive load. It is capable of processing raw gaze data streamed from various eye trackers in real-time and calculating eye gaze metrics such as fixation count and fixation duration. The original architecture of this pipeline is shown in Figure 1.

Since computer vision techniques can be adopted 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

[1] https://www.tobiipro.com/learn-and-support/learn/tobii-pro-studio/

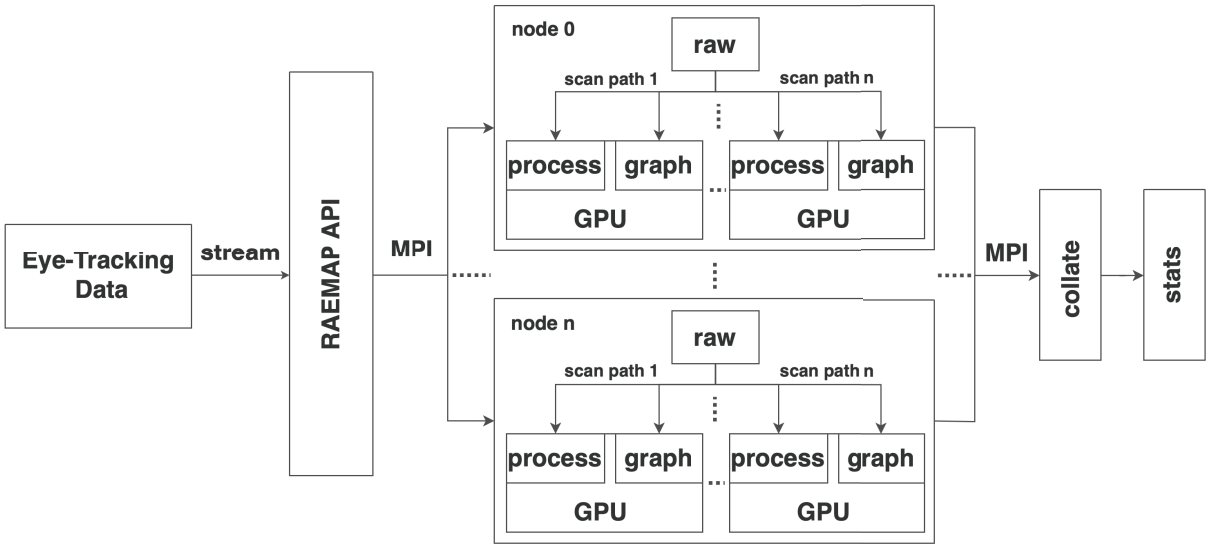


Figure 1. The Architecture of the RAEMAP [17]. The API distributes tasks among the nodes using MPI. Each node hosts an instance of the RAEMAP providing the functionality *raw* to extract raw gaze data, along with parallel processing of process and graph steps. *Process* step calculate fixations, fixations 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.

dynamic AOIs by checking if the gaze coordinate falls within any dynamic AOIs's polygonal boundaries (see Figure 2).

TABLE I
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

We select four CNN-based real-time object detectors that were pre-trained on MS COCO [31] images dataset as the baseline models for dynamic AOI-mapped gaze extraction. The selected object detectors represent the 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 YOLOv3 [32] method to represent one-stage object detector and faster region based convolutional neural networks (faster R-CNN) [33]) to represent two-stage object detectors. Table I provides a summary of each object detector used.

*1) **Faster R-CNN**:* We used three faster R-CNN [33] object detectors with backbone of depth 50 and 101 ResNets [34]. Among the three faster R-CNN object detectors we used, two had a Feature Pyramid Network (FPN) [35] constructed on top, whereas one used a ResNet conv5 backbone with dilation in conv5 i.e. Dilated-C5 (DC5) [36]. All the faster R-CNN models were trained on COCO images dataset using an image scale of 600 pixels with the 3x schedule (37 COCO epochs) [37].

Faster R-CNN object detectors have the capability of classifying image boxes at any position, scale, and aspect ratio. The architecture of Faster R-CNN is implemented with an $(n \times n)$ conv layer followed by two (1×1) conv layers [33]. ReLUs are applied to the output of the $(n \times n)$ conv layer. It uses regression to achieve the bounding-box.

*2) **YOLOv3**:* We used YOLOv3 object detector [32] with DarkNet-53 backbone. Darknet-53 uses successive (3×3) and (1×1) convolutional layers with shortcut connections and it has 53 convolutional layers. YOLOv3 has been trained on 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 objects and their class probabilities [32]. It uses a logistic regression with a threshold to calculate the class label of an object. It uses the binary cross-entropy loss for each label for the classification loss.

A. Implementation

We first load the pre-trained object detectors and COCO object names (class labels) using OpenCV and Detectron [37].

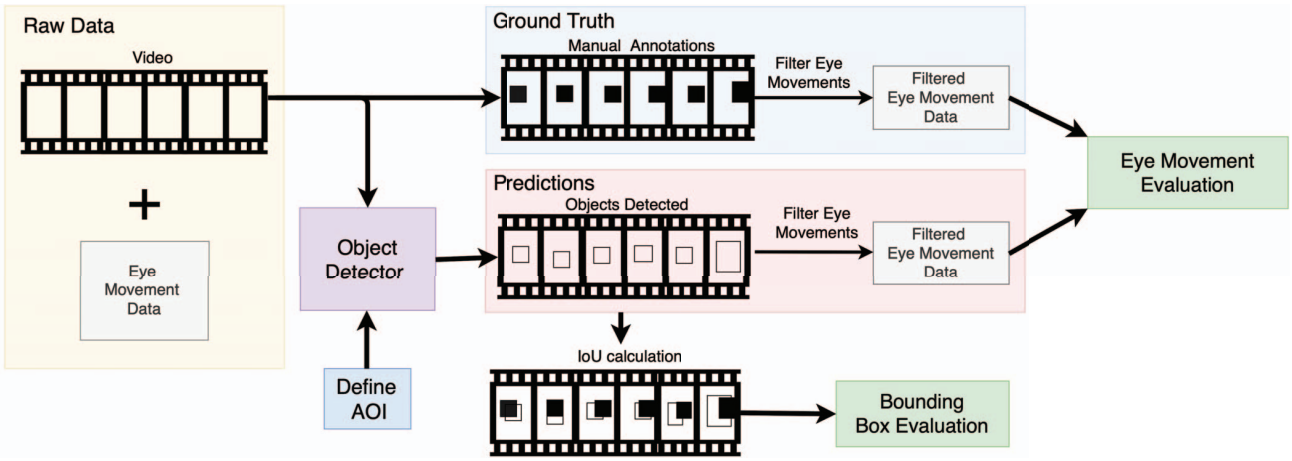


Figure 2. The Workflow of the RAEMAP which Processes Eye-tracking Data and Dynamic Eye-tracking Stimuli to Detect Dynamic AOIs. Raw video sequence is given to the object detector. Upon defining object(s) of interest, object detector outputs bounding box coordinates of each object detected. These dynamic bounding boxes 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 [30] 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 IoU and mAP, and filtered eye-movements are evaluated using precision, recall, accuracy.

Next, we configure the RAEMAP 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 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, the RAEMAP first identifies the bounding box coordinates of AOIs detected in each frame and writes them into a file. Next, the 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 manual 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.

IV. Evaluation

We evaluated our method using a publicly available eye-tracking dataset [38] that provided data collected from 15 participants while watching 12 video sequences. Participants (2 F and 13 M) were aged between 18 and 30 [38], and had normal or corrected-to-normal vision. During the study, participants have worn a Locarna "Pt-Mini" head-mounted eye tracker which had two 30 fps cameras, (1) the eye camera, and (2) the scene camera. Though the eye tracker had allowed participants to move their head naturally, all participants have been seated in front of a 19" Samsung monitor at a distance of 31.5 inches throughout the eye-tracking experiment. Participants were presented with the twelve video segments sequentially.

The 12 video segments were provided in uncompressed YUV format, at (352×288) resolution and 30 fps. The gaze data from 15 participants were provided in 12 comma separated value (CSV) files (i.e., one CSV per video). Each row in the CSV files correspond 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 in the same format as the gaze data CSV, indicating whether the gaze locations in the CSV are correct (flag = 1) or not (flag = 0). 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. Here, we filter 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 could be done separately.

A. Dynamic AOI Detection

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

- *Foreman*: Person
- *Bus*: Bus
- *Mother and Daughter*: Two People
- *Hall Monitor*: Two People

For each video sequence, we apply an object detector using the RAEMAP and dynamically detect the AOI at each frame. Upon repeating this for each object detector, we obtain a prediction for the bounding box coordinates at each frame of the video sequences, from each object detector.

When evaluating the dynamic AOIs detected, we create a ground truth dataset by manually annotating each video sequence with the expected AOI in the form of bounding boxes. We used BeaverDam[30] video annotation tool to create training and evaluation data for dynamic AOI detection. 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 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 the gaze extractions made using them.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{1}$$

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 (1)), whereas mAP is a metric used to evaluate object detectors. We use IoU to calculate how much of the boundaries predicted using each object detector overlaps with the ground truth bounding boxes. We 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 is calculated based on the classification of the predicted bounding boxes. Finally, we calculate mAP in both COCO style and Pascal VOC2008 [39] 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.

B. Eye-Movement Extraction

After detecting dynamic AOIs on video sequences, we pass the raw eye-tracking data to the RAEMAP. The RAEMAP extracts eye gaze data that falls within the bounding boxes of dynamic AOIs as generated by each object detector. It extracts gaze data in the form (x_i, y_i, t_i) from the raw gaze data. In the extracted eye gaze data, (x_i, y_i) coordinates indicate the position of the gaze point, and t_i indicates the timestamp. Moreover, we configured the RAEMAP to 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). To evaluate the dynamic AOI-mapped eye movements, we classify the eye movements according to the confusion matrix shown in Table II. We use standard information retrieval domain evaluation metrics such as precision, recall, and accuracy for the evaluation of filtered eye movements.

TABLE II
CONFUSION MATRIX FOR EYE MOVEMENTS EVALUATION

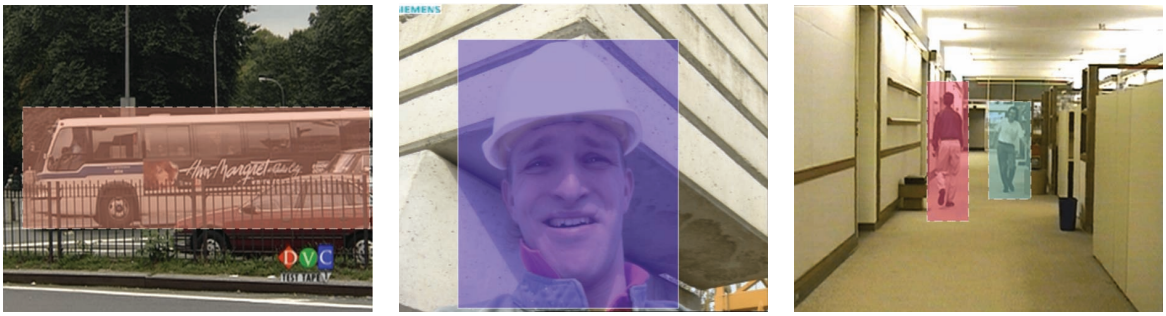
		Ground Truth AOI	
		Falls Within	**Falls Outside**
Predicted AOI	**Falls Within**	TP	FP
	Falls Outside	FN	TN

V. RESULTS

A. 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 the YOLOv3 object detector.

Table III shows the mAP values in both COCO style 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 III) with the $AP \geq 0.19$.



(a) Bus (b) Foreman (c) Hall Monitor

(d) Bus (e) Foreman (f) Hall Monitor

Figure 3. Frames with AOIs, captured from manually annotated AOIs using BeaverDam annotation tool ((a), (b), (c)) and dynamic AOIs detected from YOLOv3 object detector ((d),(e),(f)) in *Bus*, *Foreman*, and *Hall Monitor* video sequences.

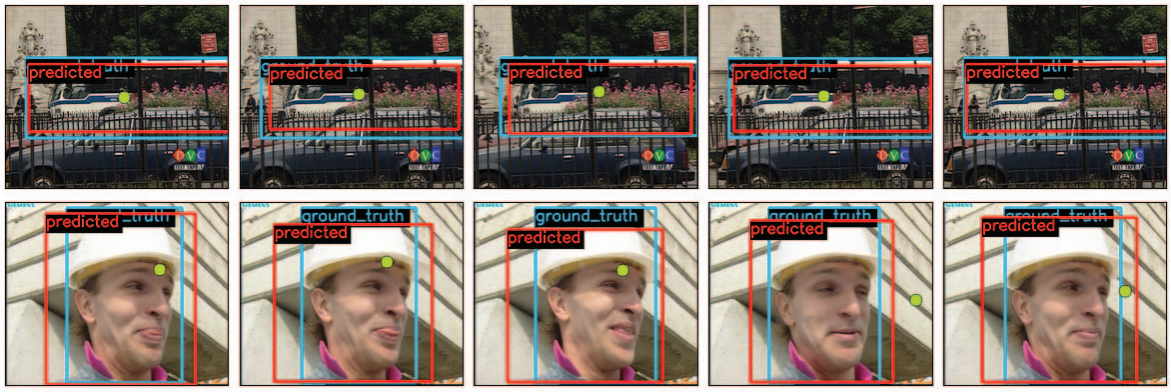


Figure 4. Visualization of manually annotated AOIs (in blue), AOIs detected by faster R-CNN object detector with ResNet-101-FPN backbone (in red), the gaze positions of a participant in five consecutive frames in *Bus* and *Foreman* video sequences.

TABLE III
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

B. Eye Movement Extraction

Figure 4 illustrates dynamic AOIs detected from 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

TABLE IV
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

boxes, whereas blue color bounding boxes indicate the ground truth. Green color circle indicates the gaze position of that frame.

Table IV 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 movements data with the highest accuracy of 64.5%.

VI. DISCUSSION

Our evaluation of dynamic AOIs generated by object detectors indicate that faster R-CNN object detector with ResNet-101-FPN backbone achieves the highest AP in both COCO style and PASCAL style (see Table III). Though faster R-CNN object detector with ResNet-101-FPN backbone achieves the highest AP rate, we observed it to be slower compared to one-stage detector YOLOv3, supporting the literature that two-stage object detectors are typically slower [40]. Two-stage object detectors are slow because they generate regions of interests 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 [40]. Our evaluation indicates that two-stage object detectors, despite being slow, performs the best in classifying objects and finding bounding-boxes as dynamic AOIs. However, this observation is highly speculative and replication of the current findings are required with a larger representation of one-stage object detectors since we only used a single one-stage object detector, YOLOv3 for the evaluation.

One limitation in our study is that we only used videos with dominant objects. We could overcome this limitation by using video sequences with dominant objects for dynamic AOIs detection. Since majority of the frames in the video sequences could be used to find the optimal object detector to optimize the performance of the object detector [40], we could further evaluate the object detectors with video sequences of varying levels of complexity.

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 observed 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 does 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 the Table IV, 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.

Interestingly, we observed in both evaluation criteria, the faster R-CNN object detector with ResNet-101-FPN backbone scored the highest. Based on the performance in both evaluation criteria, we choose faster R-CNN object detector with ResNet-101-FPN backbone as the object detector in the RAEMAP. Apart from the RAEMAP integration, 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.

VII. CONCLUSIONS

In this study, we incorporated computer vision methods for offline detection of dynamic AOIs in dynamic eye-tracking stimuli such as video streams. We presented our implementation and evaluation of object detectors to integrate in an RAEMAP to filter eye movement data within dynamic AOIs. Based on the performance evaluation, faster R-CNN with ResNet-101-FPN backbone object detector works best for the RAEMAP integration. In the future, we plan to evaluate our methodology using one-stage object detectors such as SSD [41], DSSD [42], and retinanet [37]. Moving forward in this line of inquiry, we plan to use segmentation instead of polygonal boundaries when defining dynamic AOIs for the extraction of eye movements.

REFERENCES

[1] J. S. McCarley and A. F. Kramer, "Eye movements as a window on perception and cognition," *Neuroergonomics*, vol. 3, p. 95, 2008.

[2] R. Radach, J. Hyona, and H. Deubel, *The mind's eye: Cognitive and applied aspects of eye movement research*. Amsterdam, Netherlands: Elsevier, 2003.

[3] S. Van der Stigchel, N. Rommelse, J. Deijen, C. Geldof, J. Witlox, J. Oosterlaan, J. Sergeant, and J. Theeuwes, "Oculomotor capture in adhd," *Cognitive Neuropsychology*, vol. 24, no. 5, pp. 535–549, 2007.

[4] M. A. Just and P. A. Carpenter, "A theory of reading: From eye fixations to comprehension." *Psychological review*, vol. 87, no. 4, p. 329, 1980.

[5] O. Komogortsev, C. Holland, S. Jayarathna, and A. Karpov, "2d linear oculomotor plant mathematical model: Verification and biometric applications," *ACM Transactions on Applied Perception (TAP)*, vol. 10, no. 4, p. 27, 2013.

[6] K. Krejtz, A. T. Duchowski, A. Niedzielska, C. Biele, and I. Krejtz, "Eye tracking cognitive load using pupil diameter and microsaccades with fixed gaze," *PloS one*, vol. 13, no. 9, p. e0203629, 2018.

[7] T. Kosch, M. Hassib, D. Buschek, and A. Schmidt, "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*. New York, NY, USA: Association for Computing Machinery, 2018.

[8] N. A. Gehrer, M. Schönenberg, A. T. Duchowski, and K. Krejtz, "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*, ser. ETRA '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3204493.3204543

[9] G. Jayawardena, A. M. P. Michalek, Michael, A. T. Duchowski, and S. Jayarathna, "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*, ser. ETRA '20. New York, NY, USA: Association for Computing Machinery, 2020.

[10] R. S. Hessels, C. Kemner, C. van den Boomen, and I. T. Hooge, "The area-of-interest problem in eyetracking research: A noise-robust solution for face and sparse stimuli," *Behavior research methods*, vol. 48, no. 4, pp. 1694–1712, 2016.

[11] D. Noton and L. Stark, "Eye movements and visual perception," *Scientific American*, vol. 224, no. 6, pp. 34–43, 1971.

[12] C. M. Privitera and L. W. Stark, "Algorithms for defining visual regions-of-interest: Comparison with eye fixations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 9, pp. 970–982, 2000.

[13] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International journal of industrial ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[14] K. F. Van Orden, W. Limbert, S. Makeig, and T.-P. Jung, "Eye activity correlates of workload during a visuospatial memory task," *Human factors*, vol. 43, no. 1, pp. 111–121, 2001.

[15] A. Nguyen, V. Chandran, and S. Sridharan, "Visual attention based roi maps from gaze tracking data," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 5. IEEE, 2004, pp. 3495–3498.

[16] A. S. Aydin, S. Feiz, V. Ashok, and I. Ramakrishnan, "Towards making videos accessible for low vision screen magnifier users," in *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 2020, pp. 10–21.

[17] G. Jayawardena, "Raemap: Real-time advanced eye movements analysis pipeline," in *Symposium on Eye Tracking Research and Applications 2020*. Stuttgart, Germany: ACM, 2020.

[18] G.Jayawardena, A. Michalek, and S. Jayarathna, "Eye tracking area of interest in the context of working memory capacity tasks," in *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, 2019, pp. 208–215.

[19] P. Marchant, D. Raybould, T. Renshaw, and R. Stevens, "Are you seeing what i'm seeing? an eye-tracking evaluation of dynamic scenes," *Digital Creativity*, vol. 20, no. 3, pp. 153–163, 2009.

[20] R. B. Goldstein, R. L. Woods, and E. Peli, "Where people look when watching movies: Do all viewers look at the same place?" *Computers in biology and medicine*, vol. 37, no. 7, pp. 957–964, 2007.

[21] M. D. Crossland, G. S. Rubin *et al.*, "The use of an infrared eyetracker to measure fixation stability," *Optometry and vision science*, vol. 79, no. 11, pp. 735–739, 2002.

[22] G. T. Timberlake, M. K. Sharma, S. A. Grose, D. V. Gobert, J. M. Gauch, and J. H. Maino, "Retinal location of the preferred retinal locus relative to the fovea in scanning laser ophthalmoscope images," *Optometry and vision science*, vol. 82, no. 3, pp. E177–E187, 2005.

[23] K. Kurzhals and D. Weiskopf, "Space-time visual analytics of eye-tracking data for dynamic stimuli," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2129–2138, 2013.

[24] G. Tien, M. S. Atkins, and B. Zheng, "Measuring gaze overlap on videos between multiple observers," in *Proceedings of the symposium on eye tracking research and applications*, 2012, pp. 309–312.

[25] S. Lessing and L. Linge, "Iicap? a new environment for eye tracking data analysis," 2002.

[26] S. Stellmach, L. Nacke, and R. Dachselt, "Advanced gaze visualizations for three-dimensional virtual environments," in *Proceedings of the 2010 symposium on eye-tracking research & Applications*, 2010, pp. 109–112.

[27] N. Weibel, A. Fouse, C. Emmenegger, S. Kimmich, and E. Hutchins, "Let's look at the cockpit: exploring mobile eye-tracking for observational research on the flight deck," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2012, pp. 107–114.

[28] K. Krejtz, A. Duchowski, T. Szmidt, I. Krejtz, F. González Perilli, A. Pires, A. Vilaro, and N. Villalobos, "Gaze transition entropy," *ACM Transactions on Applied Perception (TAP)*, vol. 13, no. 1, p. 4, 2015.

[29] A. T. Duchowski, K. Krejtz, I. Krejtz, C. Biele, A. Niedzielska, P. Kiefer, M. Raubal, and I. Giannopoulos, "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*. Montréal, QC, Canada: ACM, 2018, p. 282.

[30] A. Shen, "Beaverdam: Video annotation tool for computer vision training labels," Master's thesis, EECS Department, University of California, Berkeley, Dec 2016. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-193.html

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[32] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[36] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[38] H. Hadizadeh, M. J. Enriquez, and I. V. Bajic, "Eye-tracking database for a set of standard video sequences," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 898–903, 2011.

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

[40] P. Soviany and R. T. Ionescu, "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, 2018, pp. 209–214.

[41] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[42] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.