

CLERA: A Unified Model for Joint Cognitive Load and Eye Region Analysis in the Wild

LI DING^{*†}, University of Massachusetts Amherst, USA

JACK TERWILLIGER^{*}, University of California San Diego, USA

AISHNI PARAB^{*}, University of California Los Angeles, USA

MENG WANG^{*}, University of Massachusetts Amherst, USA

LEX FRIDMAN, BRUCE MEHLER, and BRYAN REIMER, Massachusetts Institute of Technology, USA

Non-intrusive, real-time analysis of the dynamics of the eye region allows us to monitor humans' visual attention allocation and estimate their mental state during the performance of real-world tasks, which can potentially benefit a wide range of human-computer interaction (HCI) applications. While commercial eye-tracking devices have been frequently employed, the difficulty of customizing these devices places unnecessary constraints on the exploration of more efficient, end-to-end models of eye dynamics. In this work, we propose CLERA, a unified model for Cognitive Load and Eye Region Analysis, which achieves precise keypoint detection and spatiotemporal tracking in a joint-learning framework. Our method demonstrates significant efficiency and outperforms prior work on tasks including cognitive load estimation, eye landmark detection, and blink estimation. We also introduce a large-scale dataset of 30k human faces with joint pupil, eye-openness, and landmark annotation, which aims to support future HCI research on human factors and eye-related analysis.

CCS Concepts: • **Human-centered computing** → *HCI theory, concepts and models*; • **Computing methodologies** → *Interest point and salient region detections*.

Additional Key Words and Phrases: Human-centered computing, cognitive load estimation, pupil detection, driver monitoring systems, computer vision, machine learning

ACM Reference Format:

Li Ding, Jack Terwilliger, Aishni Parab, Meng Wang, Lex Fridman, Bruce Mehler, and Bryan Reimer. 2023. CLERA: A Unified Model for Joint Cognitive Load and Eye Region Analysis in the Wild. *ACM Trans. Comput.-Hum. Interact.*, (June 2023), 22 pages. <https://doi.org/10.1145/3603622>

1 INTRODUCTION

Understanding the appearance and dynamics of the human eye has proven to be an essential component of various human-centered research activities and applications, e.g., visual attention modeling [5, 80], gaze-based human-computer interaction [11, 19, 48], virtual reality [9, 55], physical and psychological health monitoring [31, 43, 51], usability evaluation [29, 38], and emotion recognition [4, 45]. However, in order to assess human cognitive load or perform other visual attention modeling tasks in real-world situations, it is often required that the evaluation approach should not interfere with the natural behavior of interest such that the mental state of the individual

^{*}Work performed when the authors were at Massachusetts Institute of Technology.

[†]Corresponding author.

Authors' addresses: Li Ding, liding@umass.edu, University of Massachusetts Amherst, Amherst, MA, USA, 01002; Jack Terwilliger, jterwilliger@ucsd.edu, University of California San Diego, La Jolla, CA, USA, 92093; Aishni Parab, aishni@g.ucla.edu, University of California Los Angeles, Los Angeles, CA, USA, 90095; Meng Wang, mwang0@umass.edu, University of Massachusetts Amherst, Amherst, MA, USA, 01002; Lex Fridman, fridman@mit.edu; Bruce Mehler, bmehler@mit.edu; Bryan Reimer, reimer@mit.edu, Massachusetts Institute of Technology, Cambridge, MA, USA, 02142.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Computer-Human Interaction*, <https://doi.org/10.1145/3603622>.

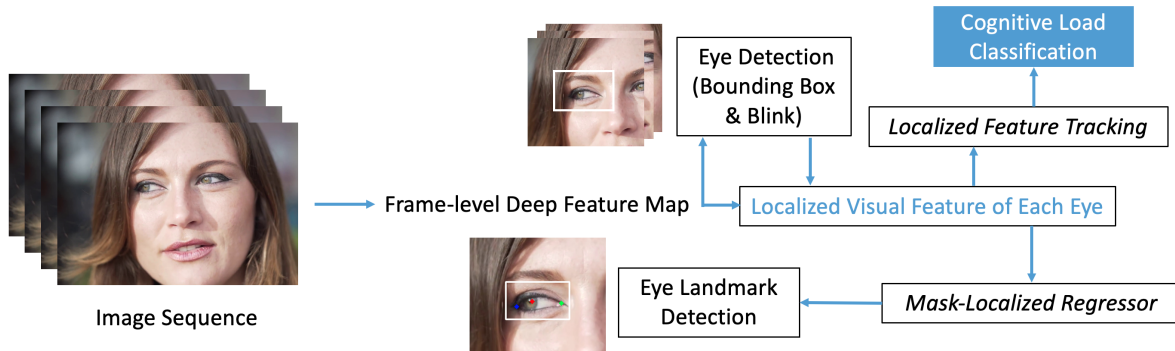


Fig. 1. Overview of the proposed model CLERA, for joint cognitive load and eye region analysis. We first perform detection of the eye on the frame-level deep feature map. Next, the architecture extends two heads: *Localized Feature Tracking* for cognitive load estimation over time and *Mask-Localized Regressor* for eye landmark detection.

being measured is not influenced by the measurement approach itself. Moreover, these assessments should also generalize to different environmental and individual-specific characteristics such as visual appearance, movement, pose, scale, perspective, time of the day, etc. Thus, developing practical, non-contact approaches that are not hindered by environmental and experimental constraints remains a challenging problem in HCI research.

Some advanced approaches [24, 25] have been proposed to take advantage of modern computer vision and deep learning technologies to estimate human cognitive load from an “in the wild” perspective through modeling pupil dynamics, which yields the potential of real-time applications such as driver attention monitoring. These approaches often require efficient and precise detection of the eye region and its landmarks, which is often achieved using out-of-the-box eye-tracking devices. However, these devices can be hard to customize and interact with, putting unnecessary constraints on exploring more efficient, end-to-end modeling of eye dynamics. For example, [24] needs either the pupil and eye landmark positions or the tracked eye-region image as the input to the cognitive load estimation model. As a result, the current methods are still incapable of making online predictions due to the latency introduced by the prerequisite of adding an eye tracker.

In this work, we focus on exploring computer-vision-based joint-learning frameworks for eye-region analysis and downstream eye-dynamics modeling tasks. Our intuition is that since both eye tracking and eye-dynamics modeling tasks can be viewed as learning tasks that take the camera image as input, they could likely share some part of the modeling and be integrated into a joint-learning framework, potentially saving considerable computation for efficiency. To validate this idea, we propose a unified deep learning model, termed *CLERA*, for joint Cognitive Load and Eye Region Analysis, as shown in Fig. 1.

The proposed model aims to make improvements on prior research from two perspectives. First, we focus on the architectural design of joint-learning deep neural networks. There is very likely to exist a large amount of computation redundancy when the cognitive load estimation model takes image sequences of the eye as input [24], because the detection of the eye region needs to be predicted by a separate system. As the tasks of eye detection, pupil localization, and cognitive load estimation all rely on extracting visual representations of the human eye region, well-learned representations could potentially be shared across these tasks. Moreover, end-to-end learning of deep neural networks usually requires large-scale data for training. However, obtaining ground truth cognitive load data is usually difficult and costly as it requires specific experimental setups, e.g., n-back tasks [62]. On the other hand, traditional computer vision tasks of eye region analysis, such as eye and

landmark detection, are well constructed and easy to annotate at scale. Since cognitive load can be estimated by modeling the physiological reactivity of the eye, successful estimation will need good visual representations of the eye and its components. This fact indicates that eye/landmark detection may serve as side supervision for the cognitive load task to improve the quality of learned representation. Based on these considerations, we propose the *Localized Feature Tracking* technique, which utilizes shared visual features for high-level tasks in the temporal domain, such as cognitive load estimation, within a joint-learning framework. With the detection tasks performed on each frame, we use a temporal tracking algorithm to track each detected eye. For each successfully tracked eye, we perform temporal modeling on the top of localized deep feature maps instead of the raw image. As a result, the whole framework is able to learn general and robust feature representations for precise eye landmarks, and blink detection, and use the same representations for cognitive load estimation, which outperforms existing methods and can efficiently run in real-time to be useful for many real-world applications.

Second, we focus on adapting modern computer vision models to better facilitate real-world applications of eye-region analysis. Existing methods for pupil and blink detection [7, 41, 47, 64] heavily rely on the assumption of environmental conditions of the training data, and usually need to work under similar controlled environments. For example, [7] requires an eye camera to be mounted on eyeglasses, which is not suitable for real-world situations. Recent advancements in common object detection and human pose prediction show exciting performance on large-scale datasets [46]. We leverage this success and frame the pupil and blink detection task as a joint instance and keypoint detection problem. State-of-the-art methods [8, 32, 54] tend to use mask-based methods, where the keypoints are predicted using heatmaps on either the whole image or particular regions of interest. The precision of keypoint outputs is thus limited by the resolution of the heatmap. Such approaches are suitable for tasks where the precision of keypoints is not highly demanding as the heatmap resolutions usually suffice, e.g., human pose estimation. However, when it comes to eye landmarks, mask-based approaches lack the required precision for eye-related tasks, such as capturing the micromovements of the pupil within the eye region. To handle such problems, we propose a method, termed *Mask-Localized Regressor*, that extends mask-based approaches to handle precise eye landmark detection that can provide sub-pixel predictions of coordinates.

In addition, we recognize the need for large-scale datasets to facilitate human factors research using modern data-driven approaches. These approaches often require diverse datasets to capture a variety of natural environments for the task. To meet this need, we propose MIT Pupil Dataset, a large-scale dataset of 30k crowd-sourced web images of human faces. The dataset includes joint annotations for pupil, eye-openness, and landmarks, and has an even distribution of images of closed and open eyes. This dataset aims to serve as an open-source benchmark and to help with the development of modern learning-based algorithms for understanding human eyes in real-world applications. Both the dataset and the algorithm proposed in this work will be released open source to contribute to the community for further research on this topic.

To summarize, the main contributions of this work are:

- (1) **CLERA**: a unified joint-learning framework for cognitive load and eye region analysis, which consists of two novel techniques:
 - (a) *Localized Feature Tracking* for using shared image features for cognitive load modeling
 - (b) *Mask-Localized Regressor* for precise eye landmark detection
- (2) **MIT Pupil Dataset**: a large-scale, open-source¹ dataset of around 30k images of human faces with joint pupil, eye-openness, and landmark annotation.

¹The dataset is not directly published online due to privacy and sensitivity concerns. For inquiries about using the dataset for research purposes, please contact the corresponding author.

2 RELATED WORK

2.1 Cognitive Load Estimation

The concept of cognitive load [52] is often used to refer to the amount of human working memory in use, and has been shown to be an important variable impacting human performance on a variety of tasks, such as machine operations, education, and driving, for which human operators are responsible for the major decision-making and action execution. Early research [30, 50, 52, 68] proposed various physiological measures that are sensitive to changes in cognitive load levels that can be characterized under controlled experimental conditions. There have been numerous studies linking eye movements to variations in cognitive load levels, especially in the driving area. Most studies use different approaches and methods on different datasets. As such, it has been hard to directly compare results. A general finding supported by previous studies is that increased levels of cognitive load often result in a narrowing of visual search space during driving, *i.e.*, gaze concentration. [59] studies the effects of mental workload on visual search and decision-making. [62] explores the impact of variations in short-term memory demands on drivers' visual attention and performance. [71] compares several methodologies for computing changes in gaze dispersion, showing that horizontal eye movements show the greatest sensitivity to variations in cognitive demand. [79] explores using machine learning methods in driver workload estimation. [24] proposes two novel vision-based methods for cognitive load estimation, and evaluates them on a large-scale dataset collected under real-world driving conditions [22]. Our work follows the direction of [24] and proposes a more integrated deep learning framework for better efficiency and generalization. Based on these successful attempts, our work steps further in this direction and integrates the task of cognitive load estimation into the general computer vision task of object detection and video understanding. Such integration allows us to design better deep learning frameworks that manage to improve both computational efficiency and generalization to "wilder" real-world circumstances.

2.2 Pupil Detection and Blink Estimation

There have been various preliminary works on using image-based computer vision methods to enhance the safety and experience of driving [12, 14, 17, 23], especially on the driver's facial analysis including detection of human eyes, eye landmarks, gaze, blink, or jointly detect some combination thereof. Traditional methods [7, 41, 47, 64] either utilize hand-crafted visual features extracted by descriptors such as SIFT and HOG, or employ image/color models based on the appearance of the human eye. These methods usually require controlled environments to work, and thus cannot handle cases in real-world uncontrolled conditions [13, 15, 16, 22], including arbitrary viewpoints, varying face appearances, and illuminance changes. In recent years, deep learning methods have been used to form better representations in order to improve the accuracy and robustness of general object detection [32, 60, 63] and keypoint detection [6, 8, 32, 54]. Some relevant papers [2, 10, 40, 65] explore using existing deep learning architectures on the tasks of blink or gaze estimation. Our work takes a further step to propose a method for precise keypoint detection and a unified framework designed specifically for joint eye, pupil, and blink detection. The whole model is optimized for multiple aspects of this task and enables real-time detection under real-world conditions.

2.3 Pupil and Blink Datasets

Table 1 shows an overview of open-source datasets that have pupil position and/or eye-openness annotated for real-world images. In general, many existing datasets [26, 28, 39, 69, 70] for pupil detection and eye tracking involve using head-mounted cameras or eye-tracking glasses, which are not applicable to many real-world applications that require practical, non-contact approaches for pupil and blink detection, and which may also involve localizing eyes of interest in the first place. It is worth noting [78] which proposes a larger-scale dataset captured with laptop webcams for gaze estimation, which also provides pupil and landmark annotations on a

subset. Although this subset has some variability in appearance and illumination, it still has many constraints such as a limited number of subjects and camera perspectives. For annotated eye blinks or closed eyes, some existing datasets [18, 21, 53, 67] are small in scale in terms of deep network training. [40] offers a larger dataset of 5k samples of closed-eye images, but only with frontal faces. A more recent work [10] provides annotations for a small subset of 480 images with semantic labels of pupil area, and 10k images of closed eyes. Our dataset aims to address the shortcomings of using restricted devices, a limited number of subjects, etc., and provides a large-scale, in-the-wild dataset of around 30k images with joint pupil and eye landmarks evenly distributed across open and closed eyes.

3 METHODS

As shown in Fig. 1, we propose a unified architecture for eye bounding boxes, eye landmarks, blink, and cognitive load estimation with sequential image input. We first extract a frame-level deep feature map using deep convolutional neural networks, and perform bounding box detection and blink estimation (binary classification of open/closed eye). We then locate the positive detection of eyes back onto the feature map and get the localized visual feature representations to perform eye landmark detection. Finally, we track the localized feature through time and use temporal modeling to perform cognitive load estimation.

3.1 Image Feature Extraction and Eye Detection

We first use a pre-trained deep convolutional network (ResNet-50 [33]) as the image-level feature extractor. ResNet-50 is a deep convolutional neural network architecture that consists of 50 layers, widely used for image recognition tasks. By pre-training on large-scale datasets, it can extract meaningful features from the image to be used for computer vision tasks. On the top of the feature map (down-scaled by 32 times from the original resolution of the image), we first perform bounding box detection, which is to predict the bounding box location (coordinates b_x, b_y) and size (width b_w , height b_h). Instead of following popular methods [61, 63] that utilize multi-scale region proposals, we (similar to [60]) predict the bounding box confidence and its parameters using an extra convolutional layer for better computation efficiency.

On the feature map, the convolutional layer predicts 5 variables on each cell: t_p, t_x, t_y, t_w, t_h . The offset of the cell from the top left corner of the image is denoted by (c_x, c_y) , and the bounding box prior has width and height of p_w and p_h . The positive predictions (cells that $t_p > 0$) finally correspond to:

$$b_x = \tanh(t_x) + 0.5 + c_x \quad (1)$$

$$b_y = \tanh(t_y) + 0.5 + c_y \quad (2)$$

$$b_w = p_w \cdot e^{t_w} \quad (3)$$

$$b_h = p_h \cdot e^{t_h} \quad (4)$$

Note that our parameterization is different from [61] because in practice, we find the sigmoid used in [61] leads to slower convergence and larger variation in prediction. It also does not allow predictions to be slightly out of the corresponding cell, which causes accuracy to decrease when the bounding box happens to locate in the middle of two cells. When a side cell has higher confidence than the center cell, the non-max suppression will select the prediction of side cell but it can never predict the accurate location of the box. Our equation solves this problem by letting each cell to predict the center of the box at most to the center of the neighboring cells (the term $(\tanh(x) + 0.5)$ ranges from -0.5 to 1.5). We also perform a binary classification for each detected eye to obtain its state (open/closed) for blink detection.

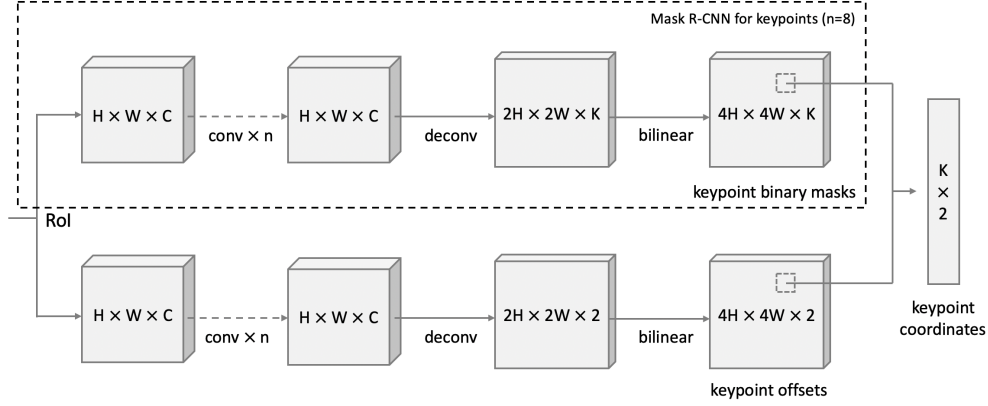


Fig. 2. Architecture of the Mask-Localized Regressor Head: We add another branch for keypoint offset regression following a similar design as the mask branch in [32]. This branch predicts keypoint-agnostic offsets, which are then added to mask-predicted indices to get precise keypoint coordinates.

3.2 Localized Feature Tracking for Cognitive Load Estimation

With the detection tasks performed on each frame, we use a temporal tracking method that sets a threshold θ for the temporal displacement of each of the detected eye, and track it through time. Namely, given that a detected eye $(b_{x_t}, b_{y_t}, b_{w_t}, b_{h_t})$ in frame t and another detected eye $(b_{x_{t+1}}, b_{y_{t+1}}, b_{w_{t+1}}, b_{h_{t+1}})$ in frame $t + 1$, we calculate the Intersection over Union (IoU) of them,

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{TP}{TP + FP + FN} \quad (5)$$

Where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. If IoU is greater than θ , they are treated as the same eye and tracking is thus established.

For successfully tracked eyes, we further extract the image feature representation from each frame, and perform temporal modeling on the top of them. We get the localized feature by locating the feature vector on the feature map by using the location of the eye. For example, for feature map l that is 16x downsampled from the original resolution, the feature vector of eye $(b_{x_t}, b_{y_t}, b_{w_t}, b_{h_t})$ will be $l[b_{x_t}/16, b_{y_t}/16]$. The localized feature vectors of the eye tracked through all the frames are then used as the input for temporal modeling, which outputs estimation of tracked properties such as cognitive load.

For temporal modeling, we use a VGG-like [66] architecture with blocks of 1D Convolution, BatchNorm [36], and ReLU. Each block consists of 3 Conv-BN-ReLU with the last one having a stride of 2 to perform down-sampling. The dimension of features in each block is set to [32,64,128,256], and finally we use global average pooling and a fully-connected layer for classification. One thing to notice is that the temporal modeling allows gradients to be back-propagated back to the feature extractor model, which can further fine-tune the whole model for better performance.

3.3 Mask-Localized Regressor for Precise Keypoint Detection

The *Mask-Localized Regressor* is designed to address the limitation of keypoint coordinate precision in mask-based keypoint detection approaches [32]. Given a heatmap f of mask coordinates of size $h' \times w'$ corresponding to the original image or image crop of size $h \times w$, $f(p_i) = 1$ if a keypoint is located at position p_i where $i \in \{1, \dots, N\}$,

$N = h' \cdot w'$. p_i is usually given as a tuple of (c'_x, c'_y) where c'_x, c'_y are rounded integer coordinates calculated as

$$(c'_x, c'_y)^T = \text{round}\left((c_x, c_y)^T \cdot \frac{(w', h')^T}{(w, h)^T}\right), \quad (6)$$

where the real coordinates are given as c_x, c_y . We then calculate the coordinate offsets t_x, t_y as

$$(t_x, t_y)^T = \frac{(c_x, c_y)^T}{(w, h)^T} - \frac{(c'_x + \alpha, c'_y + \alpha)^T}{(w', h')^T}, \quad (7)$$

where α is a fixed offset to adjust index rounding to the grid center, e.g., $\alpha = 0.5$ for zero-based indexing and $\alpha = -0.5$ for one-based.

The Mask-Localized Regressor g models the coordinate offsets t_x, t_y as the lost information during the rounding process, such that $g(p_i) = (t_x, t_y)^T$ if $f(p_i) = 1$. The model predicts on every coordinate $i \in \{1, \dots, N\}$, but only calculates loss if $f(p_i) = 1$. So given a loss function $C(\text{target}, \text{prediction})$, the loss is calculated as

$$\text{loss} = \sum_{i=1}^N C(g(p_i), (t_x, t_y)^T) \cdot f(p_i). \quad (8)$$

This method can be integrated into existing frameworks for keypoint detection. We design an architecture that extends the Mask R-CNN keypoint head to a Mask-Localized Regressor head, as shown in Fig. 2. We keep the mask branch as-is, and add a regressor branch with similar architecture for keypoint offsets. Finally, the two branches are joined together to get precise keypoint predictions.

Specifically, we perform RoI-Align using the predicted bounding box (ground truth bounding box during training) on the $8\times$ scale. Both the mask branch and offset regressor branch have $n = 4$, $H = 8$, $W = 16$, and $C = 256$.

4 MIT PUPIL DATASET

Our goal is to create a dataset suitable to train and evaluate a general-purpose eye detector with the capability to also predict the corresponding attributes of the eye, including pupil, landmark position, and openness. Such a dataset is subject to a few design decisions in order to ensure it covers a sufficiently general domain of scenarios and environmental variations, and can be efficiently and accurately annotated at large-scale. We design a pipeline where we first obtain large-scale images of human faces from different sources to ensure its variability. Then we efficiently annotate the eye and landmarks by splitting the whole annotation process into subtasks: (1) determine if there is a visible right eye present, (2) determine whether the right eye is open or closed, (3) draw a bounding box around a person's right eye, and (4) annotate the keypoints for the right eye. Examples of the dataset are visualized in Figure 4.

4.1 Dataset Structure

Our dataset is comprised of 28,039 images, each with the following attributes:

- (1) **state**: a binary variable, $state \in \{\text{open}, \text{closed}\}$, which marks an eye as either open or closed.
- (2) **bounding_box**: a quadruple $(x_{bbox1}, y_{bbox1}, x_{bbox2}, y_{bbox2})$ denoting the upper left and lower right corners of a bounding box encompassing an eye.
- (3) **lateral_canthus**: a tuple $(x_{lateral}, y_{lateral})$ denoting the location of the lateral canthus (outside corner) of the eye.
- (4) **medial_canthus**: a tuple (x_{medial}, y_{medial}) denoting the location of the medial canthus (inner corner) of the eye.
- (5) **pupil**: a tuple (x_{pupil}, y_{pupil}) denoting the location of the center of the pupil.

Table 1. Overview of open source datasets for annotated pupil and eye landmarks position as well as blink/eye-openness. (*: the actual number released for open-source)

	subjects	camera view	# of images annotated	
			pupil & landmarks	blink / closed eye
Swirski <i>et al.</i> [69]	2	head-mounted	600	-
Fuhl <i>et al.</i> [26]	17	head-mounted	38,401	-
LPW [70]	22	head-mounted	130,856	-
MPIIGaze [77, 78]	15	frontal (laptop)	37,667 (10,848*)	-
NVGaze [39]	3	head-mounted	7,128	-
OpenEDS [28]	152	head-mounted	12,759	-
ZJU [53]	20	frontal & upward	-	255 / 1,016
Kim <i>et al.</i> [40]	-	frontal	-	- / 4,891
CEW [67]	2,423	wild (internet)	-	- / 1,192
Eyeblink8 [18]	4	frontal	-	353 / -
Res. Night [21]	107	frontal (screen)	-	1,849 / -
RT-BENE [10, 20]	15	free-viewing	480	- / 10,444
Ours	>10,000	wild (internet)	24,391	- / 13,764

This dataset only includes annotations for the right eyes. By assuming that differences between the features of the left and right sides of a face vanish at the population level, the horizontally flipped image of the dataset includes only left eye annotations. This assumption allowed us to halve the effort needed to generate the dataset. We describe how we design suitable mechanisms for working with this dataset for both training and evaluation in Sec. 5.2.3.

4.2 Image Collection

We assemble a collection of approximately 30k images that consists of a wide variety of faces with equal instances of closed and open eyes. We used several existing datasets to compile a preliminary set of around 7k images, including Labeled Faces in the Wild [35], CAS-PEAL [27], Caltech Faces 1999 [72] and Closed Eyes in the Wild [67]. To collect more closed-eye images, we used search engines to locate open-source licensed real-world faces varying in head pose, gaze, race, gender and lighting conditions. These provided 17k images. In order to capture the intrinsic pupil movement of the human eye, we also gathered a set of high-resolution YouTube videos, and captured 6k images from those videos.

4.3 Annotation Process

For all annotation tasks, we employed professional in-house annotators and developed web-based custom annotation tools in order to produce high-quality and efficient annotations. The annotators are trained with a warm-up task of 100 or more images and required to pass the manual check by researchers to start annotation.

4.3.1 Bounding Box Annotator. The bounding box interface, shown in Fig. 3 on the left, shows the user one candidate image at a time. First, the annotators are asked to tell how many pairs of eyes exist in the image. If there is exactly one pair of eyes in the image (including occluded), the annotators go on to draw a bounding box around the right eye and select whether the eye is open or closed. The interface allows annotators to zoom

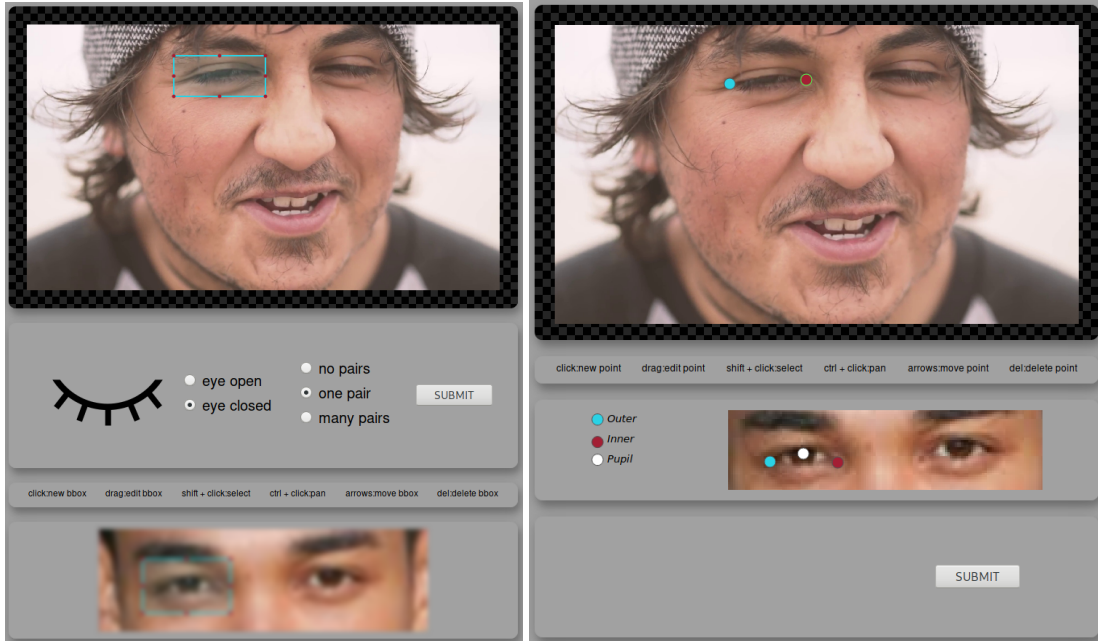


Fig. 3. The Bounding Box and Keypoint Annotation Interfaces

in, drag the box, or move it by small steps using the arrow keys in order to make annotation easier and more accurate.

4.3.2 Keypoint Annotator. As shown in Fig. 3 on the right, the keypoint interface shows the annotator an image that has been annotated with a bounding box of the right eye, and asks workers to click on the locations of eye landmarks of interest. Annotators first annotate the lateral canthus, then the medial canthus, and lastly the pupil. The annotated locations of these landmarks are shown with corresponding colors indicated in Fig. 3.

We found that by enforcing the order of landmarks to be annotated as prescribed by the interface helps to improve efficiency and eliminates errors associated with missing the order of landmarks: an annotator merely has to click their mouse three times to annotate an image as opposed to manually selecting a landmark type before each click. Annotators are also able to adjust the location of annotated landmarks by either dragging or using the arrow keys.

We also have annotators provide a redundant label for each eye state, which helps us verify the quality of annotation. This is done by asking annotators to annotate the pupil only if they (1) believe the eye is open and (2) the center of the pupil is visible or inferrable.

4.3.3 Two-pass Annotation. The annotation occurred in two passes, in which each image was annotated by two different annotators. This allowed us to measure annotation integrity and accuracy by comparing the agreement and disagreement between the two annotations. We then perform multiple experiments to determine suitable filters for each of the tasks to programmatically remove either false or ambiguous annotations from the dataset to improve the overall data quality, as described in the next section.

Table 2. Two-pass keypoint annotation statistics.

Keypoint	Euclidean distance (normalized by box width)			
	< 0.05	[0.05, 0.1)	[0.1, 0.2)	≥ 0.2
Pupil	82.8%	9.1%	1.0%	7.1%
Lateral Canthus	46.7%	30.2%	16.9%	6.2%
Medial Canthus	52.5%	23.2%	10.0%	5.3%

4.4 Data Validation

With the two-pass annotation, we measure the consistency in the annotations, *i.e.*, the degree to which annotations performed on the same image by two different workers agreed for bounding box, eye state, and keypoints.

4.4.1 Bounding Box. We first exclude all the images with less than two bounding box annotations. Then we use a common similarity metric, Intersection over Union (IoU), to compare bounding box annotations from both passes. The average IoU is 88.0%. We also observed that the bounding boxes have larger variations horizontally than vertically. The ambiguity in height of the eye caused the majority of inconsistencies in annotations.

4.4.2 Eye State. After bounding box filtering, we compare the eye state labels from two workers. 93.5% of the annotations agree with each other, and, for the rest, we labeled those as ambiguous cases and excluded them from the current dataset.

4.4.3 Keypoints. We compare keypoint annotations by measuring the Euclidean distance between the coordinates from two-pass annotations for each of the three keypoints: lateral canthus, medial canthus, and pupil. The keypoint distance is measured in pixels and then scaled to the corresponding bounding box width, which is more consistent than box height or area. The statistics are shown in Table 2. We observe that pupil annotations are more consistent than those for lateral and medial canthus. Note that for distance ≥ 0.2 , it also includes situations when one of the workers did not annotate the keypoint, which happens more often in pupil annotation cases where one of the workers rates the pupil as not visible.

4.4.4 Final Dataset. Finally, we also add filters to remove images that are out of the area of interest in this work. For the final dataset, we applied the following filters to automatically clean up the dataset:

- (1) **bounding box:** Removed all images where bounding box IoU < 0.3 , which we consider as cases where two annotators disagree with each other, *e.g.*, they annotate different eyes in the image. We then take mean box coordinate values.
- (2) **state:** removed all images where eye state annotations disagree.
- (3) **keypoints:** Removed all keypoint annotations where any of the three normalized keypoint distances ≥ 0.2 , which we consider as two annotators disagree. The box and state annotations are still kept if keypoint annotation is removed. We take mean keypoint coordinate values for the rest.
- (4) **out-of-interest:** Removed all cases where either the images have low resolution (width of the eye bounding box less than 30 pixels) or rotated over 45 degrees (inferred from keypoint positions).

5 EXPERIMENTS

In this section, we describe the empirical studies that consist of multiple tasks including cognitive load estimation, eye landmark detection, and blink detection. Since no prior work is evaluated on all these tasks, we evaluate our model separately on each task for better comparison.

Table 3. Results for the cognitive load classification task. Our proposed method outperforms previous work and shows the significance of using the localized feature tracking.

Method	Classification Accuracy
Eye Feature + SVM [44]	59.43%
Horizontal Pupil Position + HMM [24]	61.97%
CLERA (w/ Horizontal Pupil Position)	63.43%
CLERA (no fine-tune)	64.81%
CLERA	66.58%

5.1 Cognitive Load Estimation

Varied findings have been reported in the literature regarding the responsiveness of gaze concentration measures to changes in cognitive demand. We hereby describe the experiments and results regarding the validation of the proposed method on the cognitive load estimation task.

5.1.1 Dataset. In this experiment, we use an unpublished dataset (obtained and extended from [62]) of 212 30-second video clips of driver faces, each under one of two different cognitive load levels (104 low and 108 high), across 81 different subjects. The subjects needed to meet the criteria of being proficient and regular drivers, which was defined as having a valid driver’s license for at least three years and driving a minimum of three times per week. The data were collected with a Volvo XC90 vehicle, which was equipped with synchronized data collection capabilities from a range of built-in sensors including vehicle’s controller area network (CAN), cameras for recording driver behavior and the surrounding environment, and audio captured from within the vehicle cabin. The study utilized a delayed digit-recall task, known as the n-back task, with three distinct levels of difficulty to impose varying degrees of secondary cognitive workload on the drivers. In this work, we use the data from low and high levels to form a binary classification task.

Previous research on cognitive load estimation has primarily utilized synthetic or controlled environments, such as driving simulator [44, 56], tele-surgical robotic simulation [73], and simulation games [3], limiting their applicability to real-world situations. However, our work is focused on addressing this limitation by investigating the estimation of cognitive load in real-world settings. While recent work [24] has explored using real-world testing cases, we take one step further and use a significantly more challenging dataset with varying lighting conditions and camera placements. By conducting our research in naturalistic environments, we aim to capture the complexity and variability of real-world cognitive load scenarios, which could provide valuable insights for enhancing the development of cognitive load estimation models that can be applied in practical settings.

5.1.2 Comparison Methods. We use the same experimental settings as in Fridman et al. [24] that average the results over 10 random training/testing splits (80% for training and 20% for testing) across subjects. For comparisons, we first implement the SVM approach in Liang et al. [44] to serve as the baseline method. We also implement the HMM approach in [24], which is one of the state-of-the-art approach using pupil position to estimate driver’s cognitive load.

5.1.3 Results. The results are shown in Tab. 3. First, we implement the HMM model with horizontal pupil position from [24] as a baseline. To validate the effectiveness of each component in CLERA, we implement two variants of CLERA: CLERA (w/ Horizontal Pupil Position) is using horizontal pupil position as input to the temporal modeling, and CLERA (no fine-tuning) is using the proposed localized feature tracking without fine-tuning the feature extractor for temporal modeling.

We can first observe that CLERA (w/ Horizontal Pupil Position) outperforms the HMM when using the same horizontal pupil position as the input. This result aligns well with the observation in prior work [24] where the 3D-CNN model outperforms the HMM. Secondly, CLERA using the proposed localized feature tracking outperforms the one using horizontal pupil position. This indicates that there is information loss when abstracting eye movement to the change in normalized pupil position, and the localized feature can be used as a better feature with minimal extra computation cost. Thirdly, since CLERA allows end-to-end gradient learning, the full CLERA model gets a large performance gain, which suggests that the cognitive load task needs some specific visual representation that can not be learned from other vision tasks such as eye landmark detection.

In general, all the CLERA variants are able to outperform prior work, and the full model has a significant improvement. It is worth noting that while the absolute accuracy values obtained for differentiating the two cognitive load states was moderate, the test dataset consisted of data collected in a moving environment (a vehicle), with individuals having variable positioning relative to the camera and under variable lighting conditions - a very challenging real-world dataset as opposed to data collected under controlled laboratory conditions [3, 44, 56, 73]. For example, [44] claims to have over 80% accuracy in detecting driver cognitive distraction, but only have below 60% accuracy in our evaluation, which indicates that there exists a considerable difference between simulated and real-world environments, and more future efforts are required to address this issue.

Nevertheless, the primary interest for this work is to show the increase in performance across the proposed methods. We evaluate some of the broader capabilities of our methods in the next sub-section.

5.2 Eye State and Landmark Detection

5.2.1 Metrics. We adopt the Average Precision (AP) metrics for the eye localization task, which is an evaluation metric commonly used in machine learning to measure the accuracy of object detection or segmentation models. It is calculated by computing the area under the Precision-Recall curve (AP-PR) for a given set of predictions and ground truth labels. The formula for calculating AP is as follows:

$$AP = \frac{\sum_n (R_n - R_{n-1}) \cdot P_n}{R_{tot}} \quad (9)$$

where P_n and R_n are the precision and recall at the n th threshold, R_{tot} is the total number of positive examples, and R_{n-1} is the recall at the previous threshold.

As we observe evident variation in bounding box annotations, we mainly focus on keypoint metrics for evaluation, but use box metrics for ablation experiments on the detection backbone. For blink/eye-openness prediction, we simply calculate the accuracy, since the MIT Pupil Dataset is well-balanced for both cases. In terms of keypoint detection, we follow a similar principle as the existing OKS metric [46], and propose a specific metric for eye landmarks. We first measure the Euclidean distance between ground truth and predicted eye landmarks, normalized by the width of corresponding bounding boxes, which is the same metric as we used in Table 2. To calculate the AP on this distance for a range of levels, we choose to use more standard and interpretable level definitions as simply .01 : .01 : .1. To address the problem that lateral and medial canthus have a larger variance than the pupil (approximately 2×), we add a factor of 0.5 to the two and calculate the weighted mean for all three landmarks (two if closed eye) for AP calculation, meaning all the keypoints are jointly evaluated together for each detection.

In order to work with single-eye annotation, we add the following rule to the AP calculation process: ignore the first detection if it has no overlap with ground truth (for bounding box), or the weighted distance is greater than 0.5 (for keypoints). This is because we do not want to count for potentially correct detection for the other eye. For the training and testing split, we perform a random 8:2 split and use the same dataset split for all the experiments.

Table 4. Eye state and landmark detection results on MIT Pupil Dataset.

Methods	mAP	AP.1	AP.04	AP.02	State Acc.	FPS
Baseline-Regressor	53.7	76.2	55.7	7.5	98.6	38.7
Baseline-Mask	69.1	87.5	77.4	28.5	98.7	38.5
CLERA (mask-only)	70.8	90.7	79.5	28.9	98.7	38.5
CLERA	71.1	90.7	79.8	30.4	98.7	38.3

5.2.2 Baseline Methods. Since no prior work has been done on the proposed dataset, we also propose three methods for the benchmark and ablation study. In order to make fair comparisons, we use the same detection head as in Sec. 3.3 and focus on validating keypoint detection performance. The first method, called *Baseline-Regressor*, adds another regressor subnet along with two subnets in the RetinaNet detection head, which has the same subnet architecture and directly predicts the offsets of each keypoint for each anchor box. The second method, called *Baseline-Mask*, simply uses the mask branch in the mask-localized regressor head, which can be viewed as an implementation of Mask R-CNN [32] with our specific backbone. This method is only optimized on mask loss. The third method for comparison, CLERA (Mask-only), uses the proposed Mask-Localized Regressor jointly trained on mask loss and offset regression loss, but only uses the mask predictions without adding offset predictions. This is intended to show the direct performance of offset prediction. With the above three methods, we can more clearly separate and show the improvement gained from using the proposed Mask-Localized Regressor.

5.2.3 Working with Single-Eye Annotation. Since the proposed MIT Pupil Dataset only contains annotation of the right eye, in order to make the detector also capable of detecting the left eye, we use a training strategy with horizontal flipping and inferred gradient masking. During the training process, we first infer the region where potentially the other eye exists by using the position of the known eye, and generating a mask for that region. During training, that region is ignored for loss calculation in eye localization, as we do not have enough information to evaluate or penalize the detections in that region. As we add the horizontally-flipped version of the input image and corresponding annotations to generate left-eye samples, the model finally converges to detect both right eyes and left eyes at the same time. We visualize the prediction of CLERA on the training set in Figure 4, which demonstrates that the model learns to predict both eyes using this training strategy.

5.2.4 Results. Table 4 shows the experimental results for eye state and landmark detection. Since the models are jointly trained for multiple tasks, we increase the number of training iterations to 80k and the batch size to 32 for better convergence. We compare the proposed method to the comparison methods as described in Sec. 5.2.2. The overall results show that while the eye state accuracy stays similar, our method significantly outperforms the other methods on the landmark prediction task.

To dive deeper into the results, first, by comparing Baseline-Mask with CLERA (mask-only), it shows that adding the offset prediction branch helps the joint model to learn better mask predictions; secondly, by comparing the full CLERA model with CLERA (mask-only), while the loose metrics (AP.1) stay the same, we observe consistent improvements on strict metrics (AP.04 and more significantly AP.02), and also on overall performance (mAP). This result aligns well with our intuition in proposing the Mask-Localized Regressor for more precise keypoint predictions, which can be better evaluated with strict metrics.

The proposed model is also efficient and runs in real time. The FPS is calculated over the whole testing set. We benchmark all of the runtime results using the same desktop machine with Nvidia 1080Ti GPU, and the inference is carried out with one image per batch, with max dimension rescaled to 512.

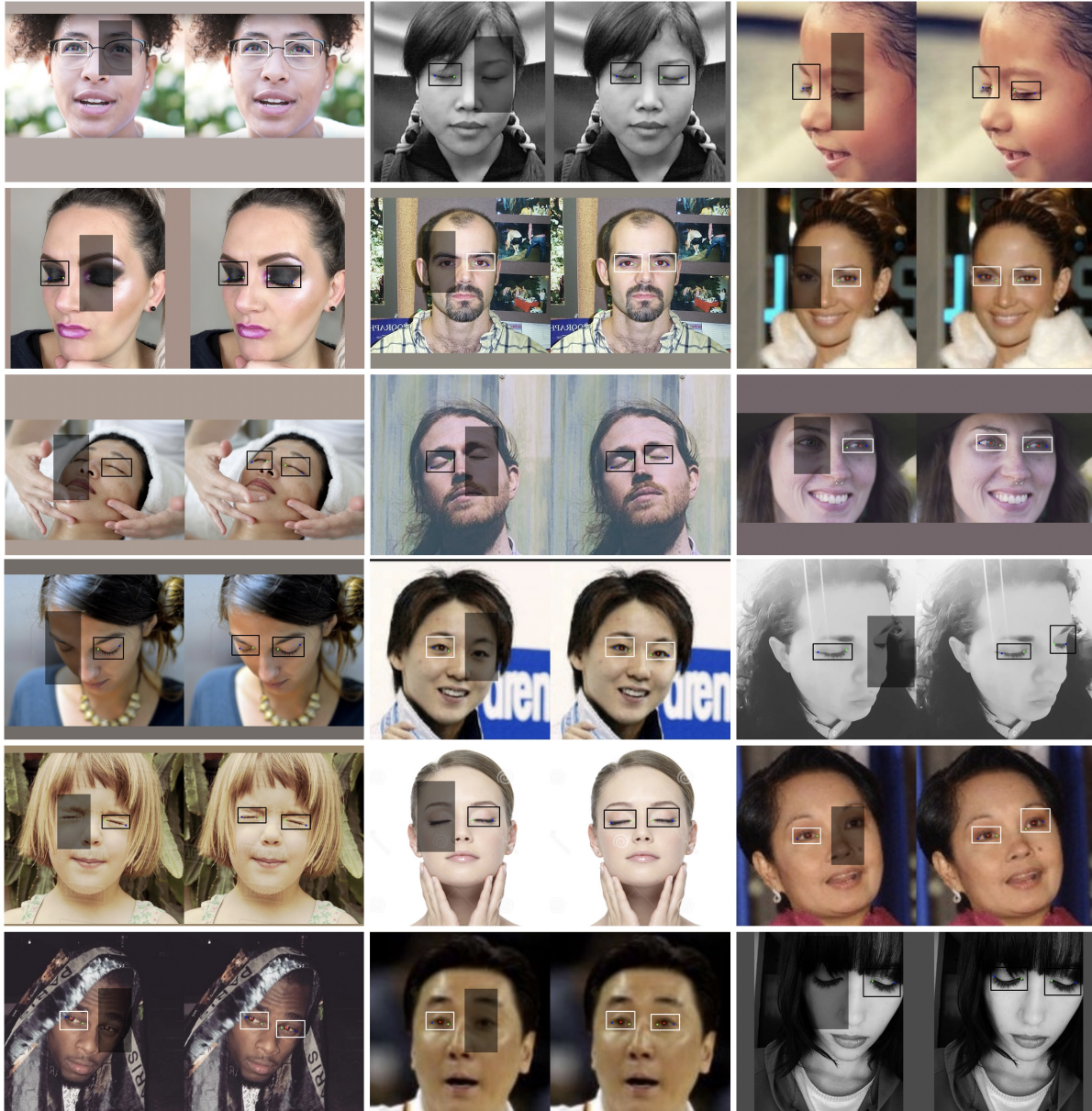


Fig. 4. Sample visualization of pairs of ground truth annotations (on the left) and predictions of CLERA (on the right) on the training set of MIT Pupil Dataset. The color of bounding boxes indicates the eye state (white for open and black for closed). The eye landmarks are visualized as colored dots (blue for lateral canthus, green for medial canthus, and red for pupil). The shadowed area indicates the gradient masking we applied during training with single-eye annotation, which is the potential area of the center of the left eye (or right eye if horizontally flipped) that we do not annotate.

Table 5. Eye landmark detection results on the MPIIGaze dataset.

Methods	mAP	AP.1	AP.04	AP.02
Baseline-Regressor	58.5	86.6	54.0	11.1
Baseline-Mask	64.0	94.7	56.2	11.4
CLERA (mask-only)	65.1	98.3	57.7	11.2
CLERA	65.5	98.4	58.9	11.7

5.3 Cross-dataset Evaluation

5.3.1 Eye Landmark Detection. While there are no similar open-source datasets for real-world non-contact eye landmark detection in unconstrained environments (Table 1), we adopt the MPIIGaze [77, 78] dataset, which also features eye landmark annotation but with a limited set of subjects and environments, and create an external testing set for eye landmark detection. More specifically, we jointly use the face images provided in [77] and the eye landmark annotation provided in [78], resulting in a dataset of 3,877 face images with annotated eye landmarks for pupil, lateral and medial canthus for both eyes.

We perform similar experiments as described in Sec. 5.2, using the same models trained on the MIT Pupil Dataset training set and evaluate on this subset of MPIIGaze dataset. Since there is no bounding box annotation provided, we normalize the errors with the distance between the corners by a factor of 1.3 as an alternative to box width. The results are shown in Table 5.

First, we observe similar overall results showing that the proposed model consistently outperforms the other methods on the landmark prediction task. However, the improvements on strict metrics (AP.04 and AP.02) are not as significant compared to the results in Table 4. In addition, while the same models perform markedly better on MPIIGaze under loose metrics (AP.1) than on the MIT Pupil Dataset, suggesting that MPII is an easier benchmark because of its constraints, the results for strict metrics are actually the opposite. After further investigation, we conclude that this is because the MPIIGaze dataset is of a lower resolution and the landmark annotations are rounded to integer. As a result, MPIIGaze is not sufficient for evaluating keypoints at high precision. We suggest future work adopt the MIT Pupil Dataset for better evaluation of eye landmark detection in terms of both precision and robustness.

5.3.2 Blink Detection. We also evaluate the performance of the proposed method on the RT-BENE dataset [10] as the testing set, which has large-scale blink annotation but with a limited set of subjects, and compare the results to existing methods. We directly use the face images provided in [20] instead of the cropped eye images in [10]. Since the images are of lower-resolution at 224×224, we apply rescaling to 448×448.

The blink prediction is obtained as the predicted state of one detected eye with the highest confidence on each image. We use the whole RT-BENE dataset with 114,490 images. The blink evaluation is only performed on images with at least one detected eye, which corresponds to 99.8% of all the samples. The results are shown in Table 6.

Comparing the RT-BENE models that require cropping of the eye region beforehand and are computationally heavy for only the blink classification task, our model (with ResNet-101 backbone) not only shows competitive performance on blink detection, but more importantly, it is a single model that handles joint eye, blink, and landmark detection in real-time with input at a 2X higher resolution. The results suggest that the proposed model successfully utilizes the shared deep features for multiple tasks. It also shows the generalization of models trained on MIT Pupil Dataset that can be applied on other datasets directly with promising performance.

Table 6. Blink detection results on the RT-BENE dataset. (The FPS of CLERA is calculated for running the full model for joint eye, blink, and landmark detection on a single Nvidia 1080Ti GPU.)

Method	Precision	Recall	AP	F1	FPS
Google ML-Kit [10]	0.172	0.946	0.439	0.290	–
Anas <i>et al.</i> [2]	0.533	0.537	0.486	0.529	408.3
RT-BENE - MobileNetV2 [10]	0.579	0.604	0.642	0.588	42.2
RT-BENE - ResNet [10]	0.595	0.610	0.649	0.598	41.8
CLERA - ResNet	0.571	0.750	0.653	0.648	42.6

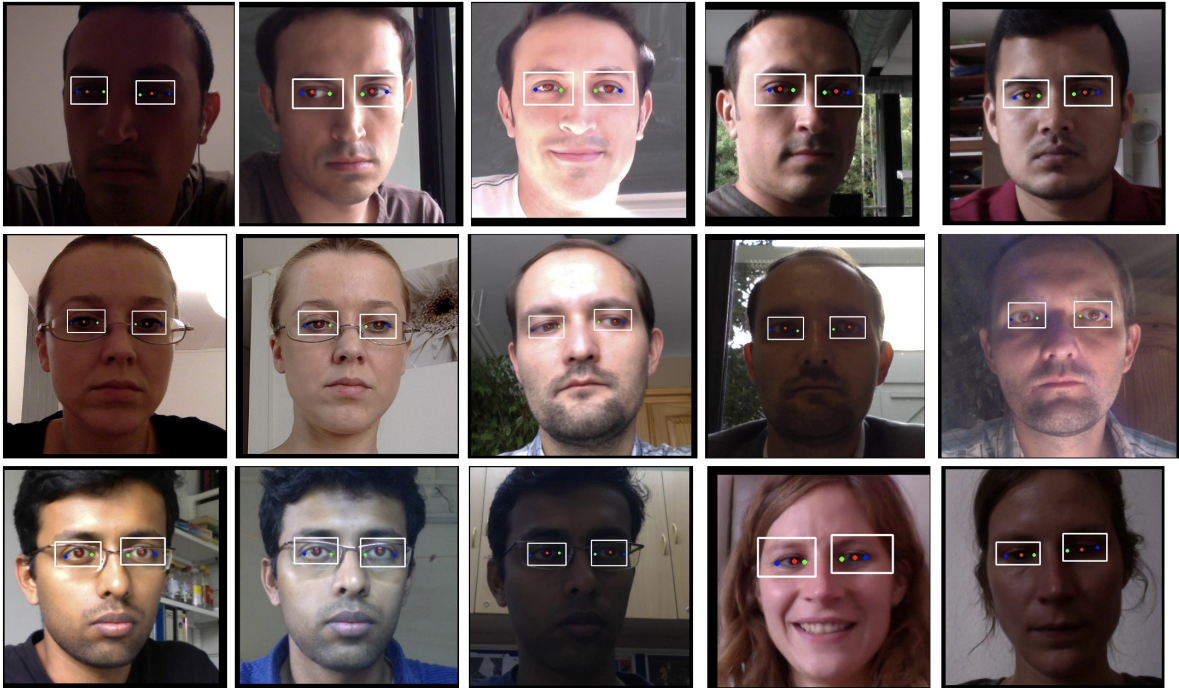


Fig. 5. Sample visualization of predictions on MPIIGaze dataset.

5.3.3 Qualitative Results. We provide example visualizations of predictions of the proposed model on the three testing datasets: MPIIGaze (Fig. 5), RT-BENE (Fig. 6), and MIT Pupil Dataset testing set (Fig. 7). The color of bounding boxes indicates the eye state (white for open and black for closed). The eye landmarks are visualized as color dots (blue for lateral canthus, green for medial canthus, and red for pupil).

6 DISCUSSION

Vision-based characterization of human attention allocation has been receiving increasing attention in recent HCI research, especially modeling related to eye dynamics, which shows great potential in real-world applications such as human-system engagement studies and applied driver monitoring. The main question we explore in

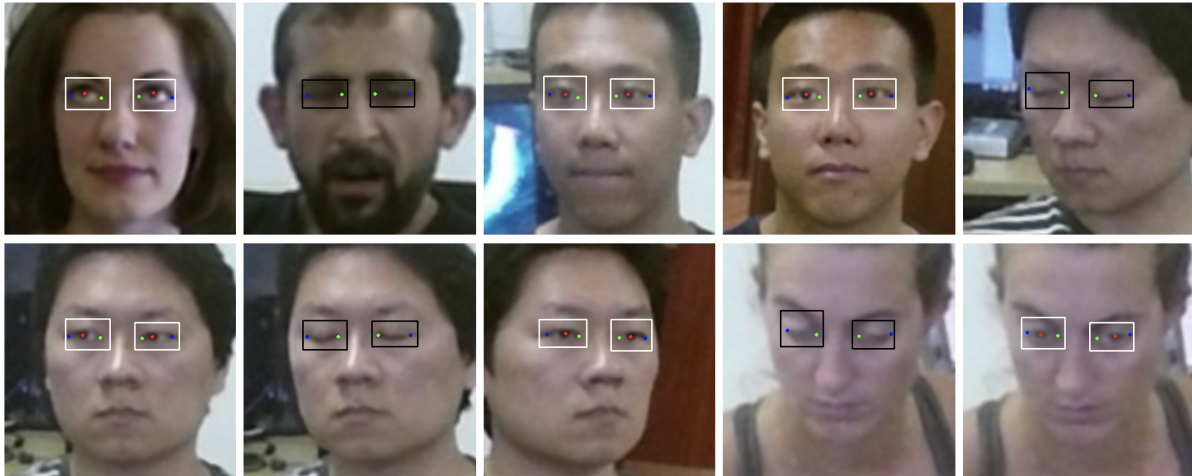


Fig. 6. Sample visualization of predictions on RT-BENE dataset.

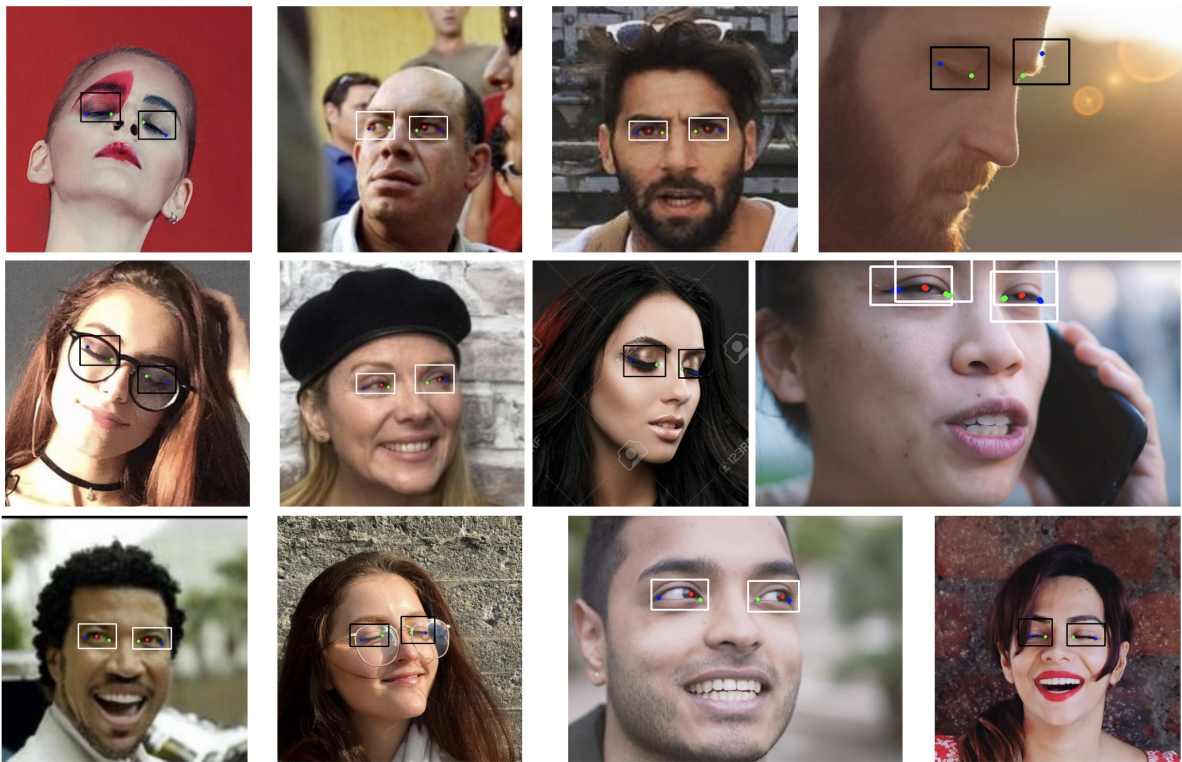


Fig. 7. Sample visualization of predictions on the testing set of MIT Pupil Dataset. Last column shows some failure cases.

this work is whether it's possible to develop a unified, end-to-end model for multiple eye-region analysis and eye-dynamics modeling tasks. Taking cognitive load estimation as an example, current approaches [24, 25] use either eye landmark positions or cropped eye images as input for an eye-dynamics modeling system, both of which require a high-accuracy eye and/or eye landmark detector to be run beforehand.

In our approach, instead of modeling eye dynamics through cropped eye images, we employ a shared deep neural network to extract image features. These are then used with different network heads to perform multiple eye-related tasks, including both low-level tasks like landmark detection and high-level tasks like cognitive load estimation. Multiple experiments show that by using a unified model, we can perform all tasks at almost no additional computational cost compared to a standard eye tracker, while also outperforming prior task-specific models in all tasks, including eye landmark detection, blink detection, and cognitive load estimation.

In the broader context of HCI research, we hope our work will inspire further investigations into more unified modeling of HCI and human factors tasks, rather than focusing solely on specific individual tasks. Our work demonstrates that by utilizing advanced deep learning techniques, the joint modeling of different yet correlated tasks can not only reduce computational cost but also improve the performance of each task. Our proposed CLERA model can support a multitude of HCI research activities involving human attention monitoring. The richness of its output and its capability for real-time monitoring can aid applications ranging from theoretical investigations of human attentional characteristics under various conditions of cognitive load or other states, to assessing the quality of engagement with different human-machine interface conceptual designs and actual implementations [37]. Furthermore, it can increase the practicality and relative cost-effectiveness of operator monitoring systems in aviation [80], air traffic control [1, 75], and power plant systems [74, 76]. It can also address the increasing safety needs of drivers as they shift from primarily active driving to roles involving more system monitoring [34, 49, 57, 58]. In summary, the CLERA model can facilitate the development of adaptive systems which account for variations in cognitive load, thus enhancing the viability of real-time operator support and fostering the improvement of human-centered systems. The large-scale dataset proposed in this work also offers a new source and benchmark for eye-region analysis, a need that has been highlighted in the literature [42], and can be employed to support other research problems related to unified modeling.

7 LIMITATIONS

Previous research on cognitive load estimation has primarily utilized synthetic or controlled environments, such as driving simulators [44, 56], tele-surgical robotic simulations [73], and simulation games [3]. Although these environments offer a controlled and safe way to experiment with cognitive load, their results may not always be generalizable to real-world situations. Our work addresses this issue by focusing on the estimation of cognitive load in real-world settings, which are more complex and variable than controlled environments.

Our study has several limitations that should be considered. Firstly, our dataset for cognitive load estimation includes varying lighting conditions and camera placements, which may impact the accuracy of the methods used. We implemented two comparison methods from [24, 44] and found that both experienced a decrease in accuracy from above 80% to around 60% for the cognitive load level classification task. While our proposed method for cognitive load estimation shows a significant improvement over previous work, it may not be appropriate to use our model directly in its current form for real-world HCI applications due to performance and safety concerns. Additionally, our study focuses solely on a specific type of cognitive load estimation method that uses computer vision models with eye-dynamics input and does not explore other possible approaches such as glance detection and physiological signals like heart rates. However, we believe that our method could work well in simple and controlled environments without specific tuning or modifications, given its superior performance in our more challenging testing circumstances.

Nonetheless, our work provides valuable insights into the challenges of estimating cognitive load in naturalistic environments and aims to inform the development of more robust models that can be applied in practical settings. Specifically, the MIT Pupil Dataset proposed in this work could assist in the development of more accurate and robust models for eye-related analysis tasks, given that it is the largest open-source dataset in the field. The method proposed in this work can be extended to other HCI tasks such as facial analysis and emotion estimation and can be improved by using better deep learning architectures from the latest computer vision research.

8 CONCLUSION

In this work, we propose CLERA - a deep learning framework for joint cognitive load and eye region analysis. By using a detection model with two novel techniques: Localized Feature Tracking and Mask-Localized Regressor, the proposed model is capable of learning visual feature representations for precise eye bounding box and landmark detection. Additionally, it can track these representations over time and apply temporal modeling for cognitive load estimation. We also introduce the MIT Pupil Dataset, a large-scale, open-source dataset comprised of around 30k images of human faces with joint pupil, eye-openness, and landmark annotations.

The main contribution of our work lies in our demonstration that the tasks of eye-region analysis and eye-dynamics modeling can be jointly modeled. This approach ensures that the computational cost is on par with that of a common eye tracker. Moreover, our model is capable of outperforming prior work in all evaluated tasks, including cognitive load estimation, eye landmark detection, and blink estimation.

In terms of future work, we look forward to exploring other tasks in the area of human factors and human-centered computing that can be modeled through eye and facial movements using the proposed framework. This work also provides a new benchmark for eye-region analysis and can be utilized to support related research areas.

ACKNOWLEDGMENTS

The dataset used in this work is from work supported by *Veoneer*. The views and conclusions expressed are those of the authors and have not been sponsored, approved, or endorsed by *Veoneer*.

REFERENCES

- [1] Ulf Ahlstrom. 2010. An eye for the air traffic controller workload. In *Journal of the Transportation Research Forum*, Vol. 46.
- [2] Essa R Anas, Pedro Henriquez, and Bogdan J Matuszewski. 2017. Online Eye Status Detection in the Wild with Convolutional Neural Networks.. In *VISIGRAPP (6: VISAPP)*. 88–95.
- [3] Tobias Appel, Peter Gerjets, Stefan Hoffmann, Korbinian Moeller, Manuel Ninaus, Christian Scharinger, Natalia Sevchenko, Franz Wortha, and Enkelejda Kasneci. 2023. Cross-Task and Cross-Participant Classification of Cognitive Load in an Emergency Simulation Game. *IEEE Transactions on Affective Computing* 14, 2 (2023), 1558–1571. <https://doi.org/10.1109/TAFFC.2021.3098237>
- [4] Claudio Aracena, Sebastián Basterrech, Václav Snáel, and Juan Velásquez. 2015. Neural networks for emotion recognition based on eye tracking data. In *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2632–2637.
- [5] Ali Borji and Laurent Itti. 2012. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 185–207.
- [6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.
- [7] Bo-Chun Chen, Po-Chen Wu, and Shao-Yi Chien. 2015. Real-time eye localization, blink detection, and gaze estimation system without infrared illumination. In *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 715–719.
- [8] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. 2018. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7103–7112.
- [9] Viviane Clay, Peter König, and Sabine Koenig. 2019. Eye tracking in virtual reality. *Journal of eye movement research* 12, 1 (2019).
- [10] Kévin Cortacero, Tobias Fischer, and Yiannis Demiris. 2019. RT-BENE: A Dataset and Baselines for Real-Time Blink Estimation in Natural Environments. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.
- [11] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine* 18, 1 (2001), 32–80.
- [12] Li Ding and Lex Fridman. 2019. Object as distribution. *arXiv preprint arXiv:1907.12929* (2019).

- [13] Li Ding, Michael Glazer, Meng Wang, Bruce Mehler, Bryan Reimer, and Lex Fridman. 2020. Mit-avt clustered driving scene dataset: Evaluating perception systems in real-world naturalistic driving scenarios. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 232–237.
- [14] Li Ding, Rini Sherony, Bruce Mehler, and Bryan Reimer. 2021. Perceptual Evaluation of Driving Scene Segmentation. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1444–1450.
- [15] Li Ding, Jack Terwilliger, Rini Sherony, Bryan Reimer, and Lex Fridman. 2020. MIT DriveSeg (Manual) Dataset for Dynamic Driving Scene Segmentation. *Massachusetts Institute of Technology AgeLab Technical Report 2020-1, Cambridge, MA* (2020).
- [16] Li Ding, Jack Terwilliger, Rini Sherony, Bryan Reimer, and Lex Fridman. 2020. MIT DriveSeg (Semi-auto) Dataset: Large-scale Semi-automated Annotation of Semantic Driving Scenes. *Massachusetts Institute of Technology AgeLab Technical Report 2020-2, Cambridge, MA* (2020).
- [17] Li Ding, Jack Terwilliger, Rini Sherony, Bryan Reimer, and Lex Fridman. 2021. Value of temporal dynamics information in driving scene segmentation. *IEEE Transactions on Intelligent Vehicles* 7, 1 (2021), 113–122.
- [18] Tomas Drutarovsky and Andrej Fogelton. 2014. Eye blink detection using variance of motion vectors. In *European Conference on Computer Vision*. Springer, 436–448.
- [19] Gerhard Fischer. 2001. User modeling in human–computer interaction. *User modeling and user-adapted interaction* 11, 1 (2001), 65–86.
- [20] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. 2018. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 334–352.
- [21] A. Fogelton and W. Benesova. 2016. Eye Blink Detection Based on Motion Vectors Analysis. *Comput. Vis. Image Underst.* 148, C (jul 2016), 23–33. <https://doi.org/10.1016/j.cviu.2016.03.011>
- [22] Lex Fridman, Daniel E. Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik, Jack Terwilliger, Aleksandr Patsek, Julia Kindelsberger, Li Ding, Sean Seaman, Alea Mehler, Andrew Sipperley, Anthony Pettinato, Bobbie D. Seppelt, Linda Angell, Bruce Mehler, and Bryan Reimer. 2019. MIT Advanced Vehicle Technology Study: Large-Scale Naturalistic Driving Study of Driver Behavior and Interaction With Automation. *IEEE Access* 7 (2019), 102021–102038. <https://doi.org/10.1109/ACCESS.2019.2926040>
- [23] Lex Fridman, Li Ding, Benedikt Jenik, and Bryan Reimer. 2019. Arguing machines: Human supervision of black box AI systems that make life-critical decisions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.
- [24] Lex Fridman, Bryan Reimer, Bruce Mehler, and William T Freeman. 2018. Cognitive load estimation in the wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–9.
- [25] Lex Fridman, Heishiro Toyoda, Sean Seaman, Bobbie Seppelt, Linda Angell, Joonbum Lee, Bruce Mehler, and Bryan Reimer. 2017. What can be predicted from six seconds of driver glances?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2805–2813.
- [26] Wolfgang Fuhl, Thomas Kübler, Katrin Sippel, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2015. Excuse: Robust pupil detection in real-world scenarios. In *International Conference on Computer Analysis of Images and Patterns*. Springer, 39–51.
- [27] Wen Gao, Bo Cao, Shiguang Shan, Xilin Chen, Delong Zhou, Xiaohua Zhang, and Debin Zhao. 2007. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 38, 1 (2007), 149–161.
- [28] Stephan J Garbin, Yiru Shen, Immo Schuetz, Robert Cavin, Gregory Hughes, and Sachin S Talathi. 2019. Openeds: Open eye dataset. *arXiv preprint arXiv:1905.03702* (2019).
- [29] Joseph H Goldberg and Anna M Wichansky. 2003. Eye tracking in usability evaluation: A practitioner’s guide. In *the Mind’s Eye*. Elsevier, 493–516.
- [30] Eija Haapalainen, SeungJun Kim, Jodi F Forlizzi, and Anind K Dey. 2010. Psycho-physiological measures for assessing cognitive load. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 301–310.
- [31] Katarzyna Harezlak and Pawel Kasprowski. 2018. Application of eye tracking in medicine: A survey, research issues and challenges. *Computerized Medical Imaging and Graphics* 65 (2018), 176–190. <https://doi.org/10.1016/j.compmedimag.2017.04.006> Advances in Biomedical Image Processing.
- [32] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2980–2988.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [34] Marika Hoedemaeker and Mark Neerincx. 2007. Attuning in-car user interfaces to the momentary cognitive load. In *International Conference on Foundations of Augmented Cognition*. Springer, 286–293.
- [35] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in ‘Real-Life’ Images: detection, alignment, and recognition*.
- [36] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

- [37] Antony William Joseph and Ramaswamy Muruges. 2020. Potential eye tracking metrics and indicators to measure cognitive load in human-computer interaction research. *J. Sci. Res* 64, 1 (2020), 168–175.
- [38] Antony William Joseph, J Sharmila Vaiz, and Ramaswami Muruges. 2021. Modeling Cognitive Load in Mobile Human Computer Interaction Using Eye Tracking Metrics. In *International Conference on Applied Human Factors and Ergonomics*. Springer, 99–106.
- [39] Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini De Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. 2019. Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [40] Ki Kim, Hyung Hong, Gi Nam, and Kang Park. 2017. A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor. *Sensors* 17, 7 (2017), 1534.
- [41] Marc Lalonde, David Byrns, Langis Gagnon, Normand Teasdale, and Denis Laurendeau. 2007. Real-time eye blink detection with GPU-based SIFT tracking. In *Fourth Canadian Conference on Computer and Robot Vision (CRV'07)*. IEEE, 481–487.
- [42] Guohao Lan, Tim Scargill, and Maria Gorlatova. 2022. EyeSyn: Psychology-inspired Eye Movement Synthesis for Gaze-based Activity Recognition. In *Proceedings of ACM/IEEE IPSN*.
- [43] Lucie Lévêque, Hilde Bosmans, Lesley Cockmartin, and Hantao Liu. 2018. State of the Art: Eye-Tracking Studies in Medical Imaging. *IEEE Access* 6 (2018), 37023–37034. <https://doi.org/10.1109/ACCESS.2018.2851451>
- [44] Yulan Liang, Michelle L Reyes, and John D Lee. 2007. Real-time detection of driver cognitive distraction using support vector machines. *IEEE transactions on intelligent transportation systems* 8, 2 (2007), 340–350.
- [45] Jia Zheng Lim, James Mountstephens, and Jason Teo. 2020. Emotion recognition using eye-tracking: taxonomy, review and current challenges. *Sensors* 20, 8 (2020), 2384.
- [46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
- [47] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato. 2014. Adaptive linear regression for appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence* 36, 10 (2014), 2033–2046.
- [48] Päivi Majaranta and Andreas Bulling. 2014. Eye tracking and eye-based human-computer interaction. In *Advances in physiological computing*. Springer, 39–65.
- [49] Bruce Mehler. 2020. Is supportive driver monitoring needed to maximize trust, use, and the safety-benefits of collaborative automation?
- [50] Bruce Mehler, Bryan Reimer, and Joseph F Coughlin. 2012. Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: an on-road study across three age groups. *Human factors* 54, 3 (2012), 396–412.
- [51] Jessica S Oliveira, Felipe O Franco, Mirian C Revers, Andréia F Silva, Joana Portolese, Helena Brentani, Ariane Machado-Lima, and Fátima LS Nunes. 2021. Computer-aided autism diagnosis based on visual attention models using eye tracking. *Scientific reports* 11, 1 (2021), 1–11.
- [52] Fred Paas, Juhani E Tuovinen, Huib Tabbers, and Pascal WM Van Gerven. 2003. Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist* 38, 1 (2003), 63–71.
- [53] Gang Pan, Lin Sun, Zhaohui Wu, and Shihong Lao. 2007. Eyeblink-based anti-spoofing in face recognition from a generic webcam. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 1–8.
- [54] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. 2017. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4903–4911.
- [55] Anjul Patney, Joohwan Kim, Marco Salvi, Anton Kaplanyan, Chris Wyman, Nir Benty, Aaron Lefohn, and David Luebke. 2016. Perceptually-based foveated virtual reality. In *ACM SIGGRAPH 2016 Emerging Technologies*. 1–2.
- [56] Marco Pedrotti, Mohammad Ali Mirzaei, Adrien Tedesco, Jean-Rémy Chardonnet, Frédéric Mérienne, Simone Benedetto, and Thierry Baccino. 2014. Automatic stress classification with pupil diameter analysis. *International Journal of Human-Computer Interaction* 30, 3 (2014), 220–236.
- [57] Prarthana Pillai, Balakumar Balasingam, Yong Hoon Kim, Chris Lee, and Francesco Biondi. 2022. Eye-Gaze Metrics for Cognitive Load Detection on a Driving Simulator. *IEEE/ASME Transactions on Mechatronics* 27, 4 (2022), 2134–2141. <https://doi.org/10.1109/TMECH.2022.3175774>
- [58] P Ramakrishnan, B Balasingam, and F Biondi. 2021. Cognitive load estimation for adaptive human-machine system automation. In *Learning control*. Elsevier, 35–58.
- [59] Miguel A Recarte and Luis M Nunes. 2003. Mental workload while driving: effects on visual search, discrimination, and decision making. *Journal of experimental psychology: Applied* 9, 2 (2003), 119.
- [60] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.
- [61] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).
- [62] Bryan Reimer, Bruce Mehler, Ying Wang, and Joseph F Coughlin. 2012. A field study on the impact of variations in short-term memory demands on drivers' visual attention and driving performance across three age groups. *Human Factors* 54, 3 (2012), 454–468.

- [63] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
- [64] Lars Schillingmann and Yukie Nagai. 2015. Yet another gaze detector: An embodied calibration free system for the iCub robot. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 8–13.
- [65] Rémy Siegfried, Yu Yu, and Jean-Marc Odobez. 2019. A deep learning approach for robust head pose independent eye movements recognition from videos. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ACM, 31.
- [66] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.
- [67] Fengyi Song, Xiaoyang Tan, Xue Liu, and Songcan Chen. 2014. Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. *Pattern Recognition* 47, 9 (2014), 2825–2838.
- [68] John Sweller, Paul Ayres, and Slava Kalyuga. 2011. Measuring cognitive load. In *Cognitive load theory*. Springer, 71–85.
- [69] Lech Świrski, Andreas Bulling, and Neil Dodgson. 2012. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 173–176.
- [70] Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2016. Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. 139–142.
- [71] Ying Wang, Bryan Reimer, Jonathan Dobres, and Bruce Mehler. 2014. The sensitivity of different methodologies for characterizing drivers’ gaze concentration under increased cognitive demand. *Transportation Research Part F: Traffic Psychology and Behaviour* 26 (2014), 227–237. <https://doi.org/10.1016/j.trf.2014.08.003>
- [72] Markus Weber. 2022. Caltech Face Dataset 1999. <https://doi.org/10.22002/D1.20237>
- [73] Chuhao Wu, Jackie Cha, Jay Sulek, Tian Zhou, Chandru P Sundaram, Juan Wachs, and Denny Yu. 2020. Eye-tracking metrics predict perceived workload in robotic surgical skills training. *Human factors* 62, 8 (2020), 1365–1386.
- [74] Shengyuan Yan, Cong Chi Tran, Yu Chen, Ke Tan, and Jean Luc Habiyaremye. 2017. Effect of user interface layout on the operators’ mental workload in emergency operating procedures in nuclear power plants. *Nuclear Engineering and Design* 322 (2017), 266–276. <https://doi.org/10.1016/j.nucengdes.2017.07.012>
- [75] Ebru Yazgan, SERT Erdi, and Deniz ŞİMŞEK. 2021. Overview of Studies on the Cognitive Workload of the Air Traffic Controller. *International Journal of Aviation Science and Technology* 2, 01 (2021), 28–36.
- [76] Jingling Zhang, Daizhong Su, Yan Zhuang, and QIU Furong. 2020. Study on cognitive load of OM interface and eye movement experiment for nuclear power system. *Nuclear Engineering and Technology* 52, 1 (2020), 78–86.
- [77] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. It’s written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2299–2308.
- [78] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence* 41, 1 (2017), 162–175.
- [79] Yilu Zhang, Yuri Owechko, and Jing Zhang. 2004. Driver cognitive workload estimation: A data-driven perspective. In *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*. IEEE, 642–647.
- [80] Gal Ziv. 2016. Gaze behavior and visual attention: A review of eye tracking studies in aviation. *The International Journal of Aviation Psychology* 26, 3-4 (2016), 75–104.