

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

1-21-2021

### Towards Robust Gaze Estimation and Classification in Naturalistic Conditions

Rakshit S. Kothari  
rsk3900@rit.edu

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Kothari, Rakshit S., "Towards Robust Gaze Estimation and Classification in Naturalistic Conditions" (2021). Thesis. Rochester Institute of Technology. Accessed from

This Dissertation is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

# Towards Robust Gaze Estimation and Classification in Naturalistic Conditions

by

Rakshit S. Kothari

B.S. Nirma Institute of Technology, 2012

M.S. Rochester Institute of Technology, 2014

A dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy  
in the Chester F. Carlson Center for Imaging Science

College of Science

Rochester Institute of Technology

January 21, 2021

Signature of the Author \_\_\_\_\_

Accepted by \_\_\_\_\_  
Coordinator, Ph.D. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE  
COLLEGE OF SCIENCE  
ROCHESTER INSTITUTE OF TECHNOLOGY  
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

---

Ph.D. DEGREE DISSERTATION

---

The Ph.D. Degree Dissertation of Rakshit S. Kothari  
has been examined and approved by the  
dissertation committee as satisfactory for the  
dissertation required for the  
Ph.D. degree in Imaging Science

---

|   |      |
|---|------|
| Dr. Gabriel J. Diaz, Dissertation Advisor | Date |
|---|------|

---

|                                       |      |
|---------------------------------------|------|
| Dr. Reynold J. Bailey, External Chair | Date |
|---------------------------------------|------|

---

|                       |      |
|-----------------------|------|
| Dr. Christopher Kanan | Date |
|-----------------------|------|

---

|                  |      |
|------------------|------|
| Dr. Jeff B. Pelz | Date |
|------------------|------|

*I dedicate this thesis to ...*

*My partner and hope, Shikha, who supported me through highs and lows*

*My mother, Yashda, who stands as a testament of everything I have achieved*

*My father, Sunil, who is a rock of stability during turbulent waves of my life*

*My sister, Vama, for lighting little candles of happiness when it appeared all dark*

*... and my brothers, Aneesh (Ramu), Sanketh (Raju) & Hobs (Ravi) for the company.*



## Acknowledgements

I would like to acknowledge my advisor and friend, Dr. Gabriel J. Diaz, who provided this wonderful opportunity to work under his guidance and support. I would also to acknowledge Dr. Jeff B. Pelz for improving this research with critical reasoning and valuable sanity checks, Dr. Reynold J. Bailey for supporting this research with valuable feedback and advice and Dr. Christopher Kanan, for essential insights towards machine learning and data analysis. I thank Research Computing at the Rochester Institute of Technology for providing resources crucial for this research. Finally, I would like to acknowledge all my collaborators for their effort and help.

# Towards Robust Gaze Estimation and Classification in Naturalistic Conditions

by

Rakshit S. Kothari

Submitted to the  
Chester F. Carlson Center for Imaging Science  
in partial fulfillment of the requirements  
for the Doctor of Philosophy Degree  
at the Rochester Institute of Technology

## Abstract

Eye movements help us identify when and where we are fixating. The location under fixation is a valuable source of information in decoding a person’s intent or as an input modality for human-computer interaction. However, it can be difficult to maintain fixation under motion unless our eyes compensate for body movement. Humans have evolved compensatory mechanisms using the vestibulo-ocular reflex pathway which ensures stable fixation under motion. The interaction between the vestibular and ocular system has primarily been studied in controlled environments, with comparatively few studies during natural tasks that involve coordinated head and eye movements under unrestrained body motion. Moreover, off-the-shelf tools for analyzing gaze events perform poorly when head movements are allowed. To address these issues we developed algorithms for gaze event classification and collected the Gaze-in-Wild (GW) dataset. However, reliable inference of human behavior during in-the-wild activities depends heavily on the quality of gaze data extracted from eyetrackers. State of the art gaze estimation algorithms can be easily affected by occluded eye features, askew eye camera orientation and reflective artifacts from the environments - factors commonly found in unrestrained experiment designs. To inculcate robustness to reflective artifacts, my efforts helped develop RITNet, a convolutional encoder-decoder neural network which successfully segments eye images into semantic parts such as pupil, iris and sclera. Well chosen data augmentation techniques and objective functions combat reflective artifacts and helped RITNet achieve first place in OpenEDS’19, an international competition organized by Facebook Reality Labs. To induce robustness to occlusions, my efforts resulted in a novel eye image segmentation protocol, EllSeg. EllSeg demonstrates state of the art pupil and iris detection despite the presence of reflective artifacts and occlusions. While our efforts have shown promising

results in developing a reliable and robust gaze feature extractor, convolutional neural networks are prone to overfitting and do not generalize well beyond the distribution of data it was optimized on. To mitigate this limitation and explore the generalization capacity of EllSeg, we acquire a wide distribution of eye images sourced from multiple publicly available datasets to develop EllSeg-Gen, a domain generalization framework for segmenting eye imagery. EllSeg-Gen proposes four tests which allow us to quantify generalization. We find that jointly training with multiple datasets improves generalization for eye images acquired outdoors. In contrast, specialized dataset specific models are better suited for indoor domain generalization. Encouraging results indicate that optimizing EllSeg on multiple datasets results in a single model generalizable across multiple domains.

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                               | <b>21</b> |
| <b>2</b> | <b>Background</b>                                 | <b>24</b> |
| 2.1      | The human visual system . . . . .                 | 24        |
| 2.1.1    | The human eye . . . . .                           | 24        |
| 2.1.2    | Eye movements . . . . .                           | 25        |
| 2.2      | Eye and Head movements . . . . .                  | 27        |
| 2.2.1    | Compensatory mechanisms . . . . .                 | 27        |
| 2.2.2    | Vestibular-Ocular Suppression . . . . .           | 28        |
| 2.2.3    | Anticipatory head movements . . . . .             | 28        |
| 2.3      | Mobile Eyetracking . . . . .                      | 28        |
| 2.3.1    | Feature based approaches . . . . .                | 30        |
| 2.3.2    | Model based approaches . . . . .                  | 31        |
| 2.3.3    | Limitations . . . . .                             | 33        |
| 2.3.4    | Machine Learning approaches . . . . .             | 34        |
| 2.4      | Event detection . . . . .                         | 38        |
| <b>3</b> | <b>Gaze in Wild</b>                               | <b>40</b> |
| 3.1      | Head-free gaze movement nomenclature . . . . .    | 41        |
| 3.2      | Methodology . . . . .                             | 44        |
| 3.3      | Hardware setup and error categorization . . . . . | 47        |
| 3.3.1    | Pupil labs eye tracking glasses (ETG) . . . . .   | 47        |
| 3.3.2    | Inertial Measurement Unit (IMU) . . . . .         | 48        |
| 3.3.3    | ZED Stereo camera . . . . .                       | 49        |
| 3.4      | System calibration . . . . .                      | 49        |
| 3.4.1    | Pupil Labs eye tracker calibration . . . . .      | 50        |
| 3.4.2    | Temporal alignment . . . . .                      | 50        |
| 3.4.3    | ETG-IMU calibration . . . . .                     | 50        |

|          |  |           |
|----------|--|-----------|
| 3.4.4    | ETG-ZED calibration . . . . .  | 52        |
| 3.5      | Operations . . . . .   | 52        |
| 3.6      | Labelling . . . . .  | 54        |
| 3.6.1    | Training labellers . . . . .   | 55        |
| 3.6.2    | Data cleaning and post processing . . . . .                            | 58        |
| <b>4</b> | <b>Gaze classification and analysis</b>                                | <b>59</b> |
| 4.1      | Classification models . . . . .  | 59        |
| 4.2      | Input features . . . . .   | 61        |
| 4.3      | Error metrics . . . . .  | 61        |
| 4.4      | Results . . . . .  | 66        |
| 4.5      | Ablation study . . . . .   | 69        |
| 4.6      | Discussion . . . . .   | 70        |
| 4.6.1    | Lower gaze pursuit classification performance by classifiers . . . . . | 70        |
| 4.6.2    | Head tracking: A pursuit or fixation? . . . . .                        | 70        |
| 4.6.3    | Head and Eye tracking can have different coordinate systems . . . . .  | 71        |
| 4.6.4    | Gaze-in-world information for classification . . . . .                 | 71        |
| 4.6.5    | General limitations . . . . .  | 72        |
| 4.6.6    | Limitations of event-based metrics . . . . .                           | 72        |
| 4.7      | Conclusion . . . . .   | 73        |
| <b>5</b> | <b>RITnet</b>  | <b>74</b> |
| 5.1      | Abstract . . . . .   | 74        |
| 5.2      | Introduction . . . . .   | 74        |
| 5.3      | Previous Works . . . . .   | 75        |
| 5.4      | Proposed Model: RITnet . . . . .                                       | 76        |
| 5.4.1    | Loss functions . . . . .   | 76        |
| 5.5      | Experimental Details . . . . .   | 77        |
| 5.5.1    | Dataset and Evaluation . . . . .                                       | 77        |
| 5.5.2    | Training . . . . .   | 77        |
| 5.5.3    | Data Pre-processing . . . . .  | 77        |
| 5.6      | Results . . . . .  | 78        |
| 5.7      | Discussion . . . . .   | 78        |
| 5.8      | Conclusion . . . . .   | 79        |
| <b>6</b> | <b>EllSeg</b>  | <b>85</b> |
| 6.1      | Abstract . . . . .   | 85        |
| 6.2      | Introduction . . . . .   | 85        |
| 6.3      | Related work . . . . .   | 87        |

|          |   |            |
|----------|---|------------|
| 6.4      | Methodology . . . . .   | 88         |
| 6.4.1    | Ellipse center . . . . .  | 89         |
| 6.4.2    | Ellipse axis and orientation . . . . .  | 90         |
| 6.4.3    | Loss functions . . . . .  | 91         |
| 6.5      | Datasets . . . . .  | 91         |
| 6.5.1    | Groundtruth ellipse fits . . . . .  | 91         |
| 6.6      | Experiments and Hypothesis . . . . .  | 92         |
| 6.6.1    | Training . . . . .  | 93         |
| 6.6.2    | Data augmentation . . . . .   | 95         |
| 6.6.3    | Evaluation Metrics . . . . .  | 95         |
| 6.7      | Results and Discussion . . . . .  | 96         |
| 6.7.1    | Comparison with state-of-the-art models . . . . .   | 96         |
| 6.7.2    | Ellipse center estimation . . . . .   | 96         |
| 6.7.3    | Improving the ellipse estimates . . . . .   | 99         |
| 6.7.4    | Center via bottleneck vs softargmax . . . . .   | 103        |
| 6.8      | Summary . . . . .   | 106        |
| 6.9      | Conclusion and future work . . . . .  | 107        |
| 6.10     | Acknowledgements . . . . .  | 107        |
| <b>7</b> | <b>EllSeg-Gen</b>   | <b>108</b> |
| 7.1      | Introduction . . . . .  | 108        |
| 7.2      | Related work . . . . .  | 110        |
| 7.2.1    | Prior shift . . . . .   | 111        |
| 7.2.2    | Covariate shift . . . . .   | 112        |
| 7.2.3    | Mitigating distribution shift . . . . .   | 112        |
| 7.3      | Methods . . . . .   | 114        |
| 7.3.1    | Datasets . . . . .  | 114        |
| 7.3.2    | Network architecture . . . . .  | 114        |
| 7.3.3    | Normalization schemes . . . . .   | 116        |
| 7.3.4    | Model Performance Metrics . . . . .   | 119        |
| 7.3.5    | Generalization tests . . . . .  | 119        |
| 7.3.6    | Predictions . . . . .   | 120        |
| 7.3.7    | Analysis . . . . .  | 121        |
| 7.3.8    | Training details . . . . .  | 122        |
| 7.3.9    | Evaluation criterion . . . . .  | 122        |
| 7.4      | Results and Discussion . . . . .  | 123        |
| 7.4.1    | Hypothesis 1: Training with multiple datasets is an optimal strategy<br>for generalization. . . . . | 123        |

|          |   |            |
|----------|---|------------|
| 7.4.2    | Hypothesis 2: Training with multiple datasets will improve within-dataset performance. . . . .                | 125        |
| 7.4.3    | Hypothesis 3: Effects of data augmentation for generalization . . . .   | 126        |
| 7.4.4    | Which single training dataset offers the best cross-domain performance across all testing datasets? . . . . . | 128        |
| 7.5      | Conclusion . . . . .  | 129        |
| 7.6      | Supplementary data . . . . .  | 129        |
| <b>8</b> | <b>Summary and Conclusions</b>  | <b>137</b> |

# List of Figures

|     |   |    |
|-----|---|----|
| 2.1 | Holistic structure of the human visual system. Image acquired from Stangore <i>et al.</i> [1] . . . . .   | 25 |
| 2.2 | Side-view cross-section of an eyeball. Image acquired from Stangore <i>et al.</i> [1]   | 26 |
| 2.3 | Photoreceptor density across the retinal field, measured in degrees from the fovea. Image acquired from Wandell <i>et al.</i> [2] . . . . .   | 26 |
| 2.4 | Commercially available eyetracker, PupilCore. Bottom left illustrates a person wearing the tracker. Bottom right illustrates a magnified image of the eye camera. The components marked are as follows: 1) scene camera (also known as world camera) 2) nose rest 3) IR eye camera 4) IR emitters (which produces bright glints on eye imagery) 5) Eye camera location . . .  | 29 |
| 2.5 | Top-down view of the Reduced Le Grand left eye model [3]. Each grid unit represents $2mm$ . . . . .   | 31 |
| 2.6 | Holistic pipeline for a mobile eyetracker. Eye and scene images with their timestamps are acquired from near-eye IR cameras and a scene camera. Computer Vision or Machine Learning techniques are employed to segment the eye image into parts of interest such as the pupil (shown in yellow overlay) and iris (shown in green overlay). This is followed by fitting an ellipse on the pupil and/or the iris segments, $x$ . Model based approaches identify the center of eyeball rotation over multiple frames and improve the ellipse estimates. During calibration, subjects are asked to fixate known calibration targets $y_c$ in the scene (example calibration pattern taken from Binaee <i>et al.</i> [4]). Finally, known targets and extracted ellipse estimates are used to estimate a mapping function $C = f(x_c, y_c)$ . For all incoming eye images, gaze is estimated using $y_g = C(x)$ . . . . . | 32 |
| 2.7 | A collection of eye images acquired with varying degrees of degradation. Top row images are extracted from the GW dataset [5]. Second and third rows contain hand selected images from the Labelled Pupils in the Wild dataset (LPW) [6]. . . . .   | 33 |



|     |  |    |
|-----|--|----|
| 2.8 | General framework for an encoder-decoder architecture. The architecture comprises of blocks which are a collection of convolutional layers. An input eye image with a resolution of $W \times H$ is passed through a series of $k$ convolutional $d$ -blocks or down-blocks which down-sample the intermediate output to a lower resolution while increasing the features extracted. The intermediate representation of the image following the encoder $d$ -blocks is passed through a series of up-sampling convolutional $u$ -blocks (or up-blocks) to produce a fine grained semantic segmentation output of the image. This bottleneck or latent representation of an image, is a low spatial resolution matrix with a large feature space. Typically, outputs from encoder blocks are also fed directly into decoder blocks. For more information on encoder-decoder networks, I refer the reader to Ronneberger <i>et al.</i> [7] . . . . . | 36 |
| 2.9 | The effect of training a network with multiple loss functions for semantic segmentation. Left: eye image acquired from the OpenEDS [8] dataset. Middle: segmentation output without structural awareness. Right: Segmentation output with structural awareness. Segmentation results are provided by RITnet [9]. Note the stray misclassifications which are effectively removed when a network is trained with two loss functions. . . . .  | 37 |
| 3.1 | Eye and head movement statistics. The top row signifies absolute eye and head velocity. The bottom row signifies the distribution of eye and head velocity in the azimuth and elevation direction. The left column illustrate fixations when subjects were stationary. The middle column illustrates fixations when subjects were in translatory motion. The right column illustrates pursuit behavior. The scale on the right shows the normalized concentration of samples and is used in all figures. . . . .   | 44 |
| 3.2 | Task selections in the GW dataset. Left to right $\rightarrow$ Indoor navigation, ball catching, visual search and tea making. . . . .   | 45 |
| 3.3 | (left) side-view, (middle) front view of hardware setup. (right) Top view of all trajectories within our world coordinate system. The red box indicates the position of the calibration pattern. The purple box signifies the region where subject stood during calibration. . . . .   | 47 |
| 3.4 | Head pose (right) and cyclopean eye distribution (left) in the azimuthal and elevation direction. The cyclopean eye distribution is reported in the Eye-in-Head coordinate system. Angles are provided in degrees. Note that head distribution peaks occur at $90^\circ$ intervals. . . . .  | 48 |

|     |   |    |
|-----|---|----|
| 3.5 | Checkerboard pattern placed in front of a participant during calibration phase. The red cross marks are used to calibrate the eye tracker in routine 1. The checkerboard corners are used for a 2-way multiview calibration between the ZED stereo camera $C_Z$ and the eye tracker world camera $C_E$ . $(R_Z^O, T_Z^O) \rightarrow$ Transformation needed to move from $C_Z$ to the calibration chart's coordinate system, $C_O$ . $(R_O^E, T_O^E) \rightarrow$ Transformation needed to move from $C_O$ to $C_E$ . . . . .   | 51 |
| 3.6 | Eye tracker accuracy vs eccentricity from the center of the calibration pattern. Color scale indicates the number of calibration samples from all subjects.   | 53 |
| 3.7 | Custom made GUI for labelling. 1: Magnitude of EiH velocity with $Az$ (Azimuthal) and $El$ (Elevation) velocity traces ( $^\circ/s$ ). 2: Magnitude of Head velocity with $Az$ and $El$ velocity traces ( $^\circ/s$ ). 3: World-view overlaid with the Point-of-Regard (PoR) and confidence score. 4: Eye-view. 5: Slider to move a window through a recording temporally. 6: Slider to change the window width. 7: <i>Go-to</i> and <i>Remove button</i> for labelled regions. 8: Interactive list of labels in a session. 9: Radio buttons to select event type and mark across a region. 10: Toggle scene and depth view. 11: Record a 10 second clip of GUI starting at the current sample. 12: Slider to shift labels forward or backward. . . . .                                | 56 |
| 4.1 | Bidirectional recurrent network model architecture. The model takes the magnitude, azimuthal and elevation eye and head velocity (6 features) as its input, passes through $k$ fully connected feature extraction layers. These features are fed into a stack of $k$ GRU layers which learn temporal patterns to classify a sample $x_t$ . The forward variant (fRNN) outputs a 24 dimensional vector instead of 48 before being reduced to 3 at the final FC layer. . . . .  | 60 |
| 4.2 | Illustration of the ELC metric on handcrafted test and reference sequences. (L1) Labels provided by labeller 1. (L2) Labels provided by labeller 2. Colors indicate the event type and whether labellers are in agreement. Dotted lines from L1 into L2 indicate the time window used in ELC for each transition point. (C1) Direct sample-sample comparison between labeller 1 and labeller 2. (L1p, L2p) Results of applying ELC to labels provided by labeller 1 and labeller 2 respectively. (C2) Event-level comparison between labeller 1 and labeller 2. Unmatched regions are given specific labels describing the misclassification type. For example, 'S-B' means that labeller 1 labelled the data as gaze shift whereas labeller 2 labelled the same data as blink. . . . . | 64 |

|     |   |    |
|-----|---|----|
| 4.3 | Sample level performance metrics. All performance curves are centered around their mean, $\mu \pm$ standard error. Left - Overall $\kappa$ score. Inner left - Gaze fixation $\kappa$ score. Inner right - Gaze pursuit $\kappa$ score. Right - Saccade $\kappa$ score. Please note the varying y-limits to accentuate the difference in performance. RNN uses memory to encode temporal patterns, and hence the RNN architectures are represented as horizontal lines as they do not operate in window sizes. We would like to highlight that all window sizes are in the velocity domain. Window sizes in angular domain can be derived by adding 10 <i>ms</i> (please refer to Section 3.5). . . . . | 68 |
| 5.1 | Comparison of model performance on difficult samples in the OpenEDS test-set. Top row, left to right, shows eyes obstructed due to prescription glasses, heavy mascara, dim light, and partial eyelid closure. Rows from top to bottom show input test images, ground truth labels, predictions from mSegNet w/BR [8] and predictions from RITnet, respectively. Compared to other methods, RITnet's output more closely matches the ground truth.  | 81 |
| 5.2 | Architecture details of RITnet. DB refers to <i>Down</i> -Block, UB refers to <i>Up</i> -Block, and BN stands for batch normalization. Similarly, $m$ refers to the number of input channels ( $m = 1$ for gray scale image), $c$ refers to number of output labels and $p$ refers to number of model parameters. Dashed lines denote the skip connections from the corresponding <i>Down</i> -Blocks. All of the Blocks output tensors of channel size $m=32$ . . . . .  | 82 |
| 5.3 | Left to right: Original image, image after gamma correction, and image after CLAHE is applied. Note that in the rightmost image, it is comparatively easier to distinguish iris and pupil. . . . .  | 83 |
| 5.4 | Generation of a <i>starburst</i> pattern from the training image 000000240768. Left to Right: Original image, selected reflections, concatenating with its 180° rotation, final pattern mask (best viewed in color). . . . .  | 83 |
| 5.5 | Our model struggles to do an accurate segmentation when eye masks are heavily blurred or defocused. Despite failure in segmentation, the segmentation output maps can be salvaged to produced plausible pupil or iris ellipse fits. . . . .   | 84 |
| 6.1 | <i>PartSeg vs EllSeg</i> . Left: A <i>Four-class</i> eye part segmentation at the pixel-level (i.e. PartSeg) produces labelled pupil (yellow), iris (green), sclera (blue) and background (purple) classes. Right: The EllSeg (three-class) modification produces labelled pupil (yellow) and iris (green) elliptical regions and the rest is marked as background (purple). . . . .  | 87 |

|     |  |     |
|-----|--|-----|
| 6.2 | Proposed EllSeg framework (region enclosed by red dotted line) builds upon existing CNN-based approaches to facilitate the simultaneous segmentation and ellipse prediction for both iris and pupil regions. The resulting ellipse parameters are highlighted in the blue box. . . . .   | 89  |
| 6.3 | Regression module architecture. The $\downarrow$ signifies average pooling to $1/2$ the resolution. Tensors are flattened after three convolutional layers and passed through two linear layers before regressing 10 values (5 ellipse parameters for pupil and iris each). . . . .  | 89  |
| 6.4 | Ellipse fitting quality on ground truth PartSeg masks. These fits are further used to generate EllSeg masks for the OpenEDS dataset. . . . .   | 93  |
| 6.5 | Summary of all experiments described in following sections (Center estimates are best viewed on screen). . . . .   | 94  |
| 6.6 | Visualization of goodness of fit metrics used in the paper. <b>(a)</b> Groundtruth ellipse (pupil or iris). <b>(b)</b> Corresponding predicted ellipse. The rectangular boxes denote ellipse-axis-aligned bounding boxes for the respective ellipses. <b>(c)</b> denotes the bounding box overlap region and <b>(d)</b> illustrates the angular difference between the two ellipses. . . . .   | 96  |
| 6.7 | <i>PartSeg vs EllSeg</i> : The pupil detection rate (top row) and iris detection rate (bottom row) as a function of the threshold for tolerated pixel error for center approximation for OpenEDS (left column), NVGaze (middle column) and RIT-Eyes (right column). Results for three architectures RITnet, DeepVOG and DenseElNet are present for both cases PartSeg (dashed lines) and EllSeg (solid lines). Note that only the pupil detection rate is shown for the DeepVOG architecture. All detection rates presented here are derived using ellipse fits on segmentation outputs on images sized at $320 \times 240$ . Here, one pixel error corresponds to 0.25% of the image diagonal length. . . . . | 98  |
| 6.8 | <i>EllSeg with and without <math>\mathcal{L}_{COM}</math> loss</i> : The pupil detection rate (top row) and iris detection rate (bottom row) for various pixel error thresholds of center approximation for three datasets. Models (RITnet, DenseElNet and DeepVOG) are trained with the EllSeg framework before the pupil center is estimated using either the ElliFit segmentation output map, or with $\mathcal{L}_{COM}$ loss. The result for non-CNN based model ExCuSe, PuRe and PuReST are also shown. One pixel error corresponds to 0.25% of the image diagonal length. . . . .   | 100 |

|      |  |     |
|------|--|-----|
| 6.9  | Violin plots of boundary overlap IoU (1st and 2nd row: top dashed box), orientation error (3rd and 4th row: middle solid box), and segmentation IoU score (last three rows: bottom dashed box) following EllSeg framework by RITnet and DenseElNet, with or without $\mathcal{L}_{COM}$ loss ( $\mathcal{L}_{COM}$ vs Ellipse), following application to the OpenEDS, NVGaze, and RIT-Eyes datasets (columns) . . . . .          | 102 |
| 6.10 | DenseElNet model prediction and its respective ground truth for OpenEDS, NVGaze and RIT-Eyes dataset. . . . .  | 104 |
| 6.11 | Figure showing 2D activation maps. Columns (L-R): Original image (1st column), activation maps for background, iris and pupil class for model DenseElNet <i>without</i> $\mathcal{L}_{COM}$ (2nd-4th column) <i>with</i> $\mathcal{L}_{COM}$ (5th-7th column). Three rows show three different cases with bottom two having the original image in the background for reference. ( <i>Best viewed on screen</i> ) . . . . .       | 105 |
| 6.12 | A horizontal line scan across the pupil center to visualize DenseElNet output behavior without $\mathcal{L}_{COM}$ (left) and with $\mathcal{L}_{COM}$ (right). The inclusion of $\mathcal{L}_{COM}$ generates characteristic peaks which do not impede the task of semantic segmentation while effectively scaling output pixel activations near the predicted pupil and iris centers ( <i>Best viewed on screen</i> ). . . . . | 105 |
| 6.13 | The difference between pupil and iris detection rate in the OpenEDS dataset. Estimates are derived from the latent space and final segmentation maps (DenseElNet). . . . .   | 106 |
| 7.1  | An illustration to visualize two different training strategies one may adopt to maximize generalization. If $D_3$ represents a test distribution, then it is intuitive to train a model using a combination of $D_1$ , $D_2$ and $D_4$ . However, if $D_4$ represents a test distribution then a model specific to $D_3$ would demonstrate optimal generalization. . . . .   | 110 |
| 7.2  | DenseElNet architecture adapted from EllSeg [10]. The number of parameters in the encoder are controlled by a base channel size of $C$ and a growth rate of $\alpha$ . . . . .   | 117 |
| 7.3  | A mock example to illustrate the difference between Batch and Instance Normalization using normalized features extracted from images sampled from domains $d_1$ , $d_2$ , $d_3$ and $d_4$ . (Left) Batch normalization centers the extracted features relative to the global mean estimate of all available domains while Instance Normalization (right) utilizes each individual image based statistic. . . . .                 | 118 |

- 7.4 Generalization test results. Each box plot highlights a model's performance centered to the within-dataset limit for each domain. The line and notch present within each box plot represents the median and 95% confidence interval respectively while the ends of each box denotes the 1<sup>st</sup> and 3<sup>rd</sup> quartile. All images are 320×240 resolution. Note that datasets which are missing groundtruth annotations do not have a boxplot entry. All measures are centered to the within-dataset performance limit. . . . . 124
- 7.5 Generalization test results to study the effects of data augmentation. Each box plot highlights a model's performance centered to the within-dataset limit for each domain. The line and notch present within each box plot represents the median and confidence interval respectively while the ends of each box denotes the 1<sup>st</sup> and 3<sup>rd</sup> quartile. All images are 320×240 resolution. Note that boxplots pertaining to datasets without a certain groundtruth annotation are missing. All measures are centered to the within-dataset threshold as seen in Figure 7.4 . . . . . 127
- 7.6 All datasets are represented as circular nodes. The arrow emerging from a node points towards its best matching dataset. Nodes colored in red are less constrained datasets with ambient reflections from their surroundings. Nodes colored in blue are constrained datasets with little to no environmental reflections. Best viewed in color. . . . . 128
- 7.7 Pupil center (in pixels) and normalized luminance distribution (in Z-scores) of each eye part across all datasets utilized in our experiments (see Section 6.5). The left and right columns contain statistics from the training and test images for each domain respectively. Due to partial annotations present in some datasets, we leverage the all-vs-one model predictions to segment all eye images into pupil, iris and background segments. Luminance statistics are then accumulated from the predicted segmentation map. . . . 136

# List of Tables

|     |  |    |
|-----|--|----|
| 2.1 | An extensive list of near eye image datasets applicable for my thesis. The Ground Truth (GT) column identifies the type of annotated modality available for each dataset. S→Annotated segments, E→ellipse parameters, C→ellipse center. Note that $C \subset (E, S)$ and $E \subset S$ . P and I correspond to pupil and iris annotations respectively. . . . .  | 38 |
| 3.1 | Sample based Cohens $\kappa$ , precision/recall $p/r$ and $F_1$ score between labellers. Note that the precision and recall values are identical (see section 3.6 for details.) . . . . .  | 56 |
| 3.2 | Inter-labeller event based metrics. All metrics are reported by their mean $\mu$ and inter-subject standard deviation $\sigma$ . $l_2$ distance of the start and end time (expressed in $ms$ ) of matched events using ELC. $O_r$ is the overlap ratio between matched events using ELC. $F_1$ score as proposed by Hooze et al [11]. Event $\kappa$ proposed by Zemblys et al [12]. Event $\kappa^*$ found using ELC event matching. For more information on each metric, please refer to Section 4.3 . . . . . | 57 |
| 3.3 | Normalized sample based confusion matrix (created by normalizing the confusion matrix with the number of samples for each event type in the ground truth) across every recording with multiple labellers. . . . .  | 57 |
| 4.1 | Comparison of event level error metrics . . . . .  | 63 |
| 4.2 | Sample based Cohen’s Kappa score $\kappa$ for each optimized classifier. . . . .   | 67 |
| 4.3 | Metrics based on various event matching techniques proposed by others. Event based $F_1$ score proposed by Hooze et al. [11] Event Error Rate (EER) proposed by Zemblys et al. [12] Event and overall $\kappa$ scores calculated using Zemblys et al. [12] . . . . .   | 68 |

|     |  |     |
|-----|--|-----|
| 4.4 | Standard metrics derived from the ELC confusion matrix. $O_r$ is the overlap ratio between matched events. $l_2$ distance between matched event start and end times and their standard deviation $l_2 - \sigma$ in <i>ms</i> . $l_2$ and $l_2 - \sigma$ are similar to RTO and RTD metrics proposed by Hooze et al. [11] . . . . .   | 68  |
| 4.5 | Sample based $\kappa$ score after removing either head movement information or directional information. . . . .  | 69  |
| 5.1 | Performance comparison on the test split of the OpenEDS dataset. The metrics and comparison models (*) are used as reported in [8]. . . . .  | 79  |
| 6.1 | Summary of datasets. $\uparrow$ and $\downarrow$ correspond to up and down sampling respectively. OpenEDS image crops are extracted around the scleral center followed by up-sampling. Note that images without valid pupil and iris fits are discarded (see Section 6.5). . . . .   | 92  |
| 6.2 | Eye Parts Segmentation: Comparison of <i>pupil</i> (and <i>iris</i> , inside parenthesis) <i>class</i> IoU scores for RITnet, DeepVOG and DenseElNet model architectures (along rows) in OpenEDS, NVGaze and RIT-Eyes dataset (along columns). Bold values indicate the best performance within each dataset. Because DeepVOG was not trained to segment the iris, we are unable to provide iris IOU scores. . . . .   | 97  |
| 6.3 | The percentage of images classified as three categories of occlusion (see Section 6.7.2) for each dataset. Values are presented as pupil (iris). . . . .   | 97  |
| 6.4 | The number of images without valid PartSeg or EllSeg ellipse fits for pupil (and iris, inside parenthesis) for DeepVOG, RITnet, and DenseElNet. The total column represents the number of valid images used for testing (as in section 6.5.1). Bold text (lower number) shows superior performance and illustrates the effectiveness of the EllSeg framework. . . . .  | 99  |
| 6.5 | Comparison of Pupil center estimate errors (in pixels) on various datasets in terms of median scores. Note all the CNN models are trained with EllSeg framework. Image size is $320 \times 240$ . . . . .  | 101 |
| 7.1 | Datasets and their respective train and test splits used to explore generalization. Each dataset is classified into two broad categories called <i>outdoors</i> and <i>constrained</i> . Eye images in outdoor datasets exhibit large proportion of environmental reflections. Constrained datasets are acquired from experiments or synthetically rendered within indoor, lab environments with little to no reflective artifacts. * Approximately a third of LPW recordings were collected outdoors. . . . . | 115 |



|     |  |     |
|-----|--|-----|
| 7.2 | Augmentation schemes applied to every single eye image with a $1/11$ probability. $\mathcal{N}$ and $\mathcal{U}$ indicate a normal and uniform distribution respectively. .         | 123 |
| 7.3 | Segmentation results of various generalization tests proposed in Section 7.3.5. All results represent the IoU metric. . . . .  | 130 |
| 7.4 | Error in pupil center prediction across various generalization tests proposed in Section 7.3.5. All results are presented in unit pixels. . . . .                                    | 131 |
| 7.5 | Error in iris center prediction across various generalization tests proposed in Section 7.3.5. All results are presented in unit pixels. . . . .                                     | 132 |
| 7.6 | Segmentation results of various generalization tests proposed in Section 7.3.5 when utilizing Batch Normalization. All results represent the IoU metric. .                           | 133 |
| 7.7 | Error in pupil center prediction across various generalization tests proposed in Section 7.3.5 when utilizing Batch Normalization. All results are presented in unit pixels. . . . . | 134 |
| 7.8 | Error in iris center prediction across various generalization tests proposed in Section 7.3.5 when utilizing Batch Normalization. All results are presented in unit pixels. . . . .  | 135 |

# Chapter 1

## Introduction

The human visual system (HVS) is arguably one of the most important sensory inputs for our survival. The primary function of the HVS is to provide visually rich, informative and relevant information to other higher order cognitive processes. To provide visual information successfully, optical elements in the eye must focus light from a target or region of interest onto a region of high acuity on the retina, the fovea. Opto-chemical stimulation by the resulting retinal image generates sensory signals which follow the optic nerve and into the Visual Cortex in the posterior of our brain. However, the fovea occupies a very small region on the retina and is incapable of sensing our entire field of vision. Moreover, in the presence of head movements our visual system must successfully stabilize this image to ensure that the relevant visual information is accessible for better decision making or involuntary actions. These difficult challenges are accomplished by strategic eye and head movements (also referred to as *gaze* movements) which operate in synergy with other sensory inputs.

The study of eye movements enable us to perceive how humans respond to their surroundings. Perhaps the biggest advantage in analyzing eye movements is that it provides a non-intrusive method of evaluating brain function and cognition. Eye movements aid us in finding physiological and psychophysical limits of the human body, particularly in context with a task or activity which demands cognitive resources. Abnormalities in eye movements can be used to identify various complex neurological problems. Furthermore, recent technological advances also rely on monitoring eye movements as an input modality for enhancing human-computer interaction and leveraging psychophysical constraints to design better and safer consumable products.

One of the central goals of the vision sciences is to understand the factors which determine where our eyes look. Although gaze movements are drawn to visually salient features in the natural world, the seminal work by Yarbus and others [13, 14, 15, 16]

showed that this effect is overwhelmed in the presence of task, when motor execution requires that attention is directed towards information-rich, task-relevant locations within the visual environment [14, 17, 18]. As a result, the dynamics of gaze coordination in the natural context are affected by properties of the task, the subject’s cognition, the spatial distribution of information in the natural environment [19, 20, 21], and of their interaction [22, 23]. Gaze movements are also driven by factors related to a subjects internal state. These include cognitive resources related to memory or higher order reasoning [24], body movement constraints that determine the dynamics of gaze shifts [25, 26, 27], and bio-mechanical constraints that influence visual strategies.

A common approach to the study of gaze behavior involves investigating the functional contributions of the different types of eye movements that are typical of natural behavior. Human visual behavior can be characterized as a progression of periods with stable visual input, known as fixations, punctuated by ballistic movements to new locations within the visual environment, known as saccades. During fixations, light refracted onto our retina forms a stable and focused image of the world. To fixate at different locations in our visual environment, the human eye performs a saccadic movement from one direction to the other. A third type of movement, known as smooth pursuits, enables us to accurately follow a continuously moving target while providing a clear image using motion compensatory and adaptive mechanisms.

Despite the importance of task related factors and their influences upon gaze behavior during visually guided action, surprisingly little amount of attention has been dedicated to the study of eye movement dynamics in more natural contexts wherein the eyes and head are free to move. Head movements are often suppressed through the use of a chin-rest, and by constraining target movement to a plane subtending only a small portion of the subject’s visual field. Target motion is often restricted to two dimensions, and sometimes viewed from a monocular view-port. A further limitation of traditional experimental setups are that they are limited to enclosed, controlled environments. In contrast, the ability to collect and analyse data in outdoor naturalistic environments enables us to study the diverse range of coordinated head and eye behavior elicited during everyday activities and sports [28], exploration of different surroundings [27], terrain [22] and goals [18, 17, 29].

The study of strategies for coordination of the eyes, head, and body have been limited, in part, due to a lack of suitable algorithms for extraction gaze-relevant features from eye imagery under less constrained lighting. Robust gaze tracking under natural lighting requires advances in two, parallel domains: the development of a robust gaze estimation (*i.e.*, eye tracker) pipeline, and the algorithms to parse the rapid stream of oculomotor events recorded by the eye trackers (*i.e.*, “event detectors”).

The aim of this dissertation is to develop robust eye tracking algorithms and event detectors for the study of eye and head coordination during natural behavior. This thesis

is organized as a systematic collection of projects as detailed in each chapter and in support of my proposed work. Following this first chapter in which the work is summarized and motivated, Chapter 2 will present the foundational works most relevant to the work at hand. Chapter 3 focuses on the development of a new dataset for the study of coordinated head-free gaze behavior - the Gaze-In-Wild dataset [5] (GW). GW was collected from 19 participants engaged in everyday activities using a spatially and temporally calibrated tracker involving a hardhat with an IMU, eye tracking glasses, and a stereo-based RGB-D (RGB imagery plus depth) sensor. A portion (approx 2 hours, 15 minutes of unique labels from 5 labellers) of the GW dataset (approx 5 hours and 10 minutes) is hand-labeled using a custom-made labeling tool (see Section 3.6). Chapter 4 expands upon this work by utilizing the GW dataset in the training of recurrent neural networks for the automated classification of gaze events embedded within the GW dataset. Chapter 5 highlights RITNet, a convolutional neural network that I collaboratively developed to segment regions of interest from near eye imagery in a robust and efficient fashion. Apart from being the best performing model on the OpenEDS dataset at the time of the model’s release, it consists of a significantly low parameter count and is capable of operating at 300Hz on a NVIDIA Titan-XP GPU for 640x480 imagery. Chapter 6 focuses on EllSeg, a three-class (pupil, iris and background) ellipse segmentation protocol on near-eye images. EllSeg demonstrates robustness to occlusion and superior pupil/iris detection as opposed to state of the art techniques such as PuRe [30], DeepVOG [31] and ExCuSe [32] across multiple datasets. While RITNet and EllSeg demonstrate robustness against structured reflections and eye lid occlusions, they do not quantify their performance on unseen images, a paramount requirement to assess how generalizable our solution is. Chapter 7 highlights the generalization capability of EllSeg trained on a broad distribution of eye images sourced from a large number of partially annotated near-eye image datasets. This work compares results across specific tests designed to explore generalization by training and evaluating dataset-specific or pan-dataset models.

## Chapter 2

# Background

In this section, I describe features of the human visual system relevant to gaze behavior and the design of mobile eye tracking systems. This section will also cover previous efforts made towards understanding eye and head movement behavior, existing gaze classification algorithms and robust gaze prediction pipelines.

### 2.1 The human visual system

A holistic structure of the early human visual system (HVS) consists of the human eyeball, the optic nerve, Lateral Geniculate Nucleus (LGN) present within the Thalamus, the striate and extrastriate areas of the visual cortex (see Figure 2.1). It accomplishes three major tasks. First, it successfully collects, quantizes and encodes information acquired from incoming light on the retina within the eye. Second, it transmits this information via neural connections into the primary visual cortex. lastly, neural information is decoded in the visual cortex into meaningful representations such as disparity and motion information along with pattern recognition.

#### 2.1.1 The human eye

Perhaps one of the finest exemplars of optical ingenuity would be the human eye which transmutes light into electrical impulses. It consists of a morphable aperture known as the pupil and an image collecting surface known as the retina. The retina consists of various photo-transducing cells known as photoreceptors. These cells convert input electromagnetic stimulation into signals that can stimulate various subsequent neural processes. Light which reaches the surface of the eye is refracted into the aqueous chamber via the corneal surface. This refraction contributes an unchanging, albeit significant, optical power to

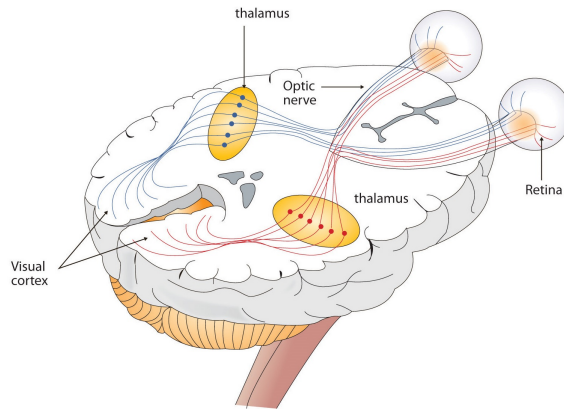


Figure 2.1: Holistic structure of the human visual system. Image acquired from Stangor *et al.* [1]

the human eye. The human eye also consists of a biological lens placed posterior to the aperture. Lens thickness can be modified by stretching or contracting the ciliary muscles which determines its contribution to the optical power of a human eye. When we change our focus from a far to a near object, the optical power of the lens changes accordingly to ensure the image subtended on the retina remains crisp. The light pathway is analogous to an imaging system wherein motor signals control lens action and pupil radius for active focusing (see Figure 2.2).

A real and inverted image of the world is produced on the retinal wall which exhibits an uneven distribution of photoreceptors. There exists a small region subtending  $1^\circ$  to  $2^\circ$  of concentrated photoreceptors known as the fovea (see Figure 2.3) [33]. Such a dense receptor distribution captures fine-grained spatial information from the light imaged on it. The spatial acuity of the retinal field rapidly decreases away from the foveal region. This means that objects imaged towards the periphery do not receive high spatial sampling which results in lack of perceived detail. The entire monocular human visual field subtends  $167^\circ$  horizontally and  $150^\circ$  vertically [34, 35] while the foveated region encompasses  $2^\circ$  ( $\sim 0.8\%$ ) within this field of view. In order to successfully resolve objects with high acuity within this range, the human eye must continuously rotate in its socket to resolve them on the fovea.

### 2.1.2 Eye movements

The function of eye movements are numerous and range from visual exploration, object tracking, world building/mapping, compensating for ocular or neurological defects to re-

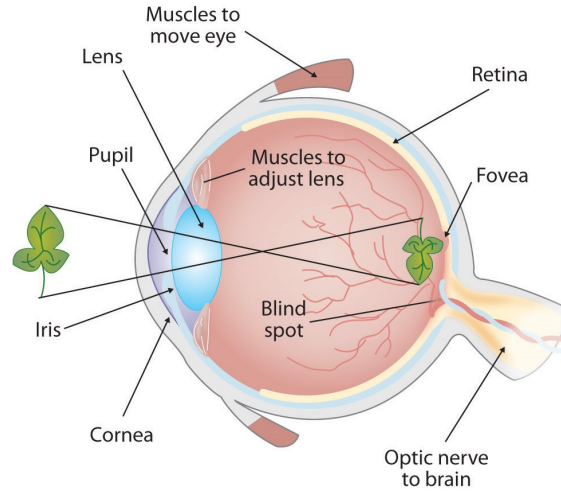


Figure 2.2: Side-view cross-section of an eyeball. Image acquired from Stangor *et al.* [1]

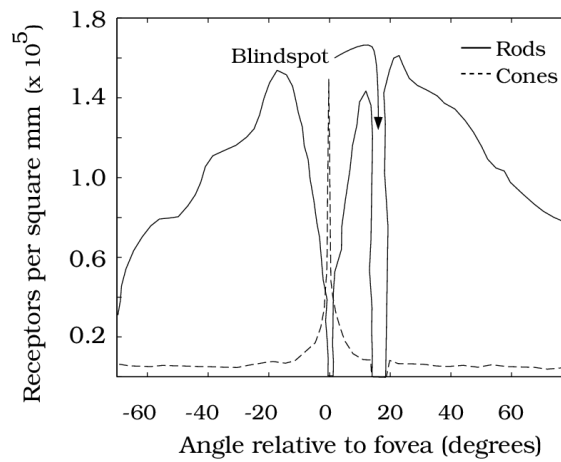


Figure 2.3: Photoreceptor density across the retinal field, measured in degrees from the fovea. Image acquired from Wandell *et al.* [2]

flecting social and behavioral patterns. Eye movement behavior, however, comprises of three fundamental movement patterns - a) fixations b) saccades and c) pursuits [36]. During fixations, eye movements are minimized and the image collected on the retinal wall is stabilized. This is easy to identify as the eyes would appear stable while maintaining their visual gaze on a target or region of interest. On the other hand, saccades are rapid and ballistic movements of the eye as gaze is shifted from one target to the next. Saccadic movements are pre-planned from an internal model of the visual world, also known as *saliency maps* generated in the visual cortex and take up to 150 - 200 ms to begin motion onset [36, 37]. The third form of eye movements, smooth pursuits, are generally regarded as an extension of fixations but entail significantly differing behavior. A smooth pursuit occurs when the eye attempts to follow a small moving target or location and attempts to resolve its image on the fovea. This results in an online, close-loop correction mechanism which relies on retinal image displacement from the fovea to trigger a smooth, corrective eye movement. Pursuit movements are typically interlaced with saccadic movements when the eye fails to keep up with the target [38, 36] moving upto an upper limit of  $100^\circ/s$  [39]. Targets moving at faster velocities are tracked entirely using sequences of predictive saccadic eye movements.

## 2.2 Eye and Head movements

Successful engagement with their surroundings results in frequent head movements in humans. Even the smallest of movements, *i.e.*, the average human step, results in a vertical movement of the head. Since the eye socket is located within the head, it naturally follows that the presence of head movements would significantly reduce retinal image stability unless there exists ocular-motor mechanisms to make compensatory eye movements.

### 2.2.1 Compensatory mechanisms

Generally, we employ two major retinal image stabilizing mechanisms, the vestibular-ocular reflex (VOR) and the opto-kinetic response (OKR). In VOR, the semicircular canals of the inner ear measure head rotation acceleration which results in eye movements in the opposing direction with near unity gain (*i.e.*, the ratio of eye and head velocity is  $\sim 1$ ). OKR is generated by motion on the retinal field which in turn leads to compensatory eye movements to reduce retinal blur. [36, 40, 41, 38] Eye and head movements efficiently employ or modulate either or a combination of these mechanisms to maximize image stability.



### 2.2.2 Vestibular-Ocular Suppression

VOR enables us to stabilize the retinal image of the world but there exist a few conditions wherein the compensatory effects are detrimental towards successful image acquisition. Head movements stimulates VOR, generating compensatory eye movements that are counterproductive to target pursuit. In order to successfully stabilize the retinal image of a moving target, the cumulative head and eye velocity, *i.e.*, the gaze velocity, must match the velocity of the moving target. This can only be achieved by suppressing VOR using an active visual feedback loop [41, 38].

### 2.2.3 Anticipatory head movements

Previous work has shown that anticipatory and predictive head movements are observed during gaze shifts while accomplishing a certain task [42]. Mann *et al.* and Kishita *et al.* have shown that the head *tracks* while the gaze can *predict* a target location in context of an outdoor activity [28, 43]. Such interplay between eye and head movements in naturalistic settings has previously been unstudied for broader contexts. Capturing a broader range of eye and head movements during everyday activities could improve our understanding of a joint ocular-motor control system and the role of prediction within it.

## 2.3 Mobile Eyetracking

To study or exploit human visual behavior requires us to measure eye and head movements. While one can conceive of simple systems to measure head movements, tracking eyes in a non-intrusive manner is a complex science with each unique solution presenting a unique set of challenges and advantages. Generally, there are three types of eyetracking techniques; a) Electro-Oculography, b) Optical tracking techniques and c) Video Oculography.

Electro-Oculography is a technique for measuring the electrical potential subtended across the eye with a pair of electrodes. This technique is sensitive to interference caused due to facial movements and is not accurate as compared to other techniques. It also involves placing each electrode to the left and right or top and bottom of the eye which could hinder naturalistic behavior. However, electro-oculography can measure eye movements with the eyes closed and does not obstruct the field of vision.

Optical tracking involves tracking the first and fourth Purkinje image caused by a projected infrared beam onto the corneal surface [44]. Eyeball orientation is measured as a function of the relative positions of these two images and requires a strict calibration protocol. This system has the ability to measure gaze positions with an accuracy within 1 minute of arc with a sampling rate up to 1000Hz. The downside of this technique is the



Figure 2.4: Commercially available eyetracker, PupilCore. Bottom left illustrates a person wearing the tracker. Bottom right illustrates a magnified image of the eye camera. The components marked are as follows: 1) scene camera (also known as world camera) 2) nose rest 3) IR eye camera 4) IR emitters (which produces bright glints on eye imagery) 5) Eye camera location

rigorous head stabilization required to obtain stable Purkinje images, often with the aid of bite bars and chin rests.

Video-oculography is a non-intrusive eyetracking technique that involves the use imaging systems such as a camera to measure ocular features of interest and correlating them with a known gaze position. Given the lack of precision and accuracy with electro-oculography and lack of mobility in optical techniques, video-oculography is the natural choice for measuring head and eye movements in an unconstrained setting. Video-oculography comprises of *head mounted* eyetracking, wherein the imaging system is placed on a wearer's head with an optical system collecting the image of their eyes, and *remote* eyetracking, wherein the imaging system is placed away from the body and requires the subject facing the imaging system. Since my aim is to measure eye and head movements while a subject is free to navigate and interact with their surrounding, this thesis will focus on head-mounted eyetracking solutions and techniques.

There exists a rich and diverse set of solutions for the problem of head-mounted eyetracking. The general approach involves the use of one or multiple infrared light sources placed next to an infrared eye camera which, using appropriate hardware design, points towards the wearers eye (see Figure 2.4). A third camera, referred to as the scene camera, points away from the wearer and into the scene. Depending on the algorithmic complexity, latency limitations and computational power, various solutions estimate gaze

descriptive features. The general steps to produce a gaze estimate from such an assembly is as follows: a) capture image(s) of the eye b) pre-process the image(s) c) apply image processing/computer vision to identify features of interest d) correlate extracted features to a measure of gaze (see Figure 2.6). While the use of a head-mounted tracking system enables unrestricted head movements, eyetracker slippage on the head degrades the accuracy of gaze estimates.

### 2.3.1 Feature based approaches

Various features in an eye image can be exploited to generate gaze estimates. Some of the earliest works on head-mounted video oculography rely on extracting the pupil center relative to the reflection center of IR light sources (also known as *glints*) from an image of the eye [45]. This approach, known as PCCR (pupil center corneal reflection) has been incorporated in a number of commercially available head mounted eyetrackers. The advantage of tracking glint location is that it directly represents the position of the eyetracker with respect to the eyeball. This renders the relative pupil center position invariant to unwanted translatory effects on the eye camera such as observed during eyetracker slippage. In the absence of glints, tracking eye corners may provide similar robustness [46].

While it is relatively trivial to estimate glint locations, estimating pupil center can be quite challenging. The general approach for identifying pupil center involves segmenting the pupil using intensity thresholds, pupil edge detection, and ellipse fitting across identified valid edges. Wang *et al.* segment the pupil using Otsu's threshold scheme [47], followed by ellipse fitting on detected pupil edges [48]. Swirski *et al.* compute an initial estimate of pupil center using Haar-like features on integral eye images, followed by k-means segmentation of the pupil using a local histogram around this estimate [49]. Santini *et al.* compute pupil edges using Canny edge detection followed by discarding edges which do not obey certain heuristics [30]. The retained edges are used to fit an ellipse and provide a confidence measure for the given eye image. Fuhl *et al.* trained a random ferns model which estimates the probability of each pixel being the pupil center. In order to train a robust classifier, they train their model with equal proportions of negative samples near the groundtruth pupil center [50]. These efforts are some of the many algorithms developed for extracting pupil center.

The movement of stable visual features, while directly correlated with gaze estimates, require a calibration routine to associate them with a known measure of gaze. Ground truth gaze information is generated by explicitly asking the wearer to fixate on predetermined points of interest. The corresponding pixel location of a point of interest on the scene camera is used to generate a polynomial mapping between the groundtruth gaze pixel in scene camera coordinates (also known as Point-of-Regard (PoR)) and the extracted gaze feature.

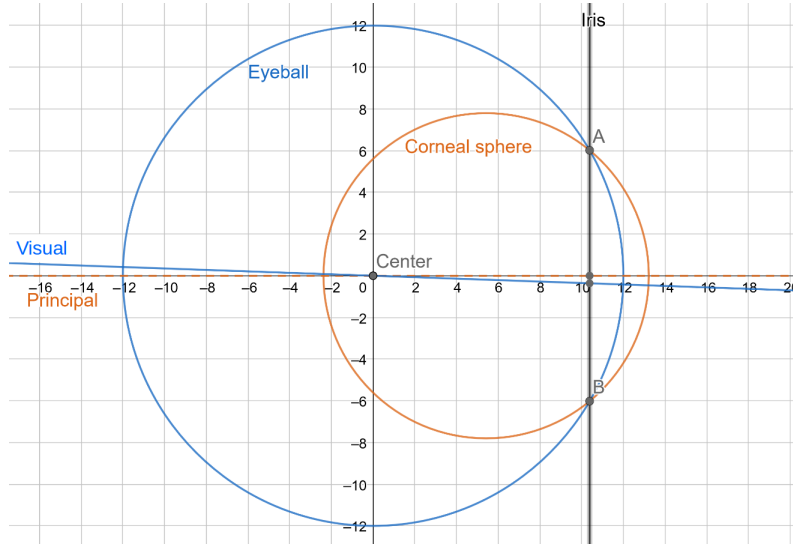


Figure 2.5: Top-down view of the Reduced Le Grand left eye model [3]. Each grid unit represents  $2mm$ .

### 2.3.2 Model based approaches

In recent times, the eyetracking community has witnessed excellent efforts which involve fitting an approximate 3D eyeball based on 2D features extracted from eye images, typically collected over a series of images. Estimating the precise physiology of the human eye is a complicated process and computationally intractable. By making certain simplifying assumptions about the human eye, geometrical constraints enable us to estimate a *reduced* optical eyeball model [51]. Once an estimate of the reduced eyeball position and orientation is obtained, the optical axis is obtained by tracing a vector from the 3D eyeball center to the 3D iris (or pupil) center. Note that fixating on a target involves placing the *foveal* axis on the target of interest (*i.e.*, the line joining the fovea and 3D eyeball center) which, in this context, is offset from the optical axis by a varied, subject specific, amount (see Figure 2.5). This offset between the optical and foveal axis is referred to as *Kappa* (Angle  $\kappa$ ) [52]. However, location of the fovea varies across people. All model based approaches focus on estimating the optical axis which lies approximately  $3 - 5^\circ$  away from the visual axis [53] which is the true representation of gaze. A single point calibration is required to estimate the rotation between optical and visual axis.

Swirski *et al.* developed a solution wherein the approximate center of eye rotation was estimated based on back-projected ellipse fits [49]. They adopted the “two circle”

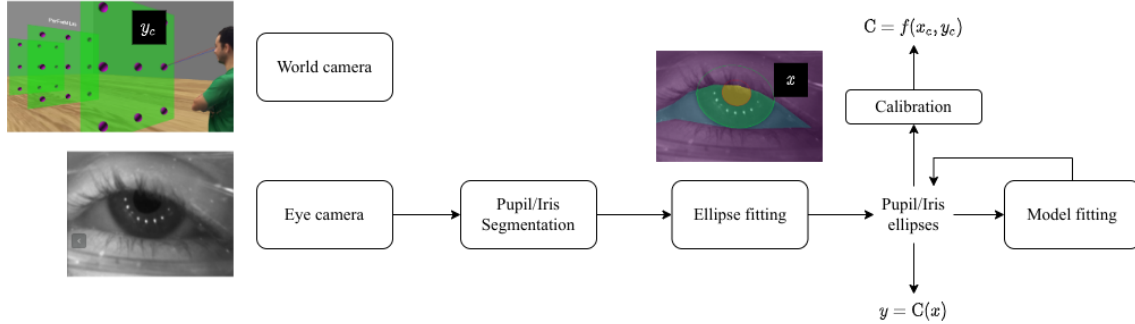


Figure 2.6: Holistic pipeline for a mobile eyetracker. Eye and scene images with their timestamps are acquired from near-eye IR cameras and a scene camera. Computer Vision or Machine Learning techniques are employed to segment the eye image into parts of interest such as the pupil (shown in yellow overlay) and iris (shown in green overlay). This is followed by fitting an ellipse on the pupil and/or the iris segments,  $x$ . Model based approaches identify the center of eyeball rotation over multiple frames and improve the ellipse estimates. During calibration, subjects are asked to fixate known calibration targets  $y_c$  in the scene (example calibration pattern taken from Binaee *et al.* [4]). Finally, known targets and extracted ellipse estimates are used to estimate a mapping function  $C = f(x_c, y_c)$ . For all incoming eye images, gaze is estimated using  $y_g = C(x)$

approach [54] which involves geometrically fitting a cone model from the camera center to the detected pupil ellipse points. It can be shown mathematically that only two 3D circles are possible which project onto a particular elliptical shape [49, 54]. Since the pupil aperture may change over time, this solution requires collecting a small number of images over a very short duration ( $\sim 100$  frames on a 120Hz eye camera) to generate a confident and accurate eyeball rotation center while assuming the pupil size remains unchanged.

A limitation of the Swirski model is that it does not compensate for refraction induced at the corneal surface. This limitation can be overcome in multiple ways: a) employing a polynomial corrective function on the estimated eyeball center [55], b) developing an iterative algorithm using ray tracing from all possible eye ball locations [56], c) estimating eyeball position using iridial edges, which do not undergo refraction if captured from appropriate eye camera position [57], or d) estimating eyeball position using multiple glints constrained to known locations [58]. Each unique solution presents a novel set of challenges and a single, refraction aware closed-loop solution is an open and active area of research.

### 2.3.3 Limitations

Head mounted eyetrackers generally employ infrared (IR) imaging sensors pointed towards a person's eyes (see Figure 2.4). To obtain a clear eye image, infrared lighting emitting diodes (LEDs) are strategically placed near the eye cameras. Since eyetrackers were initially developed for indoor and laboratory environments, this convenient design choice ensured that IR eye image exposure can be controlled by digital IR emitters and remain independent to ambient lighting and reflections in the visible domain. This observation comes from the fact that most indoor lighting have a spectral response predominantly present in the visible domain (350 to 700nm). Outdoor environments, on the other hand, degrade the quality of eye images by introducing unwanted reflections and glare from surrounding objects (see Figure 2.7). This is primarily due to the large infrared spectral power inherent to sunlight.

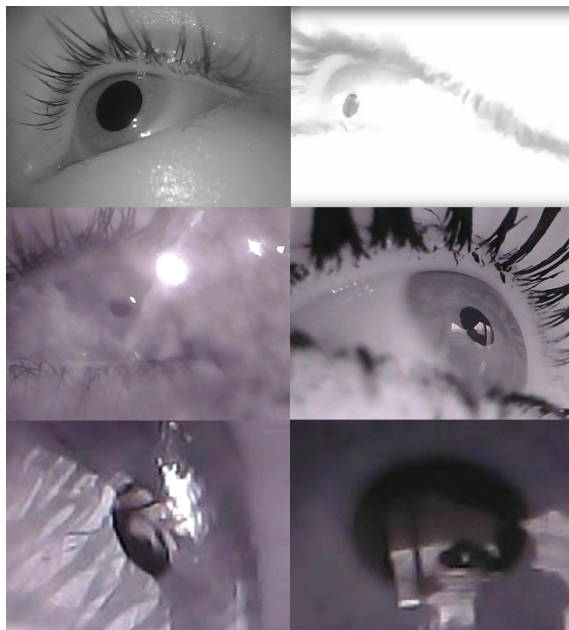


Figure 2.7: A collection of eye images acquired with varying degrees of degradation. Top row images are extracted from the GW dataset [5]. Second and third rows contain hand selected images from the Labelled Pupils in the Wild dataset (LPW) [6].

Occlusion of eye features is another major reason for the degradation of gaze signals and occur due to a variety of reasons such as eye make-up, corrective optics, partially closed eyelids and eyelashes. Handcrafted feature extractors cannot account for complex

scenarios which are prevalent in applications of eyetracking.

### 2.3.4 Machine Learning approaches

Convolutional Neural Networks (CNNs) in context of eyetracking have demonstrated resilience towards artifacts induced due to stray lighting and unwanted environmental reflections [59, 60, 31, 61, 62, 63, 64]. This resilience is acquired by an extensive training routine wherein parameters inherent to the neural network are constantly updated and tuned until network converges upon an ideal combination of weights best suited for a task. Parameters are updated based on error values derived from network predictions and annotated groundtruth for a given set of input eye images. This provides us with a unique advantage to *teach* a neural network how to extract relevant gaze features in the presence of various irregularity and image degrading artifacts. For a quick introduction to deep convolutional neural networks, I recommend the excellent work by Krizhevsky *et al.* [65]. In this section, I will cover machine learning concepts relevant to eyetracking, summarize previous efforts upon which I draw inspiration for my proposed work and provide an exhaustive list of relevant datasets beneficial for my research. Machine learning approaches for head mounted eyetracking can broadly be divided into three approaches:

#### Pupil center identification

Pupil center is one of the most common gaze feature extracted from eye imagery. The general approach towards pupil center regression involves multiple cascaded convolutional layers followed by pooling operations to regress learned features onto pupil center pixel position [66, 67]. Kim *et al.* showed that increasing the number of convolution layers and input image resolution results in improved gaze estimation with diminishing results [66]. Fuhl *et al.* proposed adding a refinement network to improve predicted pupil center position [59]. Vera-Olmos *et al.* modified this cascaded structure by adding dilated convolution layers and showed improved performance [61] over hand-crafted counter alternatives such as ElSe [68].

While the pupil center is a valuable asset to determine gaze, its position is sensitive to eyetracker slippage which reduces its reliability and requires repeated calibration procedures. Small errors in pixel estimate could result in large deviations from the actual gaze estimate. Recent work by Wu *et al.* has shown that convolutional networks can also be employed to estimate the glint positions as well [69]. This enables us to leverage the classic pupil-corneal reflection pipeline to estimate gaze (see Section 2.3.1).

### Eye parts segmentation

While estimating a PCCR signal provides some robustness to headset slippage, there are numerous limitations for the purpose of gaze estimation. Most notably, PCCR does not allow us to measure the optical axis, a crucial ingredient required to estimate the visual axis, also identified as 3D gaze vector or the line of sight. To overcome this issue, recent efforts have proposed turning gaze estimation into a semantic segmentation problem [31, 9, 63, 8]. Semantic segmentation is the process of labelling each pixel of an image into a predefined set of categories. Boundaries between segmented regions are utilized to reconstruct an approximate 3D model of the eyeball (see Section 2.3.2). The OpenEDS [8] challenge, a pioneering effort by Facebook Reality Labs, proposed segmenting an eye image into its individual constituent parts. The proposed individual parts for semantic segmentation are the pupil, iris, sclera and *background* (this category encompasses all unwanted regions such as facial hair, skin and non-body regions). Segmented pixels allows us to analyze each individual category and enable applications in biometrics, iris based feature matching and tracking [70], emotion recognition [69] and blink estimation [8, 31]. To illustrate the behavior of a segmentation network, I refer the reader to Figure 5.1 which highlights the various eye parts proposed in OpenEDS.

Numerous architectures exist for semantic segmentation [71] amongst which the encoder-decoder framework (see Figure 2.8) is the most relevant design strategy which derives a compact representation of the input data via which the decoder reconstructs a pixel to pixel mapping to semantic categories such as the pupil, iris or sclera. A notable architecture within the encoder-decoder framework is U-net [7], which has demonstrated state-of-the-art segmentation performance on various applications <sup>1</sup>. The symmetrical and minimalist design of this network has inspired numerous network architectures [9, 72, 31, 73] that enable us to explicitly manipulate intermediate representations [74, 72] with significantly lower computational demands [9], which makes it ideal for applications such as head mounted eyetracking.

### Multiple objective functions

Neural networks are generally trained using a loss function, which is an analytic representation of the error between groundtruth (typically provided by tedious human intervention or annotation) and network prediction (the output of a network). A well trained (or converged) network may excel in minimizing the particular loss function it was trained on. However, a single loss function may not ensure optimal network performance. A loss function designed to penalize a neural network for every incorrect prediction for each pixel is

---

<sup>1</sup>[lmb.informatik.uni-freiburg.de/people/ronneber/u-net](http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net)



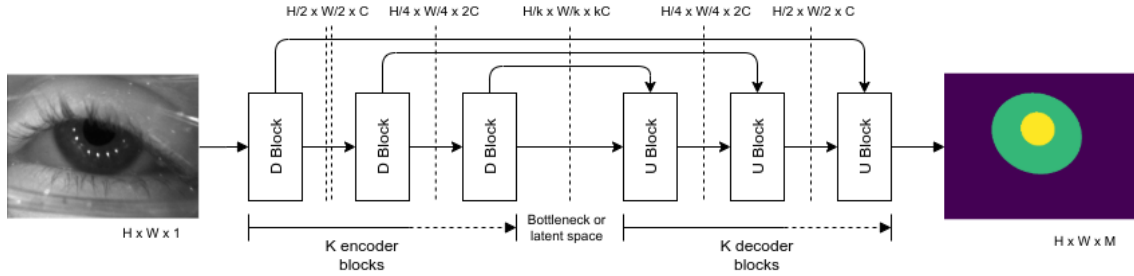


Figure 2.8: General framework for an encoder-decoder architecture. The architecture comprises of blocks which are a collection of convolutional layers. An input eye image with a resolution of  $W \times H$  is passed through a series of  $k$  convolutional  $d$ -blocks or down-blocks which down-sample the intermediate output to a lower resolution while increasing the features extracted. The intermediate representation of the image following the encoder  $d$ -blocks is passed through a series of up-sampling convolutional  $u$ -blocks (or up-blocks) to produce a fine grained semantic segmentation output of the image. This bottleneck or latent representation of an image, is a low spatial resolution matrix with a large feature space. Typically, outputs from encoder blocks are also fed directly into decoder blocks. For more information on encoder-decoder networks, I refer the reader to Ronneberger *et al.* [7]

agnostic to the global structure and semantics relevant to the problem. In order to minimize the overall loss, the neural network may misclassify arbitrary pixels without adhering to the context of a problem (see Figure 2.9). For example, misclassification of edge pixels can be detrimental to subsequent processing which rely on accurate semantic boundaries. Providing alternate loss functions which penalizes a neural network for structural misclassifications such as segments not adhering to an elliptical shape or a loss which ensures the density of a semantic category remains consistent, provides context to the problem and improves network performance [75].

Some loss functions operate in synergy and can effectively utilize shared parameters while aiding convergence. Certain loss functions are detrimental towards each other and require smart modifications to network topology which can capture a wider range of parameters required to satisfy divergent loss functions [76]. Efforts by Wu *et al.* and Park *et al.* have shown that combining different loss functions can capture better gaze features from head mounted eyetrackers [69] and remote webcams [77]. RITNet [9] and EllSeg [10] draw inspiration from these works to develop a unique combination loss functions designed to attain state-of-the-art performance on multiple datasets and across various metrics of performance.

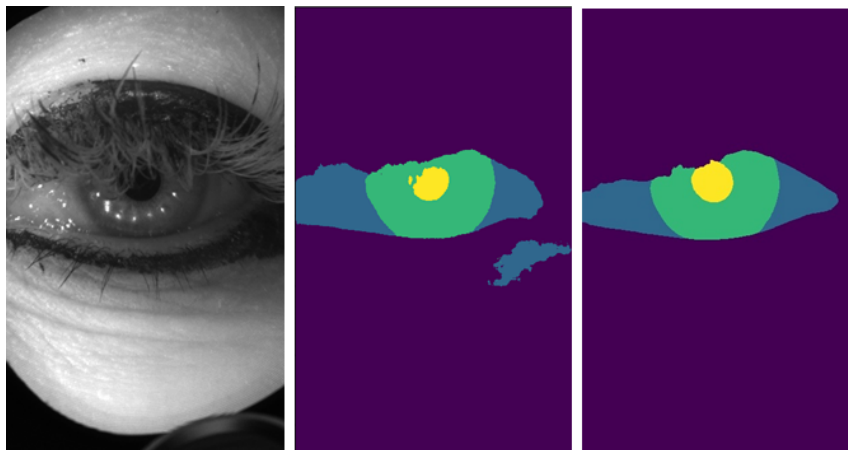


Figure 2.9: The effect of training a network with multiple loss functions for semantic segmentation. Left: eye image acquired from the OpenEDS [8] dataset. Middle: segmentation output without structural awareness. Right: Segmentation output with structural awareness. Segmentation results are provided by RITnet [9]. Note the stray misclassifications which are effectively removed when a network is trained with two loss functions.

### Domain Generalization

Convolutional neural networks effectively learn an ideal set of parameters (also known as network weights and biases) to capture pixel level statistical relationships inherent within the training data while optimizing its performance on multiple (or single) objective functions. A converged network, *i.e.*, a network which has undergone multiple update cycles beyond which we observe no change in its parameters, is guaranteed to perform reasonably well when tested on data which exhibit statistical properties similar to the training set [78].

Real-time applications during everyday activities often exhibit eye images with distortions, stray reflections and unwanted occlusions which is often not represented during training. While we may attempt to reduce external influences by applying an infrared blocking filter in front of the eyetracker [79], we cannot account for physiological differences between subjects such as varying IR skin reflectance, iris pigmentation, facial structure and occluding objects such as eye make up, eyelashes and corrective optics.

Thus, we identify that an ideal solution must a) generalize across unseen environmental factors and at the very least, generalize across subjects within similar environments and b) should the solution fail, it should do so gracefully (see Figure 5.5). In order to make these claims, one must rigorously test this solution against a large population of subjects and

environmental conditions with known groundtruth labels or targets. While not impossible, it certainly is a difficult and time consuming task. In Chapter 7, I discuss the topic of generalization and acquire a broad distribution of eye images by leveraging multiple datasets of eye images with varying environmental conditions, diverse populace of subjects and occluding artifacts. Please see Table 2.1 for a comprehensive list of publicly available datasets utilized in developing ML models for this thesis.

| Dataset                   | Resolution | Images   | GT | Eyetracker | Quality    |
|---------------------------|------------|----------|----|------------|------------|
| Swirski [49]              | 640x480    | 3,760    | PE | Logitech   | Controlled |
| PupilNet [59]             | 384x288    | 41,217   | PC | Dikablis   | Natural    |
| Fuhl <i>et al.</i> [32]   | 384x288    | 94,713   | PC | Dikablis   | Natural    |
| LPW [6]                   | 640x480    | 1,30,856 | PC | Pupil Labs | Natural    |
| OpenEDS <sup>19</sup> [8] | 400x640    | 12,759   | S  | Custom     | Controlled |
| S-Natural [80]            | 640x480    | 51,200   | S  | Synthetic  | Natural    |
| NVGaze [66]               | 1280x960   | 60,000   | S  | Synthetic  | Controlled |
| S-General [80]            | 640x480    | 51,200   | S  | Synthetic  | Controlled |
| UnityEyes [81]            | 640x480    | 50,000   | IC | Synthetic  | Natural    |

Table 2.1: An extensive list of near eye image datasets applicable for my thesis. The Ground Truth (GT) column identifies the type of annotated modality available for each dataset. S→Annotated segments, E→ellipse parameters, C→ellipse center. Note that  $C \subset (E, S)$  and  $E \subset S$ . P and I correspond to pupil and iris annotations respectively.

## 2.4 Event detection

Gaze classification is the process of segmenting a series of eye movements or eye and head movements (when the head is free) into meaningful clusters (or *events* as will be called for the rest of this thesis) of annotated data. Classified gaze sequences enable researchers to isolate events of interest and study their intrinsic properties in presence of known stimuli, tasks or their relationship to one another. Previously, gaze classification was accomplished by manually (and painstakingly) hand coding temporal sequences. Careful considerations are usually required before manually labelled these sequences and rates up to 60 seconds of hand labels per hour of labelling time were not uncommon. In order to reduce human effort and improve human efficiency, the need for automated classification algorithms was identified.

This thesis also builds upon a long history of methodologies for the automated detection of gaze events within an eye tracking signal. The simplest methods use threshold based filters and numerous descriptive features for classification [82]. Threshold based techniques require parameter tuning for each test scenario as well as being sensitive to noise and sample rate. A better solution is to use machine learning to learn a model for classifying gaze events. These algorithms have been based on a variety of machine learning algorithms and have been shown to work well when the head is fixed. Random Forests work well with high sampling rates (1000  $Hz$ ) [83]. The Naive Segmented Linear Regression (NSLR) model [84] uses the Pruned Exact Linear Time (PELT) method [85] to segment a time sequence into distinguishable segments which are then later classified using Continuous Hidden Markov Models (HMM) into events. While earlier work used hand-crafted features [83], the most recent methods have employed recurrent neural networks [12]. This approach enables algorithms to directly learn what features are relevant to the task and tend to work better than methods that rely on hand-crafted features when the amount of labeled data is fairly large. For both methods based on hand-craft features and deep learning, the algorithms have only been designed to operate when the head is fixed, and significant head movements will cause them to fail. New gaze classification algorithms are needed for datasets that incorporate both head and eye movements.

## Chapter 3

# Gaze in Wild<sup>1</sup>

Human visual behavior can be viewed as a sequence of periods of stable visual input, punctuated by saccades to new locations within the visual environment. Although saccadic targeting during visual search may demonstrate an influence of visual salience, [86] the effect is overwhelmed in the presence of a task, when motor execution requires that attention be directed towards information-rich, task-relevant locations within the visual environment. [14, 17, 18] As a result, the dynamics of gaze coordination in natural contexts are affected by a variety of extra-retinal properties of the task, the agent, the environment, and by their interaction. These include the spatial distribution of information in the natural environment, [21] cognitive resources related to memory or higher order reasoning, [24] motor constraints that determine the dynamics of gaze shifts, [25, 41, 26, 27] and biomechanical constraints that influence visual strategies for foot placement during locomotion. [22]

Despite the importance of extra-retinal influences upon gaze behavior during visually guided action, surprisingly little attention has been dedicated to the study of gaze behavior in more natural contexts. For instance, head movements are often suppressed through the use of a chin-rest, or by constraining target movement to only a small portion of the subject’s visual field. Furthermore, target motion is often restricted to two dimensions, and sometimes viewed monocularly. In part, the study of strategies for coordination of the eyes, head, and body has been limited by a lack of suitable technology. Successful tracking of the coordination between the head and eyes in unconstrained settings requires advances in two parallel domains: the instrumentation to jointly monitor the direction of the eyes and head (“eye + head tracking”), and the algorithms to parse and categorize key oculomotor events in the rapid stream of data (i.e. “event detectors”).

This thesis builds upon a variety of techniques previously used to track head orienta-

---

<sup>1</sup>This chapter appears in a published manuscript by Kothari *et al.* [5]

tion during natural behavior. Published studies have demonstrated the use of rotational potentiometers and accelerometers [41], magnetic coils [87], or motion capture [88, 24] for the sensing of head orientation. Perhaps the highest precision eye+head tracker which allowed body movement leveraged a  $5.8\text{ m}^3$  custom-made armature capable of generating a pulsing magnetic field. The subject was outfitted with a head-worn receiver capable of measuring head position and orientation within a  $1.8\text{ m}$  volume within its operational region [89]. Several systems have adopted video based head motion compensation [90, 27] and demonstrated promising results, but are too computationally expensive for real-time use, and are often subject to irrecoverable track loss following brief loss of computationally tractable regions. More recent approaches have involved the use of a head-mounted inertial measurement unit (IMU). In [91], subjects were asked to perform visual tracking tasks when watching pre-rendered stimuli projected onto a 2D screen. Most recently, Tomasi et al. used two IMU for tracking eye and head orientation relative to heading direction [92].

This work also builds upon a long history of methodologies for the automated detection of gaze events within an eye tracking signal. The simplest methods use threshold based filters and numerous descriptive features for classification. [93] Threshold based techniques require parameter tuning for each test scenario as well as being sensitive to noise and sample rate. A better solution is to use machine learning to learn a model for classifying gaze events. These algorithms have been shown to work well when the head is fixed. Pekkanen *et al.* proposed the Naive Segmented Linear Regression (NSLR) model [84] which segments a time sequence into distinguishable events which are then classified using continuous Hidden Markov Models (HMM). While earlier work used hand-crafted features, [83] more recent methods have employed recurrent neural networks (RNN) [12] which enable algorithms to directly learn what features are relevant to the task.

### 3.1 Head-free gaze movement nomenclature

Gaze classification requires distinct and separable classes that are identifiable in our daily activities. While it is *relatively* trivial to identify basic eye movements such as fixations, saccades and pursuits, it becomes difficult to identify head-free gaze movement classes. For instance, consider a situation where a person keeps fixating at a target in front of them while moving their head side to side. From a purely eye movement perspective, it would be considered a smooth pursuit of the target. However, from a head-free context, it would be considered as a fixation.

There has been some disagreement in the research community about the specific criteria for establishing a taxonomy of gaze events.[94, 95] For example, one approach is to classify events based upon specific oculomotor movements, such as the two major retinal image stabilizing mechanisms: the vestibular-ocular response (VOR), and the opto-kinetic

response (OKR). In VOR, the semicircular canals of the inner ear measure head rotation acceleration which results in eye movements in the opposite direction with near unity gain (*i.e.*, the ratio of eye and head velocity is  $\sim 1$ , see Figure 3.1). OKR is generated by retinal motion which in turn leads to compensatory eye movements to reduce retinal blur. [36, 40, 41, 38]. It is difficult to derive a classification scheme based solely on these stabilizing mechanisms, because they may be used in isolation, or in combination, for either fixation of a target that is stationary in the exocentric frame, or pursuit of a moving target.

Our approach is to adopt an exocentric classification scheme and to discuss its applicability in classifying a broad range of coordinated head and eye movements. We define movement categories by the functional role of the eye movement, as well as the motion of an object within an exocentric frame of reference. As a result, events in our dataset is classified as follows:

1. **Gaze fixation (GF)** - Gaze fixation may be brought about through stabilization of the eyes and head, or during movements of the eyes and head that are compensatory and, as a result, produce a stable gaze vector on a stationary object in the world coordinate frame. Stabilized retinal image motion lies near to the range of 0.5 to 5  $^{\circ}/s$ , a limit above which the target image starts to blur. [36] Hence, a wide range of miniature head compensated eye movements can be termed as gaze fixation. In our taxonomy, gaze fixations may be further categorized as:
  - **Tremors** - The resting eye and head rarely display perfect stability. Skavenski *et al.* identified that despite instructing subjects to remain as stationary as possible, tremor was observed in the head and eyes ( $< 1^{\circ}/s$ ,  $10Hz$ ). [96] Furthermore, the characteristics of tremor is known to vary based on the nature of the instrumentation [97] and type of restraint. [96]
  - **Drift** - Drifts are slow motions of the eye that are often punctuated by microsaccades and aid in maintaining crisp visual features across the retina. While there is some disagreement on the range of drift motion, they usually display amplitudes within  $0.25^{\circ}$  and velocities less than  $0.5^{\circ}/s$  when the head is fixed. [97]
  - **Microsaccades** - Small, rapid eye movements that occur in between fixations are termed as microsaccades and usually last about  $25ms$  with a velocity range capped at  $50^{\circ}/s$ . [97]
  - **Fixation by rotational vestibular-ocular reflex (rVOR)** - When the subject and target are stationary in the world reference frame, rotational motion of the head is compensated using rVOR. Fixations are maintained by the VOR system because it has a significantly lower response lag as compared to OKR. [36]

Generally, a rVOR event displays near unity gain unless it is modulated due to other compensatory mechanisms such as OKR or pursuit.

- **Fixation by translational vestibular-ocular reflex (tVOR)** - When a target is stationary in the world reference frame, image stability at the fovea during self motion or passive displacements is achieved by tVOR [98]. Unlike rVOR, wherein a counter rotation of the eye in head rotation can stabilize the entire retinal image, tVOR cannot accommodate for the entire visual field due to the large range of optic-flow motion experienced at different depth planes. Primarily a foveal image adjustment mechanism, it follows that properties of tVOR motion depend on the gaze direction and can be difficult to differentiate with pursuit movements. [99] OKR augments VOR to help maintain a stable image over stationary targets. Fixations are maintained by a combination of gain modulation and optokinetic stimulation. [38, 36, 98, 100] While microsaccades may be triggered for retinal image adjustment, larger saccades during fixations signify shifts in attention or an inability of gain adjustment to compensate for motion such as observed during nystagmus. These visually driven eye movements work in synergy with tVOR [99] making them difficult to observe in everyday activities as opposed to controlled experiments which are designed to isolate their behavior.
2. **Gaze pursuit (GP)** - Also known as smooth pursuit movements, [101] gaze pursuit is the visual tracking of an object that is moving through the world frame using the eyes or a combination of the eyes and head by augmenting over our compensatory systems. [36] Gaze pursuit is often interrupted by catch-up saccades in compensation of retinal error. [25] While it is somewhat trivial to identify GP events using visual imagery, it may become difficult to differentiate them with GF (for more information refer to Supplementary Figure 1).
  3. **Gaze shift (S)** - A rapid shift of gaze to a new location in the world (i.e. a saccade) using the eye or eye and head in combination.

To illustrate our nomenclature, consider a situation where a person under fore-aft motion attempts to pursue a moving target. In situations such as these, the effects of stepping are compensated using VOR in the elevation direction. Relative distance and gaze angle modulates the tVOR to maintain target image at the fovea. The moving target's retinal image motion elicits a pursuit signal punctuated by predicative saccades. The pursuit motion augments over translational VOR by modulating its gain. If the eye and head pursuit movement can be distinctly identified in their velocity traces, we would consider such an event as a gaze pursuit. However, a distant or slow moving target



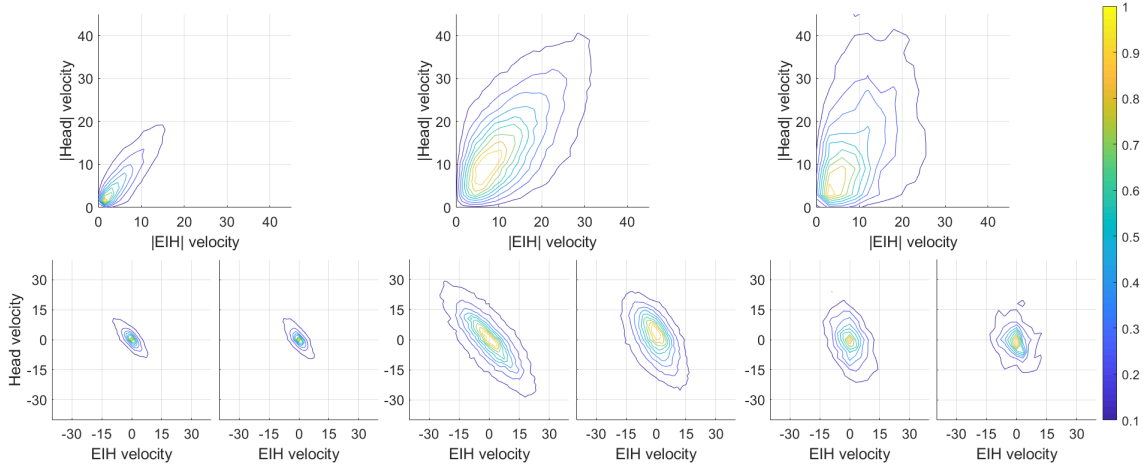


Figure 3.1: Eye and head movement statistics. The top row signifies absolute eye and head velocity. The bottom row signifies the distribution of eye and head velocity in the azimuth and elevation direction. The left column illustrate fixations when subjects were stationary. The middle column illustrates fixations when subjects were in translatory motion. The right column illustrates pursuit behavior. The scale on the right shows the normalized concentration of samples and is used in all figures.

may induce a small pursuit signal which may not be easily identifiable over opto-kinetic stabilization of the retinal image. In these situations, we would consider the event as a gaze fixation. Note that the exocentric nomenclature enables us to define multiple concurrent coordinate systems and thus requires that we specify the reference system under analysis.

## 3.2 Methodology

The aim of this work is to generate a dataset that captures complex ocular-motor strategies during natural tasks (see Figure 3.2). We recruited 19 participants (7 female, age  $\mu=28$ ,  $\sigma=12.52$ ). Informed consent was obtained from all participants prior to hardware setup to anonymously share all data recorded from them. Identifiable people in this manuscript consent to publicly share their information as presented. All methods were carried out in accordance with relevant guidelines and regulations as approved by the Institutional Review Board at Rochester Institute of Technology, FWA-00000731. Participants were tasked with performing up to four activities while wearing an eye tracker, a hardhat instrumented with sensors, and a backpack with a laptop computer (see Figure 3.3). Since task demands and interpretation have been known to guide eye movements, [14] care was



Figure 3.2: Task selections in the GW dataset. Left to right  $\rightarrow$  Indoor navigation, ball catching, visual search and tea making.

taken to ensure all participants received a standard set of instruction read aloud by the experimenter. Subjects were instructed to stand 1 to 2 meters away from a calibration chart within a predefined rectangular area. Once a participant was within the calibration region and facing the chart, they performed two calibration routines. After calibration was complete, participants proceeded to complete the given task. Table 1 in the Supplementary lists the calibration accuracy, tasks recorded, and the labelling status of each observer. Tasks were selected to create a wide range of head and eye poses as seen in Figure 3.4. Upon completion of a task, participants returned to the calibration area to prepare for the next task. The following tasks were chosen:

- **Indoor navigation:** Subjects were instructed to walk around an indoor corridor loop twice. Indoor navigation was chosen to elicit coordinated eye and head movements that occur naturally during walking. We observed various gaze shifts to objects such as text on posters, signboards, people walking by etc. As expected, subjects made very few to no gaze shifts towards the ground due to lack of terrain complexity [22] and very little attention demands [24] for foot placement accuracy [79]. While some of the subjects were familiar with the indoor corridor layout, we did not observe any noticeable difference in their behavior compared to subjects unfamiliar with the environment.
- **Ball catching:** The purpose of this task was to induce gaze pursuit behavior by asking participants to play catch with the experimenter. The experimenter would change throwing strategies in the middle of the task by either bouncing the ball on the floor, passing the ball to another experimenter or rolling the ball on the ground towards the participant. The subjects tracked the ball as a series of gaze fixations and predictive catch-up/look-ahead saccades and occasionally pursued the ball during a specific period of the ball trajectory.
- **Object search without prior subject-object interaction:** Subjects were tasked to locate and count as many objects with geometrical shapes (such as triangles, rectangles etc.) as they could find in a predetermined closed circuit corridor. This task was chosen to elicit visual search behavior in a head-free setting without biasing a subject with a particular object or shape.
- **Tea making:** As a validation for the classic tea making paradigm, [102] we instructed subjects to go to the kitchen and make themselves a cup of tea. For this task, due to the close proximity of objects, relevant information sometimes fell outside the field of view.

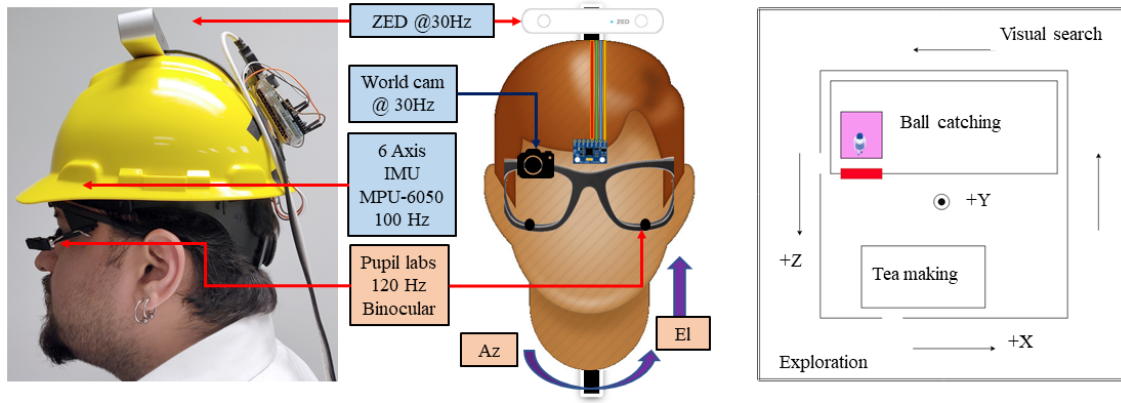


Figure 3.3: (left) side-view, (middle) front view of hardware setup. (right) Top view of all trajectories within our world coordinate system. The red box indicates the position of the calibration pattern. The purple box signifies the region where subject stood during calibration.

### 3.3 Hardware setup and error categorization

To collect naturalistic data, we instrumented participants with an MPU-6050 6-axis Inertial Measurement Unit (IMU) mounted under a hardhat, an ATmega Arduino attached behind the hardhat, a 120Hz binocular Pupil Labs eye tracking glasses (ETG) [103] and a ZED stereo camera (see Figure 3.3). To ensure its applicability in a wide variety of domains, the Gaze-in-Wild dataset provides easy access to depth of the real world stimulus calibrated from the person's FoV. Contrary to a two IMU system, [92] we chose a single IMU system to avoid using a body worn device since many applications of eye tracking are predominately head-mounted. The hardware setup weighed 700 gms (excluding laptop weight), which is similar to previous setups. [41] To reduce slippage, the hardhat was equipped with an adjustable knob to tighten its hold on a subject's head.

#### 3.3.1 Pupil labs eye tracking glasses (ETG)

Binocular eye trackers usually contain two eye cameras and a single world camera (which captures the scene in front of a person). Eye tracking solutions require some form of eye feature (derived from images captured from the eye camera) to Point of Regard (PoR - pixel position on the world camera) mapping to provide an accurate gaze estimate. This process is also known as eye tracker calibration. Mapping functions often vary from polynomial regression to multi-layer perceptron regression. Despite calibration, angular

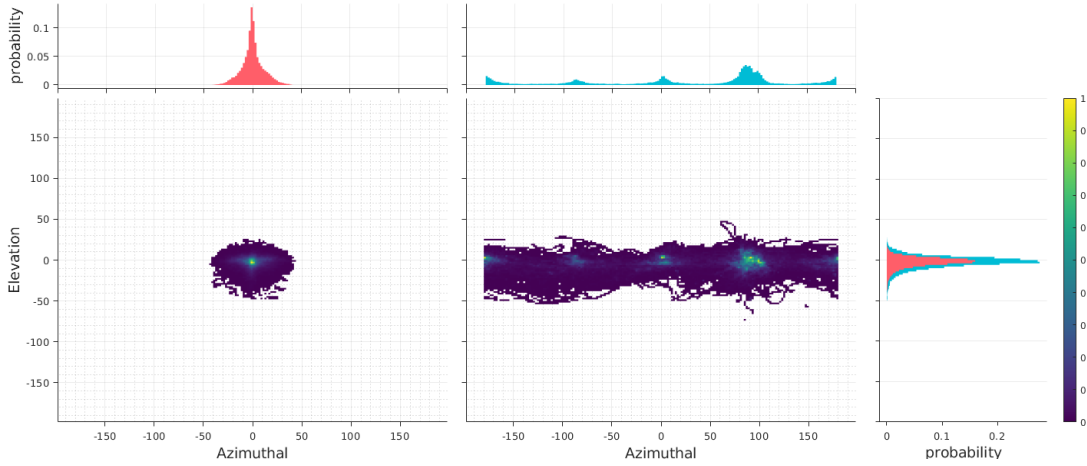


Figure 3.4: Head pose (right) and cyclopean eye distribution (left) in the azimuthal and elevation direction. The cyclopean eye distribution is reported in the Eye-in-Head coordinate system. Angles are provided in degrees. Note that head distribution peaks occur at  $90^\circ$  intervals.

error tends to remain low near to the calibration region and increases radially outwards. Furthermore, there exist many sources of error which degrade the quality of gaze tracking, [103] particularly in unrestrained settings. The Pupil Labs eye tracker estimates the approximate center of eye ball rotation and a 3D pose of the pupil (modeled as a 3D disc). This enables the extraction of 3D gaze vectors with respect to the Eye-In-Head (EiH) coordinate system  $C_E$ . The Pupil Labs software (version number 1.8.26) also provides a confidence value for each gaze sample which can be interpreted as a reliability measure. It is calculated as a ratio of the number of support pixels to the number of pixels on the ellipse fit of an imaged pupil. Support pixels are the edge points within a threshold distance away from the pupil ellipse fit. All gaze samples with confidence below 0.3 were discarded from analysis.

### 3.3.2 Inertial Measurement Unit (IMU)

The MPU-6050 is a low cost 6-axis IMU that integrates a 3-axis accelerometer and a 3-axis gyroscope to estimate its pose relative to its initial position at the onset of data acquisition. The IMU is connected to an Arduino placed behind the hardhat, which in turn, is connected to the laptop backpack. The Digital Motion Processor inside the IMU provides its pose estimate at  $100Hz$ . The I<sup>2</sup>Cdevlib open source library was used to extract information from the IMU. [104] Pose estimates using an IMU sensor are known

to drift due to error accumulation making it necessary to offset the IMU regularly to avoid drift in orientation measurements. Calibrating the IMU's offset at the beginning of data collection and fine tuning during post processing ensures accurate head pose within  $7^\circ$  ( $\sigma = 8.34^\circ$ ) of error for short recordings. Longer recordings may incur significant error in pose estimates unless externally corrected or reduced using a secondary sensor. Frequent head turns may also lead to an increase in head pose error so it is a good practice to reset the IMU following a few head turns. [92] While we do not hinder participants mid task, pose estimates for certain recordings (marked with  $\gamma$  in Supplementary Table 1) were manually corrected by a rotation operation before and after each heading change during post processing. Head angular drift and deviation in orientation are measured for all participants by the difference in head pose at the beginning and end of a task. We evaluated the sensor drift to be  $0.021^\circ/s$  ( $\sigma=0.035$ ) on average. Per participant drift can be found in Supplementary Table 1.

### 3.3.3 ZED Stereo camera

The ZED stereo camera provides a 1080p point cloud at  $30Hz$  which is calibrated and mapped onto the ETG coordinate system  $C_E$  from its own coordinate system  $C_Z$ . We found the error in depth measurement to be proportional to the distance under consideration. The euclidean 3D error was found to be less than  $0.5m$  at a distance of  $\sim 10m$  (beyond that is considered to be infinity), which is in agreement with other independent analysis. [105]

## 3.4 System calibration

All measurements in the GW setup are reported in reference to a modified checker chart which is fixed in the world coordinate system (see Figure 3.5). Prior to data collection, we instructed the participants to perform two calibration routines before each task.

**Routine 1** - This is the native offline calibration routine offered by Pupil Labs version 1.8.26 (i.e. *calibration using natural features*) following 3D pupil detection and gaze mapping. This routine required that subjects looked sequentially at red calibration targets placed in alternating boxes on the modified checkerboard chart.

**Routine 2** - In the second routine, participants were asked to maintain a comfortable head pose while fixating on one of the calibration targets. They were then asked to move their heads horizontally or vertically while maintaining fixation at that point, thus inducing a vestibular ocular response. This routine performed a system calibration by

aligning all hardware components to a common world coordinate system.

### 3.4.1 Pupil Labs eye tracker calibration

The angular error between the gaze POR and the location of the red calibration target within the world camera imagery is presented in Figure 3.6. This measure reflects  $\angle(k_e^{-1}P_x, k_e^{-1}P_c)$ , where  $P_x$  and  $P_c$  are the homogeneous coordinates of the red calibration target and gaze PoR.  $k_e$  is the intrinsic matrix of the ETG world camera. We evaluated the calibration accuracy to be within  $1^\circ$  of error within  $10^\circ$  from the center of the calibration pattern. Individual participant eye tracker calibration error can be found in Supplementary Table 1. The ETG eye camera has manual focus lenses which were readjusted for every participant to ensure sharp visual features.

### 3.4.2 Temporal alignment

Each individual component of our system has a fixed temporal offset from each other. This temporal offset is removed using normalized cross-correlation of the angular velocity traces between the IMU, ETG and the ZED stereo camera. Since the ZED camera utilizes visual odometry to derive a pose estimate, it is not uncommon to observe a poor pose estimate during the VOR calibration routine. In those situations, we tracked the checkerboard corners in the ZED and ETG world camera to derive a velocity estimate for each corner point. In the absence of ZED pose information, these velocity estimates were used to compute the offset between ETG and ZED using cross-correlation as described in the next section. It should be noted that there exists an inherent latency between head and eye movements during a VOR [106]. However, we remove all latency while correcting for temporal offsets (including biological latency).

### 3.4.3 ETG-IMU calibration

Initially, the IMU and ETG are defined in their own respective coordinate systems,  $C_H$  and  $C_E$ . When participants were fixated at a point on the calibration chart during Routine 2, their eye and head pose was defined as the Z axis of our new world coordinate system  $C_W$  using rotation operations. The IMU is placed approximately 1-2 *cm* above the cyclopean gaze origin (an imaginary point midway on the line joining both eye centers). Instead of correcting for translation offset (which can vary by subject), we choose to align  $C_H$  and  $C_E$  to  $C_W$  solely using rotation matrices  $R_H^W$  and  $R_E^W$ . These matrices were initially derived using vector rotations and manually fine tuned until the coordinate systems were satisfactorily aligned (Gaze-in-World (GiW) velocity, *i.e.*, the head compensated cyclopean

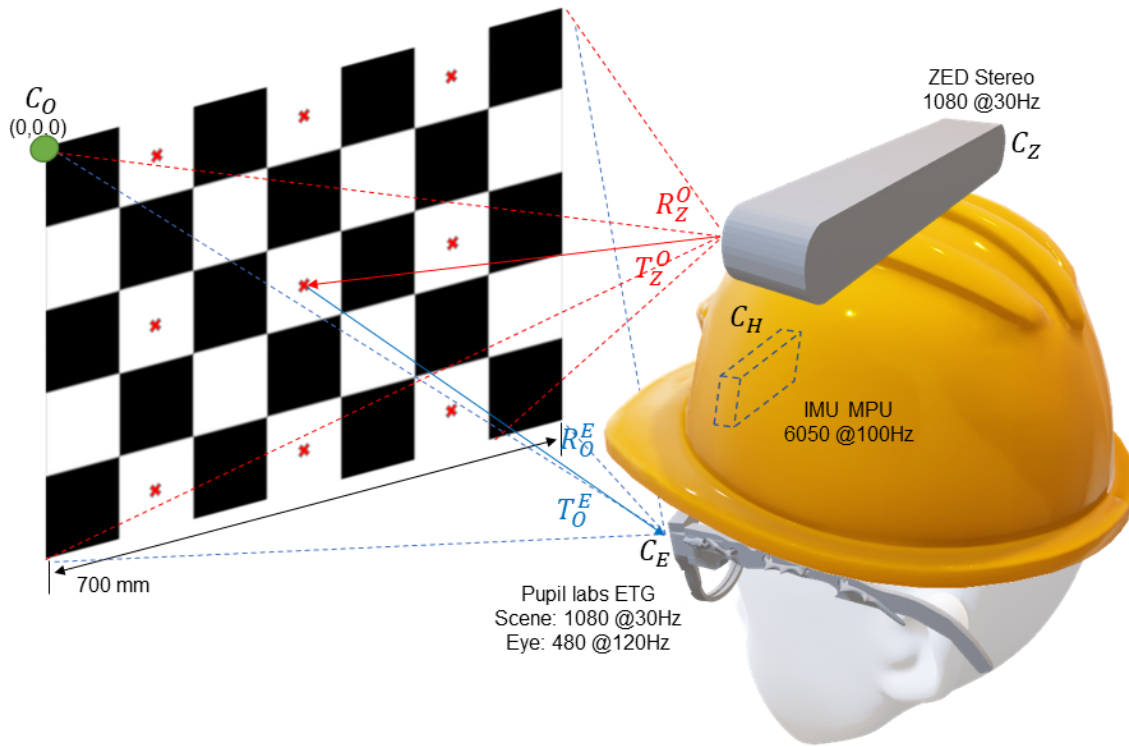


Figure 3.5: Checkerboard pattern placed in front of a participant during calibration phase. The red cross marks are used to calibrate the eye tracker in routine 1. The checkerboard corners are used for a 2-way multiview calibration between the ZED stereo camera  $C_Z$  and the eye tracker world camera  $C_E$ .  $(R_Z^O, T_Z^O) \rightarrow$  Transformation needed to move from  $C_Z$  to the calibration chart's coordinate system,  $C_O$ .  $(R_O^E, T_O^E) \rightarrow$  Transformation needed to move from  $C_O$  to  $C_E$



eye velocity is minimized). Once the head and eye orientation are defined in  $C_W$ , we rotate the EiH vector using the updated head pose to obtain the GiW vector.

#### 3.4.4 ETG-ZED calibration

Calibrating the ETG and ZED is required to register the depth point cloud from ZED's coordinate system  $C_Z$  to the ETG scene camera  $C_E$ , to obtain calibrated depth values of the visual field. The visual field is defined from the center of the world camera, hence we choose to superimpose the depth map onto the world imagery. Since the distance between each checkerboard corner point is known, we can produce a grid of corner points in world units ( $mm$ ) defined in the checkerboard coordinate system  $C_O$ . This grid can be aligned and projected on  $C_E$  and  $C_Z$  using extrinsic parameters  $(R, T)$ . Corner points extracted from time synced ETG world and ZED left camera images were used to find  $R$  and  $T$ . The extracted image points and the checkerboard grid are related using  $x_Z = k_Z(R_O^Z X_O + T_O^Z)$  and  $x_E = k_E(R_O^E X_O + T_O^E)$ . Here,  $X_O$  is the 3D checkerboard grid defined in  $C_O$ .  $k_Z$  and  $k_E$  are the left ZED and world camera intrinsic matrices. For detailed information regarding this process, we refer the reader to single camera calibration, part 1, multiview geometry by Hartley *et al.* [107]. The transformations required to align  $C_Z$  to  $C_E$  can be derived as  $R_Z^E = R_O^E R_O^Z^{-1}$  and  $T_Z^E = T_O^E - R_Z^E T_O^Z$ , which are used to transform the depth point cloud from  $C_Z$  to  $C_E$ . Once we have an aligned depth map, we trace a ray from the ETG world camera center to a subject's PoR and intersect it with the transformed point cloud to derive a 3D PoR in  $mm$ .

### 3.5 Operations

All absolute angular velocity measurements (i.e. magnitudes) are calculated using a modified Two-Point Central Difference algorithm (2-P) [108]. The angular velocity  $\omega_v$  can be derived as  $\delta\theta/\delta t$ , where  $\delta\theta$  is given by  $\angle(v_{n+1}, v_{n-1})$ . Here,  $v_n$  is a normalized unit direction vector while  $t_n$  is the timing associated with sample  $n$ .  $\delta\theta$  is the angular displacement within the elapsed time. For a fixed sampling rate  $f_s$ ,  $\omega_v = f_s \delta\theta/2$ .

Pupil tracking is usually performed in the near infrared because the human iris, regardless of color in the visible spectrum, reflects well in the near infrared. This ensures adequate contrast between the iris and the pupil, which is dark when illuminated off axis. However, noise may be introduced while tracking the pupil due to many external and internal factors such as varying illumination conditions, algorithmic artifacts, lack of contrasting eye features, occluded pupils etc. These artifacts may result in high frequency noise in the pupil positional signal. Consequentially, several steps were taken to filter the gaze signal. Since the eye was imaged with a sampling frequency of  $f_s$ , frequencies

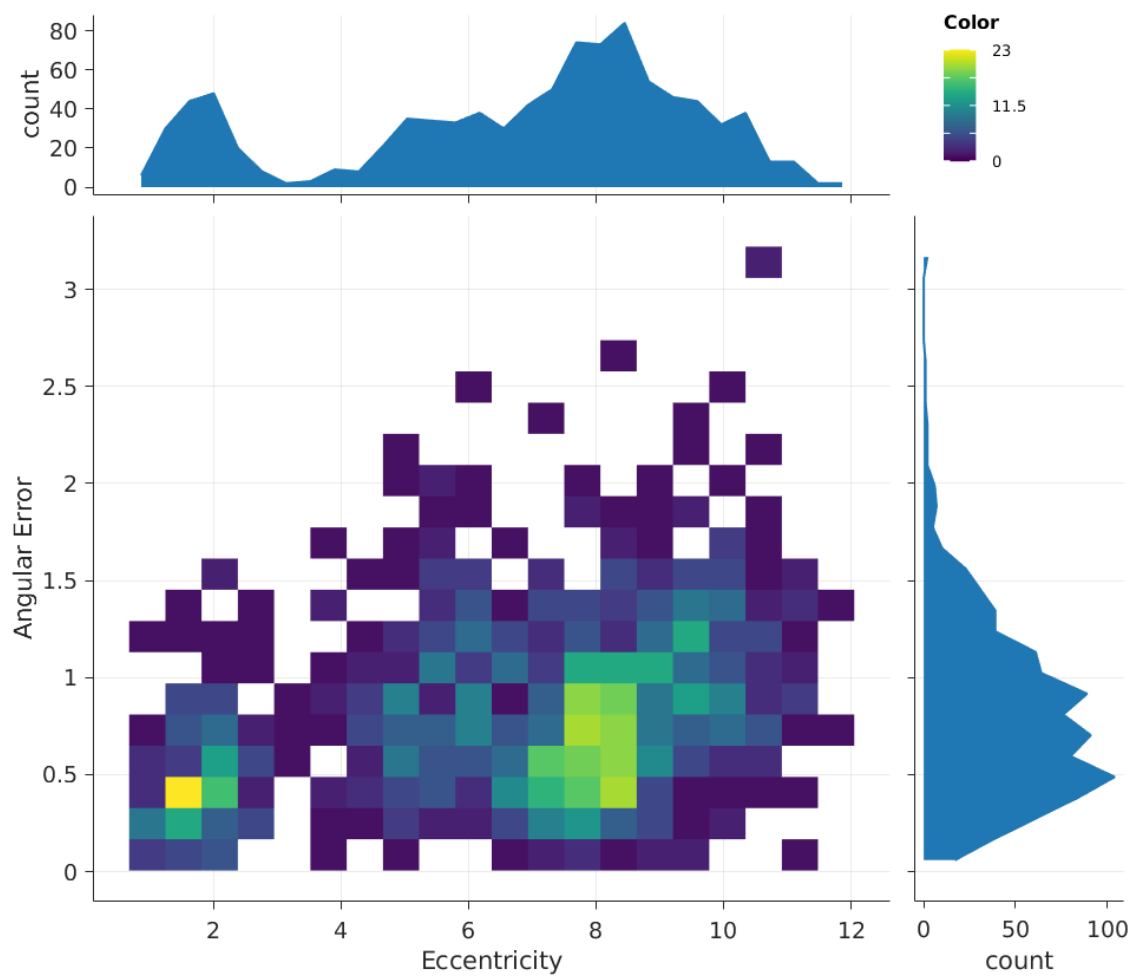


Figure 3.6: Eye tracker accuracy vs eccentricity from the center of the calibration pattern. Color scale indicates the number of calibration samples from all subjects.

higher than the Nyquist frequency ( $f_n = f_s/2$ ) were aliased into our signal as noise. To avoid aliasing, we introduced a low pass filter to suppress all frequencies higher than  $f_n$  (Kaiser window, cut-off:  $58 \pm 2$  Hz), a limit well beyond that where typical saccades exhibit significant power. [109] Furthermore, the 2-P central difference algorithm results in gain suppression near  $f_n$  without exhibiting phase shifts as opposed to other non-symmetric techniques wherein signal delay is not constant. Phase offsets due to anti-aliasing filters were removed by performing Zero-Phase filtering. [110] To further reduce noise, we utilized Bilateral filtering [111] since it provides an optimal trade-off between noise removal while maintaining characteristics of eye movements (such as preserving peak saccade velocity). Non-adaptive techniques such as Gaussian filtering suppressed saccade velocity peaks while increasing their duration and potentially produce misleading characteristics which could lead to misinterpretation of eye movements. The optimal parameters for bilateral filtering were empirically derived (window length  $50ms$ ,  $\sigma_t = 18ms$ ,  $\sigma_r = 8.75^\circ/s$ ). The azimuthal and elevation velocity components are calculated using small angle approximations because of numerical stability during quadrant changes. That is,  $\omega^{Az} = \delta\theta^{Az}/\delta t$ .  $\delta\theta^{Az}$  is approximated as  $\sin \delta\theta^{Az}$ . Small angle approximation results in 1% error in measurement at  $14^\circ$ . Assuming a maximum human angular velocity of  $900^\circ/s$ , the upper limit for human angular displacement cannot exceed  $\sim 8^\circ$  within a sample at our sampling rate of  $120Hz$ , which is within 1% measurement error.

### 3.6 Labelling

Training and evaluating a gaze event classification model requires labelling our dataset which is one of the major contributions of this work. The GW dataset was hand-labelled by five annotators who were trained to identify head-free gaze events. They produced over 140 minutes of hand-labelled head-free gaze behavior data. The dataset contains approximately 19,000 detected fixation events, 18,000 saccades, 1,300 pursuit events, and 3,500 blinks. Using a custom labelling tool (see Figure 3.7), labellers had access to eye images, scene images with PoR cross-hair, and the individual head and eye velocity traces. Using our tool, one minute of recorded data requires 45-60 minutes of annotator time. While it is possible to develop tools that allow faster labelling[112], they may bias the labeller with automated suggestive labels. Each labeller made decisions independently and they were encouraged to leave sequences where they were uncertain of the classification untouched. These sequences, along with low confidence samples (confidence below 0.3, see section 3.3.1), were treated as unlabelled and were not used to compute statistics or to train/evaluate models. While we do observe saccades as low as  $15^\circ/s$ , we do not label microsaccades or post saccadic oscillations due to system accuracy limitations (head compensated gaze tremor was found to be  $\mu=0.55^\circ/s$ ,  $\sigma=0.32^\circ/s$ ). To provide

maximum flexibility to researchers, we labelled stable fixations (caused due to tremors, drift and micro-saccades) and rVOR as a single gaze event type, stationary fixation, while labelled fixations due to tVOR and optokinetic stimulation as another gaze motion category, fixation under translation (labellers used *gaze following* as a pseudonym). This enables researchers to isolate the influence of compensatory mechanisms using a variety of statistical methods.

Cohen’s Kappa  $\kappa$  is a measure of the overall agreement between two raters classifying items into a given set of categories. [113] For a given gaze event category, *precision*  $p$  is the fraction of accurately detected samples over all retrieved samples while *recall*  $r$  is the fraction of accurately detected samples over all relevant samples in the groundtruth. These measures, along with the  $F_1$  score (the harmonic mean of  $p$  and  $r$ ) are applied by iteratively calculating agreement between each labeller and the rest of the group, and then reporting the average value. Note that the described iterative strategy results in  $p$  and  $r$  holding the same value. The average overall value of Cohen’s Kappa  $\kappa$  was  $\tilde{\kappa}$  of 0.74 ( $\sigma=0.03$ , median=0.74), and Cohen’s Kappa is reported for each event type in Tables 3.1, 3.2, and 3.3. Previous studies have shown that human coders exhibit a performance above 0.85  $\tilde{\kappa}$  while classifying head fixed eye movements, with a very low inter-rater variance. [11] While we have not managed to replicate such a high level of agreement, we can offer insights as to why. First, head-free gaze behavior is significantly more complex with a wide range of behaviors to be classified into the previously mentioned labelling scheme in Section 3.1. For instance, consider classification of head-free gaze behavior while attempting to catch a ball into periods of gaze fixations, saccades and pursuit. Subjects engaged in head-free gaze pursuit for a very small portion of the ball trajectory, primarily relying on a series of fixations and predictive saccades to track the moving ball. This distinction is not straightforward and can easily be overlooked during labelling. Secondly, relying on a single source of information such as visual imagery or gaze signals could lead to incorrect coding (see Supplementary Figure 1). Signal filtering and interpolation produces artifacts which may be interpreted differently by each rater. [114] Despite the fact that we have provided multiple sources of information, it is not uncommon for a human labeller to make erroneous decisions. Lastly, while it is accepted that human coders may change their labelling strategy over time [11] and the start and end times of coded events may vary, lack of holistic task awareness could result in data misinterpretation.

### 3.6.1 Training labellers

Our labelling team was trained using lectures on eye movements, gaze interpretation and eye-head coordination from the literature to thoroughly understand the labelling nomenclature used in GW. They were then asked to label a common, very small subset of the dataset that was then analyzed and discussed as a group with the authors. The labellers

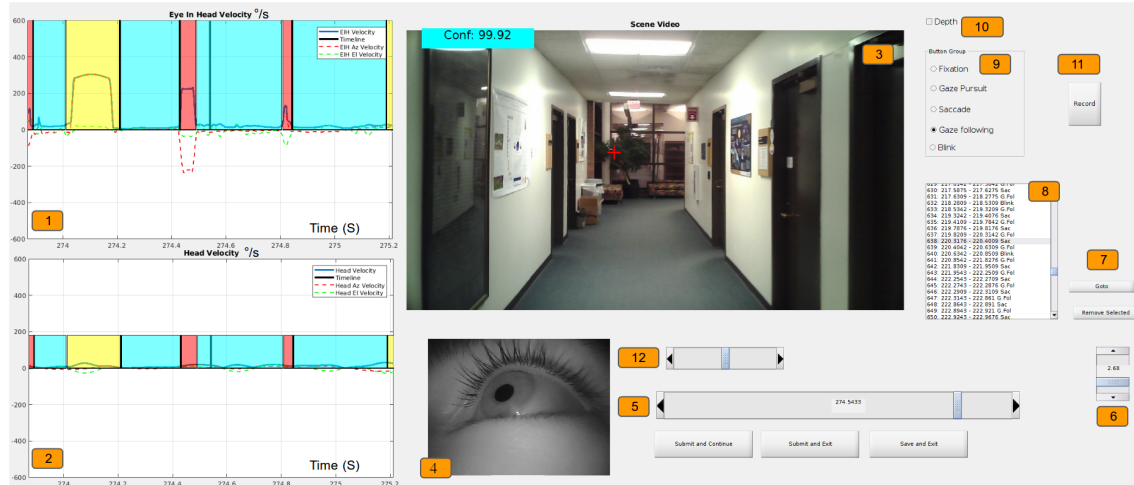


Figure 3.7: Custom made GUI for labelling. 1: Magnitude of EiH velocity with  $Az$  (Azimuthal) and  $El$  (Elevation) velocity traces ( $^{\circ}/s$ ). 2: Magnitude of Head velocity with  $Az$  and  $El$  velocity traces ( $^{\circ}/s$ ). 3: World-view overlaid with the Point-of-Regard (PoR) and confidence score. 4: Eye-view. 5: Slider to move a window through a recording temporally. 6: Slider to change the window width. 7: *Go-to* and *Remove button* for labelled regions. 8: Interactive list of labels in a session. 9: Radio buttons to select event type and mark across a region. 10: Toggle scene and depth view. 11: Record a 10 second clip of GUI starting at the current sample. 12: Slider to shift labels forward or backward.

|          | Fixational samples |          | Gaze-pursuit samples |          | Saccade samples |          |
|----------|--------------------|----------|----------------------|----------|-----------------|----------|
|          | $\mu$              | $\sigma$ | $\mu$                | $\sigma$ | $\mu$           | $\sigma$ |
| $\kappa$ | 0.74               | 0.04     | 0.73                 | 0.05     | 0.75            | 0.04     |
| $p/r$    | 0.94               | 0.03     | 0.77                 | 0.12     | 0.79            | 0.10     |
| $F_1$    | 0.94               | 0.02     | 0.75                 | 0.04     | 0.78            | 0.03     |

Table 3.1: Sample based Cohens  $\kappa$ , precision/recall  $p/r$  and  $F_1$  score between labellers. Note that the precision and recall values are identical (see section 3.6 for details.)

|            | Fixational events |          | Gaze-pursuit events |          | Saccade events |          |
|------------|-------------------|----------|---------------------|----------|----------------|----------|
|            | $\mu$             | $\sigma$ | $\mu$               | $\sigma$ | $\mu$          | $\sigma$ |
| $l_2$      | 12.84             | 2.25     | 13.39               | 2.82     | 12.96          | 1.99     |
| $O_r$      | 0.91              | 0.01     | 0.92                | 0.02     | 0.74           | 0.03     |
| $F_1$      | 0.86              | 0.04     | 0.75                | 0.04     | 0.89           | 0.04     |
| $\kappa$   | 0.71              | 0.09     | 0.54                | 0.05     | 0.79           | 0.09     |
| $\kappa^*$ | 0.54              | 0.14     | 0.47                | 0.09     | 0.61           | 0.15     |

Table 3.2: Inter-labeller event based metrics. All metrics are reported by their mean  $\mu$  and inter-subject standard deviation  $\sigma$ .  $l_2$  distance of the start and end time (expressed in *ms*) of matched events using ELC.  $O_r$  is the overlap ratio between matched events using ELC.  $F_1$  score as proposed by Hooge et al [11]. Event  $\kappa$  proposed by Zemblys et al [12]. Event  $\kappa^*$  found using ELC event matching. For more information on each metric, please refer to Section 4.3

|                      | Fixational samples |          | Gaze pursuit samples |          | Saccade samples |          |
|----------------------|--------------------|----------|----------------------|----------|-----------------|----------|
|                      | $\mu$              | $\sigma$ | $\mu$                | $\sigma$ | $\mu$           | $\sigma$ |
| Fixational samples   | 0.94               | 0.03     | 0.02                 | 0.02     | 0.03            | 0.02     |
| Gaze pursuit samples | <b>0.20</b>        | 0.10     | 0.77                 | 0.12     | 0.03            | 0.03     |
| Saccade samples      | <b>0.19</b>        | 0.08     | 0.02                 | 0.02     | 0.79            | 0.10     |

Table 3.3: Normalized sample based confusion matrix (created by normalizing the confusion matrix with the number of samples for each event type in the ground truth) across every recording with multiple labellers.

began manually annotating the GW dataset following this group exercise. Individual weekly meetings with the authors were set to discuss periods of uncertain data.

### 3.6.2 Data cleaning and post processing

To remove erroneous labels, we adapt the approach proposed by Zemblys *et al.* [83] For our dataset, fixational events with  $<0.5^\circ$  separation between them and within  $75ms$  of each other were combined into a single event. Fixations less than  $50ms$  and saccades greater than  $150ms$  in duration were automatically removed. Finally, labelled events with duration less than  $10ms$  were automatically removed.

## Chapter 4

# Gaze classification and analysis<sup>1</sup>

The aim of this work is to use labelled eye and head movement data acquired from the Gaze-in-Wild dataset to train automated classifiers. We trained two standard machine learning models for gaze event classification: a moving window based method and a recurrent neural network (RNN). The input to both classifier models is a sequence of temporally discrete sensor data vectors, i.e.,  $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \dots, \mathbf{x}_T\}$ , where  $\mathbf{x}_n \in \mathbb{R}^d$  and  $n$  is the current time step. As described in Section 4.2, these data vectors contain information from the IMU and eye tracker. For both models, we merge fixations when stationary and fixations under translation into a single gaze fixation class (see Section 3.1 and Section 3.6).

### 4.1 Classification models

The moving window model classifies a gaze sample at time  $n$  by aggregating information from a window of data vectors adjacent to  $\mathbf{x}_n$ , i.e.,  $\mathbf{w}_n = W(\mathbf{x}_{n-s}, \dots, \mathbf{x}_n, \dots, \mathbf{x}_{n+s})$ , where the vector of window features  $\mathbf{w}_n \in \mathbb{R}^g$  is computed using a window size of  $2s + 1$  samples and the function  $W(\cdot)$  computes the windowed feature vector. We chose the random forest (RF) classification algorithm since it works well for low-dimensional data, and our framework resembles state-of-the-art gaze event algorithms for controlled 2D environments.[12] RF is an ensemble learning method wherein multiple decision trees are trained on a subset of samples and their feature space.[115] A RF is easy to train and they are robust to noise and over-fitting, which are common problems for decision trees. For gaze classification in 2D controlled environments, Zembyls *et al.* showed that RF performed well with only 16 trees and 10-dimensional features up to a 200 ms window.[12] In our experiments, we use 40 trees, a minimum leaf size of 30, and we use  $\sqrt{g}$  randomly

---

<sup>1</sup>This chapter appears in a published manuscript by Kothari *et al.* [5]



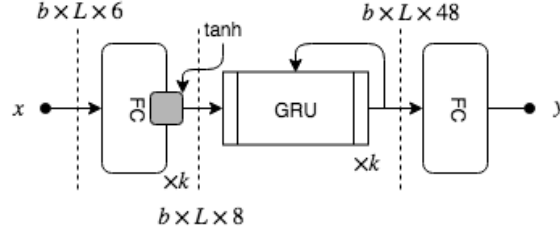


Figure 4.1: Bidirectional recurrent network model architecture. The model takes the magnitude, azimuthal and elevation eye and head velocity (6 features) as its input, passes through  $k$  fully connected feature extraction layers. These features are fed into a stack of  $k$  GRU layers which learn temporal patterns to classify a sample  $x_t$ . The forward variant (fRNN) outputs a 24 dimensional vector instead of 48 before being reduced to 3 at the final FC layer.

selected features per tree where  $g$  is the number of features for a given window size. To improve efficiency during the process of training the window-based RF classifier, we removed duplicated  $\mathbf{w}$  vectors (samples with equal value up to the second decimal). These duplicates were instead represented by a single sample that was upweighted by the number of duplicates found (e.g. the confidence measure was scaled). No duplicates were removed from the test set.

Rather than using explicit windows, the RNN model operates on the velocity data stream, i.e the absolute, azimuthal and elevation velocity (see Section 3.5). We use two variations of the RNN model. Our one directional forward RNN model (fRNN) classifies the gaze at time  $n$  using only past and present information, i.e.,  $F(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ . This model would be especially useful for real-time gaze prediction. For offline processing, we also use a bi-directional RNN (biRNN) that has past, present, and future information as input, i.e.,  $F(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n, \dots, \mathbf{x}_T)$ . Both models are implemented with gated recurrent units (GRUs),[116] which can handle longer-term dependencies than simple RNNs. A similar approach was used by the GazeNet architecture,[12] which used an RNN to classify events in a controlled 2D environment. To prevent the over-representation of samples that were labeled by  $N$  labellers (where  $N > 1$ ), these samples were weighted by  $1/N$  during the process of training.

The input to our model is a subset of windowed features  $W$ . Specifically, the model accepts the absolute, azimuthal and elevation EiH and head velocity as input. Multiple sequences,  $b$ , are stacked into a single batch of data. All sequences were padded with zeros to be of the same length as the longest sequence present in the batch,  $L$ . This  $b \times L \times 6$  dimensional data passes through  $k$  fully connected layers which generates a nonlinear

representation of EiH and head velocity. Extracted features are fed into a stack of  $k$  GRU (see Figure 4.1) units with a dropout probability of 10% which learn to associate temporal patterns with a type of gaze behavior. We use a combination of cross-entropy and generalized Dice [117] loss functions. The network was optimized using ADAM[118] for 175 epochs with a learning rate of 0.001, which we reduced linearly as the training performance improved. We experimented with the number of recurrent and linear layers and found  $k=3$  worked best. All codes related to GW is made publicly available.

## 4.2 Input features

The  $\mathbf{x}_n$  features consist of normalized EiH vectors  $v_e$  and head vectors  $v_h$  concatenated together. For the window-based RF classifier, for each time step  $n$ , we extract the following set of handcrafted features from a window of size  $2s + 1$  around the  $n$ -th sample:

1. **Mean EiH and head angular distance:**  $\Delta\theta_e, \Delta\theta_h$ . Angular distance in degrees between the mean EiH/head vector of  $s$  samples before and after the current sample of interest,  $x_n$ .
2. **Deviation in EiH and head velocity:**  $\sigma_e, \sigma_h$ . Standard deviation of magnitude of EiH and head angular velocity.
3. **Confidence:** We supply the confidence measure (see Section 3.3.1) to our classifiers as weights for each sample. High confidence and duplicate samples are assigned larger weights.

We also aggregate velocity measurements from every sample in the window. The velocity measurements are the absolute EiH  $|\omega_e|$  and head velocity  $|\omega_h|$  (angular velocity extraction has been described in Section 3.5), azimuthal EiH and head velocity  $\omega_e^{Az}, \omega_h^{Az}$  and elevation EiH and head velocity  $\omega_e^{El}, \omega_h^{El}$ . All velocity measurements are expressed in  $^\circ/s$ . Azimuthal and elevation velocity contain directional information such that a positive sign indicates a clockwise rotation and vice versa. Assimilating features as a time series results in a  $g$  dimensional window feature vector, where  $g = 4(2s + 1) + 6$ . The full window feature vector is given by  $\mathbf{w}_n = ([|\omega_e|, |\omega_h|, \omega_e^{Az}, \omega_h^{Az}, \omega_e^{El}, \omega_h^{El}]_{n-s}^{n+s}, \Delta\theta_e, \Delta\theta_h, \sigma_e, \sigma_h)_n^T$ , where  $[*]$  stands for aggregation.

## 4.3 Error metrics

Evaluating the performance of automated classification systems or human labellers is not straightforward. Traditional error metrics give sample-level measurements (*e.g.* percentage of individual samples correctly classified) and evaluate performance on a global basis,

thus oblivious to the inherent structure of the data. For instance, metrics such as accuracy, precision, recall and  $F_1$  score are widely used to evaluate the performance of head fixed gaze classification algorithms. [83, 114, 119] For evaluating agreement level among labellers or classifier performance with unbalanced data (large variation in the number of samples per class), accuracy based error metrics suffer from the *Accuracy Paradox* [12] which means that a predictive model with high sample level scores might have a lower event prediction ability. Powers observed that symmetric kappas (*e.g.* Cohen’s kappa), which are designed for inter-rater metrics, may not be directly suitable for automated classifiers. [120] Sample based measures fail to account for any temporal structure and may not reflect the severity of misclassifying a few, albeit structurally important, samples. Furthermore, it is more intuitive to reason in terms of correctly/incorrectly classified collections of continuous samples of the same class, or *events*.

Event based metrics were designed to compensate for the limitations of sample based evaluation methods. Hoppe *et al.* provided the percentage of correctly classified events by comparing the samples within the bounds of each groundtruth event. The category with the highest number of samples was matched with the reference event. [119] Hooge *et al.* proposed a set of evaluation metrics such as the event-level  $F_1$  score, the relative timing offset (RTO) and the relative timing deviation (RTD) between matched events. [11] To compute the  $F_1$  score for a particular gaze movement category, they treat every other category as a common opposite category. However, this operation removes all inter-category confusion. The first overlapping testing event of the same category as the groundtruth is considered as matched. Temporal offsets between event start and end times are calculated for all matched events, providing the added benefit of a measure for temporal alignment quality. Zemblys *et al.* proposed the event error rate (EER), which is a length normalized Levenshtein distance between event sequences. [12] Zemblys *et al.* also proposed the event-level Cohen’s kappa measure, an extension of the event-level  $F_1$  score. [12] These proposed event level metrics use the standard available measures ( $F_1$ , Cohen’s  $\kappa$ ) but vary in their event *matching* scheme. Differing from Hooge *et al.*, Zemblys *et al.* proposed that a testing event with the highest overlap ratio with a groundtruth event is to be treated as a match. Note that events of differing categories may also be considered as *matched*. This results in an event level confusion matrix which is used to generate an overall and per category Cohen’s kappa score. Existing event level metrics improve the way we evaluate the performance of temporal classifiers but have their own individual shortcomings for varying scenarios. For instance, the majority vote method gives no penalty to unexpected short events that split longer events, and significantly influence the statistical distribution. [119, 12] The event level  $F_1$  score also does not support multi-class evaluation, [12] and the EER measure does not match events and treats all event sequences as strings. It does not consider or provide insight into temporal offsets. Furthermore, it also suffers

|                       | Technique        | Timing offsets | Confusion matrix | Symmetric | Threshold dependency | Reliability of timing offsets |
|-----------------------|------------------|----------------|------------------|-----------|----------------------|-------------------------------|
| Majority vote [119]   | Sample majority  | ×              | ✓                | ×         | ×                    | N/A                           |
| Event F1 [11]         | Earliest overlap | ✓              | ×                | ×         | ×                    | low                           |
| Event kappa [12]      | Largest overlap  | ✓              | ✓                | ✓         | ×                    | low                           |
| Event error rate [12] | N/A              | ×              | ×                | ✓         | ×                    | N/A                           |
| ELC                   | Window match     | ✓              | ✓                | ×         | ✓                    | high                          |

Table 4.1: Comparison of event level error metrics

from the *Accuracy Paradox* and only returns a single value as an overall rating. Last but not least, different event-matching procedures significantly affect the RTO and RTD measurements. Zemblys *et al.* identified that the RTO and RTD measures will be compromised when using the largest overlapping event-matching strategy. [12] Similar situations may occur when utilizing the earliest overlapping matching strategy. For example, when onset of the earliest overlapping testing event is close to the offset of a reference event. Various event based metrics are summarized in Table 4.1. To address some of the shortcomings of previous approaches, we devised the Event Level Cross-Category Metric (ELC) as described below.

Consider the following taxonomy:

- Reference sequence - groundtruth sequence of labels.
- Testing sequence - predicted sequence of labels, usually the output of an automated classification process.
- Matched event - two events are considered matched when their start and end position roughly align in a predetermined window and meet the matching criterion (discussed below). As an example, consider sequences L1 and L2 in Figure 4.2. All fixation events in L1 (marked in green) are considered as matched.
- Unmatched event - All events which do not satisfy our matching criterion are considered as unmatched. Both saccades in Figure 4.2, are considered as unmatched.
- Detached event - We often find unmatched events in our ground truth which completely overlap with another test event and belong to the same gaze category. These type of events are considered to be detached. For example in Figure 4.2, the blink in L1 (marked in yellow, the start point is matched whereas the end point has no match) is considered as a detached event. Researchers may safely consider detached

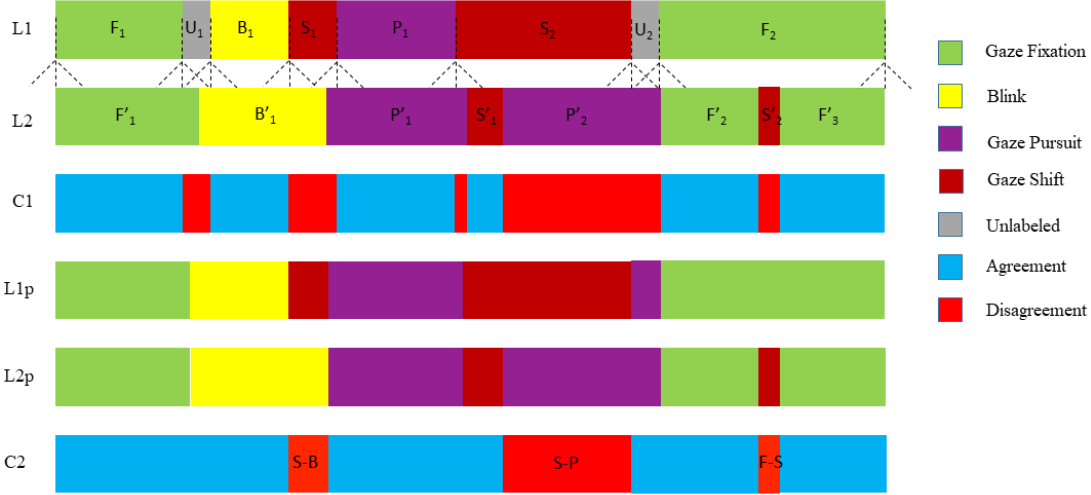


Figure 4.2: Illustration of the ELC metric on handcrafted test and reference sequences. (L1) Labels provided by labeller 1. (L2) Labels provided by labeller 2. Colors indicate the event type and whether labellers are in agreement. Dotted lines from L1 into L2 indicate the time window used in ELC for each transition point. (C1) Direct sample-sample comparison between labeller 1 and labeller 2. (L1p, L2p) Results of applying ELC to labels provided by labeller 1 and labeller 2 respectively. (C2) Event-level comparison between labeller 1 and labeller 2. Unmatched regions are given specific labels describing the misclassification type. For example, ‘S-B’ means that labeller 1 labelled the data as gaze shift whereas labeller 2 labelled the same data as blink.

events as matches per their strictness requirements and application (this operation would inflate the performance score of a classifier).

- **Transition point** - It is assumed that all event boundaries touch each other at their transition points. Transition points have samples of different gaze behavior adjacent to it. In case event boundaries do not touch, we assume the period between them to be the *none* class. All entries pertaining to *none*, i.e blinks and unlabelled periods, are removed from consideration. Note that all events have two transition points.
1. **Window-based matching:** First, we identify every transition point in the reference and testing sequence. For every transition point in reference sequence, we extend a window of a certain size (*e.g.* 50 ms) onto the test sequence and find all transition points within. The reference transition point is matched with the first (in

time) testing transition point within the match window which satisfies a particular matching criterion. For onset transition points, the event type on the right should match the reference event type. Similarly, offset transition points are matched if the event type on its left matches the reference event type. An event is matched if both its start and end transition points are matched.

2.  **$l_2$  distance calculation for matched events:** Following window-based matching, overall timing offsets are calculated for matched events. Unlike RTO and RTD, which are calculated separately for start and end points of events, we calculate the  $l_2$  distance ( $\sqrt{(start_1 - start_2)^2 + (end_1 - end_2)^2}$  where  $start_1$  and  $start_2$  is the start positions of two events,  $end_1$  and  $end_2$  are end positions of two events) for each event. The mean and standard deviation of all calculated  $l_2$  distances (per class and overall) are used as indicators of alignment quality between two labelled sequences.
3. **Overlap ratio calculation for matched events:** Since events of different categories have various ranges of duration, the severity of temporal misalignment could be different for individual event types having the same timing offset values. Therefore, we calculate the *overlap ratio* [121]  $O_r$  ( $O_r = n_1 \cap n_2 / n_1 \cup n_2$ ), where  $n_1$  and  $n_2$  are samples belonging to two matched events. The overlap ratio reflects the temporal alignment quality of two events. As with the  $l_2$  distance, the mean and standard deviation are calculated and reported.
4. **Timing offsets correction:** Once the  $l_2$  distance and overlap ratio is calculated, we remove the effects of misalignment by correcting timing offsets in both sequences. This correction is applied on all matched transition points regardless of an event's match status. For each matched transition point, timing (sample index) of two points are averaged to create a single representative transition point. If the original point is shifted away from the event center, the displaced samples are assigned the event's category. Likewise, if the transition point moves inwards, the displaced samples are assigned the external event's gaze category. If the displaced samples are unlabelled in a particular sequence, they are assigned the same gaze movement class as the corresponding sequence.
5. **Event level confusion matrix:** Comparing two labelled sequences leads to a collection of matched and unmatched events, *i.e.*, a confusion matrix, which describes inter-category event classification performance. Owing to the timing offsets correction step, event mismatches within the preset threshold are eliminated. Standard metrics such as Cohen's  $\kappa$  and  $F_1$  score can be derived from the confusion matrix for deeper insights or to summarize performance.

6. **Applying previous steps in both directions:** ELC is an asymmetrical event matching technique. It can be applied twice by interchanging the testing and reference sequences to find an average performance measure along with a sense of metric agreement. For instance, if the number of detached events is higher in a particular order, it provides insight into larger proportions of event merges in the testing sequence. Inter-labeller performance is computed by applying ELC both ways but not for human-classifier evaluation.

In Figure 4.2, sequences L1p and L2p show the results of applying ELC to the labels in L1 and L2 respectively. The application of these rules eliminates many minor (mainly temporal) disagreements between sequences and considers only the regions of major disagreement as seen in sequence C2. Event Kappa utilizes the largest overlapping strategy to match events, which results in lower RTO and RTD scores. [12] For instance, event  $F_2$  in L1 gets split into two shorter events  $F'_2$  and  $F'_3$  by an unexpected event  $S'_2$  in L2, the metric tends to match the fixation in L1 with the largest overlapping event ( $F'_3$  in this case). This leads to a poor RTO and RTD measures. However, ELC considers the start and end points of  $F_2$  in L1 and matches them with the start of  $F'_2$  and the end of  $F'_3$  respectively.  $F_2$  is considered as a matched event and the testing sequence is rewarded by increasing the F/F counter in the confusion matrix. Likewise, the testing sequence is scored negatively for the offending event,  $S'_2$ , by increasing the F/S counter in the confusion matrix. The  $l_2$  distance (functionally equivalent to RTO and RTD measurements) accurately computes the alignment quality. Interchanging L2 as the reference and L1 as the testing sequence, events  $F'_2$ ,  $S'_2$  and  $F'_3$  would be considered as unmatched events and  $l_2$  distances would not be calculated.

Overall, ELC provides a faithful indication of timing offsets using the window-based matching strategy. ELC is dependent on a parameter, *i.e.*, the window size. The window size indicates the system tolerance for timing offsets between ground truth and testing events. Since it's easier to identify the start and end points of gaze shifts as compared to other types of gaze events, different window sizes for gaze shift related events ( $\pm 25ms$ ) and non gaze shift related events ( $\pm 35ms$ ) are used. Researchers may consider using larger window sizes for situations wherein event onset and offsets conditions are relaxed.

## 4.4 Results

The two classifiers are assessed using *leave-one-out* cross validation by testing on a single person's data (the holdout subject) and training the model on remaining subjects. This process is repeated for subjects 1, 2, 3, 6, 8, 9, 12, 16, 17, 22. For each of these tests, certain steps are taken to prevent overfitting. Training data for the holdout subject is

|       | $\kappa$ | G.Fix $\kappa$ | G.Pur $\kappa$ | Sac $\kappa$ |
|-------|----------|----------------|----------------|--------------|
| RF    | 0.63     | 0.63           | 0.28           | <b>0.74</b>  |
| fRNN  | 0.54     | 0.54           | 0.29           | 0.68         |
| biRNN | 0.61     | 0.61           | <b>0.37</b>    | 0.69         |
| Human | 0.74     | 0.74           | 0.73           | 0.75         |

Table 4.2: Sample based Cohen’s Kappa score  $\kappa$  for each optimized classifier.

split into five folds with approximately equal frequencies of saccades and fixation gaze events (but an unequal number of frames) per fold. However, due to the low frequency of pursuit events within the dataset, and their unequal distribution across subjects, we did not divide gaze pursuit events between folds. For each holdout subject, four folds were used to train the RNN model while the single fold was used to fit model parameters, which were then saved. Following conversion, the parameters for the best performing model on the single validation fold were saved. The iterative process results in five sets of converged parameters per holdout subject. The best performing set of parameters on the holdout subject is accepted as the optimal set of weights for that model type, and for subsequent comparison against other model types. One notable exception to this procedure is the RF algorithm, which does not require a validation set. Instead, its parameters were chosen to maximize its performance while maintaining a manageable model footprint  $\sim 50$  mega bytes.

Classifiers are evaluated using both sample and event level metrics (see Section 4.3). Classifier output is not evaluated during blinks or for unlabelled data points. As the window size increases, RF gains increasing temporal awareness which results in higher  $\kappa$  performance with diminishing returns. It can be observed in Figure 4.3 that RF arrives at an asymptotically improving performance with a window size of 30 *ms* and above. Individual  $\kappa$  scores for each gaze class reveals that all classifiers find it difficult to distinguish gaze pursuits. Overall, sample based metrics convey that RF with a large window size outperforms RNN for detecting saccades but performs poorly on gaze pursuit samples (Table 4.2).

We report event based metrics and observe that biRNN outperforms RF on all measures. Interestingly, event  $F_1$  and event  $\kappa$  scores computed using Zembyls *et al.* shows an increase in saccade classification performance (see Table 4.3) for biRNN over RF. However, this increase is not reflected using sample based metrics (Table 4.2) or event  $\kappa$  computed using ELC (Table 4.4). Notably, RF outperforms RNN based methods in  $l_2$  scores, indicating a better ability to produce tighter fits around saccades (see Table 4.4). Overall, gaze pursuit classification baselines fall short on human level performance but the results are



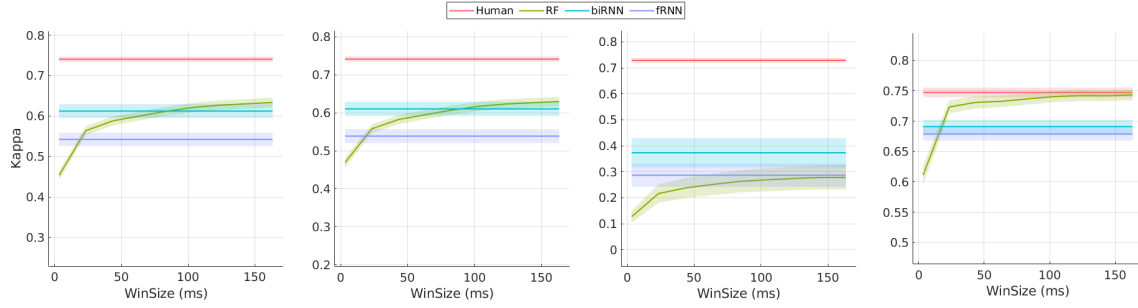


Figure 4.3: Sample level performance metrics. All performance curves are centered around their mean,  $\mu \pm$  standard error. Left - Overall  $\kappa$  score. Inner left - Gaze fixation  $\kappa$  score. Inner right - Gaze pursuit  $\kappa$  score. Right - Saccade  $\kappa$  score. Please note the varying y-limits to accentuate the difference in performance. RNN uses memory to encode temporal patterns, and hence the RNN architectures are represented as horizontal lines as they do not operate in window sizes. We would like to highlight that all window sizes are in the velocity domain. Window sizes in angular domain can be derived by adding 10 *ms* (please refer to Section 3.5).

|       | G.Fix $F_1$ | G.Pur $F_1$ | Sac $F_1$ | EER         | G.Fix $\kappa$ | G.Pur $\kappa$ | Sac $\kappa$ | Overall $\kappa$ |
|-------|-------------|-------------|-----------|-------------|----------------|----------------|--------------|------------------|
| RF    | 0.74        | 0.26        | 0.82      | 0.26        | 0.46           | 0.01           | 0.63         | 0.32             |
| fRNN  | 0.74        | 0.22        | 0.81      | 0.30        | 0.61           | 0.22           | <b>0.69</b>  | 0.47             |
| biRNN | <b>0.80</b> | <b>0.35</b> | 0.83      | <b>0.16</b> | 0.61           | <b>0.27</b>    | 0.67         | 0.47             |
| Human | 0.86        | 0.75        | 0.89      | 0.14        | 0.71           | 0.54           | 0.79         | 0.62             |

Table 4.3: Metrics based on various event matching techniques proposed by others. Event based  $F_1$  score proposed by Hooge et al. [11] Event Error Rate (EER) proposed by Zemblys et al. [12] Event and overall  $\kappa$  scores calculated using Zemblys et al. [12]

|       |                  | G.Fix       |       |              |              | G.Pur       |       |              |              | Sac         |       |              |              |
|-------|------------------|-------------|-------|--------------|--------------|-------------|-------|--------------|--------------|-------------|-------|--------------|--------------|
|       | Overall $\kappa$ | $\kappa$    | $O_r$ | $l_2$        | $l_2 \sigma$ | $\kappa$    | $O_r$ | $l_2$        | $l_2 \sigma$ | $\kappa$    | $O_r$ | $l_2$        | $l_2 \sigma$ |
| RF    | 0.37             | 0.31        | 0.93  | <b>12.97</b> | 2.10         | 0.03        | 0.88  | 15.15        | 3.44         | <b>0.54</b> | 0.75  | <b>12.80</b> | 2.16         |
| fRNN  | 0.27             | 0.21        | 0.90  | 15.09        | 3.32         | 0.03        | 0.89  | 15.70        | 3.44         | 0.44        | 0.69  | 15.01        | 3.42         |
| biRNN | 0.37             | <b>0.34</b> | 0.92  | 14.93        | 3.64         | <b>0.14</b> | 0.90  | <b>14.25</b> | 3.84         | 0.44        | 0.71  | 14.73        | 3.62         |
| Human | 0.56             | 0.54        | 0.92  | 13.85        | 3.64         | 0.47        | 0.89  | 15.23        | 3.84         | 0.61        | 0.73  | 13.67        | 3.62         |

Table 4.4: Standard metrics derived from the ELC confusion matrix.  $O_r$  is the overlap ratio between matched events.  $l_2$  distance between matched event start and end times and their standard deviation  $l_2 - \sigma$  in *ms*.  $l_2$  and  $l_2 - \sigma$  are similar to RTO and RTD metrics proposed by Hooge et al. [11]

| Cohen $\kappa$        | Overall | G.Fix | G.Pur | Sac  |
|-----------------------|---------|-------|-------|------|
| biRNN (only eyes)     | 0.56    | 0.55  | 0.24  | 0.71 |
| biRNN (only absolute) | 0.58    | 0.57  | 0.33  | 0.71 |
| biRNN                 | 0.61    | 0.61  | 0.37  | 0.69 |

Table 4.5: Sample based  $\kappa$  score after removing either head movement information or directional information.

consistent with the difficulty in classifying pursuit movements over other gaze movement types in general. [84, 122, 123] The biRNN model produces higher ratio of detached events (G.Fix: 0.21, G.Pur: 0.20, Sac: 0.05) as compared to RF (G.Fix: 0.09, G.Pur: 0.02, Sac: 0.11) which indicates that a larger number of ground truth events completely overlapped with events of the same category but their transitions did not fall within the matching window. Since it is debatable if these detached events can be considered as matches, we omit them from all measures to avoid inflating scores.

## 4.5 Ablation study

To understand the role of each feature, we systematically removed essential components from the best performing model (biRNN with 3 FC and GRU layers). The input to biRNN comprises of absolute EiH/head velocity, azimuthal and elevation EiH/head velocity. This generates a signal with 6 features. Please note that azimuthal and elevation velocity store relative direction information between the eye and head (-ve sign means an anticlockwise rotation). By comparing different conditions using sample based  $\kappa$  score, we highlight the essential components required for head-free gaze classification in Table 4.5. For a detailed comparison using all metrics, please refer to Supplementary Table 2.

As expected, the performance of biRNN with EiH information (only absolute eye velocity) did not vary while detecting gaze fixations and saccades, but drops by 35% ( $0.37 \rightarrow 0.24$ ) while detecting pursuit events. Interestingly, a few pursuit events were still detected despite the lack of head movements. This indicates that head-free eye movements during pursuit behavior show a varied velocity pattern than gaze fixations and can be differentiated without any knowledge of head motion. We also observe that there is a minor loss in performance of 10% ( $0.37 \rightarrow 0.33$ ) when we remove azimuthal and elevation components. This highlights that absolute velocity information alone can provide reasonable certainty for classification.

## 4.6 Discussion

The main purpose of this work was to build the first dataset of labelled gaze movements collected during natural behavior ‘in the wild’ (outside of the laboratory), to have multiple labellers manually label the gaze events in the dataset, and to showcase the performance of two standard temporal classification techniques, Random Forest and Recurrent Neural Networks, using some common evaluation metrics. To overcome incorrect inter-event timing offsets observed in existing metrics, we introduce the ELC metric. The usefulness of a classifier lies in its ability to generalize in unseen circumstances. Hence, all our baseline performances are evaluated using the *leave-one-out* approach, wherein a classifier is tested on a single subject’s data while trained on the rest. Despite the fact that there is variability among human labellers, there is as yet no other choice, so we rely here on their labels as the gold standard. To improve upon existing event metrics and provide a reliable measure of alignment quality, we devised a new event matching technique, ELC, which matches events based on their transition points. ELC provides some control on evaluation strictness by identifying events which belong to the same category but are not temporally aligned due to event fragmentation.

### 4.6.1 Lower gaze pursuit classification performance by classifiers

The best performing classifier for gaze pursuits is 49% lower than the average human level performance (sample  $\kappa$ :  $0.73 \rightarrow 0.37$ ) whereas fixation and saccade performance achieves an average of 87% of human performance (sample  $\kappa$  G.Fix:  $0.61 \rightarrow 0.74$ , Sac:  $0.69 \rightarrow 0.75$ ). While pursuing moving targets, we observed that participants seamlessly interchanged between fixational and pursuit movements. The distinction between these movements are difficult to observe, especially during low velocity conditions because small angular errors in orientation measurements (a phenomenon common with IMUs) could result in misinterpretation without additional context for consideration (such as scene imagery with overlaid gaze PoR), a modality currently unavailable to our classifiers. Distinction between gaze fixation and pursuit events is further compounded when the head tracks a moving target or makes anticipatory movements but gaze remains stable at a fixation point. This motion elicits a signal similar to VOR, but if we rely purely on visual inspection then these events can easily be confused with pursuit motion. Situations such as these, combined with minor orientation errors, largely contribute to fixation/pursuit confusion seen in Table 4.2.

### 4.6.2 Head tracking: A pursuit or fixation?

Previous research has shown that the head *tracks* a moving target (in our case the ball) while the eyes *predict* the ball location using predictive saccades. [28] We find numerous

instances of gaze shifts to known targets where head movements precede eye movements in an anticipatory manner [124] to ensure that upcoming eye movements do not deviate too far from the relatively tight distribution seen in Figure 3.4. Participants frequently showed tracking behavior with the head and predictive or catch-up motion with the eyes during early phase of the ball trajectory. This behavior is usually followed by gaze pursuits during the next phase, i.e. the ball height is peaked and its projected retinal velocity is low. Following the peak phase, participants made predictive saccades to their hand for successful ball interception. GW also captures instances where the head *catches-up* to the fixation location while maintaining a strict coupling with the ball trajectory. While some may argue that head tracking of a moving object constitutes a pursuit motion, we instructed labellers to mark those sequences as fixations because the signals are identical to a VOR (please refer to Supplementary Figure 1).

#### 4.6.3 Head and Eye tracking can have different coordinate systems

Based on the ablation study, we observed that providing only the absolute velocity information achieved almost the same performance as biRNN-3, our best performing model. Interestingly, it highlights that for a slight drop in performance, future end-end classification frameworks may perform reasonably well if they simply provide unaligned eye and head motion information. While gaze fixations and saccades are distinctly identifiable using only eye-in-head (EiH) information, pursuit movements would be difficult to differentiate with a fixation without head movement information. As a sanity check, we also verified that the presence of a head tracking device improves classification of head-free pursuit movements by up to 35% as opposed to without head movement information (sample  $\kappa$ : 0.24  $\rightarrow$  0.37). It is interesting to note that despite removing head movements, the RNN classifier is still able to identify a few pursuit events which indicates that they demonstrate different EiH velocity statistics as fixations (for more information, please refer to Supplementary Table 2).

#### 4.6.4 Gaze-in-world information for classification

We include head pose as an input modality for the classifiers. While it is possible to classify the gaze-in-world signal, which is the head compensated eye-in-head signal, we wanted to train algorithms which could directly capture eye and head movement dynamics along with classifying it. For instance, we often find gaze pursuit events which are dominated either by head or eye movements, a distinction which would be lost when classifying gaze-in-world information.

#### 4.6.5 General limitations

Given limitations in current technology, it is unavoidable that tracking head position using a low cost IMU will accumulate error over time. All task duration were  $\sim 3$  minutes long and the error in orientation at the start and end of a recording was found to be  $7^\circ$  on average (see Section 3.3). While this error affects the absolute velocity component by a very small margin ( $0.04^\circ/s$  on average), it leads to unwanted shifts in the azimuth and elevation velocity component (see Supplementary Figure 1). Despite the use of a ratcheted head strap, this error accrues, in part, due to slippage of the helmet on the head, which will cause a misalignment of the helmet-mounted ZED stereo camera and the Pupil Labs eye tracking glasses (see Section 3.2). Future work might further reduce slippage through using software correction, such as the estimation of rotational slip on a frame-to-frame basis by matching visual features in the stereo camera and Pupil Labs world camera imagery, or through the fusing visual pose estimates with IMU data, as is commonly used in simultaneous localization and mapping.

#### 4.6.6 Limitations of event-based metrics

Although event level error metrics give researchers a better idea of the actual performance of automated classifiers or agreement level between labellers, existing event level metrics suffer from various drawbacks. The majority vote metric by Hoppe *et al.* remains agnostic to the testing sequences' structure. It does not penalize during event fragmentation caused by unexpected short events in the testing sequence. [119, 12] Moreover, this metric could be biased by the distribution of samples. Event level  $F_1$  score does not work well in multi-category scenario [12] and gives out unreliable RTO and RTD. EER does not provide any measure of alignment quality and suffers from the *Accuracy Paradox*. [12] Event matching techniques based on the largest overlap ratio, such as the event  $\kappa$  proposed by Zemlys *et al.* do not provide a reliable measure of alignment quality. [12] ELC overcomes these issues by matching events whose transition points fall within a window. A potential drawback of ELC is its dependency on the window size. Although the window size could be carefully chosen for different types of events and transitions, the metric could generate different results due to varying window sizes. For example, if a small window size was chosen, ELC would have a lower tolerance for transition ambiguity between certain event types which could result in higher misclassification scores. Furthermore, ELC is not symmetrical. To alleviate that, we propose that metrics derived using ELC should be averaged when used to evaluate inter-coder performance. While ELC overcomes certain drawbacks from previous evaluation techniques, new event level metrics are needed which accurately reflect performance, is symmetric in nature, provides a reliable measure of temporal alignment quality and is independent of an external threshold.

## 4.7 Conclusion

This work introduces GW, a large-scale dataset for studying eye and head coordination in naturalistic conditions. Participants were asked to perform four tasks without constraining them in any manner and were free to accomplish the tasks in any manner they chose to. Approximately 2 hours and 15 minutes of gaze behavior was manually hand coded by multiple human annotators and used to train gaze classifiers. We benchmark the performance of two machine learning algorithms for classifying these events and found that both achieved near human level performance for detecting gaze fixations and saccades, but they found it difficult to distinguish gaze pursuit behavior without additional contextual information otherwise available to human coders. In an effort to produce intuitive measures for event level similarities between two sequences, we propose the ELC event matching algorithm. We verify that all commercial eye tracking solutions could benefit in classifying head-free gaze pursuit movements by including a low cost IMU. Furthermore, comparable results are observed when head-free gaze movements are classified purely based on absolute velocity information of the eye and head, which indicates that head-free gaze classification is possible without aligning the eye and head coordinate systems.

# Chapter 5

## RITnet<sup>1</sup>

### 5.1 Abstract

Accurate eye segmentation can improve eye-gaze estimation and support interactive computing based on visual attention; however, existing eye segmentation methods suffer from issues such as person-dependent accuracy, lack of robustness, and an inability to be run in real-time. Here, we present the RITnet model, which is a deep neural network that combines U-Net and DenseNet. RITnet is under 1 MB and achieves state-of-the-art results on the 2019 OpenEDS Semantic Segmentation challenge. Using a GeForce GTX 1080 Ti, RITnet tracks at over 300Hz, enabling real-time gaze tracking applications. Pre-trained models and source code are available <sup>2</sup>.

### 5.2 Introduction

Robust, accurate, and efficient gaze estimation is required to support a number of critical applications such as foveated rendering, human-machine and human-environment interactions, as well as inter-saccadic manipulations, such as redirected walking [125]. Recent non-intrusive, video-based eye-tracking methods involve localization of eye features such as the pupil [103] and/or iris [57]. These features are then regressed onto some meaningful representation of an individual’s gaze. Convolutional neural networks (CNNs) have demonstrated high accuracy [66, 69] and robustness in unconstrained lighting conditions [63] and an ability to generalize under low resolution constraints [74, 77].

In an effort to engage the machine learning and eye-tracking communities in the field of

---

<sup>1</sup>This chapter appears in a published manuscript by Chaudhary *et al.* [9]

<sup>2</sup><https://bitbucket.org/eye-ush/ritnet/>

eye-tracking for head-mounted displays (HMD), Facebook Reality Labs issued the Open Eye Dataset (OpenEDS) Semantic Segmentation challenge which addresses part of the gaze estimation pipeline: identifying different regions of interest (e.g., pupil, iris, sclera, skin) in close-up images of the eye. Such *semantic segmentation* of these regions enables the extraction of region-specific features (e.g., iridial feature tracking [70]) and mathematical models which summarize the region structures (e.g., iris ellipse [57, 63, 77], or pupil ellipse [103]) used to derive a measure of gaze orientation.

**The major contributions of this paper are as follows:**

1. We present RITnet, a semantic segmentation architecture that obtains state-of-the-art results on the 2019 OpenEDS Semantic Segmentation Challenge with model size of **only 0.98 MB**. Our model performs segmentation at 301 *Hz* for  $640 \times 400$  images on an NVIDIA 1080Ti GPU.
2. We propose domain-specific augmentation schemes which help in generalization under a variety of challenging conditions.
3. We present boundary aware loss functions with a loss scheduling strategy to train Deep Semantic Segmentation models. This helps in producing coherent regions with crisp region boundaries.

### 5.3 Previous Works

Recently developed solutions for end-to-end segmentation involve using Deep CNNs to produce a labeled output irrespective of the size of the input image. Such architectures consist of convolution layers with a series of down-sampling followed by progressive up-sampling layers. Downsampling operations strip away finer information that is crucial for accurate pixel-level semantic masks. This limitation was mitigated by Ronneberger et al. by introducing skip-connections between the encoder and decoder [7]. Jergou et al. proposed TiramisuNet [72], a progression of dense blocks [126] with skip connections between the up- and down-sampling pathways. TiramisuNet demonstrated reuse of previously computed feature maps to minimize the required number of parameters. Dangi et al. proposed the DenseUNet-K architecture [127] for image-to-image translation based on simplified dense connected feature maps with skip connections. The RITnet model presented in this paper is based on the DenseUNet-K architecture<sup>3</sup>.

---

<sup>3</sup><https://github.com/ShusilDangi/DenseUNet-K>



## 5.4 Proposed Model: RITnet

Recently, segmentation models based on Fully Convolutional Networks (FCN) have performed well across many datasets [72, 7]. That success, however, often comes at the cost of computational complexity, restricting their feasibility for real-time applications where rapid computation and robustness to illumination conditions is paramount [8]. In contrast, RITnet has 248,900 trainable parameters which require less than 1MB storage with 32-bit precision (see Figure 5.2) and has been benchmarked at over 300  $Hz$ .

RITnet has five *Down*-Blocks and four *Up*-Blocks which downsample and upsample the input. The last *Down*-Block is also referred to as the *bottleneck* layer which reduces the overall information into a small tensor  $1/16^{th}$  of the input resolution. Each *Down*-Block consists of five convolution layers with LeakyReLU activation. All convolution layers share connections from previous layers inspired by DenseNet [126]. We maintain a constant channel size as in DenseUNet-K CITE with  $K = 32$  channels to reduce the number of parameters. All *Down*-Blocks are followed by an average pooling layer of size  $2 \times 2$ . The *Up*-Block layer upsamples its input by a factor of two using the nearest neighbor approach. Each *Up*-Block consists of four convolution layers with LeakyReLU activation. All *Up*-Blocks receive extra information from their corresponding *Down*-Block via skip connections, an effective strategy which provides the model with representations of varying spatial granularity.

### 5.4.1 Loss functions

Each pixel is classified into one of four semantic categories: *background*, *iris*, *sclera*, or *pupil*. Standard cross-entropy loss (CEL) is the default choice for applications with a balanced class distribution. However, there exists an imbalanced distribution of classes with the fewest pixels representing pupil regions. While CEL aims to maximize the output probability at a pixel location, it remains agnostic to the structure inherent to eye images. To mitigate these issues, we implemented the following loss functions:

**Generalized Dice Loss (GDL):** Dice score coefficient measures the overlap between the ground truth pixel and their predicted values. In cases of class imbalance [73], weighting the dice score by the squared inverse of class frequency [117] showed increased performance when combined with CEL.

**Boundary Aware Loss (BAL):** Semantic boundaries separate regions based on class labels. Weighting the loss for each pixel by its distance to the two nearest segments introduces edge awareness [7]. We generate boundary pixels using a Canny edge detector which are further dilated by two pixels to minimize confusion at the boundary. We use these edges to mask the CEL.

**Surface Loss (SL):** SL is based on a distance metric in the space of image contours

which preserves small, infrequent structures of high semantic value [128]. BAL attempts to maximize the correct pixel probabilities near boundaries while GDL provides stable gradients for imbalanced conditions. Contrary to both, SL scales the loss at each pixel based on its distance from the ground truth boundary for each class. It is effective in recovering smaller regions which are ignored by region based losses [128].

The total loss  $\mathcal{L}$  is given by a weighted combination of these losses as  $\mathcal{L} = \mathcal{L}_{CEL}(\lambda_1 + \lambda_2 \mathcal{L}_{BAL}) + \lambda_3 \mathcal{L}_{GDL} + \lambda_4 \mathcal{L}_{SL}$ .

## 5.5 Experimental Details

### 5.5.1 Dataset and Evaluation

We train and evaluate our model on the OpenEDS Semantic Segmentation dataset [8] consisting of 12,759 images split into *train* (8,916), *validation* (2,403) and *test* (1,440) subsets. Each image had been hand annotated with four semantic labels; *background*, *sclera*, *pupil*, & *iris*.

Per OpenEDS challenge guidelines, our *overall score* metric uses the average of the mean Intersection over Union (mIoU) metric for all classes and model size (S) calculated as a function of number of trainable parameters in megabytes (MB). The *overall score* is given as  $\frac{mIoU + \min(1/S, 1)}{2}$ .

### 5.5.2 Training

We trained our model using Adam [118] with a learning rate of 0.001 and a batch size of 8 images for 175 epochs on a TITAN 1080 Ti GPU. We reduced the learning rate by a factor of 10 when the validation loss plateaued for more than 5 epochs. The selected model with the best validation score was found at the 151<sup>st</sup> epoch. In our experiments, we used  $\lambda_1 = 1, \lambda_2 = 20, \lambda_3 = (1 - \alpha)$  and  $\lambda_4 = \alpha$ , where  $\alpha = epoch/125$  for  $epoch < 125$  otherwise 0. This loss scheduling scheme gives prominence to GDL during initial iterations until a steady state is achieved, following which SL begins penalizing stray patches.

### 5.5.3 Data Pre-processing

To accommodate variation in individual reflectance properties (e.g., iris pigmentation, eye makeup, skin tone or eyelids/eyelashes) [8] and HMD specific illumination (the position of infrared LEDs with respect to the eye), we performed two pre-processing steps. These steps were based on the difference in the train, validation and test distributions of mean image brightness (Figure 11 in Garbin et al. [8]). Pre-processing reduced these differences and also increased separability of certain eye features. First, a fixed gamma correction

with an exponent of 0.8 was applied to all input images. Second, we applied local Contrast Limited Adaptive Histogram Equalization (CLAHE) with a grid size of  $8 \times 8$  and clip limit value of 1.5 [129]. Figure 5.3 shows an image before and after pre-processing.

To increase the robustness of the model to variations in image properties, training data was augmented with the following modifications:

- Reflection about the vertical axis.
- Gaussian blur with a fixed kernel size of  $7 \times 7$  and standard deviation  $2 \leq \sigma \leq 7$ .
- Image translation of 0-20 pixels in both axes.
- Image corruption using 2-9 thin lines drawn around a random center ( $120 < x < 280, 192 < y < 448$ )
- Image corruption with a structured *starburst* pattern (Figure 5.4) to reduce segmentation errors caused by reflections from the IR illuminators on eyeglasses. Note that the *starburst* image is translated by 0-40 pixels in both directions.

Each image received at least one of the above-mentioned augmentations with a probability of 0.2 on each iteration. The probability that an image would be flipped horizontally was 0.5.

## 5.6 Results

We compare our results against SegNet [8], another fully convolutional encoder-decoder architecture. mSegNet refers to the modified SegNet with four layers of encoder and decoder. mSegNet w/BR refers to mSegNet with Boundary Refinement as residual structure and mSegNet w/SC is a lightweight mSegNet with depthwise separable convolutions [8]. As shown in Table 5.1, our model achieves a  $\sim 6\%$  improvement in mIoU score while the complexity is reduced by  $\sim 38\%$  compared to the baseline model mSegNet w/SC. However, our model’s segmentation quality was impacted at higher values of motion blur and image defocus (Figure 5.5), Figure 5.1 demonstrates that our model generalizes to some challenging cases where other models fail to produce coherent results.

## 5.7 Discussion

Our model achieves state-of-the-art performance with a small model footprint. The final architecture was arrived at after exploring a number of architectural variations. Reducing the channel size from 32 to 24 and increasing the number of convolution layers in the

| Model               | Mean F1     | mIoU        | Model Size (S) | No. of parameters (million) | Overall Score |
|---------------------|-------------|-------------|----------------|-----------------------------|---------------|
| mSegNet*            | 97.9        | 90.7        | 13.3           | 3.5                         | 0.491         |
| mSegNet*<br>w/BR    | 98.3        | 91.4        | 13.3           | 3.5                         | 0.495         |
| mSegNet*<br>w/SC(B) | 97.4        | 89.5        | 1.6            | 0.4                         | 0.762         |
| <b>Ours</b>         | <b>99.3</b> | <b>95.3</b> | <b>0.98</b>    | <b>0.25</b>                 | <b>0.976</b>  |

Table 5.1: Performance comparison on the test split of the OpenEDS dataset. The metrics and comparison models (\*) are used as reported in [8].

*Down*-Block did not affect the results. Surprisingly, increasing the channel size to 40 and removing one convolutional layer in the *Down*-Block degraded performance, resulting in spurious patches in output regions. Performance was influenced by the choice of loss functions and the adjustment of their relative weights. By setting the boundary-aware loss at a relatively higher weight, we observed sharp boundary edges and consequently improved our test mIoU from 94.8% to 95.3%.

We speculate that some aspects of our model were successful because they accounted for labeling artifacts in the openEDS dataset. For example, although pupil-to-iris boundaries were defined using ellipse fits to multiple points selected on the boundaries [8], sclera-to-eyelid boundaries were created using a linear fit between adjacent points marked on the eyelids. It is perhaps for this reason that the use of nearest-neighbor interpolation outperformed bilinear interpolation in the process of upsampling. Although the smoother curves that result from bilinear interpolation resulted in more accurate detection of the iris and pupil, it was less accurate in segmentation of the sclera.

Finally, data preprocessing had a significant impact on model performance. Introduction of CLAHE and gamma correction resulted in an overall improvement of 0.2% in the validation mIoU score. Augmentation helped in noisy cases such as reflections from eyeglasses, varying contrast, eye makeup, and other image distortions.

## 5.8 Conclusion

We designed a computationally efficient model for the segmentation of eye images. We also presented methods for implementing multiple loss functions that can tackle class imbalance and ensures crisp semantic boundaries. We showed several methods for incorporating pre-processing and augmentation techniques that can help mitigate against image distortions.

RITNet currently has the best results on the OpenEDS test set, has a model size under 1 MB, and achieves an impressive 301Hz on a NVIDIA 1080Ti.

## Acknowledgements

We thank Anjali Jogeshwar, Kishan KC, Zhizhuo Yang, and Sanketh Moudgalya for providing valuable input and feedback. We would also like to thank the Research Computing group at RIT for providing access to GPU clusters.

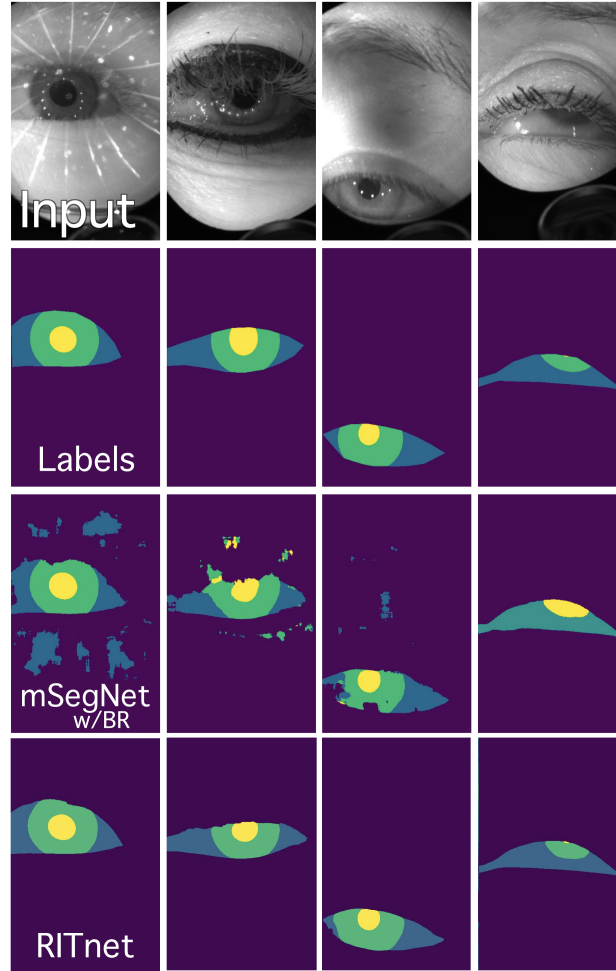


Figure 5.1: Comparison of model performance on difficult samples in the OpenEDS test-set. Top row, left to right, shows eyes obstructed due to prescription glasses, heavy mascara, dim light, and partial eyelid closure. Rows from top to bottom show input test images, ground truth labels, predictions from mSegNet w/BR [8] and predictions from RITnet, respectively. Compared to other methods, RITnet’s output more closely matches the ground truth.

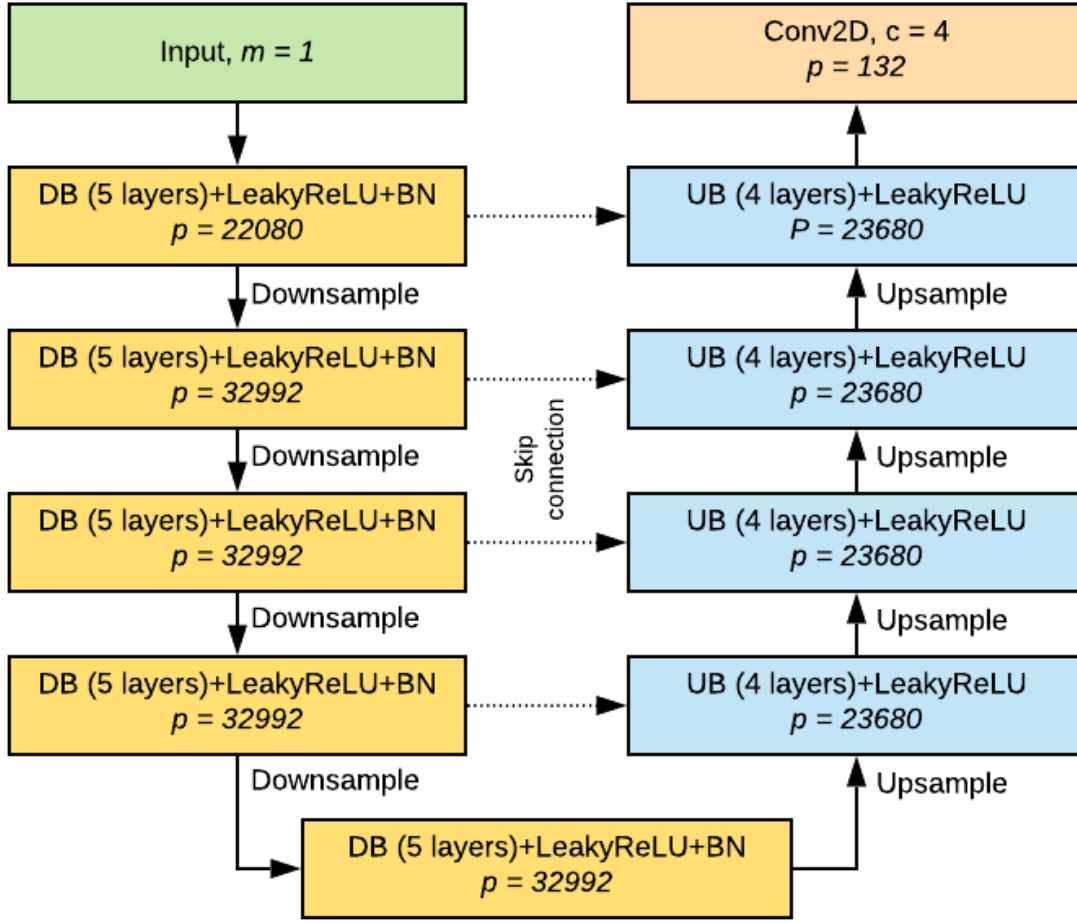


Figure 5.2: Architecture details of RITnet. DB refers to *Down-Block*, UB refers to *Up-Block*, and BN stands for batch normalization. Similarly,  $m$  refers to the number of input channels ( $m = 1$  for gray scale image),  $c$  refers to number of output labels and  $p$  refers to number of model parameters. Dashed lines denote the skip connections from the corresponding *Down-Blocks*. All of the Blocks output tensors of channel size  $m=32$ .

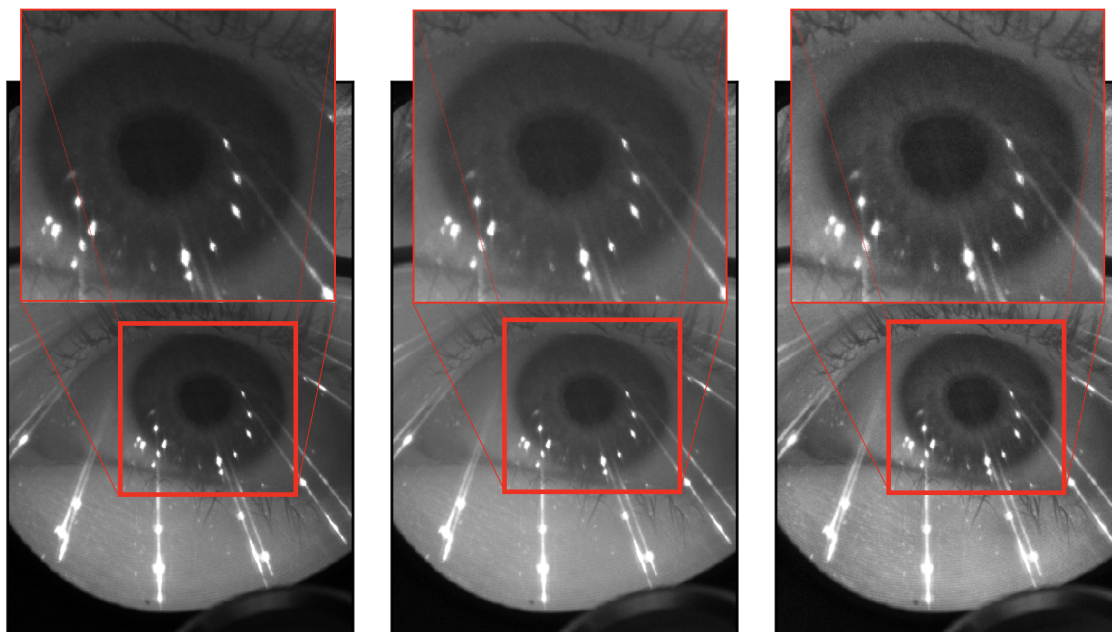


Figure 5.3: Left to right: Original image, image after gamma correction, and image after CLAHE is applied. Note that in the rightmost image, it is comparatively easier to distinguish iris and pupil.

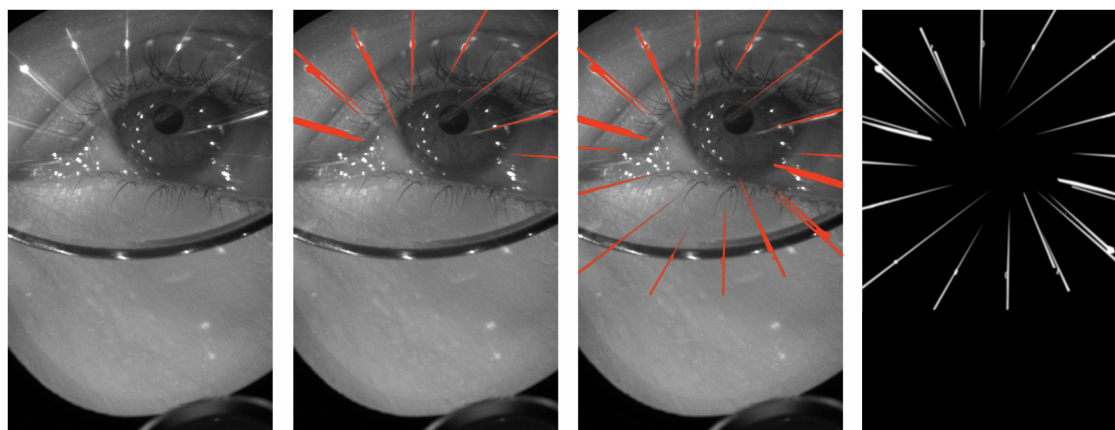


Figure 5.4: Generation of a *starburst* pattern from the training image 000000240768. Left to Right: Original image, selected reflections, concatenating with its  $180^\circ$  rotation, final pattern mask (best viewed in color).



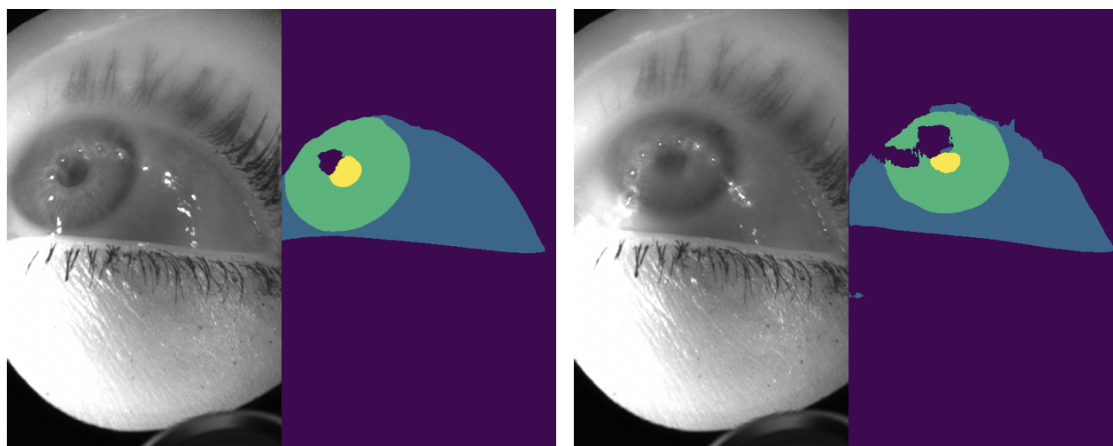


Figure 5.5: Our model struggles to do an accurate segmentation when eye masks are heavily blurred or defocused. Despite failure in segmentation, the segmentation output maps can be salvaged to produced plausible pupil or iris ellipse fits.

# Chapter 6

## EllSeg<sup>1</sup>

### 6.1 Abstract

Ellipse fitting, an essential component in pupil or iris tracking based video oculography, is performed on previously segmented eye parts generated using various computer vision techniques. Several factors, such as occlusions due to eyelid shape, camera position or eyelashes, frequently break ellipse fitting algorithms that rely on well-defined pupil or iris edge segments. In this work, we propose training a convolutional neural network to directly segment entire elliptical structures and demonstrate that such a framework is robust to occlusions and offers superior pupil and iris tracking performance (at least 10% and 24% increase in pupil and iris center detection rate respectively within a two-pixel error margin) compared to using standard eye parts segmentation for multiple publicly available synthetic segmentation datasets.

### 6.2 Introduction

There is great potential for the use of eye tracking in augmented and virtual reality (AR/VR) displays both as a means for user interaction, and for gaze-dependent rendering techniques that can both increase visual fidelity [130] while also lowering computational overhead [131]. Contemporary methods for eye tracking in VR and AR build upon techniques established in the context of head-mounted video-oculography, which involve the use of one or more infrared light sources placed next to infrared *eye* cameras. These eye cameras are pointed towards each of the wearer’s eyes while a third camera, referred to as the scene camera, points away from the wearer to capture the environment being

---

<sup>1</sup>This chapter appears in a published manuscript by Kothari *et al.* [10]

observed [132]. Existing solutions extract gaze descriptive features such as pupil center [32, 68, 59, 30, 133, 66], pupil ellipse [64, 134, 49, 135, 31], iris ellipse [57, 136, 137], or track iridial features [70, 138]. These solutions vary in algorithmic complexity, latency, and computational power requirements. Extracted features are then correlated to a measure of gaze using calibration routines [139, 4, 140], which compensate for person-specific physiological differences.

Despite many recent advances in eye-tracking technology [141], three factors continue to adversely impact the performance of eye-tracking algorithms: 1) reflections from the surroundings and from intervening optics, 2) occlusions due to eyelashes, eyelid shape, or camera placement and 3) small shifts of the eye-tracker position caused due to slippage [142]. Gaze estimation algorithms such as ExCuSe [32] and PuRe [133] which rely on hand-crafted features are particularly susceptible to stray reflections (unanticipated patterns on eye imagery) and occlusion of descriptive gaze regions (such as eyelid covering the pupil or iris). Recent appearance-based methods based on Convolutional Neural Networks (CNNs) are better able to extract reasonably reliable gaze features despite the presence of reflections [9] or occlusions [77]. Additionally, for head-mounted eye-tracking systems, the degradation of gaze estimate accuracy over time due to slippage [143] can be minimized by estimating the 3D eyeball center of rotation [144] (loosely referred at as an 'eyeball fit'). Estimating the precise physiology of the human eye is a complicated process and computationally intractable [145]. By making certain simplifying assumptions [51] about the human eye and its geometrical constraints, an estimate of a *reduced* optical eyeball model can be obtained from 2D pupil [134, 49, 103, 146] or iris [57, 136, 137] elliptical fits. These elliptical fits are derived from identified pupil and iris segments or outline [147]. Efforts by Chaudhary *et al.* [9] and Wu *et al.* [69] demonstrate that CNNs can precisely segment eye images into its constituent parts, *i.e.*, the pupil, iris, sclera and background skin regions.

In this work, we show that partially occluded pupil or iris regions can result in imprecise or degenerate elliptical fits. To mitigate this, we provide a solution, called *EllSeg*, which is made robust to occlusion by training CNNs to predict entire elliptical eye regions (the full pupil and the full iris) along with the remaining background, as opposed to the standard visible eye-parts segmentation (PartSeg) (see Figure 6.1). Additionally, we demonstrate that this approach enables us to train segmentation-based CNN architectures directly on datasets wherein only the pupil centers are available [6, 68, 59], allowing us to combine eye parts segmentation and pupil center estimation into a common framework.

The summary of our contributions are as follows:

1. We propose EllSeg, a framework that can be utilized with any encoder-decoder architecture for pupil and iris ellipse segmentation. EllSeg enables prediction of the pupil and iris as full elliptical structures despite the presence of occlusions.

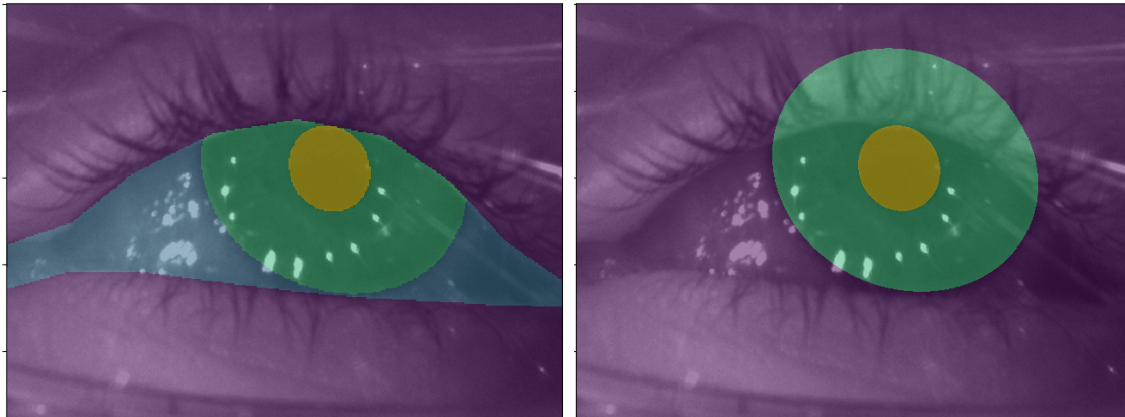


Figure 6.1: *PartSeg vs EllSeg*. Left: A *Four-class* eye part segmentation at the pixel-level (i.e. PartSeg) produces labelled pupil (yellow), iris (green), sclera (blue) and background (purple) classes. Right: The EllSeg (three-class) modification produces labelled pupil (yellow) and iris (green) elliptical regions and the rest is marked as background (purple).

2. To establish the utility of our methodology, we rigorously test our proposed 3-class ellipse segmentation framework using three network architectures, a modified Dense Fully Connected Network [72] (referred as DenseElNet), RITnet [9] and DeepVOG [31]. Performance is benchmarked with well defined train and test splits on multiple datasets, including some which are limited to labelled pupil centers only.

### 6.3 Related work

This work is primarily based on the observation that CNNs can identify which category a pixel belongs to despite conflicting appearance (*e.g.* accurately predicting a pixel as belonging to the pupil despite being occluded by eyelids or glasses). Successful segmentation in the presence of ambiguous appearance indicates that a CNN can reason over a wide range of inter-pixel spatial relationships while precise segmentation boundaries indicate successful utilization of fine-grained, high-frequency content observed in local neighborhoods. This ability to capture local information with a global context is achieved by repeatedly pooling intermediate outputs of convolutional operations within a neural network [148]. While numerous architectures can produce a “one-to-one” mapping between an image pixel and its segmentation output class, specific architectures rely on encoding an input image to low dimensional representation followed by decoding and up-sampling to a segmentation map - aptly named encoder-decoder architectures.

Researchers have demonstrated promising results using encoder-decoder architectures for image segmentation. For example, Chaudhary *et al.* [9] proposed RITnet, a lightweight architecture which leverages feature reuse and fixed channel size to maintain low model complexity while demonstrating state of the art performance on the OpenEDS dataset [8]. In this work, we designed our own encoder-decoder architecture called *DenseElNet* which incorporates the dense block proposed by RITnet while leveraging residual connections across each block as proposed by Jegou *et al.* [72]. This ensures a healthy gradient flow and faster convergence while mitigating the vanishing gradient problem [149, 150]. Similar to common encoder-decoder architectures, DenseElNet reduces the spatial extent of its input image but increases the channel size. Note that DenseElNet does not offer any particular novelty over existing encoder-decoder architectures. It is simply being used to facilitate testing of our EllSeg framework.

The primary purpose of eye image segmentation, in the context of gaze estimation, is to produce reliable ellipse fits. The DeepVOG framework by Yiu *et al.* [31] utilizes the U-net architecture [7] to segment the pupil followed by an out-of-network ellipse fitting procedure to generate a 3D model using the "two circles" approach [49, 54]. A limitation of their approach is that they segment the pupil based solely on appearance which would likely suffer from occlusion as described previously. Fuhl *et al.* [63] demonstrated that ellipse parameters can be regressed using the bottleneck representation of an input image. However they do not report any metrics for ellipse fit quality. Wu *et al.* [69] leverage multiple decoders to segment an image and estimate 2D cornea and pupil center. Multiple decoders may increase computational requirements and introduce bottlenecks in the pipeline by operating on redundant information. In contrast, we show that the iris and pupil ellipse can be generated using a single encoder-decoder forward pass.

## 6.4 Methodology

Figure 6.2 highlights the EllSeg framework on any generic encoder-decoder (E-D) architecture. First, an input image  $I \in \mathbb{R}$  is passed through an encoder to produce a bottleneck representation  $Z$  such that  $Z = E(I)$ . In our implementation of DenseElNet,  $I$  is down-sampled four times by a factor 2 at the bottleneck layer. Subsequently, the network segmentation output  $O$  is given by  $O = D(Z)$  and consists of three channels (background  $O_{bg}$ , iris  $O_{ir}$  and pupil  $O_{pl}$  output maps). Note that the segmentation outputs are also used to derive pupil and iris ellipse centers. The pupil and iris centers, along with the remaining ellipse parameters (axes and orientation), are also regressed from this bottleneck representation  $Z$  using a series of convolutional layers followed by a flattening operation and mapped to a ten-dimensional output (5 parameters for both the iris and pupil ellipses). Please refer to Figure 6.3 for the ellipse regression module architecture. We test

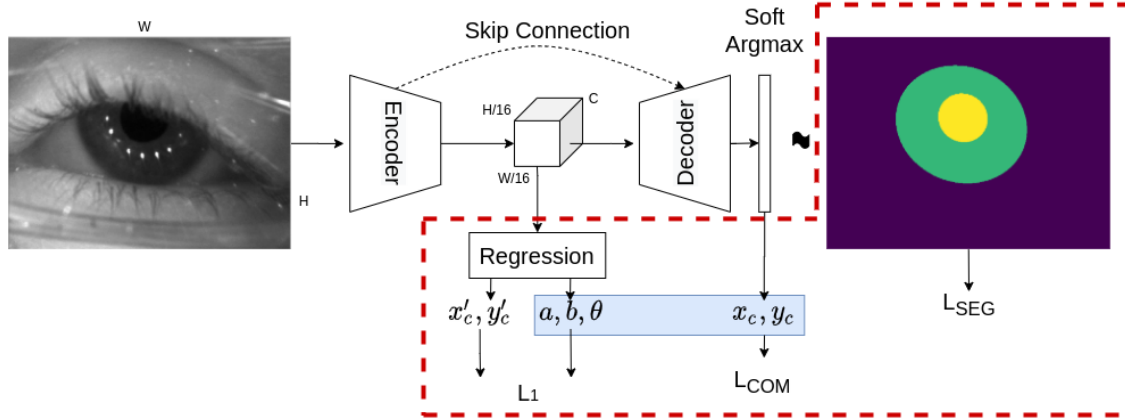


Figure 6.2: Proposed EllSeg framework (region enclosed by red dotted line) builds upon existing CNN-based approaches to facilitate the simultaneous segmentation and ellipse prediction for both iris and pupil regions. The resulting ellipse parameters are highlighted in the blue box.

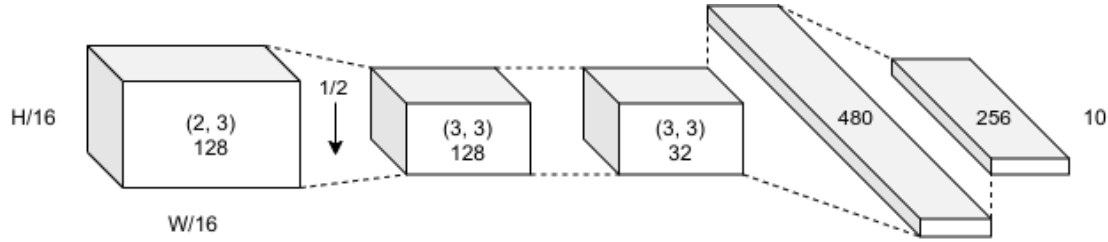


Figure 6.3: Regression module architecture. The  $\downarrow$  signifies average pooling to  $1/2$  the resolution. Tensors are flattened after three convolutional layers and passed through two linear layers before regressing 10 values (5 ellipse parameters for pupil and iris each).

the effectiveness of EllSeg framework on three architectures, DenseElNet (2.18M parameters), RITnet (0.25M parameters), and DeepVOG (3.71M parameters). Note that the regression module is trained alongside the entire network in an end-end fashion.

#### 6.4.1 Ellipse center

The center of any convex shape can be described as a weighted summation of its spatial extent (see Equation 6.1). In this context, *spatial extent* refers to all possible pixel coordinates while *weight* refers to the probability estimate of a pixel being within the convex

structure.

$$x_c^k, y_c^k = \sum_{i=1}^W \sum_{j=1}^H p_{<i,j>}^k x, \quad \sum_{i=1}^W \sum_{j=1}^H p_{<i,j>}^k y, \quad p_{<i,j>}^k \subset \mathbb{R} \quad (6.1)$$

Here,  $x_c^k$  and  $y_c^k$  correspond to the center of a particular feature class  $k$  (such as pupil). The iterators  $i$  and  $j$  span across the width  $W$  and height  $H$  of an image. The probability values  $p^k$  for each pixel are derived after a scaled, spatial softmax operation [151]:

$$p^k = \frac{\exp(\beta O_{<i,j>}^k)}{\sum_{i,j=1}^{W,H} \exp(\beta O_{<i,j>}^k)} \quad (6.2)$$

Here,  $\beta$  is a control parameter (also known as temperature [152]), which scales network output around the largest value. We empirically set  $\beta$  as 4. This formulation of ellipse center gives rise to several advantages offered by EllSeg over PartSeg discussed in Section 6.4.3 and Section 6.7.3.

While one may trivially estimate the pupil center in this manner, deriving the iris center is not straightforward due to its placement *within* the pupil. One alternative is to sum the pupil and iris activation maps before spatial softmax. However, this incorrectly results in the predicted pupil and iris sharing the same 2D center which is physiologically improbable as the pupil is not usually perfectly centered within the iris [153]. Instead, we propose leveraging the background class to predict the iris center in our 3 class segmentation framework. Encoder-decoder architectures have shown to perform exceedingly well at identifying "background" class pixels (see Supplementary Table 1 in Nair *et al.* [80] and Table 2 in Wu *et al.* [69]). To derive the iris center, we negate the background class output map in Equation 6.2, a modification which subsequently leads to an inverted peak at the predicted iris center location. This inversion ensures the background probability scores do not affect segmentation based loss functions (see Section 6.7.3).

### 6.4.2 Ellipse axis and orientation

The bottleneck representation  $Z$  is a low dimensional latent representation of the input image. This convenient representation enables us to regress parameters such as the ellipse axis and orientation (we use  $L_1$  loss in our implementation). Experiments revealed that regressing the pupil and iris centers does not offer sub-pixel accuracy (see Section 6.7.4) as opposed to deriving them from segmentation output as described in the previous section.

### 6.4.3 Loss functions

#### Segmentation losses $\mathcal{L}_{SEG}$

In the EllSeg framework, the network output  $O$  is primarily used to segment an eye image into pupil and iris ellipses, and the background (which includes scleral regions). To train such an architecture, we use the combination of loss functions proposed in RITnet [70]. This strategy involves using a weighted combination of four loss functions; cross-entropy loss,  $\mathcal{L}_{CEL}$ , generalized dice loss [117]  $\mathcal{L}_{GDL}$ , boundary aware loss  $\mathcal{L}_{BAL}$  and surface loss [128]  $\mathcal{L}_{SL}$ .

The total loss  $\mathcal{L}$  is given by a weighted combination of these losses as  $\mathcal{L}_{SEG} = \mathcal{L}_{CEL}(\lambda_1 + \lambda_2\mathcal{L}_{BAL}) + \lambda_3\mathcal{L}_{GDL} + \lambda_4\mathcal{L}_{SL}$ . In our experiments, we used  $\lambda_1 = 1$ ,  $\lambda_2 = 20$ ,  $\lambda_3 = (1 - \alpha)$  and  $\lambda_4 = \alpha$ , where  $\alpha = epoch/M$  and  $M$  is the number of epochs.

#### Center of Mass loss $\mathcal{L}_{COM}$

The  $L_1$  loss function is used to formulate an error function between the center of mass, *i.e.*, the pupil and iris ellipse centers from the segmentation output maps, to their respective ground-truth centers. This enables us to leverage datasets such as ElSe [68], PupilNet [59] and LPW [6] in a segmentation framework where only the ground-truth pupil center is available. Note that COM  $L_1$  loss (henceforth referred to as  $\mathcal{L}_{COM}$  loss) does not impede segmentation loss functions, but instead conditions the network output to jointly satisfy all loss functions. This results in the characteristic peaks observed in Section 6.7.3. The inversion of the background class results in an inverted peak at the iris center location.

## 6.5 Datasets

Combining segmentation and  $\mathcal{L}_{COM}$  losses allows the EllSeg framework to train CNNs on a large number of datasets (to the best of our knowledge, it enables the inclusion of all publicly available near-eye datasets). To demonstrate the utility of EllSeg, we choose the following datasets for our experiments: NVGaze [66], OpenEDS [8], RITEyes, ElSe [68], PupilNet [154] and LPW [6]. For more details about each dataset, available ground-truth modality, and train/test splits, please refer to Table 6.1. Note that we specifically leverage the S-General dataset from the RIT-Eyes framework [80] as it offers wide spatial distribution of *eye* camera position.

### 6.5.1 Groundtruth ellipse fits

To obtain groundtruth pupil and iris ellipse fits from the selected datasets, pupil and limbus edges are extracted from groundtruth segmentation masks using a canny edge detector.



Table 6.1: Summary of datasets.  $\uparrow$  and  $\downarrow$  correspond to up and down sampling respectively. OpenEDS image crops are extracted around the scleral center followed by up-sampling. Note that images without valid pupil and iris fits are discarded (see Section 6.5).

| Dataset         | Resolution | Train subset  | Test subset  | Groundtruth included | Image Count (train, test) | Preprocess                        |
|-----------------|------------|---|--|----------------------|---------------------------|-----------------------------------|
| NVGaze          | 1280×960   | male 1-4<br>female 1-4                                  | male 5<br>female 5                                 | All                  | 15623, 3895               | $\downarrow 4$                    |
| OpenEDS 2019    | 400×640    | OpenEDS <sup>19</sup><br>train                          | OpenEDS <sup>19</sup><br>valid                     | PartSeg              | 8826, 2376                | Crop to 400×300<br>$\uparrow 1.6$ |
| RITEyes General | 640×480    | Avatars 1-18  | Avatars 19-24                                      | All                  | 33997, 11519              | $\downarrow 2$                    |
| LPW             | 640×480    | Subjects 1-16   | Subjects 17-22                                     | Pupil center         | 93127, 33388              | $\downarrow 2$                    |
| ElSe            | 384×288    | I, III, VI, VIII, IX, XI, XIII, XV, XVII, XIX, XX, XXII | II, IV, V, VII, X, XII, XIV, XVI XVIII, XXI, XXIII | Pupil center         | 60079, 33846              | $\uparrow 5/3$                    |
| PupilNet        | 384×288    | I, III, V   | II, IV   | Pupil center         | 25471, 15707              | $\uparrow 5/3$                    |

To ensure subpixel accuracy, we consider edge pixels in the inverted mask as well. Edge pixels which satisfy pupil-iris (*i.e.*, no neighboring sclera or background pixel) or limbus (*i.e.*, no neighboring pupil or background pixel) conditions are used to determine ellipse parameters using the ElliFit algorithm [155] (see Figure 6.4). Random Sample Consensus (RANSAC) [156] is employed to remove outliers. While datasets such as RITEyes and NVGaze directly offer EllSeg compatible groundtruth semantic masks, synthetic masks for OpenEDS were generated based on elliptical fits. Images without valid pupil or iris fits (117 out of 11319) were discarded from all subsequent analysis.

## 6.6 Experiments and Hypothesis

We rigorously test various hypotheses to validate the efficacy of our proposed methodology in the field of eye-tracking. In the first experiment (Section 6.7.1), we benchmark the segmentation performance of our network, DenseElNet, on the standard PartSeg framework. Comparable or superior performance on the PartSeg task will validate DenseElNet. In the second experiment (Section 6.7.2), we test whether the EllSeg framework improves

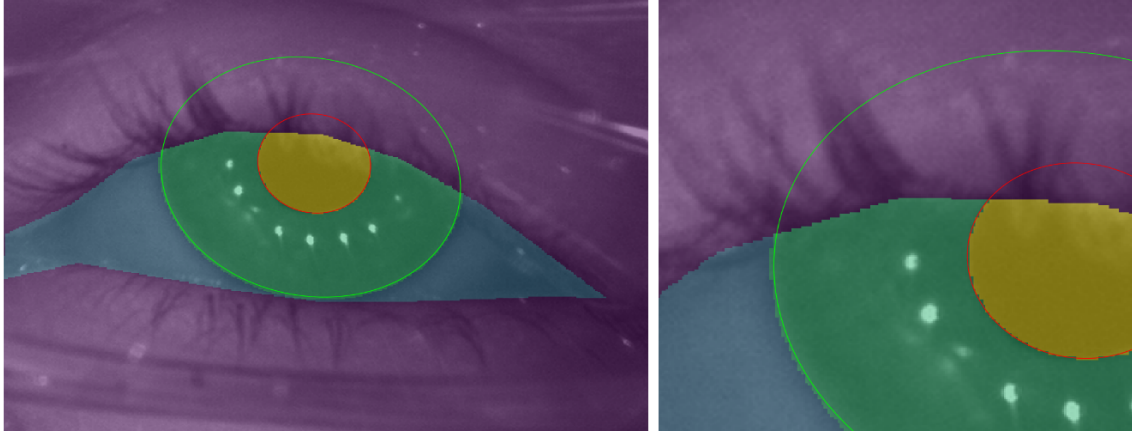


Figure 6.4: Ellipse fitting quality on ground truth PartSeg masks. These fits are further used to generate EllSeg masks for the OpenEDS dataset.

the detection of both pupil and iris estimates over its PartSeg counterpart. Finally, in the third experiment (Section 6.7.3), we compare the results of regressing elliptical parameters in the EllSeg framework to those found when estimating the ellipse parameters using RANSAC. This experiment will test whether reliable and differentiable ellipses can be directly estimated in an encoder-decoder architecture. Summary of all the experiments can be found in Figure 6.5.

### 6.6.1 Training

To ensure fair comparison, all CNN architectures are trained and evaluated with identical train/validation/test splits. The training set is divided into a 80/20 % train/validation split. Sample selection is stratified based on binned 2D pupil center position and subsets present within each dataset (see Table 6.1). This approach ensures that biases introduced due to sampling are minimized while maintaining similar statistical distributions across training and validation sets. Bins with fewer than five images are automatically discarded. All architectures are trained using ADAM optimization [118] on a batch of 48 images at 320x240 resolution with a learning rate of  $5 \times 10^{-4}$  on an NVIDIA V100 GPU.

During training, all models were evaluated with the metric:  $[4 + \text{mIoU} - 0.0025(d_p + d_i) - (\theta_p + \theta_i)/90^\circ]$ , where mIoU corresponds to the mean intersection over union (IoU) [157] score which quantifies segmentation performance,  $d_p$  &  $d_i$  are the distances between pupil and iris centers from their groundtruth values in pixels, and  $\theta_p$  &  $\theta_i$  are the angular error between the predicted and groundtruth ellipse orientations in degrees. If no improvement


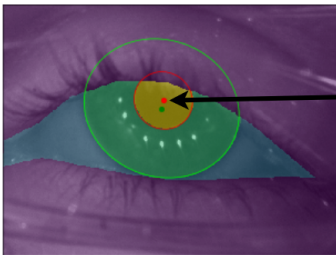
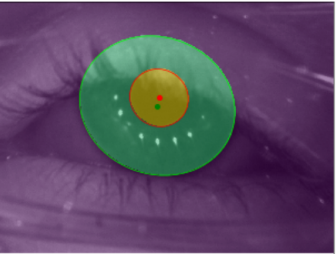
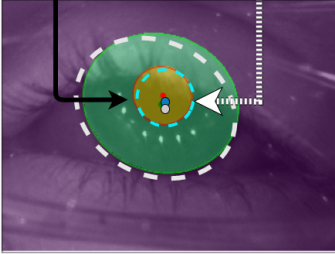
|         | (Exp 1) Benchmark of segmentation accuracy  | (Exp 2) Accuracy of pupil/iris localization  | (Exp 3) Accuracy using Ellifit + RANSAC vs Regression on LCOM loss                  |
|---------|---|--|---|
| PartSeg |  |   | Based on ellipse fit<br>Regressed (dashed ellipse)                                  |
| EllSeg  |   |  |  |
| Metrics | IoU score   | Euclidean distance of pupil/iris centers   | IoU score, Euclidean distance, goodness of ellipse fit                              |
|         | Section 6.1   | Section 6.2  | Section 6.3   |

Figure 6.5: Summary of all experiments described in following sections (Center estimates are best viewed on screen).

above 0.001 was observed on this metric for ten consecutive epochs, then a network’s parameters were deemed converged. The learning rate was reduced by a factor of ten if no improvements were identified over five epochs. To reduce training time and ensure stable training on pupil-center-only datasets, all models were pretrained on NVGaze, OpenEDS and RIT-Eyes training sets for two epochs.

### 6.6.2 Data augmentation

To increase the robustness of models and avoid overfitting, training images were randomly augmented with the following procedures with equal probability (12.5%) of occurrence:

- Horizontal flips
- Image rotation up to  $\pm 30^\circ$
- Addition of Gaussian blur with  $2 \leq \sigma \leq 7$
- Random Gamma correction for  $\gamma = [0.6, 0.8, 1.2, 1.4]$
- Exposure offset up to  $\pm 25$  levels
- Gaussian noise with  $2 \leq \sigma \leq 16$
- Image corruption by masking out pixels along a four-pixel thick line
- No augmentation

### 6.6.3 Evaluation Metrics

All segmentation performance is evaluated by IoU scores. Ellipse center accuracy is reported as the Euclidean distance in pixel error from their respective groundtruth annotations. Additionally, pupil and iris detection rate [134], *i.e.*, the percentage of ellipse centers accurately identified within a range of pixels of the groundtruth center point is also reported.

As most gaze estimation algorithms rely on ellipse fitting on the segmented pupil and/or iris, we quantify elliptical goodness of fit with metrics that effectively capture ellipse offset, orientation errors and scaling errors. In this work, we utilize a bounding box overlap IoU metric that accounts for all ellipse parameters: center, axes, and orientation. For each defined elliptical structure, an enclosing bounding box is generated. IoU scores are obtained from a comparison between groundtruth and predicted bounding boxes (Figure 6.6). Note that the orientation error (difference in ellipse orientation) of the fits is calculated for images in which the ratio of major to minor axis length exceeded 1.1 - this avoids large artifacts when elliptical fits are nearly circular.

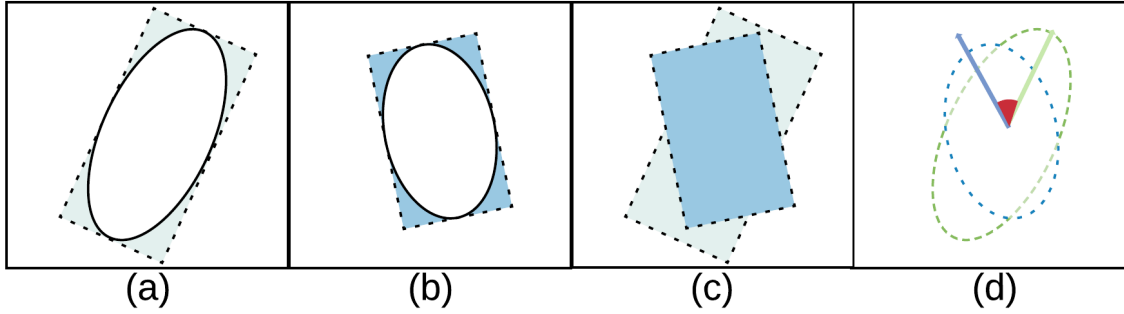


Figure 6.6: Visualization of goodness of fit metrics used in the paper. (a) Groundtruth ellipse (pupil or iris). (b) Corresponding predicted ellipse. The rectangular boxes denote ellipse-axis-aligned bounding boxes for the respective ellipses. (c) denotes the bounding box overlap region and (d) illustrates the angular difference between the two ellipses.

## 6.7 Results and Discussion

### 6.7.1 Comparison with state-of-the-art models

The DenseElNet architecture is a hybrid of RITnet and TiramisuNet, and has 2.18M parameters. We also explore the alternative possibility of utilizing other state-of-the-art encoder-decoder architectures like DeepVOG and RITNet. DeepVOG, with 3.71M parameters, segments images into two classes; *pupil* and *background*, *i.e.*, (non-pupil). RITnet, with 0.25M parameters, defines four classes; *pupil*, *iris*, *sclera*, and *background* (other). Table 6.2 highlights that both RITnet and DenseElNet models outperform DeepVOG on every dataset. Table 6.2 also demonstrates that the performance of DenseElNet and RITnet are comparable ( $< 2\%$  difference) on all datasets despite varying model complexity.

### 6.7.2 Ellipse center estimation

In this section, we explore the usefulness of the full ellipse segmentation (EllSeg) over the traditional eye parts segmentation (PartSeg) by comparing the pupil/iris center detection rates. We train three network architectures; RITnet, DeepVOG, and DenseElNet both with  $\mathcal{L}_{SEG}$  loss functions using the following training scenarios:

- Traditional, four class PartSeg (referred as *RITnet-PartSeg*, *DeepVOG-PartSeg*, and *DenseElNet-PartSeg*)
- 3-class EllSeg (referred as *RITnet-EllSeg*, *DeepVOG-EllSeg*, and *DenseElNet-EllSeg*)

Table 6.2: Eye Parts Segmentation: Comparison of *pupil* (and *iris*, inside parenthesis) *class* IoU scores for RITnet, DeepVOG and DenseElNet model architectures (along rows) in OpenEDS, NVGaze and RIT-Eyes dataset (along columns). Bold values indicate the best performance within each dataset. Because DeepVOG was not trained to segment the iris, we are unable to provide iris IOU scores.

| Model      | OpenEDS            | NVGaze             | RIT-Eyes           |
|------------|--------------------|--------------------|--------------------|
| RITnet     | 95.0 (91.4)        | <b>93.2 (91.7)</b> | 89.5/94.4          |
| DeepVOG    | 89.1 (NA)          | 90.9 (NA)          | 83.5 (NA)          |
| DenseElNet | <b>95.4 (92.1)</b> | 93.1 (91.4)        | <b>91.5 (95.4)</b> |

Table 6.3: The percentage of images classified as three categories of occlusion (see Section 6.7.2) for each dataset. Values are presented as pupil (iris).

|         | Occluded   | Partial     | Visible     |
|---------|------------|-------------|-------------|
| OpenEDS | 0.0 (0.0)  | 1.5 (17.2)  | 98.5 (82.7) |
| NVGaze  | 2.3 (0.0)  | 14.8 (75.6) | 82.9 (24.4) |
| RITeyes | 9.5 (11.1) | 70.7 (22.3) | 19.8 (66.7) |

Note that, in this section, all ellipse centers are derived by utilizing ElliFit [155] along with RANSAC outlier removal on output segmentation maps.

Figure 6.7 presents the pupil/iris detection rate as a function of the error threshold (in pixels) for DeepVOG, RITnet, and DenseElNet, using both PartSeg and EllSeg frameworks. Although all models demonstrate similar performance when tested upon the OpenEDS dataset, models trained using the EllSeg framework demonstrate superior pupil and iris detection on the NVGaze and RIT-Eyes datasets.

Analysis of the ground truth imagery suggests that this difference may be attributed to the varying amounts of pupil/iris occlusion within each dataset. In order to verify this, we compute *occlusion magnitude*,  $O_m$ , which is defined as one minus the IoU of PartSeg and EllSeg ground truth maps. Based on this magnitude, each image is classified into 3 categories of occlusion (shown in Table 6.3) based on empirical thresholds, a) fully occluded ( $O_m \geq 0.7$ ) b) partially occluded ( $0.3 \leq O_m < 0.7$ ) and c) fully visible ( $O_m < 0.3$ ).

Dramatic improvements can be observed for the NVGaze and RITeyes datasets wherein a large percent of images demonstrate partially occluded iris or pupil. Since a smaller percent of images are occluded in the OpenEDS dataset, we observe a small but consistent improvement in the iris detection rate between 3-6 pixel error threshold (see Figure 6.7,

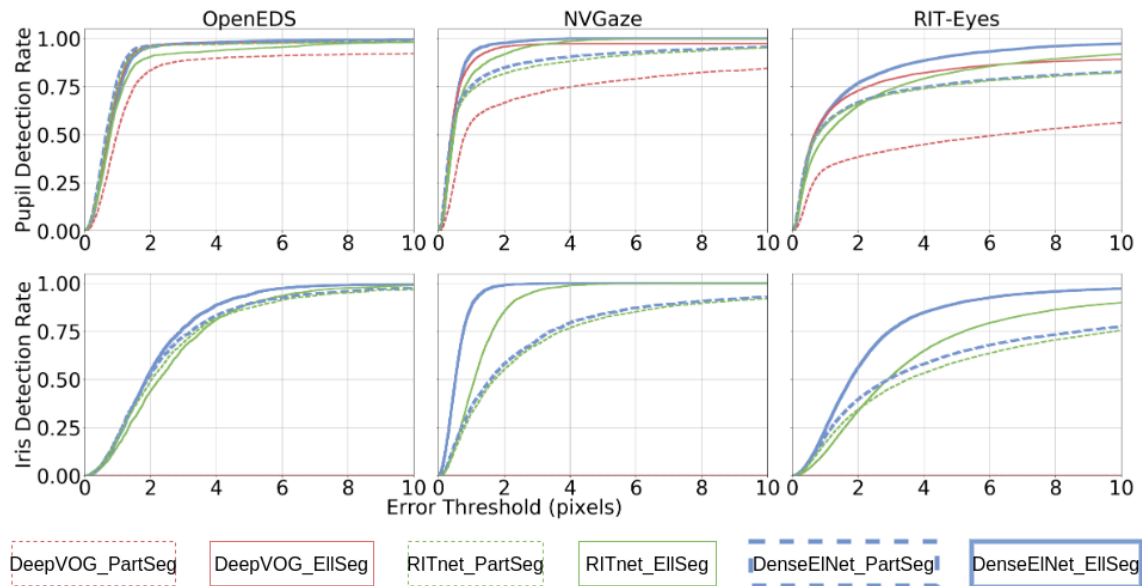


Figure 6.7: *PartSeg vs EllSeg*: The pupil detection rate (top row) and iris detection rate (bottom row) as a function of the threshold for tolerated pixel error for center approximation for OpenEDS (left column), NVGaze (middle column) and RIT-Eyes (right column). Results for three architectures RITnet, DeepVOG and DenseElNet are present for both cases PartSeg (dashed lines) and EllSeg (solid lines). Note that only the pupil detection rate is shown for the DeepVOG architecture. All detection rates presented here are derived using ellipse fits on segmentation outputs on images sized at  $320 \times 240$ . Here, one pixel error corresponds to 0.25% of the image diagonal length.

second row-first column). These results and subsequent analysis clearly demonstrate that EllSeg is robust to occlusions.

In addition to improving ellipse center estimates, Table 6.4 demonstrates that the EllSeg protocol reduces the number of images with invalid ellipse fits on the predicted segmentation output.

Table 6.4: The number of images without valid PartSeg or EllSeg ellipse fits for pupil (and iris, inside parenthesis) for DeepVOG, RITnet, and DenseElNet. The total column represents the number of valid images used for testing (as in section 6.5.1). Bold text (lower number) shows superior performance and illustrates the effectiveness of the EllSeg framework.

|         | Model    | Total | DeepVOG         | RITnet                  | DenseElNet            |
|---------|----------|-------|-----------------|-------------------------|-----------------------|
| PartSeg | OpenEDS  | 2376  | 17 (NA)         | 1 (0)                   | 2 (0)                 |
|         | NVGaze   | 3895  | 10 (NA)         | 0 (0)                   | 0 (0)                 |
|         | RIT-Eyes | 11519 | 1072 (NA)       | 287 (69)                | 353 (62)              |
| EllSeg  | OpenEDS  | 2376  | <b>6</b> (NA)   | 1 (0)                   | <b>0</b> (0)          |
|         | NVGaze   | 3895  | <b>0</b> (NA)   | 0 (0)                   | 0 (0)                 |
|         | RIT-Eyes | 11519 | <b>215</b> (NA) | <b>60</b> ( <b>18</b> ) | <b>1</b> ( <b>0</b> ) |

### 6.7.3 Improving the ellipse estimates

In this section, we analyze the impact of  $L_{COM}$  on segmentation output maps, ellipse shape parameters and ellipse center estimates.

Ellipse center estimates results are shown in Figure 6.8. All models (RITnet, DeepVOG and DenseElNet) are trained with the EllSeg framework *with* and *without*  $\mathcal{L}_{COM}$ . Ellipse centers *without*  $\mathcal{L}_{COM}$  loss are estimated using ElliFit on segmentation output maps. Models trained *with*  $\mathcal{L}_{COM}$  loss estimate their centers ( $x_c$  and  $y_c$ ) as shown in Figure 6.2.

Figure 6.8 also includes the results of non-CNN based algorithms ExCuSe [32], PuRe [133], and PuReST [30] which rely on filtered edges, morphological operations and hand-crafted features using computer-vision based methods. Note that none of these methods were designed for OpenEDS, NVGaze, or RITeyes datasets. To facilitate application, pixels with a ground truth label identifying them as a member of the "background" class are converted to a uniform grey (digital count=127). This step minimizes the chance of false detection of the pupil within the background, which is a common issue for images within the OpenEDS and NVGaze datasets, which have black regions in the periphery.



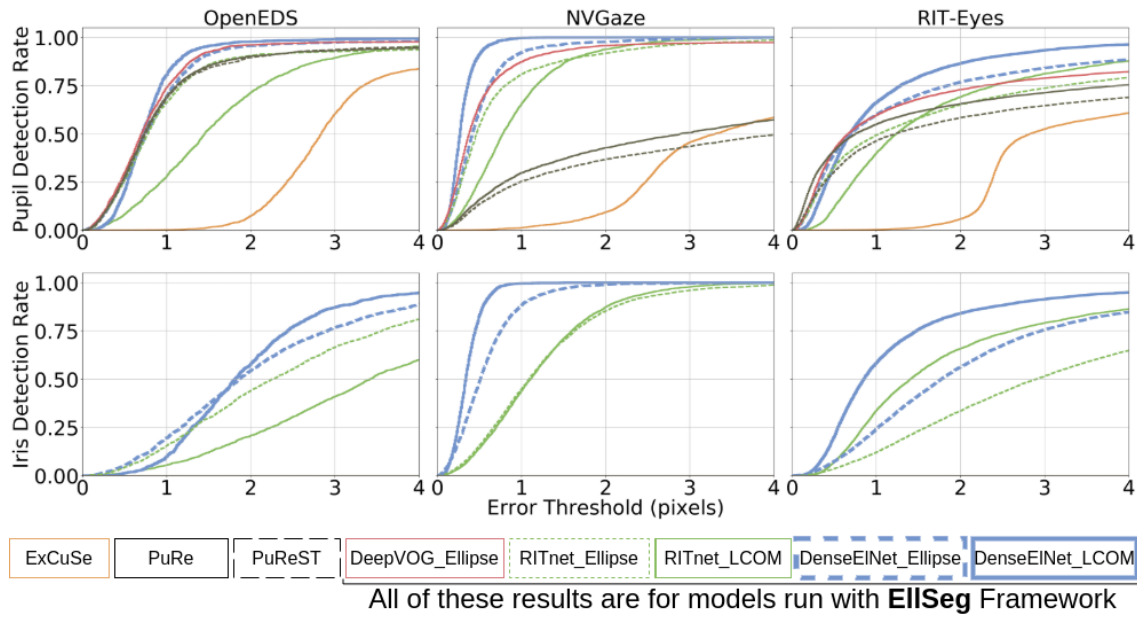


Figure 6.8: *EllSeg* with and without  $\mathcal{L}_{COM}$  loss: The pupil detection rate (top row) and iris detection rate (bottom row) for various pixel error thresholds of center approximation for three datasets. Models (RITnet, DenseElNet and DeepVOG) are trained with the EllSeg framework before the pupil center is estimated using either the ElliFit segmentation output map, or with  $\mathcal{L}_{COM}$  loss. The result for non-CNN based model ExCuSe, PuRe and PuReST are also shown. One pixel error corresponds to 0.25% of the image diagonal length.

Table 6.5: Comparison of Pupil center estimate errors (in pixels) on various datasets in terms of median scores. Note all the CNN models are trained with EllSeg framework. Image size is  $320 \times 240$ .

| Model    | RITnet      |                     | DenseElNet  |                     |
|----------|-------------|---------------------|-------------|---------------------|
| Method   | Ellipse fit | $\mathcal{L}_{COM}$ | Ellipse fit | $\mathcal{L}_{COM}$ |
| OpenEDS  | 0.8         | 1.5                 | 0.8         | 0.7                 |
| NVGaze   | 0.5         | 0.8                 | 0.4         | 0.3                 |
| RIT-Eyes | 1.0         | 1.2                 | 0.7         | 0.7                 |
| Fuhl     | -           | 73.4                | -           | 1.7                 |
| LPW      | -           | 4.7                 | -           | 0.8                 |
| PupilNet | -           | 77.6                | -           | 1.6                 |

Note that for ExCuSe, images are resized to the author-recommended size (384x288). The predicted center is then remapped to (320x240) to facilitate comparison. For PuRe and PuReST, the EyeRecTool [158] is used to compute pupil center using the original image size (320x240).

Figure 6.8 reveals that, although introduction of  $\mathcal{L}_{COM}$  often degraded the performance of RITnet, it improved performance for our model, (DenseElNet). Further, for pupil detection, the models trained using CNN outperforms all the non-CNNs based models ExCuSe, PuRe and PuReST.

Table 6.5 shows the comparison of median values of pupil center estimates *with* and *without*  $\mathcal{L}_{COM}$  loss in regards to both models RITnet and DenseElNet. There is a slight improvement in the median values in the DenseElNet model with the introduction of this loss function. However, for the RITnet model, the inclusion of  $\mathcal{L}_{COM}$  deteriorated the performance by 57%, 19%, and 19% for OpenEDS, NVGaze, and RIT-Eyes datasets respectively (within one-pixel error range for Pupil center). We suspect this behavior is due to the relatively limited channel size and low parameter count of RITnet when compared to DenseElNet.

The analyses presented up to this point focus on the accuracy of pupil/iris center estimates. However, many algorithms for gaze estimation rely on accurate estimation of pupil and iris ellipses for the construction of 3D geometric models of the oriented eye [31, 103, 49, 57]. This necessitates a quantitative measure for the goodness of an ellipse fit. The methodology presented in Section 6.6.3 and represented in Figure 6.6 is used to calculate the *boundary IOU* - a measure used to estimate the quality of boundary estimation. Boundary IoU was calculated for both the pupil and the iris after application

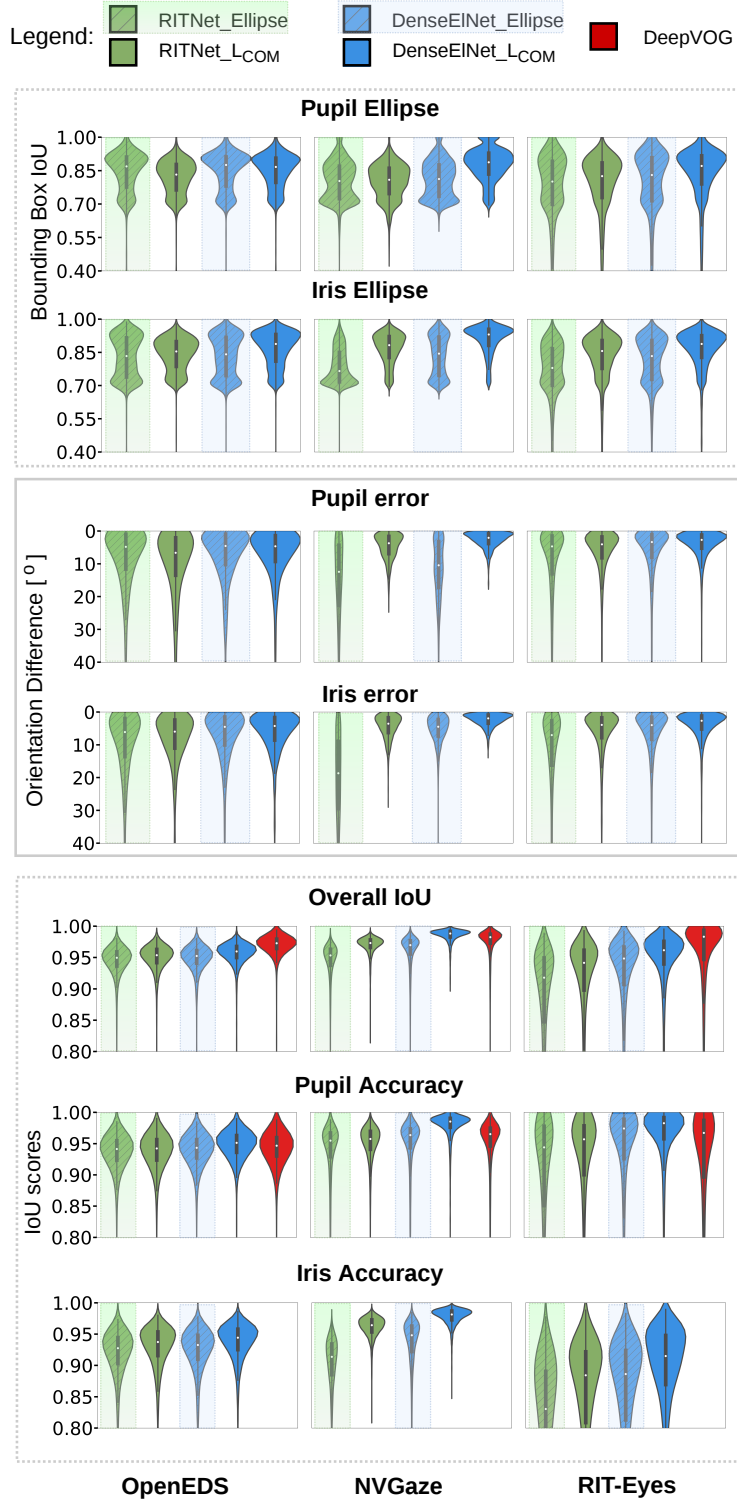


Figure 6.9: Violin plots of boundary overlap IoU (1st and 2nd row: top dashed box), orientation error (3rd and 4th row: middle solid box), and segmentation IoU score (last three rows: bottom dashed box) following EllSeg framework by RITnet and DenseElNet, with or without  $\mathcal{L}_{COM}$  loss ( $\mathcal{L}_{COM}$  vs Ellipse), following application to the OpenEDS, NVGaze, and RIT-Eyes datasets (columns)  
*(Best viewed on screen)*

of RITnet and Densenet to several datasets, either with or without  $\mathcal{L}_{COM}$ . When  $\mathcal{L}_{COM}$  is used, ellipse orientation and axis parameters are regressed via the bottleneck layer, and when it is not, the ellipse is fit to the segmented mask.

The result of this analysis are presented in Figure 6.9, and reveal that that DenseElNet *with*  $\mathcal{L}_{COM}$  outperforms *without*  $\mathcal{L}_{COM}$  in terms of boundary IOU and orientation error for both, the pupil and iris, on almost all datasets.

The pixel-wise IOU score of iris and pupil segmentation is presented in Figure 6.9 (last three rows). This analysis reveals that DenseElNet also outperforms other models in the segmentation of the pupil and iris. Although DeepVOG has the highest overall IoU score, one must also consider that the DeepVOG model is a two-class (binary) classifier (pupil vs. background) being compared against models of three-class segmentation (pupil, iris, background) and, in the former case, the IoU score is inflated by the presence of a large number of background pixels. This analysis also demonstrates that segmentation performance is improved by the inclusion of  $\mathcal{L}_{COM}$  for all cases. Some examples of segmentation outputs with the inclusion of  $\mathcal{L}_{COM}$  for OpenEDS and RIT-Eyes datasets are shown in Figure 6.10.

### Qualitative Analysis: Effectiveness of $\mathcal{L}_{COM}$ loss

Here, we study the impact of the  $\mathcal{L}_{COM}$  loss function with the DenseElNet architecture. Figure 6.11 shows the activation maps generated (*with* and *without*)  $\mathcal{L}_{COM}$  for three eye images. On closer observation of the pupil class, we observe a high intensity peak in the region around pupil center in the *with*  $\mathcal{L}_{COM}$  condition (last column) compared to the *without*  $\mathcal{L}_{COM}$  condition (fourth column from left). This peak around the pupil center is also evident in Figure 6.12 which shows a horizontal scan through the pupil center of one of the eye images illustrating the relative activation value for background, pupil, and iris *without* (left) and *with* (right)  $\mathcal{L}_{COM}$ .

Note that in Figure 6.11, the iris activation maps appear even when the iris is occluded by the eyelids in both *with*  $\mathcal{L}_{COM}$  (second column from right) and *without*  $\mathcal{L}_{COM}$  (third column from left) conditions.

Figure 6.12 shows relatively flat activation values near the iris centers for the iris class in both *with* and *without*  $\mathcal{L}_{COM}$  cases; no peak is evident in the iris activation values. Note that the minimum in the background activation value localizes the center of the *iris* representing the inverse of the background (non-iris) region.

#### 6.7.4 Center via bottleneck vs softargmax

To help provide an intuition regarding future network designs, we observe the impact of regressing the pupil and iris center estimates from the bottleneck (latent) layer [63],

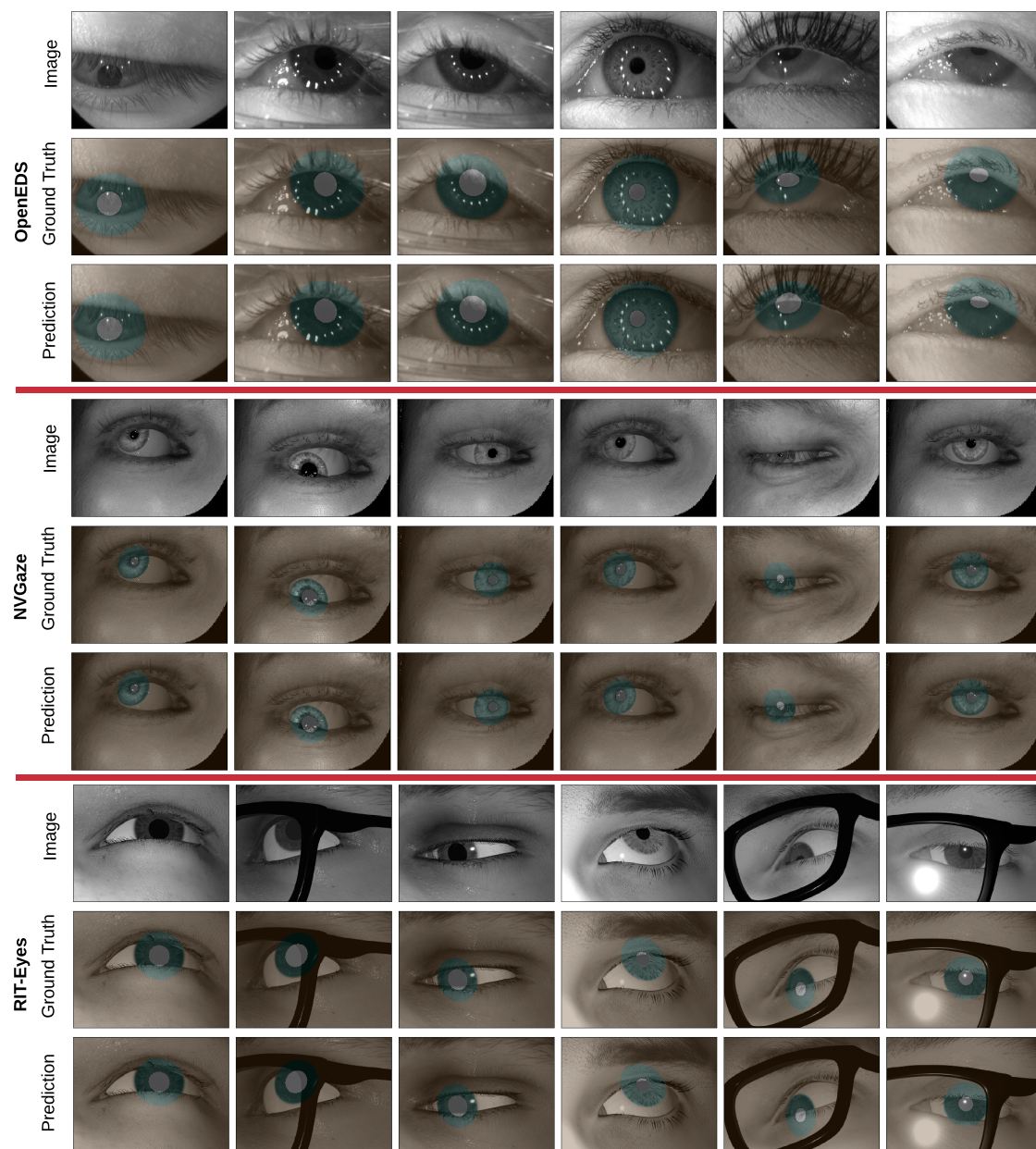


Figure 6.10: DenseElNet model prediction and its respective ground truth for OpenEDS, NVGaze and RIT-Eyes dataset.

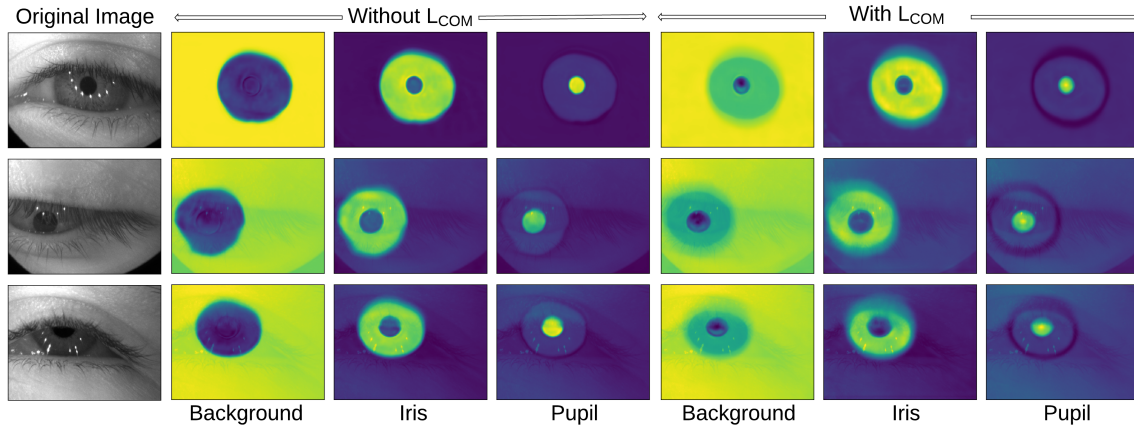


Figure 6.11: Figure showing 2D activation maps. Columns (L-R): Original image (1st column), activation maps for background, iris and pupil class for model DenseElNet *without*  $\mathcal{L}_{COM}$  (2nd-4th column) *with*  $\mathcal{L}_{COM}$  (5th-7th column). Three rows show three different cases with bottom two having the original image in the background for reference. (*Best viewed on screen*)

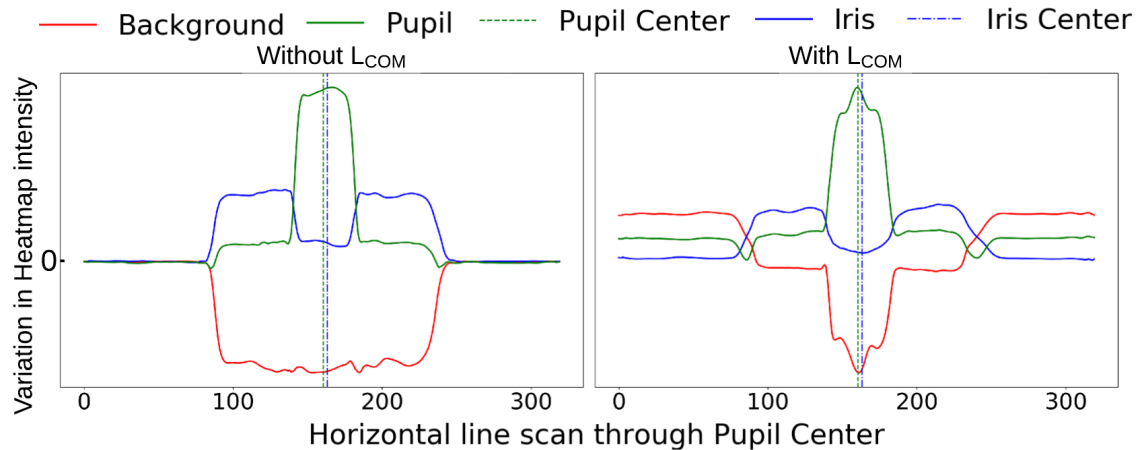


Figure 6.12: A horizontal line scan across the pupil center to visualize DenseElNet output behavior without  $\mathcal{L}_{COM}$  (left) and with  $\mathcal{L}_{COM}$  (right). The inclusion of  $\mathcal{L}_{COM}$  generates characteristic peaks which do not impede the task of semantic segmentation while effectively scaling output pixel activations near the predicted pupil and iris centers (*Best viewed on screen*).

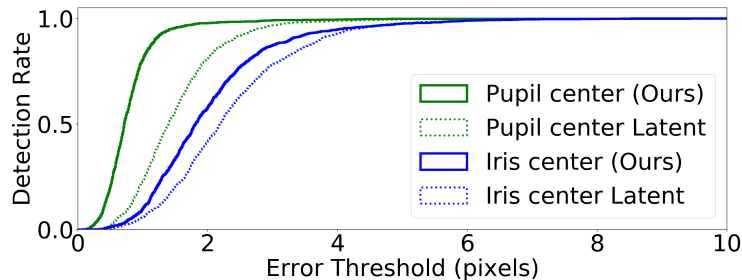


Figure 6.13: The difference between pupil and iris detection rate in the OpenEDS dataset. Estimates are derived from the latent space and final segmentation maps (DenseElNet).

as opposed to estimating them using soft-argmax on the output segmentation maps (see Figure 6.2). Estimates from segmentation outputs are observed to be better than those regressed from latent space (pupil 81%  $\rightarrow$  98% and iris 42%  $\rightarrow$  58% detection at the two-pixel error margin) (see Figure 6.13). We hope that this intuition can help guide future efforts for CNN based near-eye feature extraction.

## 6.8 Summary

This paper presents EllSeg, a new framework for training a CNN to directly segment the entire elliptical structures of the pupil and iris. This framework was applied to RITnet [9], DeepVOG [31] and a custom designed hybrid model, DenseElNet, for segmentation as well as predicting pupil/iris ellipse estimates from eye images.

In Section 6.7.1, we benchmark our custom designed network architecture, DenseElNet, and achieve better baseline PartSeg performance to state-of-the-art encoder-decoder architectures, RITnet and DeepVOG (see Table 6.2). Our un-optimized forward pass implementation of DenseElNet operates at 30Hz on an NVIDIA 1080 Ti, Intel-7800K. In Section 6.7.2, we show that our proposed framework *EllSeg* outperforms part-segmentation networks, *i.e.*, *PartSeg*, for pupil and iris center detection across three test datasets (OpenEDS, NVGaze, and RIT-Eyes). Additional analysis reveals that the accuracy of EllSeg can be attributed to greater robustness to occlusion of the iris and pupil by the eyelids. Section 6.7.3 demonstrates that the addition of  $\mathcal{L}_{COM}$  loss function to the EllSeg framework results in improved pupil/iris ellipse estimates (10% pupil and 24% iris center detection rate within a two-pixel error margin) and segmentation performance ( $> 0.6\%$ ,  $> 1.5\%$ ,  $> 2\%$  for OpenEDS, NVGaze and RIT-Eyes respectively). Visual inspection of output EllSeg activation maps reveals high confidence conditioned around the



pupil and iris centers. Lastly in Section 6.7.4, we determine that deriving pupil and iris centers using softmax is better than regressing the same via the bottleneck layer.

## 6.9 Conclusion and future work

To conclude, we present EllSeg, a simple 3-class full ellipse segmentation framework intended to extend conventional encoder-decoder architectures for the segmentation of eye images into pixels that represent the pupil, iris, and background. The EllSeg framework was benchmarked on multiple datasets using two network architectures: RITnet and our custom CNN design, DenseElNet. Results demonstrate superior estimation of the pupil and iris centers and orientation compared to their eye part segmentation models. An added benefit of the EllSeg framework is that it extends model training to image datasets in which only the pupil center has been labelled. Superior performance by the EllSeg framework can be attributed to greater robustness to occlusion of the pupil or iris.

While we evaluate EllSeg on multiple datasets collected from a large pool of individuals (see Table 6.1), a user based evaluation was not performed due to the time consuming nature of manual data collection and labelling. For future work, we intend on performing a comprehensive user study of our model on a wide range of subjects to further quantify the performance of our framework. We also intend on exploring other models with varying complexity to evaluate the efficacy of EllSeg. Pretrained models, code and other related resources will be made publicly available <sup>2</sup>.

## 6.10 Acknowledgements

We thank Research Computing [159] at the Rochester Institute of Technology for providing all necessary hardware required for this project. We also thank Dr. Christopher Kanan for his helpful feedback and guidance. Lastly, we would like to thank Dr. Thiago Santini and Dr. Wolfgang Fuhl for their guidance in setting up the framework for their PuRe and PuReST algorithms.

---

<sup>2</sup><https://bitbucket.org/RSKothari/ellseg>



## Chapter 7

# EllSeg-Gen

The study of human gaze behavior during unconstrained, in the wild activities requires access to reliable gaze estimation. However, the ability to identify gaze features such as the iris and pupil centroid in eye imagery suffers in the presence of varying degrees of reflective artifacts and occlusions, for example from the eyelids and eyelashes. Robustness to such artifacts and occlusions on unseen imagery is a salient requirement of an ideal eyetracking solution. This work explores the topic of model generalization for segmenting eye imagery.

### 7.1 Introduction

Eye tracking solutions frequently employ computer vision or machine learning (ML) algorithms to extract features of interest from images captured using eye cameras. These features facilitate the estimation of a subject’s gaze position. While numerous efforts have explored both approaches, recent works [9, 10, 31, 61, 63, 8, 160] report that ML systems demonstrate state of the art performance in identifying gaze relevant features for head-mounted eyetracking. Contrary to computer vision approaches where features are identified using handcrafted algorithms and heuristics, superior performance by ML is partly achieved by making minor adjustments in a ML system’s internal parameters with the objective to maximize the probability of predicting known outputs for given training inputs [161]. A ML system with millions of parameters could theoretically demonstrate perfect performance over the distribution of data it was trained on. These systems however often fail to generalize to out-of-distribution samples that are dissimilar to ones seen during training but plausible under the overarching goals of the problem. For example, ML systems trained to segment eye images acquired with a small geographic subset of the human population, or optimized for particular imaging hardware, may fail to generalize

onto the average use case.

In the context of eyetracking, high performance across subjects, environmental reflection, camera quality, camera placement and eye occluding artifacts can be cast as a Domain Generalization problem [162, 163, 164, 165]. Although the intuitive thought is that a broader training set will always produce the most generalizable model, it may come at the cost of performance [166, 167, 168] as a significantly broader distribution must be captured by limited network complexity [169]. A relevant and alternative approach to solving the generalization problem would be to assemble a suite of specialized, domain-specific models. At run-time, one could select from them the model trained on a dataset that maximizes statistical similarity to the intended testing data. The notable drawback to this approach is that it assumes the existence of a hypothetical method which finds the best matching model without any test-time labels or annotations. Figure 7.1 is graphical example to illustrate these approaches one may adopt for optimal generalization.

Researchers often tweak parameters or re-train a ML model specifically for their application. This generally means collecting and annotating data with a fixed hardware setup and environment while training a model that can generalize across subjects or gaze positions. Despite sampling from the same data distribution, a model may not generalize well beyond a few subjects or gaze positions [80]. We hypothesize that jointly training with multiple datasets may expand the available training distribution and in turn improve generalization across data collected from different subjects under the same conditions.

In this work, we explore the relationship between model generalizability and performance within the context of eye tracking. The specific contributions of this work are as follows:

- Sometimes, an engineer must design a model that is optimized for generalizability to unseen conditions rather than specialization to a specific set of conditions. We test the hypothesis that a single model trained on data drawn from multiple heterogeneous domains can generalize better than a specialized, dataset-specific model when evaluated on domains **unseen during training**. If we accept this hypothesis, then it suggests there is a benefit in expanding the breadth of the training distribution by accumulating more data. If we reject this hypothesis, then the better approach to achieve generalizability is by exploring test-time techniques to find the optimal model from a pool of dataset-specific models.
- In other contexts, an engineer aim to design a system for a specialized use-case that is represented by a pre-existing dataset of ground-truth imagery. Indeed, the best possible performance is attained when we train and evaluate a model on data drawn from the same distribution. Nonetheless, distribution shift can still exist due to biased sampling. We hypothesize that training with multiple heterogeneous

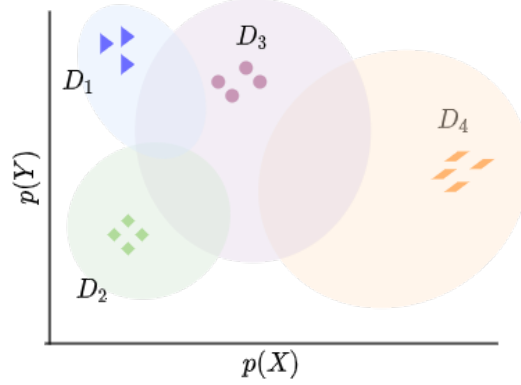


Figure 7.1: An illustration to visualize two different training strategies one may adopt to maximize generalization. If  $D_3$  represents a test distribution, then it is intuitive to train a model using a combination of  $D_1$ ,  $D_2$  and  $D_4$ . However, if  $D_4$  represents a test distribution then a model specific to  $D_3$  would demonstrate optimal generalization.

distributions could improve ceiling performance by mitigating distribution shift. If we accept this hypothesis, then it suggests that one may adopt a multiset training approach to identify limitations in a dataset. If we reject this hypothesis, then it suggests our model has limited complexity or the multiset training approach avoids overfitting to a particular domain.

- In addition to the central hypothesis related to the principles of model generalization, we present multiple contributions and insights of practical use to the eye tracking community. These include analyzing the effects of increasing model complexity and data augmentation.

## 7.2 Related work

Issues related to domain generalization arise when a ML model trained on a particular domain generally does not perform well on out-of-domain samples [170, 162, 171]. This well known phenomenon is known as distribution shift and generally occurs when the distribution of data points used to train the network does not match the distribution of data we are evaluating the model on. In this section, we give an overview of two major types of distribution shifts and the theoretical justifications for potential gains using joint optimization on multiple sets.

We are ultimately interested in minimizing the risk of incorrect predictions [161, 172]. Given a source distribution of data, we estimate the target risk as follows:

$$\begin{aligned} R_\tau(h) &= \sum_{y \in Y} \int_X l(h(x)|y) p_\tau(x, y) dx \\ &= \sum_{y \in Y} \int_X l(h(x)|y) p_s(x, y) \frac{p_\tau(x, y)}{p_s(x, y)} dx \end{aligned} \quad (7.1)$$

Here,  $R_\tau(h)$  estimates risk for a model  $h$  for an unknown test domain  $\tau$  and train domain(s)  $s$ . A domain can be represented as a joint probability distribution  $p(x, y)$  over input data  $x$  and its associated labels  $y$ . The function  $l$  computes the error between the prediction  $h(x)$  and the known label  $y$ .

### 7.2.1 Prior shift

Prior shift occurs when the distribution of output labels between the train and test domains vary [172, 173]. This can be represented in Equation 7.1 by decomposing the joint distribution,  $p(x, y)$ , as  $p(x|y)p(y)$ . Prior shift describes the specific case where  $p_s(y) \neq p_\tau(y)$  but the conditional distributions are equivalent,  $p_s(x|y) = p_\tau(x|y)$ .

$$R_\tau(h) = \sum_{y \in Y} \int_X l(h(x)|y) p_s(x, y) \frac{\overline{p_\tau(x|y)p_\tau(y)}}{p_s(x|y)p_s(y)} dx \quad (7.2)$$

Where the ratio  $p_\tau(y)/p_s(y)$  represent the change in class proportions. Consider the case of eye image segmentation, a critical step in the context of head-mounted eyetracking (see Figure 6.1 - an eye image and its ground-truth segmentation). In this context, a class could represent an annotated eye part at the pixel level (e.g. pupil, iris, sclera, other). A change in class proportions could occur when the eye camera is displaced at different distances for the train and test domains. This form of domain shift assumes that the conditional distributions  $p_\tau(x|y)$  and  $p_s(x|y)$  are equal. This assumption generally is not true since for a given label  $y_i$ , eye image appearance can widely vary depending on the environment, subject physiology, pupil dilation, gaze and camera positions. Therefore, the statistics related to the spatial distribution of group membership at the pixel level is unlikely to match across training/testing domains that reflect a difference in eye-to-camera geometry, camera intrinsic parameters, or across different populations.

To measure and mitigate prior shift, we require access to the test domain labels. Access to test labels allows us to draw samples non-uniformly in a manner such that  $p(y)$  of both domains align. Domain generalization however assumes no access to test labels. Data

augmentation schemes such as artificially translating, rotating and scaling images from a domain can mitigate a portion of prior shift [174, 175, 176, 177].

### 7.2.2 Covariate shift

Covariate shift occurs when image appearances vary between the train and test domain due to biased sampling. For example, a model trained to segment eye images captured indoors may not generalize well onto outdoor environments despite enforcing other variables such as subject physiology, gaze position and camera pose to be fixed.

$$R_\tau(h) = \sum_{y \in Y} \int_X l(h(x)|y) p_s(x, y) \frac{\overline{p_\tau(y|x)} p_\tau(x)}{\overline{p_s(y|x)} p_s(x)} dx \quad (7.3)$$

There are numerous approaches which attempt to minimize covariate shift but they require access to target imagery in order to model or approximate  $p_\tau(x)$ . Domain generalization assumes no access to target imagery.

### 7.2.3 Mitigating distribution shift

Empirical Risk Minimization [161] is one of the earliest algorithms which serves as a baseline for generalization. Simply stated, simultaneous optimization across multiple domains results in a model with the least empirical risk on the training set acquired from multiple domains. While simple in design, multiset training serves as a baseline to measure other techniques for domain generalization.

Previous work in domain generalization follow the intuition that a model may generalize better if internal network activations are invariant across domain-specific factors [162, 171, 178]. In the context of eyetracking, we want to encourage a network to learn a generalized representation of an eye image and its subsequent mapping to semantic categories which are invariant to camera quality, occluding artifacts, gaze position and eye camera location. This is achieved by penalizing a network when the learned latent representation of eye images statistically align themselves to the domain the images were sampled from.

One of the earlier attempts to align representations between any two domains involves minimizing the Maximum Mean Discrepancy metric [179, 180, 181]. This is achieved by enforcing that the mean latent representation for each individual domain must be equal. Constraining predicted features in this manner results in learning a distribution of features centered around a common statistical mean while semantic information is encoded in the deviation from the predicted mean.

Other approaches have demonstrated better results using adversarial learning [182, 183, 181]. The key insight within adversarial techniques is to modify a model’s parameter update rule, also known as it’s gradient, to prevent a model from learning domain specific features. This is typically accomplished by leveraging a secondary network known as a discriminator with the task of identifying the specific domain that the predicted image features belong to. The primary model is tasked with confusing the discriminator while simultaneously learning to accomplish its primary task (*e.g.* the segmentation of eye images). There are multiple methods one might use to minimize discriminator error. Min-max learning [184] attempts to minimize the error made by the discriminator in predicting a feature’s correct domain while optimizing the primary network so that it maximizes the error made by the discriminator. In contrast, gradient reversal [185] attempts to minimize the discriminator’s error while negating the gradient used to optimize the primary network. Although techniques for distribution shift using adversarial learning can be effective, inappropriate placement of adversarial operations within a network can penalize features which are domain dependant and correlated to the primary task. This results in the extraction of sub-optimal features which negatively impacts a network’s performance.

Meta-learning approaches have also proven to be effective for domain generalization [186, 187]. Gradients are derived from randomly selected training sets but optimized to minimize the loss on randomly selected validation sets [188]. This is accomplished during every parameter update operation wherein a temporary network is created with the gradient trajectory determined by the training domains. This temporary network is then optimized to maximize performance on the validation domains, resulting in a modified gradient trajectory. Gradient updates in meta-learning approaches require computing the gradient via the temporary network on the validation domains and channeled into the original gradient derived from the training domains.

Domain generalization can also be achieved by following a curriculum based learning approach [189]. Li *et al.* decomposed a network into a feature extraction module consisting of convolutional operations and a classification module which estimates the probability of a category via extracted features. Their work proposes training domain specific modules which are randomly connected with a domain-agnostic model during cross-domain episodes. Cross-domain episodes alternate with within-domain episodes which further optimize models on their respective domains.

While these techniques have demonstrated better generalization performance in their respective applications, their effectiveness for head-mounted eye image segmentation remains unclear. Extensive experiments by Gulrajani *et al.* reveal that Empirical Risk Minimization [161] results in similar or better performance as compared to state of the art approaches when implemented and evaluated correctly. This work adopts multiset training to explore two hypotheses by leveraging insights from Gulrajani *et al.*

## 7.3 Methods

We describe the datasets, encoder-decoder convolutional architecture and various tests which allow us to test our hypothesis that jointly optimizing using multiple datasets results in better generalization than finding an optimal dataset to train a model.

### 7.3.1 Datasets

Tests of generalizability are complicated by practical limitations in dataset acquisition. Acquiring data which represents the general populace requires large-scale data collection with annotated ground truth. This needs many hours of tedious work, careful labelling, and may be impractical without the investment of significant time, financial resources, and coordinated effort across multiple laboratories. To mitigate this limitation, this work exploits multiple pre-existing and publicly available datasets of near-eye images acquired from a large number of subjects and different eye trackers which are captured under varying environmental conditions.

We acquire eye imagery from nine publicly available, annotated and heterogeneous datasets (see Table 7.1 for overall statistics and Figure 7.7 for individual distribution plots). Eye images from each dataset are assigned to a *train* or *test* split. Each split contains images acquired from different subjects or recording IDs. Images present in each dataset are scaled to a common resolution of 320 x 240 pixels. OpenEDS eye images were cropped vertically to maintain a constant aspect ratio across all datasets. Cropping was performed in a manner which ensured that the entire iris ellipse is visible. Note that the sets published in ExCuSe [32], ElSe [68] and PupilNet [59] are combined into a single dataset as the source of eye imagery are from the same collection [190]. We refer to the combination of these datasets as the *Fuhl* datasets.

### 7.3.2 Network architecture

Drift-free [144] and parallax-free [191] eyetracking requires modelling the approximate 3D center of rotation of an eyeball from 2D pupil [103, 56, 55] or limbus ellipses [57]. Convolutional encoder-decoder architectures have successfully been deployed to segment an eye image and extract pupil and iris ellipses [9, 10, 31, 61, 8, 63] for datasets with pixel-level semantic annotations. Most publicly available datasets do not provide access to pixel-level ground truth annotation but instead provide the pupil center only. To circumvent this limitation, we adopt with minimal changes the recently published EllSeg framework [10] that uniquely allows us to train a network using datasets with partial annotations. This is primarily achieved by tasking a convolutional network to segment entire elliptical masks instead of visible eye parts. The center of mass of the predicted

| Cond        | Res      | Status | Dataset   | Subject ID  |   | # of Subjects |      | # of Images |       |
|-------------|----------|--------|---|---|---|---------------|------|-------------|-------|
|             |          |        |   | Train   | Test  | Train         | Test | Train       | Test  |
| Constrained | 640×480  | Synth  | S-General [80]  | 1 - 18  | 19 - 24   | 18            | 6    | 34254       | 11582 |
|             | 400×640  | Real   | OpenEDS'19 [8]  | Train set   | Valid set   | 95            | 28   | 8827        | 2386  |
|             | 1280×960 | Synth  | NVGaze [66]   | Male 1-4,<br>Female 1-4   | Male 5,<br>Female 5   | 8             | 2    | 16000       | 4000  |
|             | 640×480  | Real   | LPW* [6]  | 2,4,5,7,10,<br>11,13,14,17,<br>19,21,22   | 3,6,8,9,12,<br>15,16,18,20  | 12            | 10   | 24000       | 17730 |
|             | 640×480  | Real   | Swirski [134]   | 1   | 2   | 1             | 1    | 298         | 298   |
|             | 384×288  | Real   | BAT [123]   | 1,2,3   | 4,5,6   | 3             | 3    | 3662        | 3541  |
| Outdoors    | 640×480  | Synth  | S-Natural [80]  | 1 - 18  | 19 - 24   | 18            | 6    | 34267       | 11548 |
|             | 640×480  | Synth  | UnityEyes [81]  | -   | -   |               |      | 16000       | 2000  |
|             | 384×288  | Real   | (Fuhl)<br>ExCuSe [32] +<br>ElSe [68] +<br>PupilNet [59] | I, III, V, VII,<br>VIII, XI, XII,<br>XIV, XVI, XVIII,<br>XIX, XX, XXI,<br>XXIV,<br>New II, New IV | II, IV, VI, IX,<br>X, XIII, XV,<br>XVII, XXII,<br>XXIII, New I,<br>New III, New V | 16            | 13   | 73053       | 57496 |

Table 7.1: Datasets and their respective train and test splits used to explore generalization. Each dataset is classified into two broad categories called *outdoors* and *constrained*. Eye images in outdoor datasets exhibit large proportion of environmental reflections. Constrained datasets are acquired from experiments or synthetically rendered within indoor, lab environments with little to no reflective artifacts. \* Approximately a third of LPW recordings were collected outdoors.



elliptical maps allows us to optimize the entire architecture using only pupil and/or iris center annotations, conveniently allowing us to train a network on eye images with partial annotations [10]. For a complete breakdown of the architecture, please see Figure 7.2.

### 7.3.3 Normalization schemes

Normalizing the input to a convolutional layer significantly speeds up training, improves peak accuracy performance, and reduces the dependence on weight initialization and hyper-parameter searches [192, 193]. Batch normalization in particular is a widely accepted technique to reparameterize the underlying optimization problem by providing a smoother loss landscape [193]. The batch normalization function is:

$$f_{BN}(x^k) = \left( \frac{x^k - \mu_k}{\sigma_k} \right) \gamma^k + \beta^k \quad (7.4)$$

Where,  $x^k$  is the  $k^{\text{th}}$  extracted image feature. Parameters  $\mu_k$  and  $\sigma_k$  are the global mean and standard deviations of feature  $k$  across all training images. Parameters  $\gamma$  and  $\beta$  are learnable affine transformations. Computing the global mean and standard deviation across the entire training set is impractical when using stochastic learning [192]. Instead, batch normalization computes  $\mu_k$  and  $\sigma_k$  within a population sample (e.g. a *batch*) and approximates the global statistic by accumulating these values as training progresses. Statistics accumulated during the training phase are fixed during model evaluation and are generally provided alongside network parameters to facilitate inference.

Batch normalization is effective if statistics of the population sample accumulated during training are approximately equal to the global statistics of features extracted from the test set. This assumption is often violated in the context of domain generalization as the statistics accumulated from the training domain(s) may encourage learning sub-networks which are specifically aligned towards individual domains and which may not be transfer onto an unknown test domain (see Figure 7.3).

To overcome this limitation, we adopt Instance Normalization [194] as a drop-in replacement for Batch Normalization and observe improvements to generalization (see Tables 7.6, 7.7 and 7.8). Instance Normalization computes the mean and standard deviation on a per image basis which are used to normalize and describe image features, irrespective of their domain, to the same range. Batch normalized features may belong to distributions that vary in mean and standard deviation, while instance normalized features are cast onto the unit-normal distribution. Figure 7.3 visualizes the perils of Batch Normalization and how replacing it with Instance Normalization mitigates this problem.

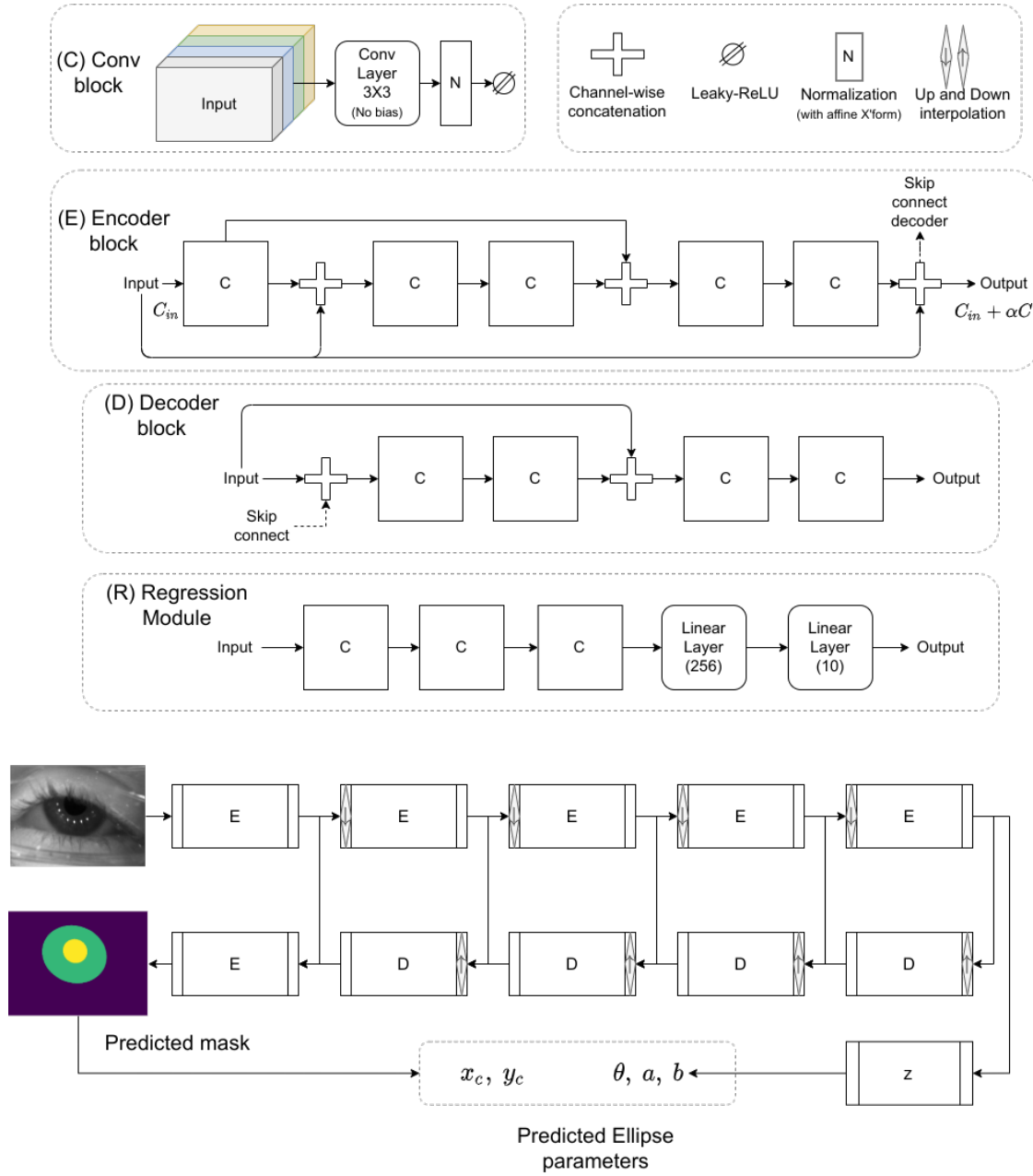


Figure 7.2: DenseElNet architecture adapted from EllSeg [10]. The number of parameters in the encoder are controlled by a base channel size of  $C$  and a growth rate of  $\alpha$ .

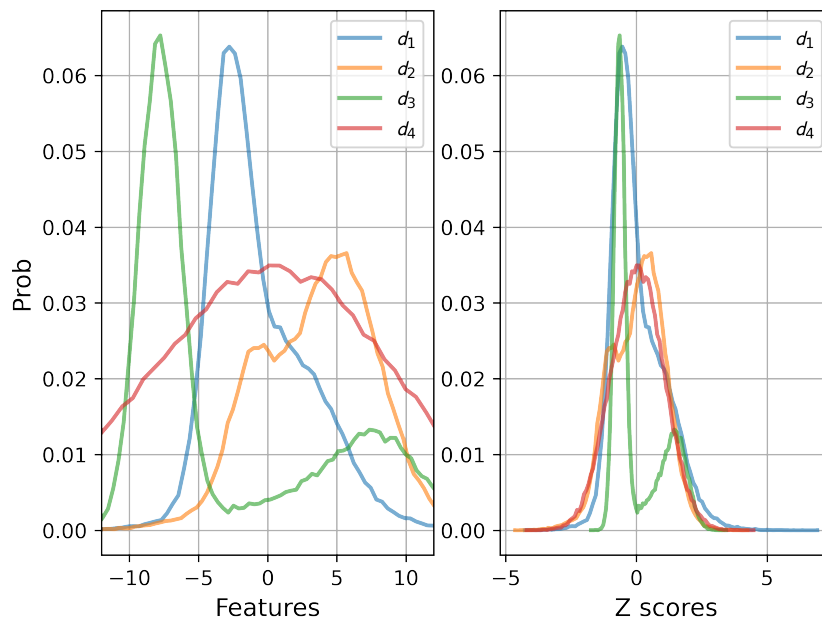


Figure 7.3: A mock example to illustrate the difference between Batch and Instance Normalization using normalized features extracted from images sampled from domains  $d_1$ ,  $d_2$ ,  $d_3$  and  $d_4$ . (Left) Batch normalization centers the extracted features relative to the global mean estimate of all available domains while Instance Normalization (right) utilizes each individual image based statistic.

### 7.3.4 Model Performance Metrics

We analyze three separate metrics of performance:

- Mean Intersection-Over-Union (mIoU) score [195, 196] for the segmented pupil, iris and background class. This metric is measured as  $\frac{1}{K} \sum_{k=1}^K y_k \cap \hat{y}_k / y_k \cup \hat{y}_k$  where  $y$  and  $\hat{y}_k$  are predicted and groundtruth masks for the  $k^{\text{th}}$  semantic category. Higher mIoU score indicates better performance.
- Pupil center error measured in units of pixels. Lower pixel error is better.
- Iris center error measured in units of pixels. Lower pixel error is better.

### 7.3.5 Generalization tests

The primary goal of this work is to determine if training a model using multiple datasets results in better performance when generalized to an unseen domain as opposed to selecting the best performing domain-specific model. The former follows our intuitive understanding that training a model on multiple domains expands the available distribution we draw samples from. The latter is possible when the domain we are evaluating overlaps with an existing dataset. The adopted approach draws inspiration from Koshla *et al.* [171] and proposes four tests which allow us to explore our hypotheses. Every test below will be run on each individual domain’s test set.

#### Within-dataset

This test is intended to measure the ceiling performance for a given dataset’s test set. Evaluating a model on the same dataset it was optimized on returns the upper performance limit. Any performance exceeding this measure indicates that a distribution gap between the train and test exists. This could occur if the dataset is limited by insufficient variability due to a biased sampling of its data distribution, insufficient number of subjects or limited gaze positions due to constrained tasks.

#### Cross-dataset

This test is intended to measure the performance of a model under conditions of cross-dataset Domain Generalization using a single training dataset. Cross dataset results indicate the performance of models when evaluated on domains not utilized during training. This test allows us to quantify how dissimilar two domains are based on their cross dataset performance measures. For every available dataset, this test allows us to empirically find the closest matching dataset.

### All-vs-one

This test is intended to measure the ceiling level of performance when training utilizes all available datasets, including the within-domain dataset. This scheme trains a model using combined imagery from all available datasets. Comparing the performance of a model trained on all available distributions against its equivalent within-dataset performance gives us a clue about a single model’s ability to capture information from multiple heterogeneous distributions. Deteriorated performance indicates insufficient network capacity and serves as a test to ensure that our results are not influenced by network architecture.

### Leave-one-out

This test is intended to measure the performance of a model trained using multiple datasets except on a given test set that is used to evaluate the model performance. For example, leave-one-out results on the OpenEDS test set would involve training a model with all datasets except the OpenEDS training images. Comparing the results of this leave-one-out test with the cross-dataset performance provides evidence for the optimal strategy which maximally generalizes on the OpenEDS dataset.

### 7.3.6 Predictions

In this section, we briefly summarize expected results using the four proposed tests. Model performance will be reported on their relative Intersection-over-Union (IoU) scores, and their relative ability to estimate the pupil and iris centers, in units of pixels (see Section 6.6.3). Systematic comparison across the proposed tests allows us to generate specific predictions directly related to our hypotheses.

#### Hypothesis 1: Cross-domain generalization

A single model trained on data drawn from multiple heterogeneous domains (the leave-one-out test) can generalize better than a specialized, dataset-specific model when evaluated on domains **unseen during training** (the cross-dataset test). This is achieved by comparing the leave-one-out test with the cross-dataset test. If the leave-one-out test leads to better generalization than the cross-dataset test, this would indicate that adding more datasets and acquiring a broader distribution will improve generalization, presumably asymptotically until the within-dataset performance limit is reached. If the cross-dataset model outperforms the leave-one-out test then it suggests that the best approach to achieve generalizability is to explore test-time techniques to find an optimal model from a pool of dataset-specific models, rather than to rely upon a single broadly-trained and general-purpose model.

### **Hypothesis 2: Within-domain generalization**

Researchers often retrain models for their specific applications with a fixed hardware setup and environmental conditions. However, a model may not generalize across subjects or gaze positions due to distribution shifts caused by limited or biased sampling of their specific distribution. The within-dataset performance represents the distribution shift exhibited when the train and test splits are sampled from the same distribution. We hypothesize that any remaining distribution shift due to biased sampling will be mitigated by increasing the breadth of training distribution. If this is true, then the all-vs-one test will outperform the within-dataset test. Conversely, if the within-dataset test outperforms the all-vs-one test then it suggests that a) the all-vs-one model has insufficient complexity (parameters) to capture the entire breadth of the training data or b) the all-vs-one model has learned a generalized representation of eye images and a multiset training approach avoids overfitting to domain-specific features. These two possibilities are explored in post-hoc tests, and addressed in the discussion section.

### **Hypothesis 3: Effects of data augmentation**

Data augmentation has numerous benefits for machine learning applications [197, 198]. It improves generalization by combating overfitting, expands the training distribution and reduces domain gap by combating prior shifts. Previous work has demonstrated that domain-specific data augmentation significantly improves cross-subject performance in context of head-mounted eyetracking [60]. Data augmentation is generally accepted as standard practice within Machine Learning and we quantify its effects on generalization (see Table 7.2).

It is plausible that within the context of eye image segmentation, data augmentation sufficiently reduces distribution shift by expanding the available training distribution while removing the need for multiset training. If this is true, then we expect to see improvements in the within-dataset and cross-dataset tests when compared against experiments which do not involve data augmentation. If this is false, then we expect to observe little to no improvements on all tests. Table 7.2 summarizes all augmentation schemes explored in this manuscript.

#### **7.3.7 Analysis**

Exploring our hypotheses for each dataset requires us to compare model performance between the proposed tests while keeping the within-dataset performance as baseline. However, datasets differ in complexity. A 1 pixel error improvement in pupil center estimation on the NVGaze and Fuhl dataset cannot be compared objectively without considering the

variability inherent within the ground truth, for the reason that one would expect more variability in model estimates for an inherently more variable test dataset. To mitigate this limitation, we consider the dispersion of test results (reported in median absolute deviation units, or MADs, due to its robustness to outliers) which are derived by normalizing a performance metric to the within-dataset test result.

### 7.3.8 Training details

This work relies on a modified version of DenseElNet [10], a standard encoder-decoder convolutional neural network inspired by RITNet [9], TiramisuNet [72] and UNet [7]. Eye images are passed to the encoder which consists of 4 densely connected convolutional blocks. Each block extracts features from eye images while down-sampling their spatial extent by a factor of 2. Latent representations rich with semantic features are then fed into the decoder which produces a segmentation output mask for each eye image. For a complete breakdown of architecture, please refer to Figure 7.2. Latent representations are also fed into a regression module which regresses pupil and iris ellipse parameters.

For experiments which involve joint optimization on multiple datasets, an equal number of randomly selected samples from each domain are concatenated as input to our neural network. This process alleviates the concern of unequal representation due to a disproportionate number of samples in each domain (see Table 7.1). The validation set comprises 20% samples held out from the training set, either from a single or multiple domains. All models are optimized with ADAM [118] for 80 epochs. This work utilizes the loss functions proposed in EllSeg (Section 3.3 in Kothari *et al.* [10]). To curb overfitting and to ensure our model can be applied to both, left and right eye images, we horizontally flip each eye image and its associated annotations randomly.

Experiments involving the augmentation of training data modify input eye images and its associated annotations by randomly selecting an operation from a pool of augmentation schemes. Table 7.2 provides a summary of all schemes explored in this manuscript. Each scheme has a  $1/11$  chance of being selected.

### 7.3.9 Evaluation criterion

Gulrajani *et al.* summarizes three model selection methods for domain generalization which limit access to the test domain for fair evaluation [200]. We adhere to strict evaluation protocols by adopting the leave-one-domain out method when quantifying generalization. The performance of a model on a dataset is evaluated on a set of eye images from human subjects that were never present during training (see Table 7.1). The best performing model is selected as the configuration which maximizes an average of mIoU and  $d_i + d_p$  on the validation set. Here,  $d_p = 1 - \alpha e_p$  and  $d_i = 1 - \alpha e_p$  where  $e_p$  and

| Gauss blur                                   | Motion blur                                | Gamma                             | Exposure   | Gauss noise                            |
|--|--|-----------------------------------|--|--|
| w=7<br>$\sigma=\mathcal{U}(2,7)$             | w=7<br>$\theta=\mathcal{U}(0, \pi)$        | $\gamma=\mathcal{U}(0.6, 1.4)$    | $\Delta L_{max}^i = 0.8 \times \tilde{L}^i$<br>$\Delta L_{min}^i = 0.8 \times (255 - \tilde{L}^i)$<br>$L = L + \mathcal{U}(-\Delta L_{min}^i, \Delta L_{max}^i)$ | $\mu=0$<br>$\sigma=\mathcal{U}(2, 16)$ |
| Synth lines                                  | Scale                                      | Rotation                          | Translation  | Synth fog                              |
| $\mathcal{U}(1,10)$<br>random<br>white lines | Scale<br>factor<br>$\mathcal{U}(0.5, 0.9)$ | $\theta=\mathcal{U}(-\pi, \pi)/4$ | Horz: $\mathcal{U}(-W, W)/3$<br>Vert: $\mathcal{U}(-H, H)/3$   | Please<br>refer to<br>imgaug [199]     |

Table 7.2: Augmentation schemes applied to every single eye image with a  $1/11$  probability.  $\mathcal{N}$  and  $\mathcal{U}$  indicate a normal and uniform distribution respectively.

$e_i$  are pixel errors in predicting the pupil and iris centers and  $\alpha = 240$  is the smallest image dimension. All models are evaluated every 2000 *iterations*, wherein an iteration is defined as a single network parameter update operation based on a batch of eye images. Annotations not present in a dataset are ignored while computing the evaluation metric. For experiments which involve training on multiple datasets, each batch consists of 3 eye images extracted randomly from every dataset included in the experiment. For experiments involving training on a single dataset, each batch consists of 24 eye images.

## 7.4 Results and Discussion

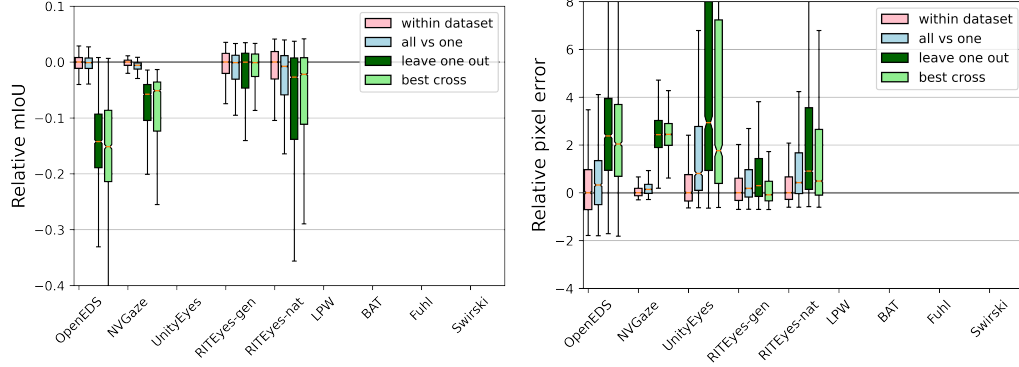
In this section, we provide results for tests listed in Section 7.3.5.

### 7.4.1 Hypothesis 1: Training with multiple datasets is an optimal strategy for generalization.

Ground truth pupil center annotations are available for the majority of datasets. Hence, our analysis primarily focuses on pupil center performance with specific observations made for datasets with annotated iris centers and segmentation masks. Comparing pupil center accuracy across all datasets suggests that results depend upon the conditions present during data collection.

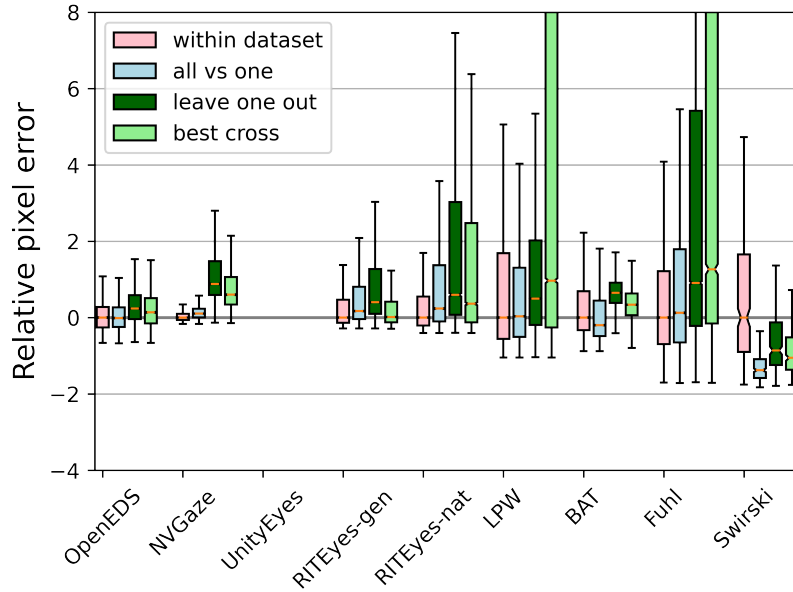
Pupil center generalization on datasets collected in naturalistic, outdoor conditions supports the hypothesis that a model trained on multiple datasets outperforms all dataset specific models. For example, the Fuhl datasets (ExCuSe + ElSe + PupilNet) represent an outdoor use case with unconstrained reflections from the surrounding environment. Comparing the best performing cross-dataset test result with the leave-one-out test (see





(a) Performance measured as the mIoU metric. The Y axis represents IoU score relative to the control condition, *within-dataset*. Higher is better.

(b) Error in iris center,  $e_i$ , in pixels. The Y axis represents pixel distance relative to the control condition, *within-dataset*. Lower is better.



(c) Error in pupil center,  $e_p$ , in pixels. The Y axis represents pixel distance relative to the control condition, *within-dataset*. Lower is better.

Figure 7.4: Generalization test results. Each box plot highlights a model’s performance centered to the within-dataset limit for each domain. The line and notch present within each box plot represents the median and 95% confidence interval respectively while the ends of each box denotes the 1<sup>st</sup> and 3<sup>rd</sup> quartile. All images are  $320 \times 240$  resolution. Note that datasets which are missing groundtruth annotations do not have a boxplot entry. All measures are centered to the within-dataset performance limit.

Figure 7.4) reveals that a multiset model offers improvements to generalization ( $\uparrow 0.42$  MADs) over a model trained on its best matching dataset, RITEyes-natural. Said another way, the results suggest that training with breadth will increase the chances of generalizing to outdoor imagery. Note that the model size is kept constant across tests. Similar behavior ( $\uparrow 0.66$  MADs) is observed on the LPW dataset which was collected both indoors and outdoors.

In contrast, pupil center prediction error for datasets collected indoors (OpenEDS and NVGaze) indicate that the optimal strategy is to utilize a dataset-specific model. This is evident when the best performing dataset-specific model performance is compared against a multiset model ( $\downarrow 0.37$ ,  $\downarrow 3.75$  MADs on OpenEDS and NVGaze respectively).

One possible explanation for this difference between indoor and outdoor datasets is that uncontrolled, outdoor data contains larger variability in eye image appearance. This interpretation is supported by a post-hoc analysis of luminance distribution of individual eye parts to illustrate the wide variability in pupil appearance existent in the Fuhl datasets (see Figure 7.7). The normalized luminance distribution of pupils in the Fuhl datasets demonstrate a large spread and significant presence of intensities above the mean luminance of an eye image. Note that due to the absence of groundtruth, predicted segmentation masks are utilized to identify individual eye-parts in Figure 7.7. Another possibility is that both LPW and Fuhl datasets exhibit a wide distribution of pupil center positions which is sufficiently represented during multiset training. We anticipate that data augmentation techniques, such as random affine transformation (see Table 7.2) could alleviate the gap in performance between the best cross-dataset and leave-one-out model. This is addressed in the test of Hypothesis 3.

The multiset training paradigm also demonstrates an improvement in segmentation performance on the OpenEDS ( $\uparrow 0.95$  MADs). This evidence supports the intuition that multiset training offers a broader range of iris appearances which in turn has the potential to improve iris segmentation.

#### 7.4.2 Hypothesis 2: Training with multiple datasets will improve within-dataset performance.

The within-dataset performance represents the distribution shift exhibited when the train and test splits are sampled from the same distribution. We hypothesize that training with multiple heterogeneous distributions could improve upon within-dataset performance by mitigating this latent distribution shift. That is to say we hypothesize that in the presence of distribution shift, the shift will be mitigated by increasing the breadth of the training distribution, as would be suggested if the all-vs-one test were to outperform the within dataset test. However, results indicate that this is not the case for the majority of tests which imply a lack of distribution shift between training and testing sets.

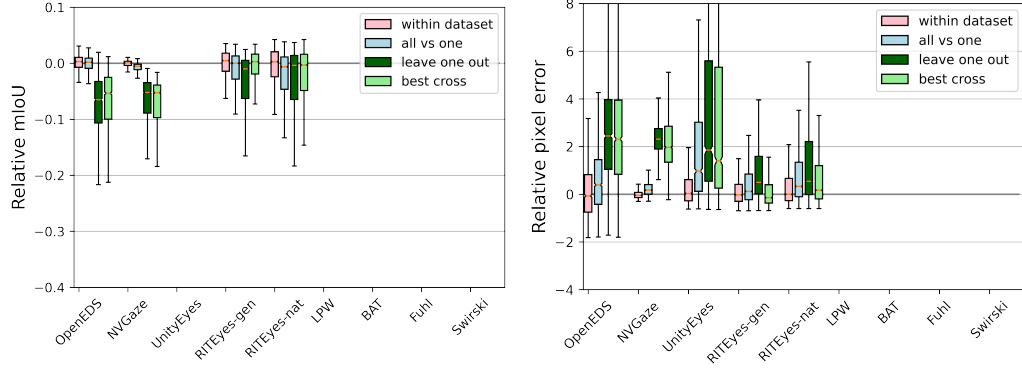
For the majority of datasets, (see Figure 7.4) the all-vs-one model demonstrates only a slight reduction in performance from the expected upper bound indicated by the within-dataset test. The highest disagreement with the upper bound is exhibited in the NVGaze dataset with  $\downarrow 1.40$ ,  $\downarrow 1.38$  and  $\downarrow 0.96$  MADs from the IoU, pupil and iris center upper bound respectively. This result suggests that, for these datasets, a single model can nearly achieve the best expected performance despite training with numerous available images, even when model size is kept constant across tests.

It is only on the BAT and Swirski datasets that the all-vs-one test outperforms the within-dataset performance, by 0.45 and 1.34 MADs, respectively, indicating distribution shift between the training and testing datasets. This comparison is a relatively simple test to identify a biased sampling of the training data-distribution either due to limited subjects or limitations in the data acquisition process such as narrow gaze angles. Closer inspection of the Swirski dataset reveals that it is comprised of  $\sim 300$  eye images from a single subject while the BAT dataset consists of 3.5K eye images from 3 subjects. Results indicate that these datasets are insufficient for cross-subject generalization despite their images being drawn from the same environment and eyetracker hardware.

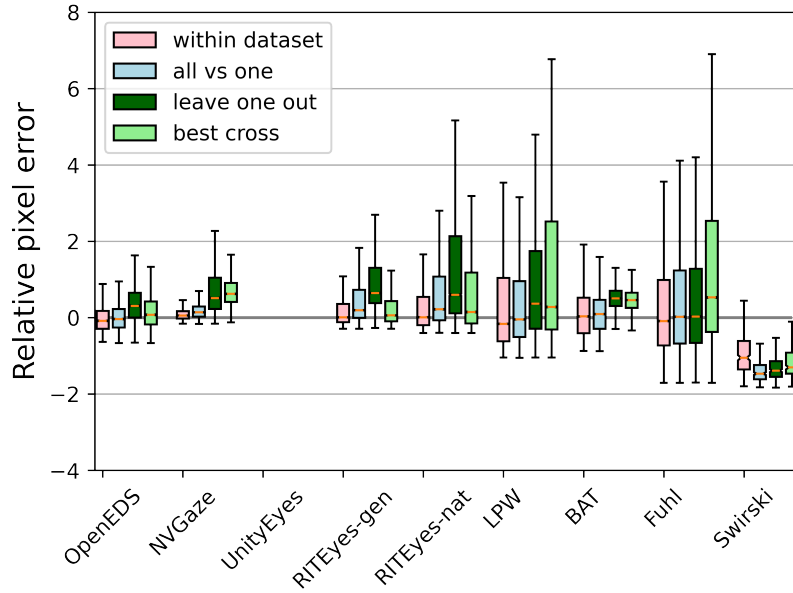
### 7.4.3 Hypothesis 3: Effects of data augmentation for generalization

Figure 7.5 summarizes performance gains observed by augmenting eye images, and reveals that data augmentation improves performance across all testing sets. This is indicated by reductions in the inter-quartile range (IQR) and improvements to the median performance across all metrics and tests. Improvements are particularly significant for outdoor datasets with environmental reflections, both real and synthetic. Notable improvement ( $\uparrow 3.00$  MADs) due to data augmentation can be observed on the Swirski dataset where the distribution shift (see Figure 7.7) is almost entirely eliminated. Data augmentation also improves generalization across the board. Significant improvements to generalization can be observed on the Fuhl datasets wherein a combination of multiset training and data augmentation achieves performance on par with the within-dataset baseline.

We find that our previous established hypotheses (see Section 7.4.2) regarding multiset training remains unaffected by the presence of data augmentation for the Fuhl datasets. Generalizing to outdoor datasets seem to favor a multiset training approach while a specialized, dataset specific model offers better pupil center detection for indoor datasets. In hypothesis 1, we identified that the LPW dataset favors multiset training due to the possibility of a wide distribution of pupil center locations. As expected, data augmentation sufficiently reduces the performance gap between the multiset model and the best performing cross-dataset model (albeit with a higher spread), indicating that either approach for generalization returns similar performance.



(a) Performance measured as the mIoU metric. Higher is better. (b) Error in iris center,  $e_i$ , in pixels. Lower is better.



(c) Error in pupil center  $e_p$  in pixels. Lower is better.

Figure 7.5: Generalization test results to study the effects of data augmentation. Each box plot highlights a model's performance centered to the within-dataset limit for each domain. The line and notch present within each box plot represents the median and confidence interval respectively while the ends of each box denotes the 1<sup>st</sup> and 3<sup>rd</sup> quartile. All images are 320×240 resolution. Note that boxplots pertaining to datasets without a certain groundtruth annotation are missing. All measures are centered to the within-dataset threshold as seen in Figure 7.4

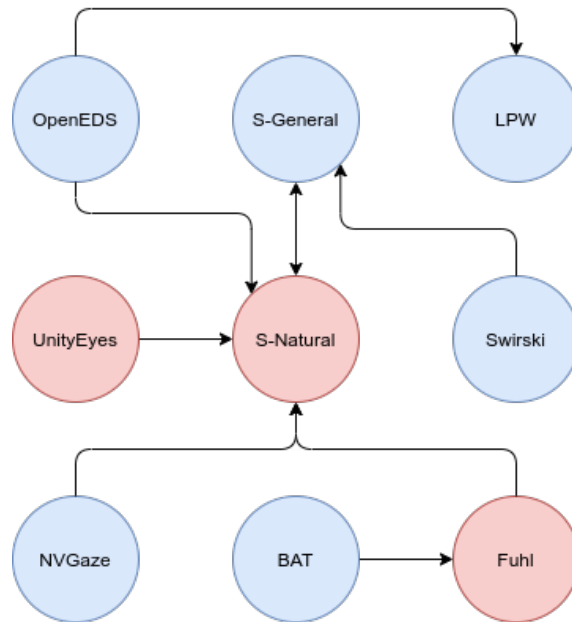


Figure 7.6: All datasets are represented as circular nodes. The arrow emerging from a node points towards its best matching dataset. Nodes colored in red are less constrained datasets with ambient reflections from their surroundings. Nodes colored in blue are constrained datasets with little to no environmental reflections. Best viewed in color.

#### 7.4.4 Which single training dataset offers the best cross-domain performance across all testing datasets?

A model trained on the artificially rendered RITEyes-Natural dataset offers the best cross dataset performance when evaluated on the remaining datasets (see Tables 7.4, 7.5 and 7.3). This is not surprising as it represents a relatively wide and densely sampled distribution of pupil centers (see Figure 7.7). Their synthetic images are rendered using a broad range of gaze positions, camera locations and eye image appearances from 24 artificial subjects. Furthermore, the synthetic eye images were rendered with naturalistic environmental reflections and lighting, and these augmentations may have also improved generalization to the datasets present in our experiments. Figure 7.6 is a graphical representation of how datasets align by model performance.

## 7.5 Conclusion

In this work, we have explored the task of Domain Generalization for segmenting near infrared eye images. This was achieved by jointly optimizing an encoder-decoder network on multiple eye image datasets with the intuition that a model would learn a generalized representation of eye images and elliptical eye parts. This work evaluated two approaches towards generalization, a) a multiset training approach to produce a single, robust model or b) pick the best performing model from a pool of pretrained, dataset-specific models. Generalization results indicate that outdoor datasets which exhibit higher appearance variability significantly benefit from multiset optimization. In contrast, dataset-specific models generalize better onto indoor datasets, which we speculate are more representative of constrained use-cases, as in the case of eye tracking integration into AR/VR headsets in which camera properties are fixed, camera pose is heavily constrained, and lighting conditions are controlled.

In contrast, one may be motivated to adopt a different approach to model design with training for a use case that is represented by a pre-existing dataset of ground-truth imagery. Although a model's peak performance can be attained when it is trained and evaluated on data sampled from the same source distribution, limited or biased sampling often leads to a distribution shift and deteriorated model performance. Results indicate that multiset training can be utilized to identify and mitigate such a distribution shift, if it exists. This work also validates data augmentation as an alternative approach to reduce model overfitting and further improve generalization by observing improvements in model performance across the board. Results indicate that a combination of data augmentation and multiset training attains the peak performance recorded for outdoor eye images. While data augmentation does improve performance, it does not invalidate the contributions of multiset learning but instead, complements it. All models and code will be made publicly available.

## 7.6 Supplementary data

| Train on / Test on | OpenEDS | NVGaze | UnityEyes | RITEyes-Gen | RITEyes-Nat | LPW | BAT | Fuhl | Swirski |
|--------------------|---------|--------|-----------|-------------|-------------|-----|-----|------|---------|
| OpenEDS            | 0.957   | 0.670  |           | 0.576       | 0.419       |     |     |      |         |
| NVGaze             | 0.652   | 0.988  |           | 0.563       | 0.362       |     |     |      |         |
| UnityEyes          |         |        |           |             |             |     |     |      |         |
| RITEyes-Gen        | 0.796   | 0.936  |           | 0.962       | 0.933       |     |     |      |         |
| RITEyes-Nat        | 0.806   | 0.932  |           | 0.961       | 0.955       |     |     |      |         |
| LPW                |         |        |           |             |             |     |     |      |         |
| Santini            |         |        |           |             |             |     |     |      |         |
| Fuhl               |         |        |           |             |             |     |     |      |         |
| Swirski            |         |        |           |             |             |     |     |      |         |
| all-OpenEDS        | 0.815   | 0.985  |           | 0.977       | 0.965       |     |     |      |         |
| all-NVGaze         | 0.960   | 0.930  |           | 0.974       | 0.965       |     |     |      |         |
| all-UnityEyes      | 0.960   | 0.983  |           | 0.973       | 0.961       |     |     |      |         |
| all-RITEyes-Gen    | 0.961   | 0.983  |           | 0.962       | 0.958       |     |     |      |         |
| all-RITEyes-Nat    | 0.960   | 0.984  |           | 0.968       | 0.928       |     |     |      |         |
| all-LPW            | 0.960   | 0.984  |           | 0.974       | 0.962       |     |     |      |         |
| all-Santini        | 0.961   | 0.984  |           | 0.976       | 0.965       |     |     |      |         |
| all-Fuhl           | 0.961   | 0.984  |           | 0.975       | 0.964       |     |     |      |         |
| all-Swirski        | 0.960   | 0.985  |           | 0.977       | 0.968       |     |     |      |         |
| all                | 0.956   | 0.982  |           | 0.961       | 0.947       |     |     |      |         |

Table 7.3: Segmentation results of various generalization tests proposed in Section 7.3.5. All results represent the IoU metric.

| Train on / Test on | OpenEDS | NVGaze  | UnityEyes | RITEyes-Gen | RITEyes-Nat | LPW    | BAT   | Fuhl   | Swirski |
|--------------------|---------|---------|-----------|-------------|-------------|--------|-------|--------|---------|
| OpenEDS            | 0.683   | 1.925   |           | 7.970       | 23.625      | 2.462  | 2.406 | 7.736  | 1.732   |
| NVGaze             | 1.039   | 0.168   |           | 17.104      | 30.998      | 10.507 | 1.227 | 70.052 | 4.736   |
| UnityEyes          |         |         |           |             |             |        |       |        |         |
| RITEyes-Gen        | 0.845   | 0.850   |           | 0.292       | 0.768       | 2.019  | 1.540 | 12.642 | 0.987   |
| RITEyes-Nat        | 0.818   | 0.767   |           | 0.305       | 0.407       | 2.122  | 2.366 | 2.985  | 1.623   |
| LPW                | 1.152   | 3.037   |           | 12.014      | 32.724      | 1.052  | 1.308 | 17.489 | 0.788   |
| Santini            | 2.912   | 94.254  |           | 31.814      | 54.246      | 12.778 | 0.892 | 94.609 | 2.105   |
| Fuhl               | 1.060   | 0.867   |           | 4.122       | 8.896       | 3.133  | 1.444 | 1.719  | 1.731   |
| Swirski            | 4.272   | 169.359 |           | 35.463      | 71.910      | 13.786 | 2.911 | 98.026 | 1.844   |
| all-OpenEDS        | 0.918   | 0.240   |           | 0.378       | 0.529       | 0.894  | 0.412 | 1.717  | 0.240   |
| all-NVGaze         | 0.666   | 1.048   |           | 0.453       | 0.624       | 0.960  | 0.438 | 1.786  | 0.374   |
| all-UnityEyes      | 0.625   | 0.279   |           | 0.385       | 0.580       | 0.949  | 0.427 | 1.675  | 0.274   |
| all-RITEyes-Gen    | 0.659   | 0.326   |           | 0.692       | 0.778       | 1.062  | 0.622 | 1.789  | 0.323   |
| all-RITEyes-Nat    | 0.598   | 0.283   |           | 0.503       | 1.003       | 0.936  | 0.439 | 1.763  | 0.265   |
| all-LPW            | 0.661   | 0.293   |           | 0.542       | 0.756       | 1.550  | 0.484 | 1.707  | 0.414   |
| all-Santini        | 0.606   | 0.265   |           | 0.343       | 0.515       | 0.896  | 1.543 | 1.689  | 0.250   |
| all-Fuhl           | 0.597   | 0.272   |           | 0.414       | 0.579       | 0.920  | 0.500 | 2.627  | 0.347   |
| all-Swirski        | 0.660   | 0.256   |           | 0.363       | 0.528       | 0.988  | 0.410 | 1.712  | 0.980   |
| all                | 0.669   | 0.272   |           | 0.461       | 0.642       | 1.085  | 0.691 | 1.842  | 0.463   |

Table 7.4: Error in pupil center prediction across various generalization tests proposed in Section 7.3.5. All results are presented in unit pixels.



| Train on / Test on | OpenEDS | NVGaze | UnityEyes | RITEyes-Gen | RITEyes-Nat | LPW | BAT | Fuhl | Swirski |
|--------------------|---------|--------|-----------|-------------|-------------|-----|-----|------|---------|
| OpenEDS            | 1.848   | 5.864  | 11.039    | 14.375      | 31.254      |     |     |      |         |
| NVGaze             | 4.249   | 0.300  | 9.760     | 15.680      | 31.174      |     |     |      |         |
| UnityEyes          | 7.438   | 3.458  | 0.651     | 9.790       | 16.928      |     |     |      |         |
| RITEyes-Gen        | 3.885   | 2.749  | 4.047     | 0.702       | 1.099       |     |     |      |         |
| RITEyes-Nat        | 4.404   | 2.741  | 2.413     | 0.618       | 0.610       |     |     |      |         |
| LPW                |         |        |           |             |             |     |     |      |         |
| BAT                |         |        |           |             |             |     |     |      |         |
| Fuhl               |         |        |           |             |             |     |     |      |         |
| Swirski            |         |        |           |             |             |     |     |      |         |
| all-OpenEDS        | 4.225   | 0.353  | 1.364     | 0.581       | 0.715       |     |     |      |         |
| all-NVGaze         | 1.860   | 2.731  | 1.643     | 0.721       | 0.853       |     |     |      |         |
| all-UnityEyes      | 1.903   | 0.367  | 3.582     | 0.663       | 0.800       |     |     |      |         |
| all-RITEyes-Gen    | 2.060   | 0.462  | 1.697     | 0.997       | 1.064       |     |     |      |         |
| all-RITEyes-Nat    | 2.121   | 0.430  | 1.928     | 0.893       | 1.513       |     |     |      |         |
| all-LPW            | 2.097   | 0.413  | 1.501     | 0.716       | 0.874       |     |     |      |         |
| all-BAT            | 1.858   | 0.399  | 1.430     | 0.698       | 0.829       |     |     |      |         |
| all-Fuhl           | 2.037   | 0.405  | 1.620     | 0.645       | 0.783       |     |     |      |         |
| all-Swirski        | 1.856   | 0.441  | 1.427     | 0.689       | 0.822       |     |     |      |         |
| all                | 2.171   | 0.441  | 1.460     | 0.883       | 1.033       |     |     |      |         |

Table 7.5: Error in iris center prediction across various generalization tests proposed in Section 7.3.5. All results are presented in unit pixels.

| Train on / Test on | OpenEDS | NVGaze | UnityEyes | RITEyes-Gen | RITEyes-Nat | LPW | BAT | Fuhl | Swirski |
|--------------------|---------|--------|-----------|-------------|-------------|-----|-----|------|---------|
| OpenEDS            | 0.956   | 0.572  |           | 0.550       | 0.427       |     |     |      |         |
| NVGaze             | 0.664   | 0.986  |           | 0.570       | 0.355       |     |     |      |         |
| UnityEyes          |         |        |           |             |             |     |     |      |         |
| RITEyes-Gen        | 0.713   | 0.883  |           | 0.960       | 0.932       |     |     |      |         |
| RITEyes-Nat        | 0.812   | 0.917  |           | 0.954       | 0.948       |     |     |      |         |
| LPW                |         |        |           |             |             |     |     |      |         |
| Santini            |         |        |           |             |             |     |     |      |         |
| Fuhl               |         |        |           |             |             |     |     |      |         |
| Swirski            |         |        |           |             |             |     |     |      |         |
| all-OpenEDS        | 0.688   | 0.486  |           | 0.822       | 0.850       |     |     |      |         |
| all-NVGaze         | 0.920   | 0.428  |           | 0.729       | 0.757       |     |     |      |         |
| all-UnityEyes      | 0.949   | 0.950  |           | 0.950       | 0.937       |     |     |      |         |
| all-RITEyes-Gen    | 0.933   | 0.442  |           | 0.685       | 0.787       |     |     |      |         |
| all-RITEyes-Nat    | 0.939   | 0.466  |           | 0.802       | 0.698       |     |     |      |         |
| all-LPW            | 0.952   | 0.967  |           | 0.954       | 0.942       |     |     |      |         |
| all-Santini        | 0.952   | 0.963  |           | 0.957       | 0.946       |     |     |      |         |
| all-Fuhl           | 0.949   | 0.958  |           | 0.955       | 0.943       |     |     |      |         |
| all-Swirski        | 0.950   | 0.954  |           | 0.952       | 0.940       |     |     |      |         |
| all                | 0.956   | 0.982  |           | 0.961       | 0.947       |     |     |      |         |

Table 7.6: Segmentation results of various generalization tests proposed in Section 7.3.5 when utilizing Batch Normalization. All results represent the IoU metric.

| Train on / Test on | OpenEDS | NVGaze | UnityEyes | RITEyes-Gen | RITEyes-Nat | LPW   | BAT   | Fuhl   | Swirski |
|--------------------|---------|--------|-----------|-------------|-------------|-------|-------|--------|---------|
| OpenEDS            | 0.614   | 2.102  |           | 8.346       | 18.620      | 1.349 | 1.951 | 4.920  | 1.583   |
| NVGaze             | 1.031   | 0.201  |           | 13.952      | 61.093      | 4.483 | 2.139 | 67.145 | 4.146   |
| UnityEyes          |         |        |           |             |             |       |       |        |         |
| RITEyes-Gen        | 0.732   | 1.262  |           | 0.302       | 0.720       | 1.735 | 2.375 | 32.363 | 1.160   |
| RITEyes-Nat        | 0.801   | 0.681  |           | 0.355       | 0.481       | 1.581 | 2.243 | 6.457  | 2.001   |
| LPW                | 0.737   | 1.448  |           | 3.205       | 8.256       | 0.638 | 1.694 | 5.471  | 1.022   |
| Santini            | 1.740   | 1.838  |           | 3.256       | 4.458       | 2.634 | 0.494 | 3.994  | 1.633   |
| Fuhl               | 1.313   | 1.796  |           | 3.469       | 5.272       | 2.696 | 1.045 | 1.724  | 1.740   |
| Swirski            | 2.003   | 2.723  |           | 3.818       | 5.200       | 4.189 | 0.747 | 7.593  | 0.616   |
| all-OpenEDS        | 1.387   | 0.637  |           | 0.717       | 0.924       | 1.260 | 0.688 | 1.771  | 0.437   |
| all-NVGaze         | 1.088   | 1.196  |           | 0.767       | 0.897       | 1.089 | 0.585 | 2.060  | 0.704   |
| all-UnityEyes      | 1.091   | 0.660  |           | 0.544       | 0.674       | 0.826 | 0.529 | 1.798  | 0.392   |
| all-RITEyes-Gen    | 1.614   | 0.820  |           | 0.957       | 1.051       | 1.166 | 0.492 | 1.701  | 0.447   |
| all-RITEyes-Nat    | 1.242   | 0.725  |           | 0.786       | 1.213       | 0.986 | 0.623 | 1.918  | 0.530   |
| all-LPW            | 1.071   | 0.353  |           | 0.462       | 0.600       | 1.310 | 0.419 | 1.718  | 0.265   |
| all-Santini        | 0.829   | 0.306  |           | 0.499       | 0.657       | 0.822 | 1.390 | 1.937  | 0.328   |
| all-Fuhl           | 0.823   | 0.369  |           | 0.487       | 0.643       | 0.791 | 0.748 | 2.374  | 0.448   |
| all-Swirski        | 1.113   | 0.491  |           | 0.435       | 0.577       | 0.855 | 0.478 | 1.777  | 0.533   |
| all                | 1.190   | 0.645  |           | 0.673       | 0.835       | 1.156 | 0.934 | 1.915  | 0.732   |

Table 7.7: Error in pupil center prediction across various generalization tests proposed in Section 7.3.5 when utilizing Batch Normalization. All results are presented in unit pixels.

| Train on / Test on | OpenEDS | NVGaze | UnityEyes | RITEyes-Gen | RITEyes-Nat | LPW | BAT | Fuhl | Swirski |
|--------------------|---------|--------|-----------|-------------|-------------|-----|-----|------|---------|
| OpenEDS            | 1.782   | 5.021  | 27.025    | 21.102      | 31.861      |     |     |      |         |
| NVGaze             | 5.354   | 0.268  | 14.095    | 16.955      | 44.660      |     |     |      |         |
| UnityEyes          | 4.629   | 2.340  | 0.807     | 2.251       | 2.988       |     |     |      |         |
| RITEyes-Gen        | 4.568   | 2.687  | 10.048    | 0.598       | 0.928       |     |     |      |         |
| RITEyes-Nat        | 4.861   | 2.564  | 5.654     | 0.743       | 0.756       |     |     |      |         |
| LPW                |         |        |           |             |             |     |     |      |         |
| BAT                |         |        |           |             |             |     |     |      |         |
| Fuhl               |         |        |           |             |             |     |     |      |         |
| Swirski            |         |        |           |             |             |     |     |      |         |
| all-OpenEDS        | 3.823   | 1.610  | 1.920     | 1.111       | 1.268       |     |     |      |         |
| all-NVGaze         | 2.828   | 2.999  | 2.161     | 1.036       | 1.124       |     |     |      |         |
| all-UnityEyes      | 2.381   | 1.526  | 3.848     | 1.181       | 1.293       |     |     |      |         |
| all-RITEyes-Gen    | 2.793   | 1.789  | 1.782     | 1.074       | 1.140       |     |     |      |         |
| all-RITEyes-Nat    | 2.824   | 2.255  | 1.687     | 1.086       | 1.476       |     |     |      |         |
| all-LPW            | 2.347   | 0.763  | 1.774     | 0.879       | 0.982       |     |     |      |         |
| all-BAT            | 2.603   | 0.987  | 2.286     | 1.026       | 1.158       |     |     |      |         |
| all-Fuhl           | 2.721   | 0.847  | 2.047     | 0.858       | 0.957       |     |     |      |         |
| all-Swirski        | 2.301   | 1.092  | 2.241     | 1.034       | 1.115       |     |     |      |         |
| all                | 2.171   | 0.441  | 1.460     | 0.883       | 1.033       |     |     |      |         |

Table 7.8: Error in iris center prediction across various generalization tests proposed in Section 7.3.5 when utilizing Batch Normalization. All results are presented in unit pixels.

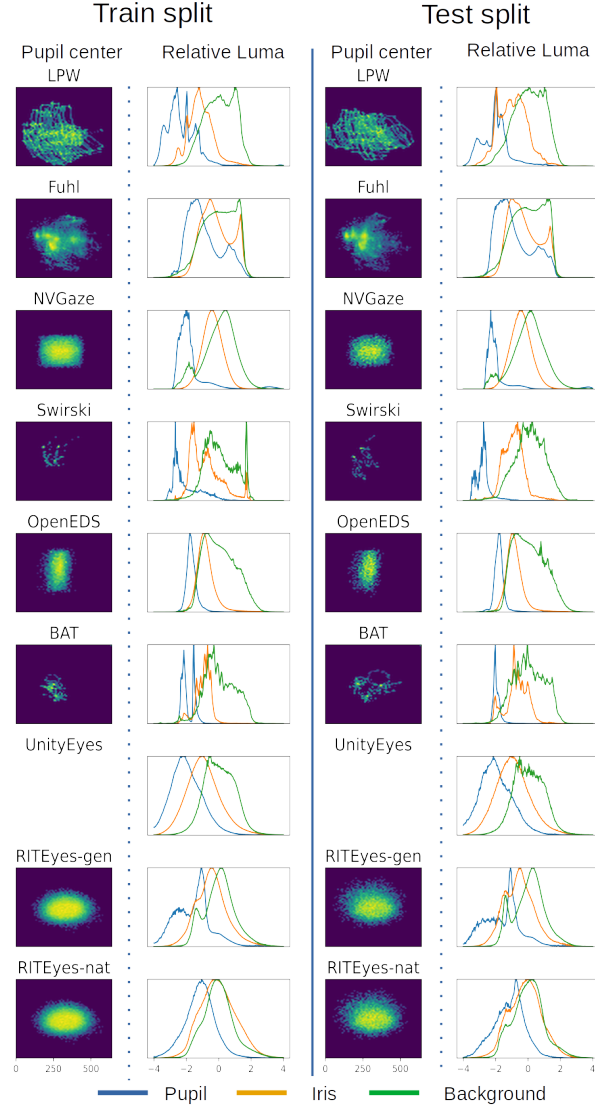


Figure 7.7: Pupil center (in pixels) and normalized luminance distribution (in Z-scores) of each eye part across all datasets utilized in our experiments (see Section 6.5). The left and right columns contain statistics from the training and test images for each domain respectively. Due to partial annotations present in some datasets, we leverage the all-vs-one model predictions to segment all eye images into pupil, iris and background segments. Luminance statistics are then accumulated from the predicted segmentation map.

## Chapter 8

# Summary and Conclusions

Applications of head mounted eyetrackers such as the study of eye and head movements are limited to controlled environments due to the lack of off-the-shelf tools for analyzing gaze events. Moreover, lack of gaze estimation solutions robust to environmental reflections and occlusions limits the application space of eyetrackers. This thesis proposal summarizes our efforts to mitigate these limitations.

For the development of head-free gaze event classifiers, we collected the Gaze-in-Wild dataset, a collection of eye and head movement data acquired from 19 participants while accomplishing every day activities such as indoor walking, ball catching, natural exploration and tea making (see Chapter 3). Manually annotated sequences of head-free gaze behavior was used to train our custom event detection algorithm based on recurrent neural networks. Chapter 4 details the performance of two machine learning algorithms for classifying these events and found that both achieved near human level performance for detecting gaze fixations and saccades, but they found it difficult to distinguish gaze pursuit behavior without additional contextual information otherwise available to human coders.

Reliable inference of human behavior during in-the-wild activities depends heavily on the quality of gaze data extracted from eyetrackers. Robustness to reflective artifacts and occlusions are necessary to ensure an uninterrupted track of reliable gaze features. Chapter 5 summarizes RITnet, an efficient encoder-decoder neural network which successfully segments an eye image into its constituent eye parts at 300Hz despite reflective artifacts from system optics. RITnet’s robustness against reflections is achieved due to well designed loss functions and data augmentation schemes.

Occlusion of the pupil or iris results in imprecise ellipse fits which can be detrimental for gaze estimation. Chapter 6 summarizes EllSeg, a pupil and iris ellipse segmentation framework which demonstrates robustness to occlusion. EllSeg improves upon various pupil and iris center estimation baselines as opposed to the standard eye part segmentation

approach. This framework can be incorporated with any encoder-decoder framework as a simple add-on module.

Chapter 7 summarizes EllSeg-Gen which explores generalization of a single model across various environmental reflections, subjects, gaze and eye camera positions. This work builds on the intuition that jointly training a model with multiple datasets learns a generalized representation of eye images and elliptical eye parts. We identify two approaches towards generalization, a) rely on multiset training to produce a single, robust model or b) pick the best performing model from a pool of pretrained, dataset-specific models. Results indicate that outdoor datasets which exhibit higher appearance variability significantly benefit from multiset optimization. In contrast, dataset-specific models generalize better onto indoor datasets which are representative of AR/VR headsets.

# Bibliography

- [1] Charles Stangor, Jennifer Walinga, et al. Introduction to psychology-1st canadian edition. 2018.
- [2] Brian Wandell and Stephen Thomas. Foundations of vision. *Psychcritiques*, 42(7), 1997.
- [3] Yves Le Grand. *Light, colour and vision*. Chapman & Hall; label on tp: Dover Publications, New York, 1957.
- [4] Kamran Binaee, Gabriel Diaz, Jeff Pelz, and Flip Phillips. Binocular eye tracking calibration during a virtual ball catching task using head mounted display. In *Proceedings of the acm symposium on applied perception*, pages 15–18, 2016.
- [5] Rakshit Kothari, Zhizhuo Yang, Christopher Kanan, Reynold Bailey, Jeff B Pelz, and Gabriel J Diaz. Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Scientific reports*, 10(1):1–18, 2020.
- [6] Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. Labeled pupils in the wild: A dataset for studying pupil detection in unconstrained environments. 2015.
- [7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 9351:234–241, 2015.
- [8] Stephan J. Garbin, Yiru Shen, Immo Schuetz, Robert Cavin, Gregory Hughes, and Sachin S. Talathi. OpenEDS: Open Eye Dataset. 2019.
- [9] Aayush K Chaudhary, Rakshit Kothari, Manoj Acharya, Shusil Dangi, Nitinraj Nair, Reynold Bailey, Christopher Kanan, Gabriel Diaz, and Jeff B Pelz. Ritnet: real-time semantic segmentation of the eye for gaze tracking. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3698–3702. IEEE, 2019.



- [10] Rakshit S Kothari, Aayush K Chaudhary, Reynold J Bailey, Jeff B Pelz, and Gabriel J Diaz. Ellseg: An ellipse segmentation framework for robust gaze tracking. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2757–2767, 2021.
- [11] Ignace TC Hooge, Diederick C Niehorster, Marcus Nyström, Richard Andersson, and Roy S Hessels. Is human classification by experienced untrained observers a gold standard in fixation detection? *Behavior research methods*, pages 1–18, 2017.
- [12] Raimondas Zemblys, Diederick C Niehorster, and Kenneth Holmqvist. gazeNet : End-to-end eye-movement event detection with deep neural networks. (2010), 2018.
- [13] Alfred L Yarbus. Eye movements during perception of complex objects. In *Eye movements and vision*, pages 171–211. Springer, 1967.
- [14] Benjamin W. Tatler, Nicholas J. Wade, Hoi Kwan, John M. Findlay, and Boris M. Velichkovsky. Yarbus, eye movements, and vision. *i-Perception*, 2010.
- [15] A. Borji and L. Itti. Defending Yarbus: Eye movements reveal observers’ task. *Journal of Vision*, 2014.
- [16] Guy Thomas Buswell. How people look at pictures: a study of the psychology and perception in art. 1935.
- [17] Mary M. Hayhoe, Travis McKinney, Kelly Chajka, and Jeff B. Pelz. Predictive eye movements in natural vision. *Experimental Brain Research*, 217(1):125–136, 2012.
- [18] Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 9(4):188–194, 2005.
- [19] Robert T. Held, Emily A. Cooper, and Martin S. Banks. Blur and disparity are complementary cues to depth. *Current Biology*, 22(5):426–431, 2012.
- [20] Sergei Gepshtein, Johannes Burge, Marc O. Ernst, and Martin S. Banks. The combination of vision and touch depends on spatial proximity. *Journal of Vision*, 5(11):1013–1023, 2005.
- [21] William W. Sprague, Emily A. Cooper, Ivana Tošić, and Martin S. Banks. Stereopsis is adaptive for the natural environment. *Science Advances*, 1(4), 2015.
- [22] Jonathan Samir Matthis, Jacob L. Yates, and Mary M. Hayhoe. Gaze and the Control of Foot Placement When Walking in Natural Terrain. *Current Biology*, 28(8):1224–1233, 2018.

- [23] Kamran Binaee and Gabriel J. Diaz. Assessment of an augmented reality apparatus for the study of visually guided walking and obstacle crossing. *Behavior Research Methods*, 51(2):523–531, 2019.
- [24] Rakshit Kothari, Kamran Binaee, Jonathan S. Matthis, Reynold Bailey, and Gabriel J. Diaz. Novel apparatus for investigation of eye movements when walking in the presence of 3D projected obstacles. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, volume 14, pages 261–266, 2016.
- [25] P. M. Daye, G. Blohm, and P. Lefevre. Catch-up saccades in head-unrestrained conditions reveal that saccade amplitude is corrected using an internal model of target movement. *Journal of Vision*, 14(1):12–12, 2014.
- [26] Edward G Freedman. Coordination of the eyes and head during visual orienting. *Experimental Brain Research*, 190(4):369–387, 10 2008.
- [27] Wolfgang Einhäuser, Frank Schumann, Stanislavs Bardins, Klaus Bartl, Guido Böning, Erich Schneider, and Peter König. *Human eye-head co-ordination in natural exploration*, volume 18. Network, 2007.
- [28] David L. Mann, Wayne Spratford, and Bruce Abernethy. The Head Tracks and Gaze Predicts: How the World’s Best Batters Hit a Ball. *PLoS ONE*, 2013.
- [29] Dana H. Ballard and Mary M. Hayhoe. Modelling the role of task in the control of gaze. *Visual Cognition*, 17(6-7):1185–1204, 2009.
- [30] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. PuRe: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding*, 170(February):40–50, 2018.
- [31] Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophey, Virginia L. Flanagan, Peter zu Eulenburg, and Seyed-Ahmad Ahmadi. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of Neuroscience Methods*, 324:108307, 2019.
- [32] Wolfgang Fuhl, Thomas Kübler, Katrin Sippel, Wolfgang Rosenstiel, and Enkelejda Kasneci. Excuse: Robust pupil detection in real-world scenarios. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9256:39–51, 2015.
- [33] Stephen Lucian Polyak. The retina. 1941.

- [34] Harry Moss Traquair and George Ian Scott. *Clinical perimetry*. London, 1957.
- [35] H Kenneth Walker, W Dallas Hall, and J Willis Hurst. Clinical methods: the history, physical, and laboratory examinations. 1990.
- [36] Michael F. Land and Benjamin W. Tatler. The human eye movement repertoire. In *Looking and Acting Vision and eye movements in natural behaviour*, pages 13–25. Oxford University Press, 7 2009.
- [37] Roger HS Carpenter. *Movements of the Eyes, 2nd Rev.* Pion Limited, 1988.
- [38] G.R. R. Barnes. Visual-vestibular interaction in the control of head and eye movement: The role of visual feedback and predictive mechanisms. *Progress in Neurobiology*, 41(4):435–472, 10 1993.
- [39] Craig H. Meyer, Adrian G. Lasker, and David A. Robinson. The upper limit of human smooth pursuit velocity. *Vision Research*, 25(4):561–563, 1985.
- [40] G. R. Barnes and J. F. Lawson. Head-free pursuit in the human of a visual target moving in a pseudo-random manner. *The Journal of physiology*, 410(1):137–55, 3 1989.
- [41] G. R. Barnes. Vestibulo-ocular function during co-ordinated head and eye movements to acquire visual targets. *The Journal of Physiology*, 287(1):127–147, 2 1979.
- [42] Anup Doshi and Mohan M. Trivedi. Head and eye gaze dynamics during visual attention shifts in complex environments. *Journal of Vision*, 12(2):1–16, 2012.
- [43] Yuki Kishita, Hiroshi Ueda, and Makio Kashino. Eye and Head Movements of Elite Baseball Players in Real Batting. *Frontiers in Sports and Active Living*, 2(January):1–12, 2020.
- [44] Hewitt D. Crane and Carroll M. Steele. Generation-V dual-Purkinje-image eye-tracker. *Applied Optics*, 24(4):527, 1985.
- [45] Laurence R Young and David Sheena. Survey of eye movement recording methods. *Behavior research methods & instrumentation*, 7(5):397–429, 1975.
- [46] Laura Sesma, Arantxa Villanueva, and Rafael Cabeza. Evaluation of pupil center-eye corner vector for gaze estimation using a web cam. In *Eye Tracking Research and Applications Symposium (ETRA)*, pages 217–220, 2012.

- [47] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [48] Jianzhong Wang, Guangyue Zhang, and Jiadong Shi. Pupil and glint detection using wearable camera sensor and near-infrared led array. *Sensors*, 15(12):30126–30141, 2015.
- [49] Lech Świrski and Neil Dodgson. A fully-automatic, temporal approach to single camera, glint-free 3D eye model fitting. *Proc. PETMEI*, 2013.
- [50] Wolfgang Fuhl, David Geisler, Thiago Santini, Tobias Appel, Wolfgang Rosenstiel, and Enkelejda Kasneci. CBF: Circular binary features for robust and real-time pupil center detection. *Eye Tracking Research and Applications Symposium (ETRA)*, 2018.
- [51] David A. Atchison and Larry N. Thibos. Optical models of the human eye. *Clinical and Experimental Optometry*, 99(2):99–106, 2016.
- [52] Pablo Artal. Optics of the eye and its impact in vision: a tutorial. *Advances in Optics and Photonics*, 6(3):340–367, 2014.
- [53] Hikmet Basmak, Afsun Sahin, Nilgun Yildirim, Thanos D Papakostas, A John Kanellopoulos, et al. Measurement of angle kappa with synoptophore and orbscan ii in a normal population. *Journal of Refractive Surgery*, 23(5):456–460, 2007.
- [54] Reza Safaei-Rad, Ivo Tchoukanov, Kenneth Carless Smith, and Bensiyon Benhabib. Three-Dimensional Location Estimation of Circular Features for Machine Vision, 1992.
- [55] Kai Dierkes, Moritz Kassner, and Andreas Bulling. A fast approach to refraction-aware eye-model fitting and gaze prediction. (June):1–9, 2019.
- [56] Kai Dierkes, Moritz Kassner, and Andreas Bulling. A novel approach to single camera, glint-free 3D eye model fitting including corneal refraction. *Eye Tracking Research and Applications Symposium (ETRA)*, (June), 2018.
- [57] Erroll Wood and Andreas Bulling. EyeTab. pages 207–210, 2014.
- [58] Arantxa Villanueva, Juan J Cerrolaza, and Rafael Cabeza. Geometry Issues of Gaze Estimation. *Intechopencom*, 4555:1006–1015, 2007.
- [59] Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci, Wolfgang Rosenstiel, and Enkelejda Kasneci. PupilNet v2.0: Convolutional Neural Networks for CPU based real time Robust Pupil Detection. 2017.

- [60] Shaharam Eivazi, Thiago Santini, Alireza Keshavarzi, Thomas Kübler, and Andrea Mazzei. Improving real-time CNN-based pupil detection through domain-specific data augmentation. pages 1–6, 2019.
- [61] F. J. Vera-Olmos, E. Pardo, H. Melero, and N. Malpica. DeepEye: Deep convolutional network for pupil detection in real environments. *Integrated Computer-Aided Engineering*, 26(1):85–95, 2018.
- [62] Jianzhong Wang, Guangyue Zhang, and Jiadong Shi. 2d gaze estimation based on pupil-glint vector using an artificial neural network. *Applied Sciences*, 6(6):174, 2016.
- [63] Wolfgang Fuhl, Wolfgang Rosenstiel, and Enkelejda Kasneci. *500,000 Images Closer to Eyelid and Pupil Segmentation*, volume 11678 LNCS of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2019.
- [64] Wolfgang Fuhl, Thiago Santini, and Enkelejda Kasneci. Fast & robust eyelid outline & aperture detection in real-world scenarios. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, pages 1089–1097, 2017.
- [65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.
- [66] Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini De Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. NVGaze. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 12:1–12, 2019.
- [67] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [68] Wolfgang Fuhl, Thiago C. Santini, Thomas Kübler, and Enkelejda Kasneci. ElSe: Ellipse selection for robust pupil detection in real-world environments. In *Eye Tracking Research and Applications Symposium (ETRA)*, volume 14, pages 123–130, 2016.
- [69] Zhengyang Wu, Srivignesh Rajendran, Tarrence van As, Joelle Zimmermann, Vijay Badrinarayanan, and Andrew Rabinovich. EyeNet: A Multi-Task Network for Off-Axis Eye Gaze Estimation and User Understanding. 2019.
- [70] Aayush Chaudhary and Jeff Pelz. Motion tracking of iris features to detect small eye movements Rochester Institute of Technology. *Journal of Eye Movement Research*, 12(6):1–18, 2019.

- [71] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation, 2018.
- [72] Simon Jegou, Michal Drozdal, David Vazquez, Adriana Romero, and Yoshua Bengio. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017-July:1175–1183, 2017.
- [73] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.
- [74] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot Adaptive Gaze Estimation. 2019.
- [75] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.
- [76] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.
- [77] Seonwook Park, Adrian Spurr, and Otmar Hilliges. Deep Pictorial Gaze Estimation. volume 11217 LNCS, pages 741–757. 2018.
- [78] M Mohri, A Talwalkar, and A Rostamizadeh. Foundations of machine learning (adaptive computation and machine learning series), 2018.
- [79] Jonathan S. Matthis and Brett R. Fajen. Visual control of foot placement when walking over complex terrain. *Journal of Experimental Psychology: Human Perception and Performance*, 2014.
- [80] Nitinraj Nair, Rakshit Kothari, Aayush K Chaudhary, Zhizhuo Yang, Gabriel J Diaz, Jeff B Pelz, and Reynold J Bailey. Rit-eyes: Rendering of near-eye images for eye-tracking applications. In *ACM Symposium on Applied Perception 2020*, pages 1–9, 2020.
- [81] Erroll Wood, Tadas Baltruaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of Eyes for Eye-Shape Registration and Gaze Estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3756–3764, New York, NY, 12 2015. IEEE.

- [82] Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the symposium on Eye tracking research & applications - ETRA '00*, pages 71–78, 2000.
- [83] Raimondas Zemblys, Diederick C. Niehorster, Oleg Komogortsev, and Kenneth Holmqvist. Using machine learning to detect events in eye-tracking data. *Behavior Research Methods*, 50(1):160–181, 2018.
- [84] Jami Pekkanen and Otto Lappi. A new and general approach to signal denoising and eye movement classification based on segmented linear regression. *Scientific Reports*, 7(1):1–13, 2017.
- [85] R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.
- [86] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
- [87] Julie Epelboim, Robert M. Steinman, Eileen Kowler, Mark Edwards, Zygmunt Pizlo, Casper J. Erkelens, and Han Collewyn. The function of visual search and memory in sequential looking tasks. *Vision Research*, 35(23-24):3401–3422, 1995.
- [88] Yu Fang, Ryoichi Nakashima, Kazumichi Matsumiya, Ichiro Kuriki, and Satoshi Shioiri. Eye-head coordination for visual cognitive processing. *PLoS ONE*, 10(3):1–17, 2015.
- [89] Robert S. Allison, Moshe Eizenman, and Bob S.K. Cheung. Combined head and eye tracking system for dynamic testing of the vestibular system. *IEEE Transactions on Biomedical Engineering*, 43(11):1073–1082, 1996.
- [90] Thomas Kinsman, Karen Evans, Glenn Sweeney, Tommy Keane, and Jeff Pelz. Ego-motion compensation improves fixation detection in wearable eye tracking. *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '12*, page 221, 2012.
- [91] Linnéa Larsson, Andrea Schwaller, Marcus Nyström, and Martin Stridh. Head movement compensation and multi-modal event detection in eye-tracking data for unconstrained head movements. *Journal of Neuroscience Methods*, 274:13–26, 2016.

- [92] Matteo Tomasi, Shrinivas Pundlik, Alex R. Bowers, Eli Peli, and Gang Luo. Mobile gaze tracking system for outdoor walking behavioral studies. *Journal of Vision*, 16(3):27, 2016.
- [93] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011.
- [94] Otto Lappi. Eye movements in the wild: Oculomotor control, gaze behavior & frames of reference. *Neuroscience and Biobehavioral Reviews*, 69:49–68, 2016.
- [95] Roy S. Hessels, Diederick C. Niehorster, Marcus Nyström, Richard Andersson, and Ignace T.C. Hooge. Is the eye-movement field confused about fixations and saccades? A survey among 124 researchers. *Royal Society Open Science*, 5(8):180502, 2018.
- [96] A. A. Skavenski, R. M. Hansen, R. M. Steinman, and B. J. Winterson. Quality of retinal image stabilization during small natural and artificial body rotations in man. *Vision Research*, 19:675–683, 1979.
- [97] Susana Martinez-Conde, Stephen L. Macknik, and David H. Hubel. The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5(3):229–240, 2004.
- [98] Dora E. Angelaki. Eyes on Target: What Neurons Must do for the Vestibuloocular Reflex During Linear Motion. *Journal of Neurophysiology*, 92(1):20–35, 2004.
- [99] Dora E. Angelaki. Three-Dimensional Ocular Kinematics During Eccentric Rotations: Evidence for Functional Rather Than Mechanical Constraints. *Journal of Neurophysiology*, 89(5):2685–2696, 2006.
- [100] MJ Mustari and S Ono. Optokinetic eye movements. In *Encyclopedia of Neuroscience*. Elsevier Ltd, 2010.
- [101] Rochelle Ackerley and Graham R. Barnes. The interaction of visual, vestibular and extra-retinal mechanisms in the control of head and gaze during head-free pursuit. *Journal of Physiology*, 589(7):1627–1642, 2011.
- [102] Michael F. Land and Mary Hayhoe. In what ways do eye movements contribute to everyday activities? In *Vision Research*, 2001.
- [103] Moritz Kassner, William Patera, and Andreas Bulling. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. *UbiComp*



- 2014 - *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1151–1160, 2014.
- [104] Jeff Rowberg. I2c device library.
  - [105] Luis Enrique Ortiz, Viviana Elizabeth Cabrera, and Luiz M G Goncalves. Depth Data Error Modeling of the ZED 3D Vision Sensor from Stereolabs. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 17(1):1–15, 2018.
  - [106] J L Vercher and Gabriel M Gauthier. Eye-head movement coordination: vestibulo-ocular reflex suppression with head-fixed target fixation. *Journal of vestibular research : equilibrium & orientation*, 1(2):161–70, 1991.
  - [107] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
  - [108] A. Terry Bahill, Jeffrey S. Kallman, and Jon E. Lieberman. Frequency limitations of the two-point central difference differentiation algorithm. *Biological Cybernetics*, 45(1):1–4, 1982.
  - [109] BL Zuber, JL Semmlow, and L Stark. Frequency characteristics of the saccadic eye movement. *Biophysical Journal*, 8(11):1288, 1968.
  - [110] Luis F. Chaparro. *Signals and Systems Using MATLAB: Second Edition*. 2015.
  - [111] Sylvain Paris. A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH 2007 courses on - SIGGRAPH '07*, 2007.
  - [112] Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. In the pursuit of (ground) truth: A hand-labelling tool for eye movements recorded during dynamic scene viewing. *Proceedings of the 2nd Workshop on Eye Tracking and Visualization, ETVIS 2016*, pages 65–68, 2017.
  - [113] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
  - [114] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods*, 49(2):616–637, 2017.
  - [115] Leo Breiman. Random Forests. *Machine Learning*, 1999.

- [116] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. pages 1–9, 2014.
- [117] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10553 LNCS:240–248, 2017.
- [118] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *AIP Conference Proceedings*, 1631(2):58–62, 12 2014.
- [119] Sabrina Hoppe and Andreas Bulling. End-to-End Eye Movement Detection Using Convolutional Neural Networks. 2016.
- [120] David M W Powers. The problem with kappa. *Conference of the European Chapter of the Association for Computational Linguistics*, pages 345–355, 2012.
- [121] Thomas Kisler and Uwe D Reichel. A dialect distance metric based on string and temporal alignment. *Elektronische Sprachsignalverarbeitung*, pages 158–165, 2013.
- [122] Oleg V. Komogortsev and Alex Karpov. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods*, 45(1):203–215, 2013.
- [123] Thiago Santini, Wolfgang Fuhl, Thomas Kübler, and Enkelejda Kasneci. Bayesian Identification of Fixations, Saccades, and Smooth Pursuits. 2015.
- [124] Mehdi Daemi and J. Douglas Crawford. A kinematic model for 3-d head-free gaze-shifts. *Frontiers in Computational Neuroscience*, 9:1–18, 2015.
- [125] Qi Sun, Arie Kaufman, Anjul Patney, Li-Yi Wei, Omer Shapira, Jingwan Lu, Paul Asente, Suwen Zhu, Morgan Mcguire, and David Luebke. Towards virtual reality infinite walking. *ACM Transactions on Graphics*, 37(4):1–13, 7 2018.
- [126] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017-Janua, pages 2261–2269. IEEE, 7 2017.

- [127] Shusil Dangi and Cristian Linte. DenseUNet-K: A simplified Densely Connected Fully Convolutional Network for Image-to-Image Translation. [https://github.com/ShusilDangi/DenseUNet-K/blob/master/DenseUNet\\_K.pdf](https://github.com/ShusilDangi/DenseUNet-K/blob/master/DenseUNet_K.pdf), 9 2019.
- [128] Hoel Kervadec, Jihene Bouchtiba, Christian Desrosiers, Eric Granger, Jose Dolz, and Ismail Ben Ayed. Boundary loss for highly unbalanced segmentation. In *International conference on medical imaging with deep learning*, pages 285–296. PMLR, 2019.
- [129] Karel Zuiderveld. Contrast Limited Adaptive Histogram Equalization. In *Graphics Gems*. 1994.
- [130] Steven A Cholewiak, Gordon D Love, Pratul P Srinivasan, Ren Ng, and Martin S Banks. Chromablur: Rendering chromatic eye aberration improves accommodation and realism. *ACM Transactions on Graphics (TOG)*, 36(6):1–12, 2017.
- [131] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):179, 2016.
- [132] Andrew T Duchowski. Eye Tracking Methodology: Theory and Practice, 2017.
- [133] Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. Purest: robust pupil tracking for real-time pervasive eye tracking. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 1–5, 2018.
- [134] Lech Świrski, Andreas Bulling, and Neil Dodgson. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 173–176, 2012.
- [135] Jianfeng Li, Shigang Li, Tong Chen, and Yiguang Liu. A geometry-appearance-based pupil detection method for near-infrared head-mounted cameras. *IEEE Access*, 6:23242–23252, 2018.
- [136] Haiyuan Wu, Qian Chen, and Toshikazu Wada. Conic-based algorithm for visual line estimation from one image, 2004.
- [137] Alexander Plopski, Christian Nitschke, Kiyoshi Kiyokawa, Dieter Schmalstieg, and Haruo Takemura. Hybrid Eye Tracking: Combining Iris Contour and Corneal Imaging, 2015.

- [138] James K.Y. Ong and Thomas Haslwanter. Measuring torsional eye movements by tracking stable iris features. *Journal of Neuroscience Methods*, 2010.
- [139] Yuta Itoh and Gudrun Klinker. Interaction-free calibration for optical see-through head-mounted displays based on 3d eye localization. In *2014 IEEE symposium on 3d user interfaces (3DUI)*, pages 75–82. IEEE, 2014.
- [140] Marcus Nyström, Richard Andersson, Kenneth Holmqvist, and Joost Van De Weijer. The influence of calibration method and eye physiology on eyetracking data quality. *Behavior research methods*, 45(1):272–288, 2013.
- [141] Wolfgang Fuhl, Marc Tonsen, Andreas Bulling, and Enkelejda Kasneci. Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. *Mach. Vis. Appl.*, 2016.
- [142] Dan Witzner Hansen and Qiang Ji. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, 2010.
- [143] Susan M. Kolakowski and Jeff B. Pelz. Compensating for eye tracker camera movement. *Eye Tracking Research and Applications Symposium (ETRA)*, 2005(March):79–85, 2005.
- [144] Thiago Santini, Diederick C. Niehorster, and Enkelejda Kasneci. Get a grip: Slippage-robust and glint-free gaze estimation for real-time pervasive head-mounted eye tracking. *Eye Track. Res. Appl. Symp.*, 2019.
- [145] Pascal Bérard, Derek Bradley, Markus Gross, and Thabo Beeler. Lightweight eye capture using a parametric model. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.
- [146] Wolfgang Fuhl, Hong Gao, and Enkelejda Kasneci. Neural networks for optical vector and eye ball parameter estimation. In *ETRA Short Papers*, pages 4–1, 2020.
- [147] Wolfgang Fuhl, Hong Gao, and Enkelejda Kasneci. Tiny convolution, decision tree, and binary neuronal networks for robust and real time pupil outline estimation. In *ETRA Short Papers*, pages 5–1, 2020.
- [148] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

- [149] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.
- [150] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [151] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.
- [152] E. Riba, D. Mishkin, D. Ponsa, and G. Bradski E. Rublee. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020.
- [153] Samuel Arba Mosquera, Shweta Verma, and Colm McAlinden. Centration axis in refractive surgery. *Eye and Vision*, 2(1):4, 2015.
- [154] Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci, and Enkelejda Kasneci. PupilNet: Convolutional Neural Networks for Robust Pupil Detection. 2016.
- [155] Dilip K. Prasad, Maylor K.H. Leung, and Chai Quek. ElliFit: An unconstrained, non-iterative, least squares based geometric Ellipse Fitting method. *Pattern Recognition*, 46(5):1449–1465, 2013.
- [156] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [157] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkö, et al. The 2005 pascal visual object classes challenge. In *Machine Learning Challenges Workshop*, pages 117–176. Springer, 2005.
- [158] Thiago Santini, Wolfgang Fuhl, David Geisler, and Enkelejda Kasneci. Eyerectoo: Open-source software for real-time pervasive head-mounted eye tracking. In *VISIGRAPP (6: VISAPP)*, pages 96–101, 2017.
- [159] Rochester Institute of Technology. Research computing services, 2019.

- [160] Priya Kansal and Sabarinathan Devanathan. Eynet: Attention based convolutional encoder-decoder network for eye region segmentation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3688–3693. IEEE, 2019.
- [161] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.
- [162] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.
- [163] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.
- [164] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. *30th International Conference on Machine Learning, ICML 2013, (PART 1)*:10–18, 2013.
- [165] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.
- [166] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *arXiv preprint arXiv:2003.02460*, 2020.
- [167] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.
- [168] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.
- [169] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- [170] Jean Ponce, Tamara L Berg, Mark Everingham, David A Forsyth, Martial Hebert, Svetlana Lazebnik, Marcin Marszalek, Cordelia Schmid, Bryan C Russell, Antonio Torralba, et al. Dataset issues in object recognition. In *Toward category-level object recognition*, pages 29–48. Springer, 2006.

- [171] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7572 LNCS(PART 1):158–171, 2012.
- [172] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.
- [173] Wouter Marco Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [174] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
- [175] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.
- [176] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.
- [177] Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.
- [178] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.
- [179] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.
- [180] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [181] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2018.

- [182] Yiru Shen, Oleg Komogortsev, and Sachin S Talathi. Domain adaptation for eye segmentation. In *European Conference on Computer Vision*, pages 555–569. Springer, 2020.
- [183] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.
- [184] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [185] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by back-propagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.
- [186] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):998–1008, 2018.
- [187] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [188] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.
- [189] Bincheng Huang, Si Chen, Fan Zhou, Cheng Zhang, and Feng Zhang. Episodic Training for Domain Generalization Using Latent Domains. *Communications in Computer and Information Science*, 1397 CCIS:85–93, 2021.
- [190] Enkelejda Kasneci, Katrin Sippel, Kathrin Aehling, Martin Heister, Wolfgang Rosenstiel, Ulrich Schiefer, and Elena Papageorgiou. Driving with binocular visual field loss? a study on a supervised on-road parcours with simultaneous eye and head tracking. *PloS one*, 9(2):e87470, 2014.
- [191] Agostino Gibaldi, Vasha DuTell, and Martin S Banks. Solving parallax error for 3d eye tracking. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–4, 2021.



- [192] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [193] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Proceedings of the 32nd international conference on neural information processing systems*, pages 2488–2498, 2018.
- [194] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
- [195] Tom Eelbode, Jeroen Bertels, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops, and Matthew B Blaschko. Optimization for medical image segmentation: theory and practice when evaluating with dice score or jaccard index. *IEEE Transactions on Medical Imaging*, 39(11):3679–3690, 2020.
- [196] TT Tanimoto. An elementary mathematical theory of classification and prediction, ibm report (november, 1958), cited in: G. salton, automatic information organization and retrieval, 1968.
- [197] Alex Hernández-García and Peter König. Further advantages of data augmentation on convolutional neural networks. In *International Conference on Artificial Neural Networks*, pages 95–103. Springer, 2018.
- [198] Jakub Nalepa, Michal Marcinkiewicz, and Michal Kawulok. Data augmentation for brain-tumor segmentation: a review. *Frontiers in computational neuroscience*, 13:83, 2019.
- [199] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. <https://github.com/aleju/imgaug>, 2020. Online; accessed 01-Feb-2020.
- [200] Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization. 2020.