Fall 12-16-2017

RGB Image-Based Pupillary Diameter Tracking with Deep Convolutional Neural Networks

Chatchai Wangwiwattana
Southern Methodist University, cwangwiwatta@smu.edu

Follow this and additional works at: https://scholar.smu.edu/engineering_compsci_etds

Recommended Citation
https://scholar.smu.edu/engineering_compsci_etds/1

This Dissertation is brought to you for free and open access by the Computer Science and Engineering at SMU Scholar. It has been accepted for inclusion in Computer Science and Engineering Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
RGB IMAGE-BASED PUPILLARY DIAMETER TRACKING
WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Approved by:

Dr. Eric C. Larson
Assistant Professor of Computer Science

Dr. Sukumaran Nair
Professor of Computer Science

Dr. Jennifer Dworak
Associate Professor of Computer Science

Dr. Michael Hahsler
Assistant Professor of Computer Science

Dr. Panagiotis Papamichalis
Professor of Electrical Engineering
RGB IMAGE-BASED PUPILLARY DIAMETER TRACKING
WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

A Dissertation Presented to the Graduate Faculty of the
Lyle School of Engineering
Southern Methodist University
in
Partial Fulfillment of the Requirements
for the degree of
Doctor of Philosophy
with a
Major in Computer Science
by
Chatchai Wangwiwattana

B.S., Computer Science, University of the Thai Chamber of Commerce
Master of Interactive Technology, Southern Methodist University

Dec 16, 2017
Copyright (2017)

Chatchai Wangwiwattana

All Rights Reserved
ACKNOWLEDGMENTS

I am very grateful to be around fantastic people. I would not able to be here without the support from my family: to my parents (Bang-on and Ponchai) and to my brothers (Passkron and Sittichai). This work could not have been accomplished without the wisdom of my advisor, Prof. Eric C. Larson, who gave me help and guidance before I was even in the program. I am also grateful to the members of my committee for their patience and support. I would like to thank my friends Sohail Rafiqi, Xinyi Ding, and Rita Enami, for their feedback, cooperation, and of course friendship. Last but not least, I would like to thank the University of the Thai Chamber of Commerce, who financially supported me for the entire program.
Pupillary diameter monitoring has proven successful at objectively measuring cognitive load. This work presents three robust RGB video based pupillary diameter trackers and compares them for measuring cognitive load using commodity cameras. We investigate the use of modified starburst algorithm from previous work and propose two algorithms: 2-Level Snakuscules and a convolutional neural network which we call PupilNet. Our results show PupilNet outperforms other algorithms in measuring pupil dilation, is robust to various lighting conditions, and robust to different eye colors. In addition, this work explores various deep learning architecture for extracting the pupillary response as well as understand the network’s behavior with three visualization techniques: feature and filter map visualization, gradient ascent, and occluded heat map. The study shows promising results for extracting the pupillary response as objective cognitive load measurement.
TABLE OF CONTENTS

LIST OF FIGURES .................................................................................................................. ix
LIST OF TABLES ..................................................................................................................... xiv

CHAPTER

1. INTRODUCTION ................................................................................................................. 1
   1.1. Motivation ..................................................................................................................... 1
   1.1.1. Pupillometry as a Measurement of Brain Functions .............................................. 1
   1.1.2. Pupillometry in Human Computer Interaction ...................................................... 3
   1.1.3. Challenges with Pupillometry Today ................................................................. 4
   1.2. Broader Impacts of This Work ................................................................................. 5
   1.3. Contributions ........................................................................................................... 5
   1.4. Structure of the Dissertation ................................................................................... 8

2. BACKGROUND: THE BASIC HUMAN PHYSIOLOGY OF THE AUTONOMIC NERVOUS
   SYSTEM AND RELATIONSHIPS WITH PUPILLOMETRY AND HUMAN COGNITIVE
   PROCESSING .................................................................................................................... 9
   2.1. Autonomic Nervous System .................................................................................... 9
   2.2. Pupillary Anatomy and Pupillary Responses ....................................................... 10
      2.2.1. Pupillarity Light Reflex .................................................................................. 11
      2.2.2. Pupillary Accommodation Reflex ................................................................. 12
      2.2.3. Task-Evoked Pupillary Responses .................................................................. 12
   2.3. Linkage of Pupillary and CL-NE System ............................................................. 13
   2.4. Cognitive Load Measurement .............................................................................. 21
   2.5. Summary .................................................................................................................. 25

3. BACKGROUND: CONVOLUTION NEURAL NETWORKS AND DEEP LEARNING .......... 26
### 3.1. Artificial Neural Networks And Deep Learning Overview .................................. 26

3.1.1. Overview of Convolutional Neural Networks (CNNs / ConvNets) .................. 29

### 3.2. Optimization ................................................................. 34

3.2.0.1. Momentum ................................................................. 36

3.2.0.2. Nesterov Accelerated Gradient ........................................ 36

3.2.0.3. Adagrad ................................................................. 36

3.2.0.4. RMSprop ................................................................. 37

3.2.0.5. Adadelta ................................................................. 38

3.2.0.6. Adam ................................................................. 38

### 3.3. Activation Functions ....................................................... 40

3.3.1. Sigmoid (Logistic) ......................................................... 40

3.3.2. Hyperbolic Tangent (Tanh) .............................................. 40

3.3.3. Rectified Linear Unit (ReLU) ........................................... 40

3.3.4. Maxout ................................................................. 41

### 3.4. Loss Functions ............................................................. 42

3.4.1. L1 and L2 Loss ............................................................. 42

### 3.5. Regularization ............................................................. 43

3.5.1. L1 and L2 Regularization ................................................. 43

3.5.2. Max-Norm Constraints .................................................. 44

3.5.3. Dropout ................................................................. 44

### 3.6. Summary ................................................................. 45

### 4. PUPILLARY DIAMETER TRACKING ALGORITHMS ................................. 46

4.1. Introduction ................................................................. 46

4.2. Related works ............................................................. 47

4.3. Preprocessing and Segmentation ........................................... 48

4.4. Improved Modified StarBurst ............................................. 49
4.5. 2-Levels Snakuscules ................................................................. 51
4.6. PupilNet: Convolution Neural Network Structure .............................. 54
  4.6.1. Training and Optimization ...................................................... 56
4.7. Conclusion ................................................................................. 57
5. APPARATUS AND DATA COLLECTION .............................................. 58
  5.1. Introduction ................................................................................ 58
  5.2. Experimental setup ..................................................................... 58
  5.3. Environment setup ...................................................................... 59
  5.4. Equipment and Gaze Signal Conditioning ........................................ 60
    5.4.1. Tasks .................................................................................. 61
    5.4.2. Digit Span Task ................................................................... 62
    5.4.3. Writing Tasks ....................................................................... 63
  5.5. Ground Truth Selection ............................................................... 65
  5.6. Conclusion .................................................................................. 68
6. RESULTS OF PUPILLOMETRY TRACKING ALGORITHMS .................... 69
  6.1. Cross Validation .......................................................................... 69
  6.2. Results: Characterizing Pupil Dilation in the Digit Span Task .......... 69
    6.2.1. Within-subject $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$ Across Lighting .... 72
    6.2.2. Within-subject $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$ Across Different Eye
           Contrasts ........................................................................... 77
    6.2.3. $\Delta_{PN}$ for Within-subject and Across-subject Training ............. 78
  6.3. Results: Within-subject Modeling for Typing Behavior .................... 79
  6.4. Conclusion .................................................................................. 82
7. GOING DEEPER: OPTIMIZING PUPILNET WITH MODERN DEEP LEARN-
   ING TECHNIQUES ........................................................................... 83
  7.1. Experimental Convolutional Neural Network Architectures ............... 83
 LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Autonomic Nervous System Available at <a href="http://www.merckmanuals.com/professional">http://www.merckmanuals.com/professional</a>. Accessed April 15, 2017.</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Pupil Constriction and Dilation [74]</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>Original data used by Moon and Spencer showing Pupillary diameter changes to different luminance levels [80]</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Pupillary response during the digit span short-term memory task. The result is reported by Kahneman and Beatty [7]</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>Locus Coeruleus is located at the pond in midbrain. It distributes Norepinephrine to most of the brain [2]</td>
<td>17</td>
</tr>
<tr>
<td>2.6</td>
<td>Tonic LC activity and Yerkes-Dodson relationship which is too low or too high LC activity leads to lower in performance [2]</td>
<td>19</td>
</tr>
<tr>
<td>2.7</td>
<td>Relationship between tonic pupil diameter and baseline firing rate of an LC neuron in monkey. Pupil diameter measurements were taken by remote eye-tracking camera at each instant in time when the monkey achieved fixation of a visual spot during the signal [2,96].</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>A cartoon drawing of a biological neuron [52]</td>
<td>27</td>
</tr>
<tr>
<td>3.2</td>
<td>A mathematical model of a neuron</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>(Left) A regular 3-layer Neural Network. (Right) A typical CNN.</td>
<td>29</td>
</tr>
<tr>
<td>3.4</td>
<td>This example has three $3 \times 3$ filter and zero-padding (gray zero area around edges of the image). The stride has two data points. The input image is on the left most side (a blue matrix), and the outputs are on the right side (green matrices). At each convolution step, a filter starts at the most top left of the input, then shifts a window to the right two blocks at a time (stride is set to two). At each step, it calculates a sum of the element-wise multiplication of the input and filter to produce each output. [52]</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>(Left) A pooling layer. (Right) A max pooling layer</td>
<td>32</td>
</tr>
<tr>
<td>3.6</td>
<td>A typical convolutional neural network classifying a dog [67]</td>
<td>33</td>
</tr>
</tbody>
</table>
3.7 SGD optimization on loss surface contours SGD with momentum [52] ......... 39

3.8 (left) a standard neural network, (right) a neural network after applying dropout [109] ................................................................. 45

4.1 A comparison of an high resolution infrared image (a) (CASIA-Iris database V4. Available: http://www.cbsr.ia.ac.cn/china/IrisDatabasesCH.asp), and a RGB image from a phone front facing camera (b) ...................... 46

4.2 Extracting eyes the eyes with our pre-processing algorithm. ................. 49

4.3 Original Pupilware Algorithm [95] ........................................... 50

4.4 Difference in pixel intensity caused by shadow ......................... 51

4.5 An energy map that a snake can see. (left) the original image, (mid) an iris energy map, (right) a pupil energy map ............................... 53

4.6 Convolution Network ............................................................ 56

5.1 High contrast eye image(left) and low contrast eye image(right) . Here, eye color is used a proxy for eye contrast because eye color is typically the best indicator of contrast between pupil and iris. ...................... 59

5.2 Eye images in different lighting conditions. Histogram equalization has been applied to increase contrast. ........................................ 60

5.3 (a) blurred image caused by rapid movement. (b) rapid eye movement while recalling memory. (c) occlusion when participants looked down at the keyboard. (d) large screen reflection occluding an entire pupil when participant move too close the screen. ........................................ 61

5.4 Experimental setup summary ................................................. 62

5.5 This image shows typing task user interface. The green cursor moves to a next character when a participant types correctly, and the error is reported to the participant in real-time. .................................. 65

5.6 Averaged absolute pupil size affected by pupillary light reflex across each iteration and each participant in three lighting conditions ............. 67

5.7 Averaged percent change in pupil size affected by pupillary light reflex across each iteration and each participant in three lighting conditions ............. 67

5.8 Average of pupillary diameter in three lighting conditions ................ 68

6.1 Cross Validation Strategy ..................................................... 70
6.2 Comparison of PupilNet and gaze tracker for two different participants. Also shown are the measures used in evaluating pupil dilation inference. ........ 75

6.3 Summary of $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$. Results are separated by the average lighting level in the room, bright ($\approx 100$ lux), dim ($\approx 7.5$ lux), and dark ($\approx 5$ lux). Please note that the plots vary from -30% up to 50%. ............ 75

6.4 Results of correlation coefficient ................................................................. 76

6.5 Bland-Altman plot between With-in subject PupilNet and Gaze Tracker ...... 76

6.6 Summary of dilation percentage differences for individual with high contrast (5 subjects) and low contrast eyes (6 subjects). The distributions for each boxplot are calculated only in the bright lighting configuration. ...... 78

6.7 Results of across-subject leave one out cross validation for each model. An asterisk (*) indicated that the distributions are statistically different based on a two-tailed T-test ($p<0.01$) ........................................ 79

6.8 Gaze tracker data for each of the fours users during a subset of the typing task. ................................................................. 80

6.9 A box and swarm plot of the number of peaks for each 30-second interval grouped by touch and non-touch typing. ......................... 81

7.1 Original PupilNet .................................................................................. 85

7.2 An example of wider PupilNet ............................................................... 85

7.3 An example of deeper PupilNet ............................................................... 85

7.4 Typical residual block [39]. Input $X$ is features from previous layer. Weight layers are convolutional layer. [39] ........................................ 87

7.5 Basic Structure of Residual PupilNet .................................................... 87

7.6 Correlation coefficient of nine variance of PupilNet separating by lighting conditions. Each box contains 198 data points (best view in digital version) ................................................................. 89

7.7 Percent change of mean square error of nine variance of PupilNet separating by lighting conditions. Each box contain about 79,200 data points. (best view in digital version) .................................................. 90

7.8 A comparison of correlation coefficients between high contrast (top) and low contrast (bottom) eye. ................................. 92

7.9 A comparison of mean absolute error of percent change from baseline between high contrast (top) and low contrast (bottom) eye. ...................... 93
7.10 Modified residual plot comparison for nine model algorithms. Each horizontal line indicates a different level of the boundary of the critical area. What we see is that most of the data points are close to zero. PupilNet Residual2 and Residual3 are more spread out to the side, while Wide64 and Wide128 undergo almost no change. This means that PupilNet Residual2 and Residual3 are able to capture subtle changes caused by the TEPR, while Wide64 and Wide128 are unable to capture such changes.

7.11 A comparison between within-subject and across subject on PupilNet Residual2.

7.12 A comparison between within-subject and across-subject on PupilNet Residual2 separated by eye types.

8.1 left: a filter map of a first Conv layer of AlexNet network. right: a filter map of a fifth Conv Layer of AlexNet network [62].

8.2 The 32 3×3 learned filters of the first convolutional layer of Residual2. There are dominant colors in some neurons, highlighted in red. These might be filters that detect certain colors in the eye images. Unfortunately, they are too small to be able to gain meaningful insights from.

8.3 32×32 3×3 learned filters of the second convolutional layer of Residual2.

8.4 left: a feature map of a first AlexNet Conv layer. right: a feature map of a fifth AlexNet Conv Layer [62].

8.5 First ten layers of PupilNet Residual2 feature maps. Feature maps in most layers are localized and sparse. This might imply that the PupilNet Residual2 model is more likely to be successfully trained.

8.6 Artificially generated images that maximize neuron activation in the first convolutional layer (left). These neurons seem to react to vertical and horizontal edges and various colors (right). Neurons in the fourth layer are able to react to more complex structures around the pupil and iris.

8.7 Artificially generated images that maximize neuron activation in the tenth convolutional layer. Many pixels are clustered around the iris edges and pupils.

8.8 Left: the network considers the dogs face as a key area to classify a Pomeranian. Right: the network looks mostly at the dog area, and some parts of the woman facial area. [141].

8.9 (left) 5×5 patch over an eye image. (right) 1×1 patch over an eye image.

8.10 Attribution heat map of 5×5 patch image overlay on top of an input eye image (best viewed in electronic version).
8.11 Attribution heat map of 1×1 patch images overlay on top of an input eye image (best viewed in electronic version) .......................................................... 111

8.12 Sample heat maps of a participant with high contrast eyes. The top row shows heat maps from the original PupilNet at 10, 100, and 200 epochs. The bottom row shows heat maps from the Residual2 model. .................. 112

8.13 Sample heat maps of a participant with low contrast eyes. Original PupilNet at the top failed to detect the pupil area, but Residual2 is able to detect the iris area successfully. .......................................................... 113

8.14 Heat maps of dim and dark images. The network is able to search the iris and pupil area, but it is not able to extract the exact location .............. 113

8.15 A comparison of Residual2 3×3 and 7×7 filter size .............................. 115

8.16 Filter map visualization of the first convolutional layer of Residual2 7×7. High activation filters are highlighted in red. ......................... 116

8.17 (top) Features that each Residual2 7×7 neuron wants to see. It has a higher number of neurons activated in the first and fourth convolutional layers. It is also better at responding to large edges than Residual2. (bottom) The activating neurons in Residual2 3×3. ........................................ 117
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Pupillary Pharmacology</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Sources, descriptions, and documentation regarding variation in pupil size</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>from Warren’s work [121]</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Participant Demographics</td>
<td>59</td>
</tr>
<tr>
<td>5.2</td>
<td>Lighting conditions</td>
<td>59</td>
</tr>
<tr>
<td>7.1</td>
<td>Table summary of PupilNet architectures. Each column represents each</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>architectures of various version of PupilNet</td>
<td></td>
</tr>
<tr>
<td>7.2</td>
<td>Table summary of PupilNet architectures. Each column represents each</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>architectures of various version of Residual PupilNets</td>
<td></td>
</tr>
<tr>
<td>7.3</td>
<td>t-test values compared against each pair of networks. (* indicates significant</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>different ($p &lt; 0.05$))</td>
<td></td>
</tr>
</tbody>
</table>
dedication
1.1. Motivation

Psychologists for over 50 years have known the eyes can be used to communicate a tremendous amount of information about who we are, which is underutilized knowledge. The eyes are part of a complicated mechanism connecting with many areas of the nervous system and revealing behavioral cues through both subtle and overt changes. Pupillary monitoring (Pupillometry) is a process of observing pupil behaviors. It is the ubiquitous computing equivalent of infrastructure-mediated sensing [90]. The sub-system that we aim to improve our understanding in is the complexities associated with the nervous system and how it integrates with sensory signaling related to pupil function. For example, as we concentrate, focus, relax, become aroused, and generally think signaling noise can be observed originating from the pupil. While we cannot explain every artifact, we can map many pupillary responses to different processes of the brain.

1.1.1. Pupillometry as a Measurement of Brain Functions

Pupillometry is a well-established laboratory technique used in various areas of medicine and science including neuroscience, psychiatry, psychology, and even computer science [26, 36, 65]. Pupillometry involves the measurement of pupil size and relating changes to examinations of brain function. In this context, the pupil has a venerable history in medicine. Pupil examination has been used as a method for diagnosing various neurological syndromes such as Addison disease (adrenal glands produce insufficient levels of aldosterone and cortisol), Horner syndrome (a disrupted nerve pathway on one side from the brain to the face and eye.), and anisocoria (a condition characterized by an unequal size of the eyes’ pupils caused
from several serious brain dysfunction). In clinical settings, pupillometry can also be used for diagnosing certain drug effects (i.e. anesthesia), detecting optic nerved dysfunction, and evaluating brainstem integrity and autonomic functions [70, 78, 132] in patients with traumatic brain injury and stroke. Pupillometry has also been reportedly used to measure the depth of anesthesia [130]. Kahneman has also demonstrated that pupillometry can be used to objectively measure cognitive processing where pupillary diameter increases during high cognitive load, but decreases when increased cognitive load is removed. This phenomenon is known as task-evoked pupillary response [6, 7]. During 1980s pupillometry was extensively studied as the window to psychiatric disorders including both cognitive and affective disturbances [110]. Pupillometry is non-invasive and can be used in patients demonstrating verbal communication limitations (e.g., developmental populations such as autism spectrum disorders (ASD)). For example, pupils demonstrate decreased responsiveness in the setting of social reward (e.g., a picture of a smiley face) in children with autism or Asperger syndrome (a developmental disorder that causes difficulties in social interaction and non-verbal communication) [18, 29, 63, 63]. Use of pupillometry has also shown potential in being able to objectively measure improvement in behavioral treatments for Fragile X syndrome (a genetic condition that caused an intellectual disability) [64].

To date it remains unclear why observing changes in pupil size is able to provide physiological and psychological information. The underlying reasons may be attributable to a strong correlation between pupil size and tonic activation of the Locus-Coeruleus-Norepinephrine (LC-NE) system [3]. The LC-NE is responsible for modulating signals entering the neocortex, which is why the LC-NE is hypothesized to be involved in processes of cognition and arousal [2, 11, 48]. This has been taken to suggest that the pupil fluctuates in size with constant excitation of low level brain function. Although a direct link between these phenomena and the responses of the LC-NE is unclear, a growing body of evidence supports that a relationship exists. For example, the LC-NE is hypothesized to regulate arousal and help to focus attention through supplying norepinephrine to the pre-frontal lobe, which neuroscientists believe is the center of executive function. This is exciting because it means such
high level “thought” functions might partially manifest through the LC-NE while generating changes in pupils (discussed in detail in chapter 2).

1.1.2. Pupillometry in Human Computer Interaction

Human-computer interaction is reliant on technology in being able to recognize human emotion and cognitive state. Pupillometry is used as a general measure of thought processing activities, but can also be used to assess simultaneous physiological responses. In this context, pupillometry can be used as a tool to assess cognitive, affective, and contextual computing. For example, computers that may be able to understand the underlying mood of an individual would be an effective aid for amyotrophic lateral sclerosis patients (a nervous system disease that weakens muscles and impacts physical function). Pupillometry is not only useful for individuals with physical disabilities. For instance, brain-computer interface may be used to augment human cognitive ability. For example, in 2001, Defense Advanced Research Projects AugCog(AugCog) aimed to develop robust tools for monitoring cognitive state and integrating them with computer systems [104].

Because a large part of human communication is non-verbal, understanding human state of mind can be used to advance human communication in the digital age. For example, this may be used to advance our understanding of human to machine communication. Picard argued that computers need to understand human emotions to react more intelligently [92]. For example, a self-driving car might need to observe a driver before returning control of the car back to a human and, hence, avoid situations where drivers are distracted.

There are other methods of measuring human emotions and cognitive states, but pupillometry is more robust for measuring cognitive load. In general, computers can be used to perceive user emotion and cognitive processing by physiological measures such as heart rate(HR), electroencephalograph(EEG), galvanic skin response(GSR), and many other techniques. Among all forms of physiological responses, eyes are proven reliable to communicate human cognition [6,7,49]. Pupillometry is non-invasive and can be used without confounding environmental influences, whereas for example, fMRI requires a subject to be assessed while
wearing electrodes in a noisy machine.

1.1.3. Challenges with Pupillometry Today

While pupillometry has been studied for >50 years, understanding and utilization of this technique outside of laboratories and hospitals has been limited. The potential cause for this lack of translation may be because pupillometry is sensitive to external and internal stimuli [121]. The most prominent factors that may confound interpretation of pupillometry are lighting and eye fixation distance. It has been demonstrated that those factors independent or together can lead to changes in pupillary diameter up to 9 millimeters independent of changes in cognitive processing [7]. As such, those and other confounding influence can be minimized via designing tools to exclude external stimuli such as integrating a fixed length tube with a fixed internal light source (i.e. Medical pupillometers). Emotional stimuli such as associated with sex, stress, interest, and excitement can also lead to changes in pupillary diameter. Cost of pupillometry may also be a limitation. A monocular pupillometer typically costs upwards of $5000 USD. A pupillometer covers the eye with a silicon cylinder and bathes the eye in infrared light. This device is commonly used by ophthalmologists and used for eye exams and other diagnostic purposes such as to detect neurosyphilis. Similarly, another option is infrared-based eye trackers, which cost upwards of $10,000 USD and is the device of choice for most modern pupillometry studies. While our lab uses the Tobii Pro glasses 2, which retails for about $12,000 USD, most organizations and labs are unable to afford such equipment. Therefore, many hospitals still use subjective measures with a pupil diameter chart and a pen light. A recent study shows subjective measuring with a pen light to detect anisocoria performed by nursing staff is significantly worse than using a pupillometer [16].

In the future, we suggest that it would be cheaper and more pervasive if we can use a phone camera to monitoring pupil size. Our previous work has shown that a normal web camera under certain condition shows reasonable performance compared with a more expensive gaze tracker [94]. Nonetheless, RGB video frames tend to have lower resolution, signal to noise ratio, and are more sensitive to lighting conditions. Therefore, prior to
implementing phone cameras into the field further testing needs to be performed.

1.2. Broader Impacts of This Work

Having a robust and cost effective pupillometer would open up the potential for more studies across research settings. Cognitive science and psychology or even the medical community may be able to gain further insight into complex behaviors by analyzing pupillogram on a much larger scale. For example, doctors and psychiatrists may be able to use a smartphone to detect cognitive disorders such as ADHD, dyslexia, or Alzheimer as well as to evaluate medical interventions. Educators may also be able to objectively measure cognitive load for to improve instructional design. Game designers may be able to optimize players engagement or even incorporate pupillogram to improve game mechanics. Advertisers may be able to detect target audiences engagement for new advertising campaigns. A football coach may be able to use this technology for detecting concussion. These are just a handful of examples that pupillometry analysis may be useful for in the real-world.

1.3. Contributions

My dissertation work encompasses overlapping contributions with the dissertation work of Dr. Sohail Rafiqi. In particular, I collaborated with Dr. Rafiqi in creating the first PupilWare paper that investigated using RGB cameras to detect pupil size. We jointly proposed the Modified Starburst algorithm and a calibration procedure used in mobile devices. This algorithm has demonstrated reasonable effectiveness for tracking in a constant lighting condition, low reflection, and high contrast iris color. However, it was still unclear how the system performed in challenging situations such as low lighting conditions, noisy environments, with dark eyes, and with low-resolution images. All other work outside of the modified starburst algorithm is the sole contribution of this dissertation and not part of the collaboration between Dr. Rafiqi and myself.

In this dissertation, we propose methods to increase the robustness of RGB imaging based pupillary diameter tracking. That is, we ask, “How might we make RGB image based
pupillary tracker more robust to lighting and eye color?" We aim to assess whether our system can produce results with accuracy comparable to a suitable ground truth across various settings and environments. As a ground truth measurement device, we use a commercial-grade, high-quality gaze tracker. Using a gaze tracker for ground truth pupillometry has been long accepted in the scientific community and was popularized by Klinger et al. [58]. The contributions presented in this dissertation are enumerated as follows:

1. (Joint contribution with Rafiqi et al., [95]) This contribution includes the Modified Starburst (MSB) algorithm that enables web cameras and smartphones to track pupillary diameter in limited scenarios. This contribution also includes a classification procedure used to predict three levels of cognitive load in stable lighting conditions.

2. (Joint contribution with Rafiqi et al., [94]) In this work, we improve the robustness of the Modified Starburst against lighting variation for mobile devices. We utilize a data-driven approach to dynamically adjust hyper-parameters. As a result, the model improves robustness to eye color and eliminates the need for regular system calibration.

3. (Novel Contribution, algorithms) In this study, we present a new model-based algorithm, 2-Levels Snakucules (2-LS). This technique is based on the Snakucules algorithm originally proposed by Thevenaz and Unser [119], and was later used for iris detection [30]. We developed Snakucules in such a way that it not only detects iris location but also estimates pupil size with sub-pixel accuracy. The novelty of our work is a pupil snake. The pupil snake generates an energy map over the iris region, then estimates pupil size and location from the energy map. These novel changes help the 2-Levels snakucules algorithm perform better than a traditional Snakucules for tracking small changes in pupillary diameter in a low-resolution image.

4. (Novel Contribution, architecture) We present a Convolutional Neural Network architecture to track pupillary diameter (PupilNet). Whereas the Modified Starburst and 2-Levels Snakucules use traditional model-based segmentation techniques where features are hand-crafted by engineers, a convolutional network optimizes feature extraction for a given learning task. We show that the convolutional architecture significantly
outperforms the Modified Starburst and 2-Levels Snakucules algorithms.

5. (Novel Contribution, human subjects evaluation) We systematically compare the three algorithms from this dissertation. We compare performance of three algorithms in three lighting conditions (i.e. bright, dim, dark) and iris contrast (i.e. light, medium, and dark). Finally, we examine model generalization by testing with between-subject cross-validations. The result shows that convolutional neural network (PupilNet) outperforms other feature-based algorithms in all situations for real-world low-resolution RGB videos. This suggests that data-driven approach might be the technique to be further developed for automatic cognitive load measurement system.

6. (Novel Contribution, architecture) We investigate recent deep learning techniques for improving PupilNet performance including (i) wide convolutional neural networks, (ii) deep convolutional neural networks, and (iii) deep residual networks. Then, we evaluate whether or not it enhances the accuracy compared to original PupilNet architecture. The result show that deep and residual networks perform better than the original PupilNet algorithm in a variety of settings. Surprisingly, wide networks perform worse than original the PupilNet architecture. We hypothesize that this is because wide networks have more tendency to “memorize” the training data, thus overfitting.

7. (Novel Contribution, analysis of architecture) To better understand what the PupilNet architecture learns, we employ various neural network visualization techniques including feature and filter map visualization, maximized neuron output gradient ascent, and occluded 2D heat maps. The results show that PupilNet learn features related to pupil location discovery, iris edges finding, and iris color. This suggests that PupilNet performance is indeed from successful learning of the pupil edges, not random noise or artifacts of the experimentation. Finally, we use the visualizations to hypothesize useful changes in the PupilNet architecture. We demonstrate that these changes do indeed improve the network performance.

8. (Secondary contributions of importance to the field, but not directly related to pupil measurement) We discuss privacy implications of an automatic cognitive load sensing
system and a mitigation protocol to help keep cognitive load of an individual private.

1.4. Structure of the Dissertation

This proposal is structured in the following manner. In chapter 2, we introduce the background information for this dissertation concerning with basic physiology, pupillometry, and cognitive load measurement. In chapter 3, we introduce neural networks and convolutional neural networks. In chapter 4, we explain algorithms of the new Modified Starburst, 2-levels Snakucules, and PupilNet. In chapter 5, we describe data collection, experiment setup, and experiment apparatus. In chapter 6, we evaluate and compare a new Modified Starburst, 2-levels Snakucules, and PupilNet. In chapter 7, we experiment further on modern deplearning architecture. In chapter 8, we explore inside PupilNet with various visualization techniques. Finally, chapter 9 discussed what we learned and privacy implication.
BACKGROUND: THE BASIC HUMAN PHYSIOLOGY OF THE AUTONOMIC NERVOUS SYSTEM AND RELATIONSHIPS WITH PUPILLOMETRY AND HUMAN COGNITIVE PROCESSING

This chapter briefly explains the basic human physiology of the autonomic nervous system and relationships with pupillometry and human cognitive processing. It necessary to understand the contributions of this work.

2.1. Autonomic Nervous System

The two main branches of the human nervous system are the central nervous system and peripheral nervous system. The central nervous system consists of the brain and spinal cord, whereas the peripheral nervous system consists mainly of nerves. In contrast to the central nervous system, the peripheral nervous system is made up of three sub-systems, which are termed the somatic, enteric, and autonomic nervous systems. The somatic nervous system is associated with controlling voluntary movements, while the enteric nervous system is mainly in control of the gastrointestinal system. By contrast, the autonomic nervous system, which is responsible for controlling activities such as pupillary function, heart and sweat gland function, and stress and emotional responses, is further divided into two separate systems, which include the sympathetic and parasympathetic nervous systems.

The sympathetic nervous system is activated for emergency mobilized energy. The physiological origins of the sympathetic nervous system are the thoracic and lumbar regions of the spinal cord, while playing an important role in regulating many homeostatic mechanisms in the human body. When activated, the sympathetic nervous system prepares for a crisis that requires intense physical activity commonly known as the “fight-or-flight” response. As a result, the sympathetic nervous system innervates tissues in almost every organ system.
and leads to changes in organ function such as increased pupil diameter, heart rate, arousal, and energy production.

In contrast, the parasympathetic nervous system is activated when organisms are in a relaxed state. The origin of the parasympathetic nervous system is the brainstem and spinal cord regions. One of the primary responsibilities of the parasympathetic nervous system is to stimulate visceral activity associated with relaxation. This function of the parasympathetic nervous system is known as the “rest and digest” response. In general, organ system responses to parasympathetic nervous system activation are typically in contrast to organ system responses to stimulation of the sympathetic nervous system.

Nevertheless, although the sympathetic and parasympathetic nervous systems typically work independently, they may act in concert with one another to control a given stage of a complex process such as sexual activity. Thus, in summary, the sympathetic nervous system functions during actions requiring survival responses such as emergency, stress, or exertion, whereas the parasympathetic nervous system functions during activities requiring no immediate organ reaction (i.e., under resting conditions).

2.2. Pupillary Anatomy and Pupillary Responses

The human pupil is a nearly circular aperture at the center of the iris through which light passes to the retina. Pupillary constriction and dilation are controlled by two opposing sets of muscles, which include the circular fiber (sphincter pupillae) and radial fiber (dilator pupillae). Both circular and radial fiber muscles are governed by both branches of the autonomic nervous system [7].

The sympathetic nervous system is charge of dilating pupils. By contrast, the parasympathetic nervous system causes excitation of the dilator muscles causing the pupil to constrict. The motor nucleus for these muscles is the Edinger-Westphal nucleus located in the midbrain and is stimulated by norepinephrine release [131]. Activation of the sympathetic nervous system brings about pupillary dilation via two mechanisms: (i) activation of the cervical sympathetic nerve to the dilator muscles, and (ii) inhibition of parasympathetic
Table 2.1. Pupillary Pharmacology

<table>
<thead>
<tr>
<th></th>
<th>Parasympathetic (constricts)</th>
<th>Sympathetic (dilates)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receptor</td>
<td>muscarinic</td>
<td>adrenergic</td>
</tr>
<tr>
<td>Neurotransmitter</td>
<td>acetylcholine</td>
<td>norepinephrine</td>
</tr>
<tr>
<td>Agonist</td>
<td>pilocarpine</td>
<td>hydroxyamphetamine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>apraclonadine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cocaine(indirect)</td>
</tr>
<tr>
<td>Antagonist</td>
<td>scopolamine</td>
<td></td>
</tr>
</tbody>
</table>

nervous system activation.

An additional important pupil behavior is pupillary hippus or pupillary unrest, which is the rhythmic but irregular changing in pupillary diameter (usually less than 0.04 Hz). These types of oscillations are independent of eye movements or changes in illumination [77]. Therefore, these oscillations are typically interpreted as random noise and be discarded from analyses [89]. In contrast, Beatty and Brennis suggest pupillary hippus or pupillary unrest might reflect brain processes triggered by spontaneous or unobserved thought [7], suggesting analyzing hippus activity may have scientific relevance. For example, it has been demonstrated that oscillation activity becomes more profound in sleep-deprive individuals [71, 85, 128, 129, 137].

2.2.1. Pupillary Light Reflex

Light intensity determines pupillary response. The main function of the pupil is to control the amount of light coming into the eye. Pupil diameter defines the optical properties of the eye and also determines the amount of light falling on the retinal surface. The pupillary light reflex designates the mechanism by which the human pupil changes its diameter, which is inverse to the amount of light striking on the retina [12]. Under normal conditions, when light intensity increases stronger activation of the parasympathetic nervous system occurs, which leads to constriction of the circular arranged pupillary sphincter muscle (with a delay).
The consequence is reduced pupil diameter. Conversely, during low light, parasympathetic nervous system stimulation acts to constrict dilator muscles causing pupils to dilate.

Pupil diameter can vary from less than 1 mm to more than 9 mm depending on the amount of light enters the eye. These constrictions can occur withing 0.2 sec, with peak response occurring from 0.5 to 1.0 sec later. [7] In darkness, the pupil can enlarge to an average size of about 7 mm with a standard deviation of about 0.9 mm [73]. In standard light condition, its average size is about 3 mm [134]. In addition, figure 2.3 shows the relationship of pupillary diameter and light intensity is not linear [80, 89].

Pupil diameter can vary from less than 1 mm to more than 9 mm depending on the amount of light that is allowed to enter the eye. Pupil constriction can occur within 0.2 sec, with peak response times occurring from 0.5 to 1.0 sec later [7]. In darkness, the pupil can be expected to enlarge to an average size of about 7 mm with a standard deviation of 0.9 mm [73]. In standard light conditions, the average size of the pupil is about 3 mm [134]. Additionally, figure 2.3 illustrates that the relationship between pupillary diameter and light intensity is not linear [80, 89].

2.2.2. Pupillary Accommodation Reflex

The accommodation response is elicited when the eye is focused on an object that is nearest to the viewer (near reflex). This response function allows the eye to produce sharply focused images. During near accommodation, the pupil constricts to increase the depth of focus of the eye blocking the light spread by the periphery of the cornea [23]. The opposite effect occurs when eyes are focused on far away objects.

2.2.3. Task-Evoked Pupillary Responses

Task-Evoked Pupillary Responses (TEPRs) is a state where pupils dilate as a function of increasing in cognitive processing. Numerous studies demonstrate that pupillometry is reliable as a measurement of cognitive load. The pupil dilates when the brain stores short-term memories, processes language, reasoning, and perception, and sustain attention and
selective attention all independent of PLR [6].

Performing short-term memory tasks require cognitive resources to store and process information. During the rest period, pupillary diameter is influenced by light and fixation, but in the absence of cognitive processing contributing to its size. Following the stimulus, a phasic mode is activated causing the baseline to drop to the preloading stage. Pupillary diameter then increases in size when cognitive load increases. Once cognitive load reaches maximum capacity or finishes processing, pupillary diameter gradually decreases to baseline size. Lower cognitive load is reflective of smaller pupil size. Figure 2.4 shows a report from Kahneman and Beatty. The pupillary diameter of 5-digits is smaller than 7-digits. Our experiment also shows similar result [95].

There is a long list of factors which affect pupillary diameter in literature. Table 2.2: summarized some pupillary variation in literature [121].

2.3. Linkage of Pupillary and CL-NE System

The Locus Coeruleus (LC) is a part of the brain structure that constitutes the sympathetic system’s hub to the whole brain [2]. The LC is located on each side of the rostral pons in the brainstem (Figure 2.5). The LC supplies the neuro-transmitter norepinephrine (NE) to almost all parts of the brain. Release of NE is associated with increased arousal and alertness, promotes vigilance, enhances formation and retrieval of memory, focuses attention, and increases restlessness and anxiety.

The LC responds to stress by releasing NE by altering the activity of prefrontal cortex. These neurons play critical roles in regulating brain function. The LC is engaged during memory retrieval [111] and slow-wave sleep [28]. This suggests the locus ceruleus norepinephrine system (LC-NE system) consolidates with memories. In addition, LC sends signals to brain areas associated with selective attention processing [36], while optimizing behavior to maximize utility (e.g. searching for new resources or exploiting existing resources) [2].
A study in 1986 demonstrated a tight correlation between online pupillary response and activation of the LC-NE system in a monkey while showing the same relationship in behavioral performance as LC tonic activity. The latter finding has recently been corroborated in humans [32].

According to Aston-Jones and Cohen, two different modes of LC activity correspond to different patterns of animal behavior: Phasic and Tonic. In the phasic mode, LC cells are activated when processing task-relevant stimuli. This mode is consistently associated with high levels of task performance. In the the tonic mode, animals exhibit poor performance on tasks requiring focused attention while showing increased distractibility [2]. However, in the tonic mode an animal is more likely to detect novel stimuli. These associate with two well-established cognitive mechanisms termed “focused” (or “exploitation”) mode and “diffuse” (or “exploration”) mode. Focused mode is the optimized performance during a specific task or event, whereas the diffuse mode is optimized for shifts of performance between tasks or events.

LC-NE system is known to regulate task engagement and arousal. There are evidence that TEPRs are also responsible to task engagement and arousal, although it is unclear how these two systems are linked. Murphy et al. provides evidence demonstrating that pupil diameter and scalp electrophysiological event-related potentials P3(00) (used to detect task engagement) is correlated with TEPRs [81]. Several other studies also demonstrated a similar relationship between pupillary tonic response, mental effort, and the state of arousal or vigilance. For instance, sustained processing increases in the pupil tonic response [105]. As difficulty and arousal increase, performance gradually degrades and substantial increases in pupillary baseline are concomitantly observed. [6, 35, 66, 91, 112]. In contrast, when the tonic state is low, as in a person who is fatigued after sustained attention or is sleepy, the pupil begins to fluctuate considerably while its average diameter gradually decreases [51]. Pupillographic sleepiness testing has been used as an objective alertness assessment in European sleep research and sleep medicine centers. [128, 129]. Thus, observing pupil diameter allows researchers to externally map changes in LC activation [61].
Figure 2.1. Autonomic Nervous System Available at http://www.merckmanuals.com/professional. Accessed April 15, 2017.
Figure 2.2. Pupil Constriction and Dilation [74]

Figure 2.3. Original data used by Moon and Spencer showing Pupillary diameter changes to different luminance levels [80]
Figure 2.4. Pupillary response during the digit span short-term memory task. The result is reported by Kahneman and Beatty [7].

Figure 2.5. Locus Coeruleus is located at the pond in midbrain. It distributes Norepinephrine to most of the brain [2].
Table 2.2. Sources, descriptions, and documentation regarding variation in pupil size from Warren’s work [121]

<table>
<thead>
<tr>
<th>Sources</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light Reflex</td>
<td>Pupil constricts with increased intensity of illumination and dilates with decreased intensity of illumination.</td>
</tr>
<tr>
<td>Darkness Reflex</td>
<td>Momentary dilation due to interruption a constant adapting light. Different from the light reflex.</td>
</tr>
<tr>
<td>Consensual Reflex</td>
<td>Stimulation of one eye affects both eyes equally. Failure called dynamic anisocoria.</td>
</tr>
<tr>
<td>Near Reflex</td>
<td>Constriction due to decreasing the point of focus.</td>
</tr>
<tr>
<td>Lid-closure Reflex</td>
<td>Momentary contraction followed by redilation.</td>
</tr>
<tr>
<td>Pupillary Unrest (Hippus)</td>
<td>Continuous changes in pupil diameter.</td>
</tr>
<tr>
<td>Psychosensory Reflex</td>
<td>Restoration of diminished reflexes due to external stimulation.</td>
</tr>
<tr>
<td>Age</td>
<td>Decreased diameter and increased variability with age.</td>
</tr>
<tr>
<td>Habituation</td>
<td>Pupil diameter decreases, speed of contraction increases, magnitude of reflex decreases.</td>
</tr>
<tr>
<td>Fatigue</td>
<td>Diameter decreases, amplitude and frequency of hippus increases. Age amplifies these effects.</td>
</tr>
<tr>
<td>Alertness &amp; Relaxation</td>
<td>Alertness suggestions decrease and relaxation suggestions increase pupil size.</td>
</tr>
<tr>
<td>Binocular Summation</td>
<td>Construction is greater when both eyes are stimulated.</td>
</tr>
<tr>
<td>Wavelength</td>
<td>Larger dilation to chromatic than achromatic stimuli. As intensity of illumination is increased proportionately more constriction is elicited by shorter wavelengths.</td>
</tr>
<tr>
<td>Alcohol</td>
<td>Dilates the pupil in proportion to the percentage of alcohol in the blood.</td>
</tr>
<tr>
<td>Sexual Preference</td>
<td>Dilation to sexually stimulating material.</td>
</tr>
<tr>
<td>Psychiatric Diagnosis</td>
<td>Abnormal pupillary responses in schizophrenics, and neurotics.</td>
</tr>
<tr>
<td>Pupil Size</td>
<td>Stimuli involving larger pupils elicit more dilation.</td>
</tr>
<tr>
<td>Political Attitude</td>
<td>Dilation for preferred political figures.</td>
</tr>
<tr>
<td>Somatic Stimuli</td>
<td>Small pupil diameters are associated with high recognition thresholds.</td>
</tr>
<tr>
<td>Taste</td>
<td>Pleasant taste elicits dilation.</td>
</tr>
<tr>
<td>Information Processing Load</td>
<td>Increasing dilation to increasingly difficult problems.</td>
</tr>
<tr>
<td>Task Relevant Response</td>
<td>Having to make a motor response augments pupillary responses.</td>
</tr>
<tr>
<td>Incentive</td>
<td>Increases diameter on easy problems only;</td>
</tr>
</tbody>
</table>

18
Figure 2.6. Tonic LC activity and Yerkes-Dodson relationship which is too low or too high LC activity leads to lower in performance [2]
Figure 2.7. Relationship between tonic pupil diameter and baseline firing rate of an LC neuron in monkey. Pupil diameter measurements were taken by remote eye-tracking camera at each instant in time when the monkey achieved fixation of a visual spot during the signal [2,96].
2.4. Cognitive Load Measurement

This section we introduce common methods available for measuring the cognitive load of users.

Cognitive load measurement is typically divided into three major categories: objective indirect (e.g., measuring tasks performance), subjective measurement (e.g., self-report), and physiological observing [115]. All techniques have their advantages and disadvantages. Many researchers use one or more cognitive load measurement methods with their studies. The following section discusses the top five cognitive load measurement found in the literature.

Self-report is the most widely used measurement of cognitive load. This can be as simple as a 9-point Likert scale ranging from very low mental effort (1) to very high mental effort (9) [88]. It is based on the assumption that users can introspect the amount of mental effort during specific tasks. This ‘intensity of effort’ may be considered to be an ‘index’ of cognitive load. A subjective measure is a standard approach for measuring cognitive load because it is simple, requiring no extra equipment, and less intrusive than techniques requiring wearable sensors. The 9-point scale has been found to be highly reliable [88]. For example, a famous self-evaluation is the NASA-TLX (NASA Task Load Index). The self-evaluation was developed by NASA’s Ames Research Center in 1986 [38, 83]. It was designed to measure overall workload based on a weighted average of ratings on various subscales: mental demand, physical demand, temporal demand, performance, and frustration. Users rated a specific task that they had immediately completed using each subscale (each subscale ranged from 0-100). There are both pen-and-paper and digital versions of the NASA-TLX. This type of measurement might be useful for instructional design, usability testing, and marketing research application. However, self-report may only be measured after tasks have been fully completed, so cannot provide real-time measurement during a task or for multiple sub-tasks. This is essential for real-time applications such as driving assistance, reading, or learning.

Dual-task analysis is performed by asking users to perform two tasks at the same time: a secondary task, and a primary task. The primary task is the task that researchers want
to measure levels of cognitive load, and the secondary task is mostly for pulling attention away from the primary task in a predictable way. For example, if a researcher needs to know which mathematics questions induce higher cognitive load, he or she would ask participants to perform mathematics questions as a primary task while playing audio of sounds and asking the user to detect certain types of sounds as a secondary task. Higher cognitive loads from the primary task will cause a lowering of performance in the secondary task. Dual-task analysis is based on three assumptions. Firstly, humans can focus on only one task at a time (selective attention) [82]. Secondly, humans can hold a limited amount of information at a time [17]. Thirdly, necessary information needs to be stored in short-term memory before processing cognitive tasks [115]. Thus, this technique is also used to measure cognitive load objectively. However, compared to subjective measurement, this method is far less used because it requires substantial preparation and some specialized equipment. Dual-task analysis might be suitable for instructional design or research in a lab environment. Although dual-task methodology provides somewhat real-time cognitive load measurement by monitoring secondary tasks performance over time, the interference of the secondary tasks makes it unsuitable for many applications. Moreover, the secondary task does not mirror typical day-to-day evocations of cognitive load.

Physiological observation is based on the observation that increases in cognitive load cause physiological changes. Those changes are more likely to be caused by the automatic nervous system, parasympathetic and sympathetic divisions. Thus, any sensor that can track such changes might be able to give a real-time cognitive load monitoring. Physiological observation is performed by special sensors attached to the participant’s body. The resulting signal is further analyzed by specialists afterward visually and with semi-automated time series analysis. The primary benefit in physiological measurement is that it does not intentionally distract users and does not require user input. However, it requires specialized equipment and an environment that mitigates other psychological interference. Thus, it is challenging to separate out sensor changes in response to cognitive load changes from other psychological stimuli that cause sensor fluctuations. Physiological observation is a
preferred method for implementing in mobile devices, because machines can observe the changes autonomously without interfering with normal activities. As the measurement of cognitive load, only eye behavior analysis, fMRI, and heart rate monitoring are commonly used because they are more reliable and consistent than other tested physiological sensors. Each physiological monitoring method has its advantages and disadvantages when it comes to implementing them on mobile devices.

Oculomotor analysis is a study of eye behavior. It is frequently found in the literature for measuring cognitive load. Cognitive load can be measured by a gaze tracker. Eye fixation [122], gaze direction [123], and pupillary dilation is analyzed (i.e., pupillometry). For example, Kahneman and Beatty demonstrated that pupillary dilation increases according to increasing levels of cognitive load, called task-evoked pupillary response [6,49]. Eye behavior analysis is commonly monitored by a gaze tracker, which is more expensive than other physiological measurement equipment. Like other physiological observations, pupillometry can be interfered with by other stimuli such as light reflex, accommodation reflex, and other psychological causes [121]. Moreover, gaze trackers have a high amount of measurement noise caused by variance in head pose, head and eye movements, and software errors. Despite those limitations, the outstanding benefit of eye analysis is that it’s the least intrusive method. This is because there is no need to attach a device to users (excepted when using glasses-worn gaze tracker), it can be quickly calibrated, and has high response rate (about 500ms 2s). More importantly, it does not interfere with normal activities and does not require verbal communication. This enables the device to work with infants, animals, and users with a form of audible disability. In a controlled situation, it is reliable and reproducible. Thus, the oculomotor analysis is suitable for many mobile applications. Another recent, commonly used eye-based measure is the index of cognitive activity (ICA) developed by Marshall [75]. This is a proprietary measure that uses wavelet analysis of the pupillary response curve to measure cognitive activity. It has been shown to be useful for assessing cognitive load of driving activity, but is not widely employed in studies of cognitive psychology because the methods used are not fully established in the literature.
Heart rate (HR) and heart rate variability (HRV) analysis [76] rely on the fact that higher cognitive load causes an increase in heart rate and makes the heart rate more consistent. On the contrary, lower cognitive load results in more heart rate beat-to-beat variability. This information can be acquired by electrocardiogram (ECG) or photoplethysmogram (PPG) depending on what sensors are attached to users’ body. These data have considerably less interference from lighting condition compared to pupillometry. However, this method is sensitive to movement due to sensor sensitivity, or sympathetic activation caused by physical movement. Furthermore, psychological stimuli still affect the data. This physiological sensing is more intrusive because a sensor has to be attached to users. Heart rate is affected by many physiological phenomenon, so heart rate variability is considered a superior measure for cognitive load because less stimuli interfere with its measurement. Even so, heart rate variability is not completely real-time. This is because reliable measurements of heart rate variability take about 120 seconds of data collection to be reliable. HR and HRV analysis are simple and low-cost. It is possible to be implemented in mobile via Bluetooth, USB, or the headphone port. Some applications such as computer games, e-learning, or office applications might be suitably measured with this method.

The final commonly used measure of cognitive activity is through the use of brain computer interfaces. One such method is functional magnetic resonance imaging (fMRI). This measures brain activity by detecting changes associated with blood flow to an area of the brain. It is measured using infrared reflectivity and absorption, typically on the forehead. It is reported that it does not only measure a level of cognitive load [87,126] but also shows different types of cognitive load [45]. It is reliable, but it is also the least used method (compared to eye behavior analysis and HR monitoring) because it is large, noisy, expensive, and requires an expert’s interpretation. It is almost impossible to be implemented on mobile devices (at present day).

In summary, the top five cognitive load measurements are self-report, dual-task analysis, eye behavior analysis, heart rate monitoring, and fMRI. The most widely used method is self-report because of its simplicity and effectiveness. Although self-report is limited to
non-real-time applications, it is possible to be used in online learning. The most suited methods to mobile applications are eye behavior analysis and heart rate monitoring. Eye behavior analysis and pupillometry are highly reliable and reproducible. These methods are suitable for indoor stationary tasks such as when interacting with a computer or watching entertainment media. While heart rate monitoring is less reliable compared to pupillometry, it is more economical and more robust to changes in lighting conditions. It is suitable for classroom use, research, and entertainment. Eye behavior analysis and HR monitoring methods could be used together to increase measurement accuracy, but no current research has attempted this type of sensor fusion.

2.5. Summary

This chapter described the physiological background related to pupillometry and human cognitive processing involved in attention and behavior, which was followed by a basic description of the autonomic nervous system, pupil anatomy, and pupillary reflexes. Finally, we discuss about common methods of measuring cognitive load. These knowledge are a fundamental to cognitive processing sensing system, which we will explain in later chapters.
This chapter briefly introduces artificial neural networks and deep learning.

3.1. Artificial Neural Networks And Deep Learning Overview

An Artificial Neural Network (ANN) is a machine learning model inspired by the connections between neurons in the human brain. A typical human brain has approximately 86 billion neurons, and they are connected through approximately $10^{14} - 10^{15}$ synapses [22]. Figure 3.1 shows a drawing of a biological neuron. Biologically, each neuron receives input signals from dendrites. If the electro-potential is strong enough, then the neuron gets excited and fires output signals at axon terminals. The output signals connect to other connected neurons. This chain reaction of a neuron’s activation controls the biological functions of animals (i.e., the ability to see, think, behave in certain ways). Similarly, an artificial neuron (figure 3.2) receives signals from artificial dendrites (input $x$). All input signals are collected at the artificial cell body and get evaluated; they decide whether the neuron should fire. This evaluation depends on an activation function. If the sum $\sum w_i x_i + b$ is more than a certain threshold, then the neuron fires and the output is sent. This chain-reaction-of-activation can predict certain outcomes at the end of the network. This is known as a feedforward network.

A neural network finds a set of weights and biases that maximize prediction performance. The simplest search strategy to determine a good approximation of the set of networks weights is a brute force search. We assign all possible numbers $R$ to the networks weights and biases until the best performance is achieved. This method is realistically infeasible because resources are limited. Another method randomly assigns sets of weights. This method is simple and fast, but is unpredictable, unreliable, and less likely to achieve the best performance. A more intelligent randomization method, such as the genetic algorithm, has
been found to improve training performance. Nevertheless, the more popular and arguably more effective technique is backpropagation. Backpropagation begins with an educated guess of what might be a good set of network-weights for the next iteration. This is regarded as a loss function. Intuitively, if we assume that the current set of weights characterize a ball, and the loss function is a surface, then backpropagation is characterized by a ball rolling down a concave surface until reaching the bottom. The bottom of the loss surface is the set of weights that produces the least loss. The gradient of the loss function indicate the direction and slope of the surface at any point. A gradient can be obtained by calculating the derivative of a loss function. During training, the backpropagation algorithm distributes the gradients through the network via a series of rules until all weights and biases in the network are fully updated. As a result, backpropagation and the proper optimization algorithm guarantees the network will reach the local minimum.

Early ANNs were made up of a single neuron and a simple threshold function known as perceptron [99]. This method was unstable and could not solve problems that were not linearly separable (i.e. XOR). Since then, ANNs have undergone several improvements over the past 60 years. For example, a simple thresholding was replaced with a sigmoid function to increase stability. This implied that backpropagation was differentiable. Nevertheless, the performance was not better than competing machine learning methods (i.e. support
vector machine or naive Bayes). Moreover, ANNs were harder to train because they have many hyper-parameters. If one attempts to create a larger network by stacking multiple layers, the networks are many times unable to learn at all. This is because gradients can be saturated during training; thus, they are reduced to small numbers until the value is too small to efficiently propagate back through the network. This phenomenon is known as vanishing gradients. Since then, ANNs had little to no progress in academic literature. Until around 2012, ANNs had significantly gained popularity. AlexNet outperformed other traditional computer vision techniques for large-scale image classification called ImageNet challenge [62]. The winning technique was deep learning. Deep learning is a large stack of representation learning modules. The premise of representation learning is to pre-train the network with unsupervised learning techniques aimed at reducing dimensionality; thus, discovering important features before stacking them up to a larger network. Examples of representation learning are the Restricted-Boltzmann machine, the deep belief network, and the auto-encoder [44]. The pre-training step was believed to be the key to success of deep learning. It turns out that it is not the case. Later studies proved that a large network could be trained without any pre-training intervention. A deep neural network can be successfully trained by simply choosing non-saturated activation functions and the appropriate initialization step [62]. The reason for the high performance of deep learning
might just be that we have a sufficiently large amount of data and a computation power not available at the time ANNs were developed.

Deep learning is an active area of research because it significantly outperforms most traditional feature-based engineering techniques. The highly flexible method excels at finding intricate patterns in high dimensional space. Deep learning is actively being used for pattern recognition problems such as image classification [27,62,117,120], natural language processing [14,47,114], speech recognition [43,103,124], and many other techniques. For a more in depth review of deep learning, we suggest reading the deep learning text by Goodfellow, Bengio and Courville [33].

3.1.1. Overview of Convolutional Neural Networks (CNNs / ConvNets)

Convolutional Neural Networks (CNNs or ConvNets) are a type of neural network designed to discover patterns in spatial data (i.e., images, videos, sound spectrograms). Thus, CNNs are commonly used in image classification and speech recognition. Compared to ANNs, CNNs can be described as ANNs that have convolutional layers attached. Figure 3.3 shows a comparison between a normal three-layer ANN and a simple CNN.

![Figure 3.3. (Left) A regular 3-layer Neural Network. (Right) A typical CNN.](image)

Most CNN’s consist of one or more convolutional layers and a few fully connected layers. A convolutional layer (Conv layer) is the layer that performs a convolution operation on the
input data. The data is convolved with \textit{feature filters}. The filters are composed of learnable weight matrices used for the convolutional operation. The outcome of the operation is a \textit{feature map}. Feature maps usually pass through an activation layer. Common activation functions include Linear, Sigmoid (Logistic), Hyperbolic Tangent, and Rectified Linear Unit (ReLU) functions.

Common hyper-parameters of the Conv layers are \textit{kernel size}, \textit{number of filters}, \textit{zero-padding} and \textit{strip}. The \textit{kernel size} is the size of the filter. \textit{Stride} is the number of gaps between a current convolving location and the next location. Finally, a \textit{zero-padding} is a method of padding zero around each input while performing a convolution. Figure 3.4 shows an example of the convolution operator in a convolutional layer. This example has three $3 \times 3$ filters and zero-padding (gray zero area around edges of the image). The \textit{stride} has two data points. The input image is on the left most side (a blue matrix), and the outputs are on the right side (green matrices). At each convolution step, a filter starts at the most top left of the input, then shifts a window to the right two blocks at a time (stride is set to two). At each step, it calculates a sum of the element-wise multiplication of the input and filter to produce each output. Once all filters finish convolving, the outputs are sent to the subsequent layers, which could be another convolution layer or an activation layer.
Figure 3.4. This example has three $3 \times 3$ filter and zero-padding (gray zero area around edges of the image). The *stride* has two data points. The input image is on the left most side (a blue matrix), and the outputs are on the right side (green matrices). At each convolution step, a filter starts at the most top left of the input, then shifts a window to the right two blocks at a time (stride is set to two). At each step, it calculates a sum of the element-wise multiplication of the input and filter to produce each output. [52]
CNN architectures usually contain pooling layers. A pooling layer is a layer which aims to down-sample feature maps (3.5 (left)). The purpose of down-sampling is to reduce over-fitting, noise and to accentuate important features regarding learning outcome. For example, for an image classification model, edge pixels are more important than pixels that are on a smooth surface. Pooling layers are more likely to discard those surface pixels. The pooling algorithm works by sliding a 2D-window over a feature map, then picking the highest number in the window as an output. This method is called max pooling (3.5 (right)). Another technique is mean pooling. Mean pooling takes an average of the values in the sliding window as an output. This means that a $2 \times 2$ pooling layer downsamples the input to exactly one quarter the original size (half the rows and half the columns). Note that down-sampling can also be achieved via a convolution layer with larger stride [107]. Thus, some networks might not have explicit pooling layers, but still employ down sampling.

![Figure 3.5. (Left) A pooling layer. (Right) A max pooling layer.](image)

CNNs extract abstract representations optimized for specific problems. Each higher layer represents a more abstract representation of the previous layers. For example, the first convolutional layer might detect lines and edges, while the next convolutional layer detects basic shapes. A layer at the end of the network might be able to detect complex shapes like human faces, assuming the problem is face or human detection. As mentioned above, the representation of convolutional neurons varies depending on the problem. For instance, if the problem is to detect a bookshelf in an image, then some neurons might learn to detect
text, which commonly appears in a book instead of human faces. Figure 3.6 shows a typical CNN classifying a dog. The input image is at the bottom of the network. First, the feature maps show the results of the convolution filters, which learned to detect edges. In the higher layers, filters learned to detect a specific part of the image. In this example, it learns to detect a dog’s eye as one of many features to detect a dog.

Once the data reaches the end of the network, it flattens out, then connects to a fully-connected layer (a dense layer) for final classification.

Figure 3.6. A typical convolutional neural network classifying a dog [67].

By looking closely, deep neural networks are a stack of building blocks (i.e. CNNs, Pooling, Dense (Fully connected), etc.). Deep neural networks can add, remove, modify or even introduce new type of layers. Some architectures seek simplicity. A study shows replacing the Max-pooling layers with Convolutional layers, with more strides, yields a similar result [107]. Other methods seek versatility by having a diverse set of layers [50].

Although deep neural networks (deep nets) can be used in supervised, unsupervised, and reinforcement learning, we mainly use deep nets for supervised learning problems in this work. There are four features in common in supervised learning: activation functions, optimization functions, loss functions, and regularization. In the next section, we review those elements commonly used in convolutional neural networks and deep learning.
3.2. Optimization

An optimization is a process of finding the value of an argument that minimizes or maximizes a function. In a backpropagation algorithm, a network learns by adjusting the weights and biases of an optimization function in the gradient flow direction. Backpropagation algorithms propagate gradients back through a series of networked layers from the back. This section introduces common optimization functions found in convolutional neural networks and deep learning literature.

Gradient Descent is the most popular optimization method for deep learning. The method starts by calculating gradients of a loss function. The resulting gradient indicates a tangent of the loss function $\nabla J(\theta)$. The network’s weight is iteratively updated in the opposite direction of the gradient flow of the objective function $\nabla_\theta J(\theta)$ until the network converges.

A hyper-parameter of Gradient Descent is a learning rate ($\eta$), which indicates the length of a learning step in the direction of the gradient flow for each iteration. The learning rate is the speed that a network learns. A low learning rate implies the network learns slowly. In contrast, a high learning rate implies the network learns faster. Too high a learning rate can prevent the network from learning at all because the gradient can become too large and subsequently arrive at local minima. In the worst case, the learning rate could overestimate and diverge and explode the entire network. If the learning rate is low enough, the network will eventually converge. There are mainly three gradient descent variances. First, the Batch gradient descent (GD) computes the gradients for the entire training set at a specific time. A typical gradient descent optimization can be described as follows:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$$  \hspace{1cm} (3.1)

where $\theta$ is parameters. GD is guaranteed to converge to a global-minima for convex loss surfaces, or to local-minima for non-convex surfaces. The downside of this algorithm is that most deep learning algorithms are nonconvex. GD also requires a lot of computational
power and memory because the entire training set must be loaded into memory. In many deep learning cases, the training set is too large to fit the memory of a single computer node. Stochastic gradient descent (SGD) might solve this memory problem. SGD (or online gradient descent) performs the update for each training point chosen randomly. The revised equation is depicted as follows:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)})$$ (3.2)

where $x^{(i)}, y^{(i)}$ represents an individual sample at index $j$. The performance is like batch GD, assuming the learning rate is reduced over time. A big benefit of SGD is that the training data does not need to be stored in the memory all at once. Moreover, because of its randomness, SGD could escape local minima; this is not possible in batch GD. Several jumps around the loss value could occur while training with SGD. To smooth out this effect and lower computation resources, researchers bundle input into a small batch. This is called Mini-batch gradient descent. Mini-batch gradient descent performs the update for a small batch at a time. Typical mini-batch sizes range between 32 and 256. The deep learning community uses the term SGD and mini-batch gradient descent interchangeably.

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i:i+n)}; y^{(i:i+n)})$$ (3.3)

The main challenge of GD is choosing the appropriate learning rate. Choosing too small a learning rate leads to painfully slow convergence. On the other hands, choosing too large learning rate causes the network to fluctuate around local minima and might not converge or even diverge. Although we could set the learning rate to decay over time, this step needs to be set ahead of time; otherwise, it might not reflect the real nature of data. Another challenge for vanilla mini-batch gradient decent is that the network can fall into a ravine. Ravines are areas where the surface curves much more steeply in one dimension than in another [20]. The next section reviews some proposed techniques to address these challenges, from choosing the appropriate learning rate to escaping ravines.
The deep learning community addresses these challenges by proposing various extensions of gradient descent algorithms. We review widely used algorithms developed by the deep learning community in this section.

3.2.0.1. Momentum

Momentum [93] is designed to allow the learning rate to be dynamically adjusted, based on previously accumulated gradients; this can be viewed as a ball rolling downhill in the real world. The learning rate increases if the direction of the current gradient is approximately equivalent to the previous gradient. The method achieves this effect by adding an acceleration term to the learning rate. This fix also helps SGD to escape ravines. As a result, the network achieves faster convergence. The momentum equation is shown below:

\[ v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \]
\[ \theta = \theta - v_t \]

(3.4)

The momentum term \( \gamma \) is typically set to 0.9. Despite this benefit, a downside of momentum is that learning rate could gain too high a momentum. Thus, it can diverge, surpassing the convergence point.

3.2.0.2. Nesterov Accelerated Gradient

Nesterov Accelerated Gradient (NAG) [84] is proposed to fix the momentum diverging problem. NAG adds a \( \gamma v_{t-1} \) term to the \( \theta \) parameter to approximate the next position. This anticipatory update prevents the learning rate from growing too large. As a result, the method is more responsive. Bengio et al. [9] show that NAG increases in the performance of RNNs. A NAG optimizer is characterized as follows:

\[ v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1}) \]
\[ \theta = \theta - v_t \]

(3.5)
3.2.0.3. Adagrad

An Adaptive Gradient Algorithm (Adagrad) \[25\] is designed to provide SGD with a variety of learning rates for different parameters. Parameters that are frequently updated will have smaller learning rates. Thus, Adagrad is well-suited for sparse data. Another strength of Adagrad is that it eliminates the need to manually tune the learning rate.

\[
\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} \\
\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}
\]

where the term \(g_{t,i}\) refers to a gradient for each parameter \(i\) at time step \(t\). \(G_t \in \mathbb{R}^{d \times d}\) is a diagonal metric, where each diagonal element \(i, i\) is the sum of squares of the gradients with regard to \(i\) up to time step \(t\) \[101\]. \(\epsilon\) is a small constant added to avoid division by zero.

The authors note that without a square root, the network performs worst in practice.

The major weakness of Adagrad is diminishing learning rates caused by the accumulative sum of squared gradients, which can grow too large. It grows so large that it makes the learning rates infinitesimally small. As a result, the network will no longer learn new information at that point.

3.2.0.4. RMSprop

Root Mean Square Propagation (RMSprop) \[42\] is developed by Geoff Hinton to solve the diminishing learning rates problem of Adagrad. Instead of storing all previous square gradients, RMSprop only stores the latest previous square gradients. \(\gamma\) is a momentum term. Hinton suggests using 0.9 for \(\gamma\). The equation is shown in Equation 3.7.

\[
E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma) g_t^2 \\
\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t
\]
The term $\sqrt{E[g^2]_t + \epsilon}$ can be viewed as the root mean squared error criterion of the gradients. We can change the term $(E[g^2]_t)$ to $RMS[g]_t$ as shown in Equation 3.8.

$$RMS[\Delta \theta]_t = \sqrt{E[\Delta \theta^2]_t + \epsilon}$$

Then, the final RMSprop equation is shown in Equation 3.9.

$$\Delta \theta_t = -\frac{\eta}{RMS[g]_t}g_t$$

3.2.0.5. Adadelta

Adadelta [140] is similar to RMSprop, although they were developed independently. Adadelta goes a bit further. Adadelta does not require a default learning rate. Adadelta replaces the learning rate $\eta$ with a root mean square of parameters $\theta$ at the previous time step $t$ ($RMS[\Delta \theta]_{t-1}$). The final Adadelta is shown in Equation 3.10.

$$E[\Delta \theta^2]_t = \gamma E[\Delta \theta^2]_{t-1} + (1 - \gamma)\Delta \theta^2_t$$

$$RMS[\Delta \theta]_t = \sqrt{E[\Delta \theta^2]_t + \epsilon}$$

$$\Delta \theta_t = -\frac{RMS[\Delta \theta]_{t-1}}{RMS[g]_t}g_t$$

$$\theta_{t+1} = \theta_t + \Delta \theta_t$$

3.2.0.6. Adam

Adaptive Moment Estimation (Adam) [56] can be viewed as a combination of RMSprop and momentum. Adam stores an exponentially decaying average of past gradients like momentum and an exponentially decaying average of past squared gradients like RMSprop. Adam works well compared to other adaptive-learning-method algorithms.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g^2_t$$
\[ m_t = \frac{1}{1 - \beta_1^t} \]
\[ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \]

\[ \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \] (3.13)

Figure 3.7 is created by Alec Radford and shows a comparison of six optimization function behaviors on loss surface contours. Momentum first overshoots and then comes back down. NAG overshoot when the optimization function is smaller than Momentum. Adadelta and RMSprop perform equally well.

Figure 3.7. SGD optimization on loss surface contours SGD with momentum [52]
3.3. Activation Functions

Activation functions are an essential part of ANNs. These functions introduce nonlinearity to a network. Nonlinearity enables networks to have the capacity to train high-complexity data. Next, we introduce common activation functions for deep learning.

3.3.1. Sigmoid (Logistic)

Sigmoid functions are preferable over a simple thresholding function (only 0 or 1). These functions normalize terms into the range between 0 and 1, as shown in equation 3.14. In recent years, the popularity of Sigmoid declined because of two major downsides in the training deep learning models. First, the function saturates gradients known as vanishing gradient. A vanishing gradient is a situation where gradients become smaller while propagating back to the network, until the network stops learning within a reasonable time. Another problem is that a sigmoid is a non-zero-centered function. This function could lead to a covariance shift, where the gradient distribution is shifting from positive to negative, back and forth during training, causing the network to learn slowly.

\[
sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.14}\]

3.3.2. Hyperbolic Tangent (Tanh)

Hyperbolic Tangents (Tanh) are like a Sigmoid function, but they are zero-centered. Which suggests that the covariance shift is less likely to occur. However, Tanh still has the same saturated gradient problem that sigmoid functions have.

\[
tanh(x) = 2\sigma(2x) - 1 \tag{3.15}\]

By comparing to sigmoid functions, tanh is a better default pick for most shallow artificial neural networks.
3.3.3. Rectified Linear Unit (ReLU)

A Rectified Linear Unit (ReLU) is one of the most popular activation functions for deep learning. ReLU works by rounding up any negative number to zero. Otherwise, the input number is unchanged, as shown in equation 3.16. The function is simple and computational inexpensive compared to sigmoid and tanh functions. More importantly, it works well in practice. Krizhevsky et al. [62] show that training with ReLU is six time faster, when compared with tanh. The authors suggested that the fast training is due to ReLUs linear non-saturating form.

\[ f(x) = \max(0, x) \]  

A problem of ReLU is dying ReLU. Dying ReLU is the situation that occurs when a ReLU neuron receives a negative input. As a result, weights in the neuron will not get an update during backpropagations \((f(x) = 1x > 0)\). This leads the neuron to stay zero or die. However, if the prior neurons of the dying ReLU are still alive, it could potentially retrieve the dying ReLU and bring it back to life. Dying ReLU is usually caused by learning rates that are too large.

There are some variances of ReLU proposed to solved dying ReLU. Leaky ReLU [72], for example, tries to address the issue by adding a small slope \(\alpha\) (i.e. about 0.02) instead of using hard zero threshold. The other three methods are parametrized ReLUs (PReLUs) [40], Exponential Linear Units (ELUs), which use exponentials [15] and ThresholdedReLU [60]. These techniques are new and there is insufficient evidence to suggest that they work better than ReLU in most situations.

3.3.4. Maxout

Maxout is introduced by Goodfellow et al. [34]. Instead of changing a nonlinear term, maxout uses learnable weights. Maxout can be compared to the general form of ReLU and Leaky ReLU. Maxout has become more popular in recent years.
\[ f(x) = \max(w_1^T x + b_1, w_2^T x + b_2) \]  
(3.17)

where \( w \) is a weight that can change according to the learning rate. \( x \) represents input. Maxout is a more data driven approach, compared to traditional ReLU. The network has a choice to create its activation behavior to fit a specific problem. It is possible that the network could learn to create a traditional ReLU on its own \((w_1, b_1 = 0)\), or create various types of ReLU based on data.

3.4. Loss Functions

This section introduces loss functions, which are commonly used in deep learning architectures. Although there are more loss functions in the literature (i.e., Tanimoto loss (tan), and Cauchy-Schwarz Divergence), they are not all widely utilized in deep learning. We define common mathematics notations as follows:

\[
\begin{align*}
W &= \text{Weight matrix} \\
i &= \text{a sample vector} \\
j &= \text{a individual element in a vector} \\
L &= \text{a loss function} \\
y &= \text{a true value} \\
f &= \text{a predicted value. The activations of the output layer in a Neural Network.}
\end{align*}
\]

Additional notations are a bold lowercase is a vector, a bold uppercase is a matrix, and a non-bold notation is scalar.

3.4.1. L1 and L2 Loss

L1 (Equation 3.18) is also known as least absolute deviations (LAD), least absolute errors (LAE). and L2 (Equation 3.19) is also known as least squares error (LSE). It is often used in regression, but they are also found in classification problems. Janocha and Czarnecki
proved that L1 and L2 could be used in classification problems. They provide a probabilistic interpretation regarding expected misclassifications [46].

\[
L_i = \| f - y_i \|_1 = \sum_j | f_j - y_j^{(i)} | \tag{3.18}
\]

\[
L_i = \| f - y_i \|_2^2 = \sum_j (f_j - y_j^{(i)})^2 \tag{3.19}
\]

### 3.5. Regularization

The power of deep learning comes from its ability to learn complex patterns in high dimensional spaces, simply by stacking more layers. Stacking more layers allows the network to generate more complex structure during training. However, this might not be robust to unseen data because it deeply learns every small detail in a training set, as opposed to learning essential patterns. This is known as overfitting. A first intuition is to simplify the network to deal with overfitting; however, it turns out that this is not a good approach because reducing the complexity of the network, reduces its capacity to learn more complex data in the future. A better technique is to help reduce the overfitting via regularization. This section reviews common regularization methods. Note that, while this work is produced, deep learning is a highly active area. This review is only a list of regularization techniques used in this work.

#### 3.5.1. L1 and L2 Regularization

L1 regularization (Equation 3.18) is another common technique used in many machine learning algorithms. L1 adds \( \lambda \| w \|_1 = \sum_j | w_j | \) to the objective function, where \( \lambda \) is the regularization strength, and \( w \) is a weight vector. Intuitive interpretation suggests that L1 leads the weight vectors to become sparse during training. This means that neurons, which react to noise, have less impact on the predicted outcome.

L2 regularization, known as "weight decay," ridge regression or Tikhonov regularization,
is the most common form of regularization. This regularization method is used in many machine learning algorithms, not only for deep learning. L2 adds $\frac{1}{2} \lambda \|w\|_2^2$ to the objective function, where $\lambda$ is the regularization strength, and $w$ is a weight vector. L2 tries to defuse weights. It prefers weights to be spread out over the entire network, rather than having a few neurons that dominate the predicting outcomes. The higher the $\lambda$, the smoother the classifier will be.

L1 and L2 can be used to complement each, this is known as elastic net regularization.

3.5.2. Max-Norm Constraints

Max-norm constraints [108] are implemented by limiting the incoming weight vector to the upper bound via a constant $c$. While training, the network is optimized under the constraint $\|w\|_2 \leq c$, where $w$ is a weight vector. Many studies suggest that the max-norm term prevents exploding gradients, even through a network is trained with a high learning rate. The reason is that weights in each hidden neuron are limited to a fixed constant number. $c$ is normally set between 3 and 4.

3.5.3. Dropout

Dropout is commonly found in many deep learning architectures because of its simplicity and effectiveness. This method was first introduced by Srivastava et al. [109]. Dropout is implemented by setting a random set of weights to zero, with some probability $p$. Normally, $p$ is set to 0.5 during training. Figure 3.8 shows the network without dropout (left) and with dropout (right). There is a movement that randomly sets the neurons to zero. One interpretation is that during this dropout force, the network avoids relying on a few pathways to dictate the final predictions. As a result, the network is less likely to assign large weights to noise in the training set. Dropout can be incorporated with other regularization methods to increase regularization strength. Srivastava et al. observed that by using dropout, max-norm constraints, large decaying learning rates, and high momentum boost regularization performance.
3.6. Summary

This chapter provides an overview of deep learning in the literature. The review begins with a brief history of artificial neural networks and deep learning. Then, it is followed by an overview of recent deep learning methods in the literature. Finally, we introduce a small building block in modern neural networks and deep learning.
4.1. Introduction

Pupil segmentation is part of preprocessing before performing iris recognition. There is a plethora of iris recognition techniques in the literature. However, the process is done with infrared images which are high resolution and high contrast between pupil and iris (figure 4.1). Simple edge detection and Hough transform would be enough for segmenting pupils for these conditions. In contrast, RGB images produced from off-the-shelf web camera are more challenging. They have low resolution, low contrast and high amount of noise. The segmentation algorithms used with infrared images are too unstable. We developed new algorithms to address these issues.

Figure 4.1. A comparison of an high resolution infrared image (a) (CASIA-Iris database V4. Available: http://www.cbsr.ia.ac.cn/china/IrisDatabasesCH.asp), and a RGB image from a phone front facing camera (b)

In this chapter, we describe three algorithms: (1) Modified Starburst (MSB), (2) Two-Levels Snakuscules (2LSN), and PupilNet. These are specifically designed to work with an off-the-shelf RGB camera.
4.2. Related works

A large body of literature on pupil segmentation is focused on biometric iris identification. Pupil segmentation is considered a preprocessing step before the unwrapping step of an iris and sending it to a recognition model. Most pupil segmentation algorithms perform in high-resolution infrared images. We review these algorithms to understand their strengths and limitations. There are two well-known algorithms commonly used for pupil segmentation. The first one is applying Hough transform [127]. Hough transform is a technique, which is used to detect particular shapes in an image. The algorithm starts by roughly segmenting a pupil with a binary thresholding, then apply a canny edge detector to find an edge of pupil. Following, the image is transformed to parameter space or Hough space. In doing so, the algorithm scans all edge pixels to find a shape that satisfies its parametric form. When located, it votes one point to that parameter set. Local maxima in parameter space are shapes within an image. Since Hough transform relies on the quality of edge detection, it performs well in high contrast images such as in infrared images. However, most pupils in RGB images are low contrast, and shapes are not presented. Therefore, Hough transform performs with high false-positive rates in such conditions. The second common algorithm is integro-differential operator proposed by Daugman [19]. The integro-differential operator is based on the fact that the difference in illumination between a pupil and an iris is maximum. The algorithm searches over the 3D space (x, y, and size) finding the parameter settings that maximizes pixel intensity differences. This technique is arguably simpler than Hough transform; however, it is computationally expensive. Many combinations of these two techniques with other techniques have been proposed. For example, [41, 59, 100] estimate the rough boundaries of the iris with Hough transform then place the active contour to find the pupil boundary. This reduces computational cost because an active contour only searches within iris area of interest. This technique heavily relies on the performance of Hough transform and the complexity of traditional active contour algorithm.

In addition to Hough transform and integro-differential operator, another well-known algorithm is Starburst algorithm. Li et al. [69] proposed the hybrid Starburst algorithm
that combines feature-based and model-based algorithms. The algorithm starts by removing
the corneal reflection from the image and estimating the iris boundaries with the derivatives
along 18 rays from the center of the eye. After detecting the edges, an RANSAC algorithm is
run to eliminate outliers. Later the algorithm extension was proposed by Ryan et al. [102].
Good results in reported but it needs to be adjusted to work with noisy low-resolution
images. However, their works were evaluated using infrared images which uniquely increase
the contrast of the pupil and iris boundary.

Literature discussing video based pupillary diameter tracking is scarce. The closest stud-
ies are pupil boundary detection in high resolution still eye images [55]. Although high-
resolution eye images present more challenging problems compared with infrared images,
similar pupil segmentation methods can be effortlessly applied. Besides, there is no pupil seg-
mentation in the literature for estimating pupil size in sub-accuracy levels for low-resolution
images from commodity devices.

Other video based related pupil research in literature is pupil location detection. They
aim to estimate eye gazing, not tracking pupil boundary [133]. An example of these works
is, OpenFace [5] is an open source tool attempted to extract various facial features such as
head pose detection, face tracking, facial landmark detection and gaze tracking. They used
a Constrained Local Neural Fields (CLNF) [4] model to detect both the eye and center of
the eye. While gaze tracking shares some similarities to our work, the measurement of pupil
size has not been investigated or reported by these previous works.

4.3. Preprocessing and Segmentation

Because all proposed algorithms only perform within eye ROI, before applying pupil
segmentation algorithms, eye areas needed to be extracted. First, we detected a face using
a face detector algorithm. The area around an eye was extracted by using facial landmark
detector trained with iBUG 300-W face landmark dataset and an ensemble of regression
trees [54]. Once eye ROI was detected, we scaled eyes to the same size. The motivation to
this was to reduce frequency adjusting algorithm hyperparameters caused by uneven image
size, and standardized pupil segmentation procedure across users. To scale eyes, we detected eye corners of both left and right eyes, then calculate Euclidean distance between these two points, then we scale the image to equally 120 pixels long. The number was acquired by an average of eye distance in our dataset. We then extracted regions of interest (ROI) of both the left and right eyes resulting in 64 by 64 pixel frames, 3 RGB channels. These eye images were further processed with our algorithms. Figure 4.2 illustrates the image preprocessing steps used by all three algorithms.

Figure 4.2. Extracting eyes the eyes with our pre-processing algorithm.

4.4. Improved Modified StarBurst

We proposed the modified starburst algorithm [95] in previous work. What follows is an abridged description of the algorithm and changes to the algorithm implemented in this analysis. The modified starburst algorithm starts by identifying the rough pupil area using binary thresholding to extract the dark area of the pupil. Noise pixels are eliminated with a morphological closing operation. We calculate the center of mass of the largest binary extracted area and use it as a seed point. From this seed, the algorithm marches in different directions, looking for strong edges (i.e., the edge of the pupil). The algorithm only marches from 45 degree to 135 degree and each direction is separated by 20 degrees. The motivation for restricting the march is to avoid the shadow from eyelashes and be more robust to
squeezing.

The pixel intensity difference between the seed point and the current point is calculated. If the pixel intensity difference is greater than a threshold $t$, we record it as an edge point. Our previous work, the threshold is a fixed number obtained from a calibration process. Once the algorithm marches all directions and has stored all the edge points it uses the average of the x and y points to calculate a new seed for the next iteration. We randomly add $\pm 2$ pixels to the new seed point to prevent the algorithm from stopping in a local optimum (similar to simulated annealing). These edge points from all iterations are sent to an ellipse-fitting algorithm which returns a best-fit ellipse. We run this algorithm separately on each eye.

As described, modified starburst has many hyperparameters. These parameters are optimized for individual users and situation with our calibration procedures proposed in PupilWare-M paper [95]. Among those parameters (degree offset, the number of steps, edge thresholding, etc), one parameter has high sensitivity and have a major impact on the algorithm performance: edge thresholding. This because the original starburst used fixed
thresholding to find pupil edge. We adopted the same method for RGB images. The problem arises when environment lighting changes. It impacts overall pixel intensity causing us to recalibrate frequently. Moreover, when the direction of a light source changes to the side, shadow shade from a user’s nose create unequal intensity between both eyes leading the algorithm performs differently. Figure 4.4 shows shadow impact modified starburst performance.

![Figure 4.4. Difference in pixel intensity caused by shadow](image)

To solve this, we calculate threshold value from a normalized cumulative histogram. We specify the pupil edge thresholding from the percentage of dark pixels in a normalized cumulative histogram. From our experiments, 0.14% of cumulative histogram performs considerably well at detecting the edges of a pupil as a default value. However, this parameter can be calibrated to fit each individual. Moreover, it performs on left and right eye ROI independently; as a result, threading can be dynamically adjusted based on input data independently between both eyes. We show later that dynamic thresholding greatly improves modified starburst robustness.

4.5. 2-Levels Snakuscules

Improving robustness is the main objective of this work. 2-Levels Snakuscules is designed to increase robustness in mind. Snakuscules [119] is an adaptation of traditional active contours tracing algorithm [53]. Traditional active contours use planar parametric curves and their parameters are optimized with existing data, monitoring shape, and prior knowledge.
Thevenaz et al. proposed a simplified version of traditional active contours optimization for detecting circular structures in an image [119]. Later, Garg et al. manipulated this algorithm to locate eye center in an image [30]. 2-Levels Snakuscules exploits the circular structure of pupil and iris. Instead of solely relying on detecting the edge of the pupil, the algorithm uses the pixel intensity of the entire iris and pupil area to estimate the pupil location and size. Figure 4.3 (right) shows an overview of 2-Levels Snakuscules algorithm.

The algorithm starts by applying a masking truncation threshold to reduce specular reflection and bright objects such as white frame glasses. This essentially clips the bright areas of an image to a maximum threshold and it reduces the chance of creating multiple local optimums. The value of thresholding is 3% of cumulative histogram. This parameter can be adjusted depending on image resolution. After clipping, a Gaussian filter is applied to the eye images. This step is essential to increase the stability of a snake moving smoothly to the center of eye. The size of a kernel depends on image resolution (A larger image requires bigger kernel size). We used 11×11 kernel size for locating the iris. Next a circle “snake” is placed at the center of the image. Snakuscules exploits the circular structure of iris where an cornea color is lighter than iris color, on average. A snake that has maximum energy is the one that has the greatest difference between average pixel intensity of the outer ring and the inner ring. Equation 4.1 shows how to calculate the energy of a snake:

$$E = \frac{\iint_{\rho R < \| x - x_0 \| < R} f(x, x_0) \, dx_1 \, dx_2}{\iint_{\| x - x_0 \| < R} \, dx_1 \, dx_2} - \frac{\iint_{\| x - x_0 \| < \rho R} \, dx_1 \, dx_2}{\iint_{\| x - x_0 \| < \rho R} \, dx_1 \, dx_2}$$  

Where $E$ is the energy of the snake with outer radius $R$ and inner radius $\rho R$. This equation ensures that we can adjust the snake body with fine-grained precision (inner circle). This equation is also used by Garg et al. [30] in finding eye position. Figure 4.5 shows the energy map after applying an iris snake (middle) and a pupil snake (right) on a gray eye image. The brightest area is the iris/pupil location.

A snake iteratively searches for more energy in four directions (up, down, left, right). If a nearby location has more energy, it greedily moves along that direction, searching for the highest energy until converged. In this work, we set a maximum of 15 iterations to increase...
Figure 4.5. An energy map that a snake can see. (left) the original image, (mid) an iris energy map, (right) a pupil energy map

the speed of the algorithm. Once the iris and eye center are located, a smaller second snake (a pupil snake) is applied only in the iris area. Note that a pupil snake is placed in an original eye image, not the blurred image so that contrast is preserved. A pupil snake can only grow to be a maximum of 90% of the iris snake to avoid detecting edges of the iris and cornea. All energy points are memorized in an energy vector ($E$). This energy vector is used to estimate pupil boundary with sub-pixel accuracy. Once a pupil snake is converged (about 5-15 iterations), the final pupil size is determined by a linear combination of radius and normalized energy weights. Intuitively, a higher energy snake boundary is more likely to be a pupil boundary. If there is more than one high energy step, then the pupil boundary might be somewhere in between those pixels as shown in equation 4.2 and 4.3.

$$E_{\text{norm}} = \frac{E}{\sum_{i=0}^{n} E_i}$$  \hspace{1cm} (4.2)

$$r_{\text{final}} = \sum_{i=0}^{n} E_{\text{norm},i} \cdot r_i$$  \hspace{1cm} (4.3)

$E_{\text{norm}}$ is a normalized vector of energy. $E$ is an energy vector. The algorithm is performed on both eyes separately.
This technique has components similar to the integro-differential operator. However, the major difference lies on both algorithms behaviors. The motivation is similar. They search for a circular ring structure that maximizing pixel intensity difference between pupil and iris. Integro-differential operator scans the entire space, while snakuscules prefer to search locally and iteratively adjust. Thus, an integro-differential operator could find global maxima but requires high computational resources. In contrast, Snakuscules use less computational resources, but it could trap in local maxima. Thus, for Snakuscules to successfully converge, it is mandatory that seeding point that is around the pupil area. A strength of snakuscules is the ability to modify an energy function. We can adjust an energy function to fit particular problem. 2-Levels Snakuscules goes beyond snakuscules on finding pupil area. 2-Levels Snakuscules inherits Snakuscules’ benefits, and it can also estimate pupil size in sub-pixel accuracy as described in this section.

The MSB hyper-parameters (primer, a percentage of cumulative histogram cutoff, and degree offsets) and 2LSN hyper-parameters (Gaussian blur kernel size, and kernel standard deviation, and maximum growth percentage) reported in the earlier section were set at a default setting to get started. We have optimized hyper-parameters using a random grid search and have chosen the best parameters based upon the mean square error for later sections.

4.6. PupilNet: Convolution Neural Network Structure

We propose a convolution neural network PupilNet for measuring pupil size. Figure 4.6 shows the network architecture of PupilNet. We noted that other architectures were investigated, but did not significantly increase or decrease the reliability of the network. The input image data used by PupilNet is tightly segmented around the iris area as a result of our preprocessing. As such, there is not much variation in the overall structure of the input images. Because of this, we found that two convolutional layers are sufficient to extract features related to pupil size. Using only two layers is advantageous regarding preventing over-fitting. Using more than two layers only marginally increased the network performance
results were not statistically significant) but training required much more computation power. With more data, a larger model may better extract pupillary features; however, with the current amount of training data, we did not find that more convolutional layers increase performance. We also experimented with larger eye area segmentations (i.e., 64×64 input image size that encompassed more of the eyes including surrounding skin). To attain the same performance as the 32×32 image inputs, the architecture required 3-4 convolutional layers. The overall result, however, was not significantly increased compared to the 32×32 image size. Segmenting only the iris area allows us to have much fewer data and a simpler network. This supports a conclusion that a simpler network using tighter segmentation has about the same expressivity as a larger segmentation with a more complex network. With the amount of training data in our dataset, that additional complexity is not warranted.

Before training our model, we preprocessed the eye images by applying histogram equalization. This helps to enhance the contrast between pupil and iris, and helps reduce the impact of different lighting conditions in the images. Even so, for dimly lit images, it also greatly increases noise artifacts. We applied the filters of size 3×3 to capture variation in each color channel of the input images. We also investigated training only on one color channel and using the hue-saturation-value representation instead of the RGB representation. In both cases, training with all three RGB channels performed better, and the network converged faster. After the convolutional layers, we flattened the architecture and used two fully connected layers. We used Rectified Linear Unit (ReLU) activation functions and 50% dropout rate in the hidden layer before the output layer. The loss function employed was the mean squared error between calculated pupil diameter and ground truth pupil diameter. For additional regularization, we employed data expansion techniques including a random rotation (up to 10 degrees) to the images as well as vertical and horizontal perturbations (up to 10% of the image size).

We implemented the network using Python 3.5 and Tensorflow r0.12. We trained on a machine with Dual 12 Core Intel Xeon@2.6GHz CPU and 320GB of memory. We trained on within-subjects for about 400 iterations. Beyond this, the networks tend to over-fitting,
and the performance does not have significant improvement.

![Convolution Network Diagram]

Figure 4.6. Convolution Network

The output of PupilNet is an estimate of the pupil dilation for each frame of the video sequence. Once we received these frame-level estimates from PupilNet, we post-processed the signal as follows: We eliminated blinks from the time series in the same manner as performed for the gaze tracker (i.e., via median filter comparison). Each participant performed 36 total trials for the digit span task, each of which has about 800 frames. Left eye images and right eye images were treated as separate examples while training the PupilNet architecture. During post-processing, we averaged the result of the left eye and right eye pupil estimates. We then applied a median filter of length approximately 2 seconds to the estimated pupil size signal over time. This filtered signal was then compared to the output of the gaze tracker pupil size. The total number of eye images is about 633,600 ($800 \times 36 \times 2 \times 11$). We used a batch size of 100 images per training batch.

4.6.1. Training and Optimization

PupilNet structure is presented in figure 4.6, based on a LeNet-5 network [68]. The number of layers is six layers. This is different than many modern architectures that might have 20-100 layers. We still use the nomenclature of “deep” to describe the network because historically anything above five layers has been called “deep.” We acknowledge that modern notions of “very deep” would not include PupilNet. It uses 2D convolution neural network
layers and mean pooling. We apply the filters of size $3 \times 3 \times 3$ to capture variation in each color channel of the input images. Finally, we flatten the architecture and use two fully connected layers. Throughout the network, we employ Rectify Linear Unit (ReLU) activation functions.

We train the network with Stochastic Gradient Descent with randomly select 128 data points each batch. We also choose Adaptive Moment Estimation (Adam) optimization function. Adam allows learning rate to be dynamically adjusted during training time. To regularize the network we also employ 50% dropout rate and L2 regularization in the hidden layer before the output layer. The loss function employed is the Mean Square Error between calculated pupil diameter and ground truth pupil diameter.

4.7. Conclusion

This chapter shows Modified Starburst, 2-levels Snakuscules algorithms, and PupilNet network architecture designed to track pupillary diameter for RGB web camera. It also described the motivation behind these techniques. The most important question remains, how this algorithm performs. To answer that question, we design experiments in chapter 5 and results are shown in chapter 6.
5.1. Introduction

Pupillary diameter trackers can be used in several domains. If you are an investor wishing to use in a hospital. You may ask is it robust enough for medical or home used. In this chapter, we will talk about the evaluating methodology designed to examine strengths and weaknesses of these three algorithms in various situations.

5.2. Experimental setup

All human subjects experiments in the evaluation of PupilNet were approved by the university human subjects board. Participants were recruited by mouth, email, and flyers and were compensated with a $10 gift card for their time. In all, we recruited 11 participants. Table 5.1 gives an overview of the participants. All participants reported normal or corrected to normal vision. One of the participants wore eye glasses and one wore colored contact lenses. We group the participants by their iris color. Because eye color is the largest contributor to contrast between the pupil and iris, we further subgroup the participants as follows: low contrast consisted of brown and hazel eyes (6 participants) and high contrast consisted of blue and green colored eyes (5 participants). We chose to combine hazel and brown eyes because the relative image contrast from the camera was qualitatively similar, as shown in Figure 5.1. In our results, we use this high- and low-contrast grouping to understand how resilient the different algorithms are to contrast changes resulting from eye color differences.
Table 5.1. Participant Demographics

<table>
<thead>
<tr>
<th>Participant Demographics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Average age = 22.25, sd = 6.181</td>
</tr>
<tr>
<td>Gender</td>
<td>6 males, 5 females</td>
</tr>
<tr>
<td>Eye color</td>
<td>Blue 4, Green 1, Hazel 2, Brown 4</td>
</tr>
</tbody>
</table>

Table 5.2. Lighting conditions

<table>
<thead>
<tr>
<th>Lighting Conditions</th>
<th>Average lux</th>
<th>std lux</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dark</td>
<td>5.5</td>
<td>2.5</td>
</tr>
<tr>
<td>Dim</td>
<td>7.4</td>
<td>2.5</td>
</tr>
<tr>
<td>Bright</td>
<td>105.0</td>
<td>14.5</td>
</tr>
</tbody>
</table>

Figure 5.1. High contrast eye image(left) and low contrast eye image(right). Here, eye color is used as a proxy for eye contrast because eye color is typically the best indicator of contrast between pupil and iris.

5.3. Environment setup

Experiments were conducted in a closed room with controlled lighting conditions. Three lighting conditions were used to simulate real-world situations. We did not constrain participants’ heads during the test; they were free to move their heads and look around as they saw fit. In each lighting condition, we measured the light level with an ambient light meter at the eye level pointing to the iPad’s display as summarized in Table 5.2. **Bright:** Firstly, we use a bright, well-lit room to conduct our experiments. Lighting was controlled via two overhead lights that provided ample light. The lights used are fluorescent bulbs which are known to produce flicker. We did not observe any noticeable flicker in the room, but we did not explicitly control for bulb flickering during our tests. **Dim:** Next we turn off all overhead lights and use a small office lamp to illuminate the room. This lamp is situated behind the participant as to light up what they are seeing, not their face. Participants described this
lighting as dim and not an environment in which they would typically work. We chose this configuration to produce visible grain artifacts in the images of the eye. The iPad tablet we employ uses a number of “low-light” techniques to boost contrast and, as such, lighting had to be decreased considerably to collect a lower quality image. **Dark:** in this configuration we turned all lighting off in the room. The only lighting to illuminate the face came from the screen of the iPad tablet. Example images and the visible artifacts are shown in Figure 5.2. Histogram equalization is used to enhance the contrast of these images. Note that in addition to the increased grain noise, the baseline pupil size is noticeably wider in the darker environments. Finally, we did not instruct participants to hold their head constant or to maintain any other pose. As a result, we observed a number of artifacts in the dataset such as blurring from head movement, occlusion, and participants sitting too close or far from the screen. Figure 5.3 gives some samples of issues observed in the dataset. We mitigate some of the artifacts with head pose estimation and face detection. When the face is not detected in the image, we do not estimate the pupil size.

![Eye images in different lighting conditions.](image)

**Figure 5.2.** Eye images in different lighting conditions. Histogram equalization has been applied to increase contrast.

### 5.4. Equipment and Gaze Signal Conditioning

Blinks cause distinct artifacts in the raw signals of the pupil diameter measurement. We remove blinks using a custom filtering process. First, we smooth the signal with a median filter of size 101 points, about 2 seconds. We then compare the filtered and raw gaze tracker
signals, discard data that differs more than three standard deviations. This is the same procedure of blink removal we use from our previous work [94, 95]. After removing blinks, we save the raw gaze tracker signal as our ground truth output for each eye.

We collect video data of the participants from the forward facing camera of an iPad Air 2. All camera settings are left as default, 50% brightness, and auto-brightness turned off. We note that the iPad provides some basic image processing and contrast enhancement for low light images as part of its default configurations. We collect videos having 720p HD (1028×720) at 30 FPS, in all lighting conditions. The sensor compensates the lack of light by increasing sensitivity (ISO). This does not affect the device frame rate in low lighting condition, but it affects the quality of the images as shown in Figure 5.2. Because the Gaze tracker sample rate is 50 FPS and the video frame rate is 30 FPS, we downsample the gaze tracker frames (with interpolation) to match the video frame rate.

5.4.1. Tasks

Participants were asked to perform two different tasks: the digit span task (memorization) and a typing task. These tasks are described in more detail in later sections. In each task, participants sit at a comfortable distance from the tablet screen and perform typing or memorization (Figure 5.4). We use these tasks to evoke cognitive load that can be measured by the Tobii Pro Glasses 2 gaze tracker. Pupil dilation is very sensitive to the lighting...
condition, so we record lighting condition at participant’s eye pointing to the iPad before the beginning of each task. We change the lighting condition in a controlled manner before the experiment starts. That is, ambient lighting condition does not change within a specific trial of an experiment, it only changes between trials. It is no visible light flickering, shimmer or ripple. Moreover, we applied counterbalancing technique in our experimental design. Thus, task ordering, the digit sequence within each digit span task and lighting level ordering were chosen randomly.

Figure 5.4. Experimental setup summary

5.4.2. Digit Span Task

The digit span task is used by many seminal studies in cognitive psychology [6, 7, 58]. During the task, a participant is asked to memorize a sequence of digits, normally between five to ten digits [58], and, after being presented with the entire sequence, the participant repeats those digits back. This artificially induces the cognitive load of an individual which, in turn, increases the pupil diameter. In other words, the pupil diameter gradually increases when a participant memorizes each digit in the sequence. Once the digits are spoken, the
pupil size gradually decreases back to the baseline. The amount of dilation in the pupil is typically small, sometimes less than one millimeter. In our study, each participant is asked to memorize sequences of length 6, 7, and 8 digits. Each sequence length is repeated for four iterations. These iterations are repeated under three different lighting conditions (bright, dim, dark), resulting in a total of $(3 \text{ sequences}) \times (3 \text{ lighting levels}) \times (4 \text{ iterations}) = 36 \text{ trials per participant}$. When the task begins, a sequence of digits appears, one-by-one, on the screen of the iPad every second. Only one digit is displayed on the screen at a time. Participants memorize these digits and, after being presented with all the digits, orally recites the entire sequence. This results in gradually increasing cognitive load as the participant memorizes the sequence. Longer sequences induce more cognitive load and, thus, larger dilations in the pupil. The experimenter ensures that the sequence is repeated without errors to ensure that the participant is sufficiently focused on the task.

5.4.3 Writing Tasks

While we have presented results of PupilNet for capturing pupil size in the digit span task, it is still unknown if PupilNet can be used outside of this specific cognitive task. The motivation for PupilNet was to understand individual’s cognitive load in general, but the current results only display an ability to measure task evoked cognitive load. In general, using pupil diameter to understand general cognitive load is still an open research problem. However, it is important to understand if PupilNet can achieve the same conclusions as a gaze tracker for tasks that require more than simple digit memorization. To investigate this, we chose to evaluate PupilNet while subjects perform a typing exercise. We consider the results of this experiment to be exploratory because the relationship between pupil diameter and cognitive load outside of simple tasks is not a well understood phenomenon. In this way, our aim in the analysis is to discover if PupilNet and the gaze tracker deliver similar conclusions.

The digit span task is designed to evoke cognitive load and is therefore straightforward to evaluate. However, the exact degree of cognitive load for a typing task is not well defined. We
hypothesized that there would be a difference in the cognitive load exerted by individuals that can and cannot perform touch typing. Touch typing is defined as a skill in which individuals use all of one’s fingers to type, without looking at the keys. Touch type requires a mix of memorization and muscle memory. We assume the memorization will result in different pupil dilation behaviors between participants that can perform touch typing and and individuals that cannot perform touch typing. Moreover, because typing involves muscle memory, we anticipate cognitive load differences will manifest for participants that focus less on the movement of their hands. We know that cognitive load is affected by tasks involving muscle memory from previous research by Yuksel et al. [138], who used cognitive load differences (measured via brain interface) in a system to train individuals to play piano. Therefore, we will explore if it is possible to observe similar shifts in cognitive load using PupilNet and the gaze tracker between touch typing and non-touch typing behaviors. We note that throughout this experiment we use within-subject trained models to infer pupil size because it was shown to have more consistent accuracy across a wider range of subjects.

We designed an experiment in which users carried out a typing task using a custom interface (Figure 5.5). Participants were asked to type a given paragraph of 163 words with no time limit. The interface shows 18-20 words at a time to minimize variation in lighting intensity and help mitigate up-down eye movement. A cursor moved over the characters as the user typed. The cursor would stop moving if the typed character was not the correct character. Typing experiments were carried out only in the brightly lit room and only using within-subject trained models. We recruited participants that had already completed the digit span task experiment. In this way, we could use the already trained within-subject PupilNet model for that participant to generate the PupilNet prediction of their pupillary response during typing. In this way, we used the digit span task to calibrate the PupilNet model for each participant.
5.5. Ground Truth Selection

In early 2000, Klingner demonstrated that high-resolution, remote eye-trackers could provide sufficient precision for measuring cognitive load via TEPR. TEPR is measured by monitoring percent changes in pupil size from the baseline pupil size (i.e., the pupil size after external stimuli is presented) [57]. This is described mathematically by equation 5.1:

$$d_{\text{change}} = \frac{d_i - \text{mean}(d_{\text{baseline}})}{\text{mean}(d_{\text{baseline}})}$$  \hspace{1cm} (5.1)

where $d_{\text{change}}$ denotes the percentage change in pupillary diameter from baseline (pre-stimulated stage). $d_{\text{baseline}}$ denotes pupillary diameter during a pre-stimulated stage which is typically a few seconds before stimulus is applied. TEPR researchers have found the percentage pupillary change advantageous because the unit can be standardized regardless of a unit (mm or pixels) and original baseline size. Even so, there is a changing relationship between pupil dilation in response to TEPR and the lighting in the room. A number of researchers have shown that percentage pupil dilation is not consistent across different lighting [136].

We use a Tobii Pro glasses 2 [1] for ground truth comparison in this study. The Tobii Pro glasses 2 has several advantages over remote gaze trackers. It is head mounted and therefore does not suffer from problems caused by head pose or distance. It provides fixation.
estimation, which can be useful for discounting the accommodation reflex (pupil dilation when fixating to far objects). The device also provides video of the participant’s point of view and where she or he is looking. This kind of information is useful for further cognitive load analysis. The pupil size is recorded at a resolution of 50 FPS. Our previous work showed that a more economical remote gaze tracker also works, as long as it provides high-resolution IR images with a hough circle algorithm in a controlled environment (i.e. office or classroom) [95].

Previous research has shown that remote gaze trackers are appropriate for measuring cognitive load [57, 58]; however, the Tobii Pro Glasses is not marketed or commercially tested for its ability to discern pupillary changes. To test the performance of Tobii Pro Glasses 2 in measuring pupil size, we compared it with a medical grade pupillometer. In particular, we use a Neuro-optics VIP hand-held pupillometer. This device allows static measurement of pupil size and is a gold standard device for measuring pupil diameter in millimeters. To compare the results of pupillometer and the Tobii Glasses Pro, we measured pupil size of seven volunteers in three lighting conditions each (bright, dim, dark). For each participant, we conducted ten iterations in each lighting condition for a total of $7 \times 10 \times 3 = 210$ measurements. The pupil size was captured at the same time from both devices for three seconds each (the medical pupillometer can only record three seconds of data at a time). The pupillometer contains a rubber encapsulating shield completely around the eye. To measure the pupil size from both devices at the same time, we placed the rubber shield inside the lens of the Tobii Glasses. The results, averaged across each iteration and each participant, are shown in Figure 5.6. From the graph, we can observe that the Tobii Pro Glasses 2 consistently captures pupil size slightly lower than the medical grade pupillometer by 0.1mm on average (sd=0.78). The a small difference increases when the environment lighting decrease. Because our interest is in tracking changes in pupil size (not absolute size), we calculated percent change from the pupil size in the bright condition as shown in Figure 5.7. The result show statistically insignificant difference (2-tails t-test, $p > 0.05$) on both absolute millimeter and percentage change units. This evidence suggests that the Tobii
Glasses Pro to be a suitable ground truth measure of percentage changes.

![Figure 5.6](image1.png) Averaged absolute pupil size affected by pupillary light reflex across each iteration and each participant in three lighting conditions

![Figure 5.7](image2.png) Averaged percent change in pupil size affected by pupillary light reflex across each iteration and each participant in three lighting conditions

To further examine if the Tobii glasses are an appropriate measure of cognitive load, we conducted analyses of the percentage pupil dilation for individuals when memorizing sequences of different lengths. This experiment is commonly referred to as the digit span task and is described in more detail later on (we use this task to evaluate PupilNet). We averaged the pupillary response over time (commonly called the pupillogram) from the Tobii Glasses grouped by 6-digit, 7-digit, and 8-digit across different lighting conditions and participants as shown in Figure 5.8. In this experiment, our pre-stimulus period is 10 seconds. The result is consistent with the results from other TEPR literature [6, 7, 57] where pupil dilations are increased, on average, when participants memorize longer sequences. Pupil size for 6-digit sequences is clearly separated from 7- and 8-length sequences. However, 7- and 8-digit sequences are not as clearly separated. This result is also consistent with previous research, and one theory is that human short-term memory capacity peaks at either 7 or 8 digits for most individuals [13, 79].

In summary, Tobii Pro Glasses 2 estimates pupil size consistency smaller than medical pupillometer. However, because we are interested in measuring percentage changes in pupil size and the Tobii Glasses exhibit similar behavior to previous research, we consider the Tobii Pro Glasses 2 a suitable ground truth measurement device.
Figure 5.8. Average of pupillary diameter in three lighting conditions

5.6. Conclusion

This chapter explained experiment setup and data collection for evaluating our purposed algorithms. The results of these work will be described in the next chapter.
6.1. Cross Validation

To train our machine learning algorithm, we employ two different cross-validation strategies: leave-one-subject-out (LOSO) cross-validation and within-subject cross-validation. We perform LOSO cross-validation on 11 participants (i.e., 10 models are created and tested on the left out user). This mirrors a use case for an out-of-the-box solution that requires no specific user calibration. Ideally, this is the preferred scenario for the PupilNet system, where no user calibration is needed.

However, we also wanted to investigate the benefit of personalizing the model with a calibration procedure. The within-subject cross-validation mirrors a use case where calibration is available for a specific participant. That is, we assume that there will be a calibration procedure using a remote eye tracker to retrain the PupilNet architecture. We choose half of the data for training and half of the data for testing, stratified across sequence length and lighting levels. So for one participant, 18 trials would be used for training and 18 trials for testing. In this way, the data are separated in contiguous time spans which is imperative for a realistic evaluation of time series data. Such a situation limits the utility of the PupilNet system. However, in some scenarios, we anticipate that the calibration step could be warranted.

6.2. Results: Characterizing Pupil Dilation in the Digit Span Task

During the digit span task, we expect the pupil to dilate while the participant memorizes the digit sequences and for the pupil diameter to constrict quickly after speaking the digits. As the length of the memorized sequence increases, the pupil should dilate more. However,
every individual has a slightly different “baseline” pupil size which shifts in response to fatigue, sleepiness, and lighting. Therefore, we wish to characterize the percentage increase in the pupil size above baseline. We utilize three different evaluation criteria for ascertaining this difference. The first evaluation metric we choose is a custom metric based upon the difference between peak pupil diameter and baseline diameter (discussed next). We also employ the correlation coefficient for one iteration of the digit span task between the estimate of percentage pupil diameter and actual percentage pupil diameter. Finally, we take the raw point-by-point difference between the estimated and actual percentage pupil dilation. We abbreviate each algorithm according to: PupilNet (PN), Modified Starburst (MSB), and 2-Level Snakuscules (2LSN).

We define an evaluation metric that is the difference between the percentage pupil dilation in the gaze tracker and the algorithm. To calculate this custom evaluation metric, we first take the average size in millimeters during the 5 seconds before the digits appear on the screen. For the gaze tracker we denote this as $GT_{start}$ and for PupilNet we denote this as
\( PN_{\text{start}} \). The peak pupil dilation is calculated by the maximum dilation value while the participant speaks back the memorized digits, denoted as \( GT_{\text{peak}} \) and \( PN_{\text{peak}} \) for the gaze tracker and PupilNet, respectively. Figure 6.2 shows two example digit span sequences for one participant with each measurement called out. Our evaluation metric is difference in these values. To compare the gaze tracker and an algorithm \( X \), for example, we calculate the dilation difference as follows:

\[
\Delta_X = \frac{GT_{\text{peak}}}{GT_{\text{start}}} - \frac{X_{\text{peak}}}{X_{\text{start}}} - \mu_{\Delta_X}
\]

We denote this difference for different algorithms by the subscript \( X \). For PupilNet, the dilation difference would be written as \( \Delta_{PN} \). We note that we are taking the raw subtractive difference between two percentages. This dilation difference therefore if \( \Delta_{PN} = -1\% \) it would mean that PupilNet calculated a 1\% greater increase in pupil diameter than the gaze tracker. The term \( \mu_{\Delta_X} \) is a constant, scalar correction factor for algorithm \( X \) that eliminates any constant bias in the results. We calculate this correction factor by the average difference between \( \frac{GT_{\text{peak}}}{GT_{\text{start}}} \) and \( \frac{X_{\text{peak}}}{X_{\text{start}}} \) for each algorithm \( X \). We apply a constant correction factor because we are most interested in the variation of the residuals and are less concerned with a systematic bias in the entire output. For each algorithm, we apply correction factors of \( \mu_{\Delta_{\text{MSB}}} = -5.0\% \), \( \mu_{\Delta_{\text{LSN}}} = -8.0\% \), and \( \mu_{\Delta_{PN}} = 6.2\% \). In practice, this correction factor would need to be added to the output of the pupil dilation estimate, \( \frac{X_{\text{peak}}}{X_{\text{start}}} \), when the system was deployed. Only one correction factor is applied to each algorithm. That is, we do not apply correction factors based upon subgroups of the data, only the overall average. We also note that reporting mean differences with the correction factor would be inappropriate. Instead, we report a variation of the residual differences. Smaller residual magnitudes centered around the correction factor are desired.

This metric ignores whether both PupilNet and the Gaze tracker recover the same exact millimeter measurement, as long as the percentage change from baseline to peak dilation is similar. Using \( \Delta_X \), we can understand if the dilation during an iteration of the digit span task is estimated closely by an algorithm. To help understand the expected values for
the percentage dilation for the gaze tracker, $\frac{GT_{\text{peak}}}{GT_{\text{start}}}$, we look at the variations seen by the gaze tracker in Figures 5.8. These figures show, on average, what the expected percentage difference from baseline should be for each digit sequence, and thus, the expected percentage differences for different cognitive loads. From the literature [49], the most marked cognitive differences occur between sequences of length 6 and length 8. From Figures 5.8, we can observe that the difference between these sequence lengths is anywhere from 3%–6%, on average. Therefore, a good algorithm would need to consistently have $\Delta_X < \pm 3\%$. We note that this is only an estimate of the needed $\Delta_X$, which could still discern useful differences even if $\Delta_X$ is greater than $\pm 3\%$, especially if the value constantly under- or over-estimates the value. Moreover, the average pupil dilation across participants contains considerable overlap, so many pupil diameter differences could be greater than 10% fluctuation, depending on the effort required for a particular individual to memorize a sequence.

6.2.1. Within-subject $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$ Across Lighting

We break up our results into several sections based upon the research question addressed by the experiment or the analysis. We begin with analysis from the digit span task sequences at different lighting levels. In our first question, we ask: what algorithm performed most accurately at detecting pupil dilation during the digit span task? Figure 6.3 shows boxplots of the results across algorithm for each lighting level using within-subjects training. Each boxplot represents the distribution of differences, $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$, for all 11 participants combined. They are grouped by the lighting in the room as bright, dim, and dark. Please recall that the lighting is chosen to produce noticeable, prominent artifacts in the images of the eyes representing worst-case lighting scenarios. Across all lighting conditions, PupilNet performs superior to all other algorithms. Overall, the interquartile range for $\Delta_{PN}$ is $-2.6\%–2.8\%$, compared to $-9.2\%–4.8\%$ for $\Delta_{MSB}$ and $-8.9\%–4.3\%$ for $\Delta_{2LSN}$. When looking across lighting conditions, all algorithms perform worse in low lighting conditions. However, the decrease in performance for PupilNet is markedly less than the other algorithms. In dark lighting ($\approx 5$ lux), PupilNet has an interquartile range of $-4.0\%–2.4\%$, which is only
marginally increased from the bright condition. However, the overall maximum and minimum variation (as given by the whiskers of the boxplot) is increased. While these extrema are concerning, the vast majority of trials have $\Delta_{PN}$ within 5% of the actual value. Given that we expect the cognitive load to induce dilations in the range of 3%–6%, we conclude that PupilNet follows nearly the same conclusion as the gaze tracker in brightly lit environments and is robust to most lighting changes. However, one should expect inaccurate dilation measures in dim and dark lighting conditions.

To further investigate the performance of each algorithm across lighting, we present results with a different evaluation metric, linear correlation. Correlation is agnostic to systematic biases in the outputs. For our purposes, correlation measures how closely two time-series increase and decrease together. Figure 6.4 is a boxplot of correlation coefficients between the gaze tracker and the algorithm under test: PN, MSB, or 2LSN. The correlation coefficient is calculated between the gaze tracker and the algorithm separately for each iteration of the digit span task. In this way, we seek to understand if the algorithm and gaze tracker increase and decrease linearly during each iteration. All of the correlations for all participants are grouped for the boxplot. The conclusions of correlation are similar to that of $\Delta_{PN}$. PupilNet and the gaze tracker produce similar trends for each iteration of the experiment, with overall correlation typically above 0.5. PupilNet by far outperforms MSB and 2LSN, which have significantly smaller median correlations based upon a Wilcoxon rank-sum test ($p < 0.01$).

Across lighting conditions, we also see that PupilNet has some degradation in performance for dim and dark environments. The degradation in the median correlation is significantly smaller than when in a bright environment based on a Wilcoxon rank-sum test ($p < 0.01$). This leads to a similar conclusion as previously: dim and dark environments will cause some inaccurate dilation measures, but overall, PupilNet is far more robust than other algorithms across lighting. Because of the clear performance advantage of PupilNet compared to the other algorithms, we focus primarily on the results of PupilNet for the remainder of the paper.
While our previous evaluation criteria show trends in the overall conclusion from each digit span task iteration, they do not elucidate the point-by-point accuracy of PupilNet. That is, how well does PupilNet match gaze tracker time series output. We use the percentage pupillary response as measured from baseline, $d_{\text{change}}$, that is customarily used when reporting pupillograms in studies of cognitive load (as discussed in equation 5.1). We report the results of this point-by-point difference using a modified Bland-Altman plot. The plot is shown in Figure 6.5. The vertical axis is the difference between the percentage pupil change, $d_{\text{change}}$, as measured by the gaze tracker and PupilNet.

The horizontal axis is the percentage change as measured by the gaze tracker. This type of plot helps to show any bias and variance in the estimates across the range of the ground truth measurement. We do not apply any constant correction factor to the plots, as was done with the $\Delta X$ measure. The majority of $d_{\text{change}}$ measurements are zero during the tests. This is because there is a relatively long duration of silence when the participant waits for the digits in the sequence to appear on the tablet, compared to the relatively quick memorization and speaking of the digits. We can see that PupilNet has a constant underreporting bias. However, close inspection of the plot reveals that there is also a small linear bias in the residual values that becomes more apparent in darker lighting conditions. This supports previous conclusions that darker lighting conditions degrade the performance of PupilNet. However, in bright conditions, there is almost no linear trend in the Bland-Altman plot, and the bounds of variation from percentile measures are mostly within 5% of the actual value. This supports a conclusion that PupilNet estimates follow the gaze tracker output closely.
Figure 6.2. Comparison of PupilNet and gaze tracker for two different participants. Also shown are the measures used in evaluating pupil dilation inference.

Figure 6.3. Summary of $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$. Results are separated by the average lighting level in the room, bright ($\approx100$ lux), dim ($\approx7.5$ lux), and dark ($\approx5$ lux). Please note that the plots vary from -30% up to 50%.
Figure 6.4. Results of correlation coefficient

Figure 6.5. Bland-Altman plot between With-in subject PupilNet and Gaze Tracker
6.2.2. Within-subject $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$ Across Different Eye Contrasts

In our second analysis, we wish to understand how PupilNet performs with individuals having a different contrast between iris and pupil. As discussed, gaze trackers use infrared lighting source and therefore are quite robust to different pupil/iris contrasts because the infrared light uniquely illuminates either the pupil or the iris. However, the contrast between the iris and pupil is critical for RGB-based algorithms like PupilNet.

Because the conclusions between evaluation criteria are similar, we choose to report results only for evaluation metric $\Delta_X$ for the remainder of analyses. Figure 6.6 shows boxplots of $\Delta_{PN}$, $\Delta_{MSB}$, and $\Delta_{2LSN}$ using within-subjects training in a bright environment. We group the results by the contrast of the participant’s eyes. From Figure 6.6 we can see that for participants with high contrast eyes, the algorithms have more similar performance, but PupilNet still has markedly smaller residuals. In the low contrast group, however, PupilNet is the only algorithm that maintains performance. In particular, 2LSN dramatically drops in performance for low contrast eyes but is a relatively good performer for high contrast eyes. This indicates that the performance gap in MSB and 2LSN might be explained by a combination of eye contrast and lighting condition. Figure 6.6 also shows multiple subgroups for each algorithm, lighting condition, and eye contrast group. For PupilNet, the degradation in performance due to lighting differences is about the same regardless of whether the participant was in the high- or low-contrast group. However, MSB and 2LSN had dramatically different performance across lighting, depending on the contrast group. We therefore conclude that PupilNet works reliably across different eye contrasts and the degradation in performance across lighting conditions is similar for different eye contrasts. MSB and 2LSN do not exhibit this reliability across eye contrast or lighting.
6.2.3. $\Delta_{PN}$ for Within-subject and Across-subject Training

In this analysis, we want to understand if PupilNet generalizes across subjects. That is, all results thus far presented assume that a calibration stage has been used to collect gaze tracker data and RGB eye data. In this analysis, we compare within-subjects cross-validation to leave-one-subject-out (LOSO) cross-validation. Ten participants’ data are used to train a PupilNet model and then tested on the remaining participant. The results are summarized in Figure 6.7. In the figure, p01 means the model is trained using participants p02 to p11 and tested on participant p01. This mirrors a use case when no gaze tracker calibration would be available. All lighting conditions and all eye contrast groups are combined. We show results grouped by each participant in the study and for all users combined. From Figure 6.7 we can see that most of the models perform slightly worse than the within-subjects models. However, the degradation in performance is not consistent. Some users show dramatic degradations in performance. This is especially true for p09, who is degraded in performance much more than other participants, with an interquartile range that includes 20% dilation difference. We reviewed the video for this participant and discovered that this participant had trouble wearing the Tobii Glasses frame because her head diameter was below average. This resulted in the glasses slipping down from the bridge of her nose and occluding part of the eye. The
personalized, within-subject training was able to account for this occlusion, but the LOSO cross validation had no training examples with this type of occlusion. However, when we disregard p09, there are still a number of users with significantly degraded performance. In Figure 6.7, we denote users with significantly different results between within-subject and LOSO with an asterisk (*). All but three users have statistically worse residuals.

It is unclear if gathering more data from more diverse participants would boost the performance of the LOSO algorithm. It is possible that collecting more data and creating a deeper, more complex network could improve results. This can only be determined empirically. Even so, from this result, we can conclude that PupilNet can be used as an out-of-the-box system for some individuals but not all users. From our previous results, the results can be readily and significantly improved through a calibration process.

![Figure 6.7](image)

Figure 6.7. Results of across-subject leave one out cross validation for each model. An asterisk (*) indicated that the distributions are statistically different based on a two-tailed T-test ($p < 0.01$)

### 6.3. Results: Within-subject Modeling for Typing Behavior

Figure 6.8 shows the gaze tracker data for four participants for a two minute interval during typing. There are marked differences in the pupil dilation for this task depending on the whether the user reported as being able to perform touch typing or not. The touch typing subjects (left) had considerably fewer peaks in their pupillary response than the non-touch
typing participants. It is important to note that not all of peaks are due to cognitive load increases. When users gaze down at the keyboard, the pupil size will change in response to the different focal length. Also, the illumination from the tablet interface may dilate and constrict the pupil in response to head motions. Because of the exploratory nature of this experiment, we are less concerned with the mechanism for why the pupil dilates or constricts; we want to understand if the same behavior is apparent in PupilNet estimate.

Figure 6.8. Gaze tracker data for each of the fours users during a subset of the typing task.

We predict the pupil size for each participant using PupilNet. The training examples for each participant come from the same models used to predict their digit span task from experiment one (within-subject). In this way, our method mirrors a use case where the participant has provided some calibration data at an earlier date. To understand if PupilNet can discern the differences in peaks, we extract the peaks in each time series using the continuous wavelet transform as proposed by Du et al. [24]. This algorithm convolves the signal with an array of wavelets with different frequency and time bandwidths. Peaks are uniquely apparent by setting a global threshold for the response of the wavelets. We use this algorithm to count the number of peaks in each 30 second interval for each participant during the typing task. Figure 6.9 shows a box and swarm plot of the number of peaks for each 30-second interval grouped by touch and non-touch typing. It is apparent from
the gaze tracker data that a clear difference exists in the number of peaks extracted for non-touch typing and touch typing. Non-touch typing individuals have considerably more peaks as extracted by the algorithm. This trend is also well established in the PupilNet box and swarm plots. However, the difference between touch typing and non-touch typing is not quite as pronounced. We can observe this because there is a slight overlap in the interquartile range for each group using PupilNet, but not the gaze tracker. We also investigated if there are any meaningful statistical differences in the distributions and we find that the results follow an intuitive pattern. A two tailed t-test between each group reveals that: (1) no significant difference ($p = 0.15$) exists between groups Non-touch GT and Non-touch PN, (2) no significant difference ($p = 0.24$) exists between groups Touch GT and Touch PN, and (3) a significant difference does exist ($p < 0.01$) between the touch typing group and non-touch typing group for both the gaze tracker and PupilNet.

![Box and swarm plot of the number of peaks for each 30-second interval grouped by touch and non-touch typing.](image)

Figure 6.9. A box and swarm plot of the number of peaks for each 30-second interval grouped by touch and non-touch typing.

While the gaze tracker and PupilNet do not give identical measurements for the pupillary dilation during the typing task, the results are largely similar and the conclusions are identical. We therefore conclude that PupilNet shows great promise in inferring cognitive load in cognitive tasks beyond the digit span task. We have only shown this difference for a typing task, but we hypothesize that the results extend to a number of different task where
a user interacts with a tablet. As we begin to understand how cognitive load changes for different tasks, PupilNet can be evaluated in other cognitively complex tasks. We hypothesize that PupilNet can effectively capture aspects of many pupillary response tasks but leave the evaluation of other tasks to future work. In future work we also wish to understand if PupilNet and a gaze tracker can be used to judge the quality of a number of tasks like reading, writing, and test-taking. Understanding the cognitive load of subjects during these tasks could be key to understanding confusion, attention, and engagement. We also want to explore the combination of different sensing modalities with PupilNet in these scenarios, such as facial expressions, gaze, and heart rate. Combining these different sensing protocols could be key in the creation of a robust context-aware computing environment.

6.4. Conclusion

We described the comparison of three algorithms: new modified starburst, 2-levels snakucules, and PupilNet. Modified Starburst and 2-levels snakucules perform comparatively equal. On the other hand, PupilNet shows greatly outperform in almost all situations. Nonetheless, PupilNet requires users specific calibration data to be most accurate.
Chapter 7

GOING DEEPER: OPTIMIZING PUPILNET WITH MODERN DEEP LEARNING
TECHNIQUES

Convolutional Neural Networks outperform many engineering solutions in image recognition problems [39, 62] and have almost exclusively occupied machine learning community in recent years. Both academics and industrial scientists actively research deep neural networks. In our work, PupilNet performs better than our previous hand-crafted feature engineering methods (Modified Starburst and 2-Level Snakes), even though PupilNet’s architecture is relatively simple: with two convolutional, two pooling, and one fully connected layers. Recent optimization techniques might be able to improve the performance of PupilNet. In this chapter, we experiment with modern Deep CNN architectures and compare their performance.

7.1. Experimental Convolutional Neural Network Architectures

Large networks might be better than smaller networks in terms of predicting pupil size. This is because bigger networks have a higher capacity and can store higher data complexity [39]. As a result, this may lead to higher performance assuming proper regularization strategies are applied. The current version of PupilNet only has two convolutional layers. Therefore, we asked the question “does larger PupilNet improve the network performance?”

7.1.1. Wider Versus Deeper PupilNets

Larger networks can mean wider networks and deeper networks. To clarify, we defined wider as implying the addition of more neurons while retaining the same number of layers (i.e., Figure 7.2) and deeper as implying the addition of a number of layers whilst retaining the same number of neurons for each layer (i.e., Figure 7.3). It is a common practice to add
more layers as opposed to only widening networks [8]. This because a deeper network can warp space and form a new basis that might potentially enable it to distinguish two classes. Nonetheless, it is unclear whether these suggestions can be applied to regression problems and it is worth including “Wide networks” in this experiment.

We created two wider and three deeper network architectures while leaving other hyperparameters unchanged. It is a common practice to increase the number of filters in neural networks by a factor of two. Thus, the wider network consists of 64 and 128 filters for each convolutional layer (W64 and W128 respectively), while maintaining the same number of layers in the network. For deeper networks, we add more convolutional layers while maintaining the same number of filters. A summary is shown in Table 7.1. It is worth noting that larger neural networks are more likely to suffer from over-fitting. While we do not increase the regularization parameters (50% dropout rate and L2 = 0.01), we did monitor training and validating losses and performed early stopping for all deep and wide networks (i.e., we stopped training as soon as we observed divergence between training loss and validation loss).

7.1.2. Residual PupilNet: Modern Deep Network Architectures

Utilizing modern architectures might improve the performance of PupilNet. Residual networks have shown outstanding performance in image classification [39] and are designed to solve the degradation problem where adding more layers to a deep network could lead to a higher training error. One way to solve the degradation problem is to stack multiple shallow pretrained networks which helps, but difficult to achieve in practice. Deep residual networks try to solve the problem by providing a modular residual block. A residual block consists of a small network and a short circuit. Although no mathematical proof exists, the hypothesis is that if some stacks of small networks are not easy to optimize, the optimizer can skip those networks and use a short circuit route ($x_{identity}$) instead. This allows networks to safely stack many residual blocks. For our problem, a residual block consists of a small a residual circuit ($k \times k$ Conv layer, a ReLU, and a $1 \times 1$ Conv layer), and a shortcut circuit.
Figure 7.1. Original PupilNet

Figure 7.2. An example of wider PupilNet

Figure 7.3. An example of deeper PupilNet
Table 7.1. Table summary of PupilNet architectures. Each column represents each architectures of various version of PupilNet

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Wide64</th>
<th>Wide128</th>
<th>Deep1</th>
<th>Deep2</th>
<th>Deep3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3x32</td>
<td>3x3x64</td>
<td>3x3x128</td>
<td>3x3x32</td>
<td>3x3x64</td>
<td>3x3x32</td>
<td></td>
</tr>
<tr>
<td>3x3x32</td>
<td>3x3x64</td>
<td>3x3x128</td>
<td>3x3x32</td>
<td>3x3x64</td>
<td>3x3x32</td>
<td></td>
</tr>
<tr>
<td>3x3x32</td>
<td>3x3x64</td>
<td>3x3x128</td>
<td>3x3x32</td>
<td>3x3x64</td>
<td>3x3x32</td>
<td></td>
</tr>
<tr>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3x3x64</td>
<td>3x3x64</td>
<td>3x3x128</td>
<td>3x3x64</td>
<td>3x3x64</td>
<td>3x3x64</td>
<td></td>
</tr>
<tr>
<td>3x3x64</td>
<td>3x3x64</td>
<td>3x3x128</td>
<td>3x3x64</td>
<td>3x3x64</td>
<td>3x3x64</td>
<td></td>
</tr>
<tr>
<td>3x3x64</td>
<td>3x3x64</td>
<td>3x3x128</td>
<td>3x3x64</td>
<td>3x3x64</td>
<td>3x3x64</td>
<td></td>
</tr>
<tr>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td>2x2 max pool, stride 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024 dense, ReLU</td>
<td>1 dense, MSE</td>
<td>1 dense, MSE</td>
<td>1 dense, MSE</td>
<td>1 dense, MSE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$x$ with $1 \times 1$ Conv layer. Then, these two tracks are added and are passed on to another ReLU, as shown in Figure 7.4. Although there are many variations of residual networks (i.e. Wide ResNet, Resnet in Resnet, ResNeXt [118, 135, 139]), our Residual PupilNet used deep ResNet incorporating $1 \times 1$ Conv layers in each residual block acting as feature pooling. The feature pooling layers allows a network to go deeper, reduces noise, and increases the strength of regularization whilst retaining the use of reasonable computation resources. A residual version of PupilNet is shown in Figure 7.5.

Three variances of Residual PupilNets are shown in Table 7.2. They follow the same pattern as Deep-PupilNet, meaning that the structures are similar to Deep1, Deep2, and Deep3. The only difference is that they use residual blocks instead of convolutional layers (Figure 7.5).
Figure 7.4. Typical residual block [39]. Input $X$ is features from previous layer. Weight layers are convolutional layer. [39]

Figure 7.5. Basic Structure of Residual PupilNet
Table 7.2. Table summary of PupilNet architectures. Each column represents each architectures of various version of Residual PupilNets

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Residual1</th>
<th>Residual2</th>
<th>Residual3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv 3×3×32</td>
<td>ResConv 3×3×32</td>
<td>ResConv 3×3×32</td>
<td>ResConv 3×3×32</td>
</tr>
<tr>
<td></td>
<td>ResConv 3×3×32</td>
<td>ResConv 3×3×32</td>
<td>ResConv 3×3×32</td>
</tr>
<tr>
<td></td>
<td>2×2 max pool, stride 2</td>
<td>2×2 max pool, stride 2</td>
<td>2×2 max pool, stride 2</td>
</tr>
<tr>
<td>Conv 3×3×64</td>
<td>ResConv 3×3×64</td>
<td>ResConv 3×3×64</td>
<td>ResConv 3×3×64</td>
</tr>
<tr>
<td></td>
<td>ResConv 3×3×64</td>
<td>ResConv 3×3×64</td>
<td>ResConv 3×3×64</td>
</tr>
<tr>
<td></td>
<td>2×2 max pool, stride 2</td>
<td>2×2 max pool, stride 2</td>
<td>2×2 max pool, stride 2</td>
</tr>
<tr>
<td></td>
<td>1024 dense, ReLU</td>
<td>1024 dense, ReLU</td>
<td>1024 dense, ReLU</td>
</tr>
<tr>
<td></td>
<td>1 dense, MSE</td>
<td>1 dense, MSE</td>
<td>1 dense, MSE</td>
</tr>
</tbody>
</table>

7.2. Experimental Design

We implemented the network using Tensorflow r0.12 implemented in python 3.5. All PupilNet variations were trained within subject for 400 iterations. Beyond this, the networks tend to over-fit, and the performance does not have significant improvement. We also employ L2 regularization of each convolutional and dense layer, with a scaling factor of C=0.01 for all layers. Training of the network parameters was completed on a machine with an Intel Xeon@2.6GHz CPU (24 cores) and 320GB of RAM.

7.3. Results: Within-Subject Models Evaluation

Eight PupilNet variations were trained and compared with Original PupilNet. Figure 7.6 shows the correlation coefficient for nine variances of PupilNet separated by lighting conditions. Any iteration that has \( p < 0.05 \) is excluded. Overall, wider variance has significantly lower correlation than deeper and residual variance, while deeper and residual variance have similar correlations. Networks that are inside each variance (\( i.e., \) Deep1 to Deep3) perform
similarly to each other. In bright conditions, residual versions have a slightly higher correlation although the correlation is lower in dark conditions. Table 7.3 shows t-test values for each pair of networks. Overall, deeper and residual variance have similar correlations.

![PupilNet Models Comparison: Correlation Coefficient](image)

Figure 7.6. Correlation coefficient of nine variance of PupilNet separating by lighting conditions. Each box contains 198 data points (best view in digital version)

We measured the regression error with the mean absolute error of percentage change from baseline, although our training network is in fact optimized for mean square error. The reason for this is that, during training, we wanted to punish large errors. The backpropagation algorithm generates large gradients in the networks as a result of which the networks may train faster. For evaluation purposes, the mean absolute error is more interpretable. Our previous works show that, in average TEPR cases, changes in pupillary diameter from baseline up to 15% occur. What we see in Figure 7.7 shows that all networks are within 15% overall as shown on the y-axis. All networks have lower errors in bright conditions and the errors increase when the amount of light decreases. Among all of the variations of PupilNet, Residual2 has the lowest error (median=1.83%).

As expected, the network’s performance decreases with a decrease in environmental lighting. In a dark environment, pupil data are degraded. This situation might be similar to
Table 7.3. t-test values compared against each pair of networks. (* indicates significant different (p < 0.05))

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Wide 64</th>
<th>Wide 128</th>
<th>Deep 1</th>
<th>Deep 2</th>
<th>Deep 3</th>
<th>Residual 1</th>
<th>Residual 2</th>
<th>Residual 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>4.98e-6 *</td>
<td>5.76e-6 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wide 64</td>
<td>5.0e-6 *</td>
<td></td>
<td>7.68e-7 *</td>
<td>4.0e-6 *</td>
<td>1.71e-6 *</td>
<td>5.20e-7 *</td>
<td>3.38e-7 *</td>
<td>3.01e-6 *</td>
<td></td>
</tr>
<tr>
<td>Wide 128</td>
<td>6.0e-6 *</td>
<td></td>
<td>7.93e-7 *</td>
<td>4.0e-6 *</td>
<td>1.94e-6 *</td>
<td>6.07e-7 *</td>
<td>3.81e-7 *</td>
<td>4.02e-6 *</td>
<td></td>
</tr>
<tr>
<td>Deep 1</td>
<td></td>
<td>7.68e-7 *</td>
<td>7.93e-7 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deep 2</td>
<td></td>
<td>3.62e-6 *</td>
<td>4.02e-6 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deep 3</td>
<td></td>
<td>1.71e-4 *</td>
<td>1.93e-4 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Residual 1</td>
<td></td>
<td>5.20e-7 *</td>
<td>6.06e-7 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Residual 2</td>
<td></td>
<td>3.38e-7 *</td>
<td>3.81e-7 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Residual 3</td>
<td></td>
<td>3.49e-6 *</td>
<td>3.81e-6 *</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.7. Percent change of mean square error of nine variance of PupilNet separating by lighting conditions. Each box contain about 79,200 data points. (best view in digital version)
participants with a darker iris. To find out if this is the case, we divided the dataset into two groups: high contrast and low contrast eyes. Figure 7.8 shows a comparison of the correlation coefficients between low and high contrast eyes. High contrast eye models have a higher correlation in all lighting situations compared to low contrast eyes. In particular, the Residual2 model has a consistent high correlation between 0.68 and 9.84. In contrast, Wide128 has the lowest correlation. In low contrast eyes, all models show a lower correlation. The performance between bright and dim lighting conditions is similar but is worse in dark situations. Wide128 has a low correlation in a dim situation because most of its iterations are excluded due to the fact that $p > 0.05$; of the original 198 points, on 23 data points were left. Figure 7.8 shows the mean absolute error of percentage change from baseline for high and low contrast eyes. The bright condition has the lower errors compared to the dim and dark conditions. All models are within the 15% range and the residual variance performs the best. Moreover, there is a large separation between the lighting situations. Bright, dim, and dark are significantly different in high contrast eyes. On the other hand, the separation is less profound in low contrast eye participants. This may imply that the networks learn to detect the pupil. Because the pupil information in low contrast and dark lighting conditions is degraded compared to bright and high contrast eyes, this may directly affect the network performance.

7.3.1. Inspect Model Quality with Modified Residual Plots

Until now, we have used the correlation coefficient and mean absolute error to compare the PupilNet variations. However, we have not yet determined which models are good enough for a cognitive load classification model. To find this out, we need to understand the cognitive load classification process from pupillogram. TEPR suggests that pupil size increases with increasing cognitive load [7] and thus, we can measure the percentage changes in pupil size from baseline (pre-stimulus state). We can divide these changes into cognitive load levels (i.e., low, medium, high). In our situation, the pupillogram of TEPR in digit span tasks has a smooth bell curve. Moreover, we also known from our data that the largest pupil size
Figure 7.8. A comparison of correlation coefficients between high contrast (top) and low contrast (bottom) eye.
Figure 7.9. A comparison of mean absolute error of percent change from baseline between high contrast (top) and low contrast (bottom) eye.
usually appeared on 8-digit trials. We can assume that the maximum pupillary diameter in 8 digits trials might represent high cognitive load. Hence, we can estimate a cognitive load range by calculating a 95-5 interquartile range of percentage change from baseline (to avoid small pupil fluctuations known as pupillary unrest) from the ground truth of all participants. This normalization ensures that data points are very similar across participants and we have termed it “critical section”. If the residual errors are beyond this range, we consider that the model has failed. In addition, if the errors are in between half of the critical section, the model might be sensitive enough to classify two levels of cognitive load (having some or no cognitive load). The same is true for three cognitive load levels (no, low, and high cognitive load). Note that we are more interested in comparing models and biometric data are noisy in general, so the exact numbers are not as important as a scale consistency across models.

Normally, only peaks of pupillary diameter are important for classifying cognitive load; therefore, we only use a 10-second stimulus period. Figure 7.10 presents the modified residual plot. The x-axis represents the mean between each point of ground truth $y_{norm}$ and prediction point $\hat{y}$, $(\frac{y+\hat{y}}{2})$, and the y-axis represents the difference between two points $(y - \hat{y})$. Both $y$ and $\hat{y}$ are normalized as percentage change from baseline and both are cropped only during the 10 second stimulation sections. Good models should spread along the x-axis (parallel to the horizontal line), and the y-axis should be around zero. Figure 7.10 shows a modified residual plot comparison for nine models. Each horizontal line indicates the boundary of the critical sections. What we see is that most of the data points are in between zero and all models slope downwards. This indicates that the models have a negative bias. Residual2 and Residual3 have slightly less bias than the others. Most networks are in the level-2 critical section which indicates that these models are at least sensitive enough for classifying one level of cognitive load (having some or no cognitive load), except PupilNet Wide32, where the errors are beyond the minimum critical error. In general, we do not want high variance in regression models; however, in this work, we need to see some variance that spreads to the right side of the plots. The spread to the right indicates that small errors are still present, although the mean between two points increases. PupilNet Residual2 and Residual3 are
more spread out to the side, while Wide64 and Wide128 have almost no change. This means that Residual2 and Residual3 are able to capture small changes caused by TEPR, while Wide64 and Wide128 have captured very little, if any, such changes.

Figure 7.10. Modified residual plot comparison for nine model algorithms. Each horizontal line indicates a different level of the boundary of the critical area. What we see is that most of the data points are closed to zero. PupilNet Residual2 and Residual3 are more spread out to the side, while Wide64 and Wide128 undergo almost no change. This means that PupilNet Residual2 and Residual3 are able to capture subtle changes caused by the TEPR, while Wide64 and Wide128 are unable to capture such changes.

The results in the previous section imply that the deeper and residual versions of PupilNets are close to each other in terms of prediction accuracy for within-subjects. These models might be able to classify two cognitive load levels. In contrast, Wide64 is less likely to be suitable for cognitive load classification applications.

### 7.4. Within- and Across-Subject Comparison

We selected PupilNet Residual2 for training the across-subject model to simulate an out of the box situation (no calibration required). Selecting PupilNet Residual2 has several advantages. Residual2 exhibits great trade-off between performance and computational resources. It also has a large capacity (learnable 4,336,417 parameters). Although there is no guarantee that a model with more parameters performs better than a model with a smaller number of parameters, it is more likely that a larger number of parameters has more scope to
capture more variations in the data. Our previous results also show that PupilNet Residual2 slightly outperforms the other models in all settings in terms of both errors and variation.

To answer the question regarding how PupilNet perform across subjects, we employed leave-one-out cross-validation (LOSO): the same method introduced in chapter 6, where we took one participant for testing and trained with the rest. The result are shown in Figure 7.11. The Figure shows the correlation coefficient (top), and mean absolute error (bottom) separated into three lighting conditions. Compared to the within-subject model, the across subject model has a significantly lower correlation (−0.78 to 0.67), and higher errors (1.4%-15%).

![Figure 7.11. A comparison between within-subject and across subject on PupilNet Residual2.](image)

In terms of different eye types, Figure 7.12 shows the with-in and across- subject model comparison separated by high and low contrast eyes. The across-subject model undergoes almost no change in performance in all lighting conditions. The error is beyond 15% for both high (2.23%-24.56%) and low (1.89%-15.78%) contrast eyes. The results suggest that calibration may needed.
Figure 7.12. A comparison between within-subject and across-subject on PupilNet Residual2 separated by eye types.
7.5. Discussion

From this the results of the previous sections, we have begun to see patterns regarding PupilNets architectures. Wide networks perform less well than deep networks and baseline networks. A possible reason for this might be that overfitting is more profound in wide networks compared to deep networks. For deep networks, regularization might play other roles than just reducing overfitting [142]. We performed early stopping for wide networks, and this might also have stopped the network learning filters fully in the CNN layers. It remains unclear whether stronger regularization would help to increase the performance of wide networks. We would argue that, if regularization does indeed improve wide network performance, the results would not show significant improvement after signal smoothing. Another explanation regarding the poor wide networks performance might be the lack of a pyramid structure [37,106]. The pyramid structure suggests that when a dimension decreases (via MaxPooling layers), the number of channels should increase to capture variations of image representation. Wide64 and Wide128 have a rectangular structure where the number of features are the same (64 or 128) throughout the networks.

The performance improves when the number of layers increases, consistent with Bengio’s suggestion in [8]. Nonetheless, it is unclear at which depth the network begins to no longer improve. When it comes to improving performance, the improvement is small compared to the baseline network (from 0.04 to 0.03 mean absolute error), while the larger networks require over three times the computational resources. For online application, using smaller networks might be a better choice, while for a task that requires high precision, larger networks would be appropriate.

When it comes to low contrast and darker environments, the results seem consistent: lower correlation and higher errors in a darker environment. This means that PupilNet can predict the pre-stimulus stage (baseline stage) as well, but it is more difficult to detect the small changes caused by TEPR when the pupil visibility is low. On the other hand, obscured pupils can affect the predicting performance. PupilNet might be able to successfully detect and learn pupil area, but not other regions such as the eyelashes or skin region.
Across-subject models have significantly lower performance compared to within-subject models. This result is expected considering the small variation of our dataset. The machine learning algorithm assumes that the training and testing set having an identical distribution, known as “data-generating distribution”. Within-subject models were trained with an almost identical distribution on both the training and testing set. In contrast, across-subject models were not. When looking at the results, the performance of the across-model with low contrast eyes is better than that with high contrast eyes. This might be the reason that the distribution of the low contrast eye dataset is more similar than the high contrast eye dataset. It is possible that, if the number of participants increases, the performance of the across-subject model might be improved because the distribution between training and testing set would be closer.

To understand further PupilNet’s behavior and seek opportunities to improve the across-subject model performance, we need to understand the networks’ behaviors. In the next chapter, we utilize CNN visualization techniques to unbox the inside of the network.
Chapter 8

UNBOXING PUPILNET: UNDERSTAND PUPILNET WITH VISUALIZATION

One criticism made against deep learning is that it is a black box. In other words, we know it works, but we do not know why and how. In some areas, knowing reasons behind the network’s decision is essential. For example, if a self-driving car has an accident, we need to examine the network to figure out why this happened. Financial institutions need to know the network’s reasons before making a major investment. Doctors need to know the reasons before undertaking severe operations. Thus, a new area of study emerged aiming to understand deep neural networks. In recent years, neural network visualization techniques have drawn the attention of the community. We use visualization methods primarily to gain insight, and this insight might lead to an improvement to the next version of PupilNet. Moreover, this knowledge could be expanded to adjacent problems (i.e., CNN regression).

To understand PupilNet, we explore the network on two levels: an individual neuron level and the entire network level. (i) An individual neuron examination addresses the issue of what properties of an eye each neuron learns to detect. (ii) The entire network examination answers the question regarding what part of the input image PupilNet most interested in.

8.1. Understanding Individual Neuron

In this section we explore individual neurons in the network; what they look like, what they learn, and how they react to stimuli. Recent studies reveal that CNN neurons learn to specialize to extract parts of an image. For example, one neuron may learn to detect human eyes and another detect a mouth. The aggregation of specialized neurons contributes to the probability that an input image is in the human face class. In addition, neurons in a higher layer learn to detect highly abstract representations. For instance, in the first convolutional layers, neurons learn to detect edges, while neurons in higher layers learn to detect texts,
parts of faces, or chairs. CNN neurons can even learn to detect a correspondence between multiple objects, such as a human head and a shoulder, or tires and a car body [141].

Nevertheless, it is unclear whether the neurons’ behaviors will be the same in this work because this work introduces several differences. Firstly, pupillogram extraction is CNN regression. In general, regression takes a raw activation strength at the last layer to predict the final results, while classification produces class probabilities. Thus, the exact activation strength has more impact on the regression than classification. The second difference is that this work has low data variation. Our image inputs are cropped tightly around the iris area. Although, this enables us to train a model with a much smaller dataset, it is unclear how the network learns to segment important parts of eyes. It is interesting to see what parts of an eye in which the network is interested. To determine this, we used three visualization techniques that are used for visualizing neurons in CNN: feature maps, filter map visualization, and the gradient ascent method.

8.1.1. Filter map and Filter map Visualization

We start with feature map and filter map visualization which are the most basic visualization techniques. These naive techniques are less meaningful, but at least can reveal some information about the quality of the network. Interesting neurons characteristics in the first convolutional layer were reported. The left side of Figure 8.1 shows the filter map of the first convolutional layer of AlexNet [62]. The network learns to detect certain features such as edges and colors resembling Gabor wavelets without predetermination by engineers. Unfortunately, beyond the first layer, filter maps become difficult to interpret, as shown on the right side of the Figure 8.4. The reason for this is that the first convolutional layer is connected directly to an input image, so many RGB image properties are retained. On the other hand, filters in higher layers have gone through many manipulations from earlier layers where original image properties are lost.

Figure 8.2 shows the first convolutional layer in PupilNet. Each 3 by 3 pixel is one 3 by 3 filter, and there are 32 filters total. We noticed that there are dominant colors in some
Figure 8.1. left: a filter map of a first Conv layer of AlexNet network. right: a filter map of a fifth Conv Layer of AlexNet network [62].

filters, highlighted in red. These might be filters that detect certain colors in the eye images. Unfortunately, they are too small to gain any meaningful insights from. The same situation happens with neurons in higher convolutional layers as well (i.e., Figure 8.3). Each row represent 32 filters of one neuron. Each 3 by 3 blob inside each row is an individual 3 by 3 filter in a second convolutional layer. These are almost uninterpretable.

Another technique is visualizing feature maps. Feature maps are the results of a convolutional operation on an input with learned filters. Figure 8.4 shows the feature maps of each convolutional filter of AlexNet with a cat image. The left image shows that neurons are interested in edges of the image. Visualizing feature maps and filter maps helps us to understand if a network is fully learned. Good feature maps should be clean, sparse, and localized, indicating that neurons have learned to specialize in a specific property.

Feature maps of a well-trained network should be sparse. Figure 8.5 shows the first ten layers of the Residual2 feature maps. Each row represents a feature map of each layer. For example, the first row is a input layer which consists of three channels (RGB). The second row consists of features after being convolved with learned filters. PupilNet has 32 filters, so
Figure 8.2. The 32 $3\times3$ learned filters of the first convolutional layer of Residual2. There are dominant colors in some neurons, highlighted in red. These might be filters that detect certain colors in the eye images. Unfortunately, they are too small to be able to gain meaningful insights from.

Figure 8.3. 32×32 $3\times3$ learned filters of the second convolutional layer of Residual2.
the resulting features are 32 images. Feature maps in Pooling layers have a factor of 2 lower resolution compared to the prior layers. This result shows that each layer seems to work as designed. Moreover, the network’s features are localized and sparse. This implies that the PupilNet Residual2 model seems to be well trained.

Nonetheless, these naive visualization techniques are not strong enough to determine the quality of PupilNet because we can only inspect the first layer of the network, and have to make a lot of assumptions over unclear noise. We need a method to visualize the neurons at arbitrary levels.
8.1.2. Visualizing Arbitrary Neurons with Gradient Ascent

The alternative technique to naively visualize neurons (features) is to monitor the neuron’s behavior [141]. This technique is like implanting a sensor into an animal brain, and then monitoring the neuron’s activation while showing various images to the animal. In successfully trained networks, each neuron learns to response to a specific characteristic in an input image (i.e., detecting horizontal edges or human faces). If an input image contains such a feature, the neuron will be strongly excited. For instance, if the neuron learns to detect a human face, it will show a high activation value if the image has one or more human faces. Based on this idea, we can feed forward a variety of images to the network and then monitor what images excite the target neurons. For example, if the target neuron shows high activation toward eyes, car tires, and a dog’s nose, this could mean that the neuron learns to detect a black circle in the image. Unfortunately, this technique is not suitable for this work because most images in our dataset are very similar. Neurons will become excited at almost every input. The only difference might be small changes in excitation intensity which
is mostly indistinguishable to humans. Instead, we can work backwards. We can artificially generate an input image that neurons want to see. The generated image \( I^* \) is created by a gradient ascent method to maximize activation in a target neuron, as shown in equation 8.1.

\[
I^* = \arg\max_I f(I)
\]  

Where \( I \) is an input image and \( f(I) \) is the neuron activation intensity. A regularizer term \( R(I) \) \((\text{i.e., } L^2)\) can be added to the equation to make the output image more visible \((I^* = \arg\max_I f(I) + R(I))\). In this work, \( f(I) \) is a function of an average of input \( I \), \( \frac{\sum_{i=0}^n I_i}{n} \), and we do not use any regularizer. In summary, the algorithm starts with choosing a neuron in an intermediate layer as a target. Then we feed forward a null image. A null image is normally an RGB image with Gaussian noise. Once the image reaches the target neuron, we calculate a gradient that maximizes the activation of the neuron and propagates the gradient back through the network. When the gradients reach the input space, we do not update the weights as in a normal training process, but we add the gradient to the null input image instead. Then, we feed forward the same image again. At every iteration, the null image is slightly changed to have a characteristic that maximizes activation of the target neuron. As a result, the image evolves to a feature that the target neuron learned to see. In this work, we fed an eye image instead of a noisy null image. The reason for this is that our dataset is mostly eyes. By directly feeding in an eye image to the network, we not only see what characteristics the target neuron cares about, but we also gain a rough location for where the neuron learned to detect.

Figure 8.6 shows on the top the generated images that most excited the neurons. Neurons in the convolutional layer seem to be interested in horizontal and vertical edges as well as various colors. These neurons seem to find the iris and pupil edges. Only seven neurons (from 32 neurons) respond to the input eye image. In the fourth convolutional layers, neurons can detect more complex shapes, and each one is more interested in a part of the eye including vertical, horizontal, and 45-degree angle edges. Figure 8.7 show the neurons in the tenth
layers. They seem to consist of many small pixels with various colors. This might be the result of several Max Pooling layers. What is left might be the important pixels for pupillary diameter prediction, because many pixels are clustered around the iris edges and pupils.

Figure 8.6. Artificially generated images that maximize neuron activation in the first convolutional layer (left). These neurons seem to react to vertical and horizontal edges and various colors (right). Neurons in the fourth layer are able to react to more complex structures around the pupil and iris.

These results suggest that PupilNet learns to detect eye characteristics. Neurons in higher layers seem to have similar properties to the lower layers, but they are more complex and localized. This might also imply that the filter development in the early layers may affect the effectiveness of later layers.
Figure 8.7. Artificially generated images that maximize neuron activation in the tenth convolutional layer. Many pixels are clustered around the iris edges and pupils.
8.2. Visualizing PupilNet with Attribution Heat Map

Individually visualizing neurons tells us about small parts of a network, but they cannot tell us about the overall network’s behavior. Thus, to investigate this, we used Attribution heat map [141]. The technique can show what area the network takes into the final prediction. The heat map can be generated by sliding a small zero patch occluding some parts of the image. Then, the occluded images are fed to the network to obtain a classification probability. If the occluded area is essential to the network, it will give a low prediction accuracy compared to the images. As shown in Figure 8.8, blue areas indicate regions where, if this area of the image is occluded, then the network has a low probability of classifying correctly. Because this study is a regression problem, the heat map is calculated with an absolute difference (L1) between a gaze tracker and a predicted value.

![Figure 8.8](image)

Figure 8.8. Left: the network considers the dogs face as a key area to classify a Pomeranian. Right: the network looks mostly at the dog area, and some parts of the woman facial area. [141].

In this study, we create a gray patch occluding a part of the eye image. If the iris or pupils are important, the patch size should be large enough to cover the edges. A $5 \times 5$ patch
might be enough to cover an edge of a 32×32 eye image. Moreover, because it is a regression
problem, it is interesting to see if PupilNet would consider all pixel intensities to predict a
final answer. A 1×1 patch (Figure 8.9) would show us the location of a cloud of pixels in
which PupilNet is interested, as well as the network sensitivity. Both patches moved from
the top-left to bottom-right one pixel at a time, and 1024 images were generated in total.
Then, each image was fed to a fully trained network to obtain pupil size prediction. An
absolute difference was calculated and was stored in a 2D heat map.

Figure 8.9. (left) 5×5 patch over an eye image. (right) 1×1 patch over an eye image

Figure 8.10 shows the 2D heat map of a 5×5 patch image overlay on top of an input eye
image. Red pixels would indicate high errors if a patch occluded that area. The result shows
that high errors occur in the iris area. This implies that the network needs iris information
to predict the final result correctly. In other words, the absence of the iris could affect the
size of the pupil which in turn causes high errors. This can also imply that the network
might place some weight on the iris area. Color neurons might be responsible for this. When
we look at the individual pixel with a 1×1 patch, as shown in Figure 8.11, pixels that are
close to a pupil show a higher error than the ones far away, and those in the skin area make
almost no contribution to the final prediction.
Figure 8.10. Attribution heat map of $5 \times 5$ patch image overlay on top of an input eye image (best viewed in electronic version)

Figure 8.11. Attribution heat map of $1 \times 1$ patch images overlay on top of an input eye image (best viewed in electronic version)
To understand how the network develops over the training period. We plotted 2D heat maps for every 100 epochs. Moreover, we compared an original PupilNet and a Residual2. Figure 8.12 shows that both networks learn similar patterns. They start out by searching a rough area where an iris should be, and then fine tune to develop a ring-like structure around a pupil. Deeper networks can be more tightly adjusted to the shape of the pupil. Comparing a failed case from an original PupilNet and its deeper version, Residual2 (Figure 8.13), the original PupilNet failed to identify the iris area, and detects the skin area instead, while PupilNet Residual2 is able to detect the pupil successfully. This indicates that a deeper network performs better in more challenging situations.

![Figure 8.12. Sample heat maps of a participant with high contrast eyes. The top row shows heat maps from the original PupilNet at 10, 100, and 200 epochs. The bottom row shows heat maps from the Residual2 model.](image)

Nonetheless, in extreme situations (i.e., Dim and Dark light conditions, even Residual2 finds it challenging to capture the pupils. In Figure 8.14, the network seems to be unable to pinpoint a pupil in the image. Nonetheless, it can roughly locate the pupil and iris areas which supports the fact that PupilNet does better in identifying pupil size during a baseline period.
Figure 8.13. Sample heat maps of a participant with low contrast eyes. Original PupilNet at the top failed to detect the pupil area, but Residual2 is able to detect the iris area successfully.

Figure 8.14. Heat maps of dim and dark images. The network is able to search the iris and pupil area, but it is not able to extract the exact location.
In summary, PupilNets seem to be not only interested in iris edges, but it also considers colors. This is consistent with our early experiment with PupilNet where gray eye images performed worse than an RGB dataset. When we look closely at higher convolutional layers (i.e., a second layer), those filters increase in complexity. Filter map visualization reveals that the network can learn to detect edges, color, and simple shapes. However, the network seems to find it difficult to detect large edges with a $3 \times 3$ filter size. It is unclear if the ability to detect a larger edge could contribute to the final prediction. The attentional heat maps show that successful networks have different behaviors compared to those that fail. The success cases learn to detect the pupil and iris areas, while failed cases learn to detect the skin area. It is unclear what makes the network find detecting of the skin useful rather than the pupil or iris area. One possible reason might be, in dark and low contrast situations, the pupil and iris are less reliable compared to the skin which is larger and more stable, since the networks are designed to detect patterns in high dimensional space. In addition, this evidence suggests that almost all pixels around the iris and pupil areas are essential to predicting pupil size. The networks seem to ignore reflection; however, in our dataset, reflection is usually in a similar location because of the lighting in an experimental setting. It is unclear whether changes in reflection location would cause major interference in the final prediction.

8.3. Theory to Practice: Improving Across-Subject Model

We learned from the last section that neurons seem to find it difficult to detect an edge in images which are larger than three pixels. What would happen if we increase the filter size? Does it help the network to detect edges and become less dependent on the color of the eyes and thus, in turn, to perform better across-subjects? Thus, in this section, we increase the filter size at the first convolutional layer to $7 \times 7$ to see if this can improve across-subject performance. A $7 \times 7$ filter size is used in the original Residual Networks proposed by He et al. [39]. We selected PupilNet Residual2 as our test model. The new network is the same except that there is a $7 \times 7$ filter size in the first layer. We trained the network with leave-one-
out cross-validation and we trained for 600 iterations equality on both models. The result is shown in Figure 8.15. The correlation coefficient is slightly higher and there is less variation, and the mean absolute error is slightly lower. Nonetheless, there is no significant difference when we perform a two-tailed t-test (p > 0.05).

![Residual 3x3 and Residual 7x7 Models Comparison](image)

**Figure 8.15.** A comparison of Residual2 3×3 and 7×7 filter size

When looking at filter map visualization, we found that the filters form interesting shapes and colors. Some high activating filters (highlighted in the red boxes) are moderately smooth and clear, with the exception of a green filter showing visible noise. Half of the filters are noisy. This could mean these neurons are dead; as a result, they have never been trained.

When performing gradient ascent visualization (Figure 8.17), the network seems to be better at detecting large edges in the eye images; however, it seems to lose small edge details. Colors still play an important role in the network. In addition, Residual 7×7 has more neurons that have successfully learned, and many of them learn to detect iris and pupil edges as well as the pupil area.

Increasing the filter size of the first convolutional layer to 7×7 can slightly improve CNN regression. When looking at the filter maps, they are not entirely smooth. This might indicate that there may be more room for improvement in future versions of PupilNet. For example, a 7×7 filter size appears to capture large edges but lose small edges. It might be possible to use both sizes at the same time, thereby resembling Google Inception networks.
Figure 8.16. Filter map visualization of the first convolutional layer of Residual2 7×7. High activation filters are highlighted in red.
Figure 8.17. (top) Features that each Residual2 7×7 neuron wants to see. It has a higher number of neurons activated in the first and fourth convolutional layers. It is also better at responding to large edges than Residual2. (bottom) The activating neurons in Residual2 3×3.
These visualization techniques are found to be useful not only in understanding how networks perform, but they also guide us to further improvement of the network.

8.4. Conclusion

We have unboxed a deep CNN regression using various techniques. The results suggest that PupilNet does learn pupil and iris characteristics such as location and shapes without explicit feature engineering. Filter map visualization reveals that it can learn to detect edges, color, and simple shapes. Network visualization techniques, such as filter and feature map visualization, gradient ascent, and occluded 2D heat maps are useful for understanding and validating the networks. These visualization techniques help us to understand PupilNet’s behavior and its quality. This information provides valuable insight with which to improve the network performance in the next version of PupilNet or in similar CNN regression problems.
9.1. Discussion and Limitations

In this work, we explored more robust techniques extracting pupillogram from an off-the-shelf RGB camera. These techniques are the modified Starburst algorithm (MSB), 2-Levels Snakucules (2-LS), and Convolutional Neural Network (PupilNet). We demonstrated that pupillogram could be extracted from just an off-the-shelf camera without the need for an expensive gaze tracker. In addition, we systematically compared these three algorithms to determine which technique shows potential for automatic cognitive load measurement. Despite the fact that MSB performs better than the original Starburst found in the biometric literature, we found that PupilNet completely outperforms hand-crafted feature engineering (MSB and 2-LS) in more challenging situations (high/low contrast eye color and three lighting conditions). This suggests that a machine learning approach is a promising strategy to pursue for a cognitive load sensing system. Nonetheless, this study is just getting started on its cognitive processing recognition journey. There are numerous limitations that we need to address before implementing such a system in the real world. For example, across subject generalization is still a challenge, and requires personalized training.

We visualized CNN regression whilst trying to understand the network quality and behaviors, aiming for the opportunity to improve the network further. We found that CNN regression can learn to extract parts of an eye, including the iris and pupil edges and the skin area. Deeper networks perform better than wider ones in our study. For instance, the results show that a deep version of CNN (Residual PupilNet) performs the best in our experiment (21 layers). However, at some point, adding layers meets with diminishing returns. The performance improvement is then too low to justify the high computational cost.
As William Ockham said, “entities must not be multiplied beyond necessity”. Accuracy and computational cost trade-off should be considered before implementing this technique in real-world applications. It remains unclear whether PupilNet performance would be improved if the number of data points increases. A study showed that simply adding more data to a deep convolutional neural network can improve the network performance [113]. This study is by no mean exhaustive. We only selected small samples from the plethora of neural network architectures proposed in the deep learning literature. There are many more hyperparameter optimization techniques that can improve the network performance such as random hyperparameter search [10] or even algorithms for auto-generating and to optimize the networks architectures themselves [97,143].

The current state of our study suggests that pupillogram extraction from an off-the-shelf low-resolution camera currently might not be suitable for medical and psychological research; however, it might be sufficient for using in controlled situations and with entertainment applications such as video games or semi-automatic cognitive load systems. Nonetheless, it is clear that video frames contain some pupillogram information that might be sufficient for training end-to-end cognitive classification. Meaning that instead of converting the eye videos to a pupillogram and then using the pupillogram as a feature, we might be able to train cognitive load classification models from raw eye videos directly. Blink and eye fixation might help add an extra dimension to end-to-end models for detecting cognitive tasks. Raw video eye data might contain more information related to cognitive processing than extracting the pupillogram alone. For example, a study showed that eye fixation [122], and gaze direction [123] are also a good predictors for cognitive load and attention. Some studies have shown that the blinking frequency is correlated with dopamine levels [26], a reward and motivation hormone. Combining pupillometry for monitoring tonic locus coeruleus activation and blinking patterns for dopamine might open up a new possibility for sensing more types of cognitive tasks. Blinks and gazing patterns are much easier problems to detect because they are less prone to external stimuli such as environment illumination, and head pose orientation. By training an end-to-end model, we can bypass the need for a gaze tracker
completely. Moreover, oculomotor analysis can be used to incorporate other physiological data such as the heart rate (HR), electroencephalograph (EEG) data, or galvanic skin response data (GSR). This might give us better insight into specific cognitive tasks, including human emotions. This would open up a whole new means of cognitive processing sensing.

Although we are still a long way from teaching machines to recognize human cognitive states, we are closer than ever before. Looking through the eyes is on the path for machines to be able to see. We have shown that this is possible. Cognitive processing sensing would provide important data that machines in the future cannot live without. This system would open up a new and unimaginable system in the years to come.

9.2. Privacy and Suggestion

This work potentially has privacy implications for the people being monitored. Almost everything can be misused. Even a pen can be used as a weapon. A better question to ask is what are a strategy to insure that it is hard to be misused. In this section, we briefly discuss these implications and how they could be addressed to protect privacy.

Privacy concerns related to technology advancement have a storied history. This discussion, in fact, has been vibrant since the telephone was invented in the 1800’s. The rapid changes in science and technology results in personal information being shared at the speed of light. It is not surprising, then, that it raises a concern on personal privacy; especially, a powerful system like a cognitive recognition system that can monitor, in some way, a human’s state of mind. At the same time, many systems need some user’s information to provide high-quality services. For example, Netflix needs users’ information and watching history to more accurately recommend new movies to its audience. For any technology, this motivates us to ask: Where is the middle ground? Can we only rely on users to protect their privacy? The answer might not be simple. A recent survey conducted by Gasparovic et al. shows that users are concerned with their privacy issues, but most people do not keep their privacy at a high level of protection [31]. It implies that only relying on users to protect their data might not be enough. Relying on businesses and governments might not be suitable either.
because there is an implicit conflict of interest (these entities typically want to know as much as is possible about those they are monitoring). The alternative might be law enforcement and regulation. Warren and Brande’s famous article, the Right to Privacy suggests that “the common law should protect an individual’s right to privacy under a right formulated as the right to be let alone” [125]. This idea became known as Privacy 1.0. It was simple but sufficient for advanced technology at a time when the instant camera became pervasive. Since then, technology has been significantly advanced. Many privacy-preserving addendum have been proposed trying to address this issue. In 2008, Serwin proposed privacy 3.0 [86]. He argued that privacy 3.0 must follow The Principle of Proportionality. It is the idea that tries to balance privacy protection and the voluntary release of information for a beneficial service. The principle of proportionality is based upon accountability, recognizing that neither the government nor private citizens benefit from over broad privacy restrictions. He suggests that personal information should be divided into four tiers: highly sensitive information, sensitive information, slightly sensitive information, and non-sensitive information. Then each tier should be treated differently. Highly sensitive information might require a strong regulation. On the other hand, non-sensitive information such as names can be far less regulated. On privacy 3.0, Tier I states as follow:

Tier I - highly sensitive information would be subject to strong limitations on the collection and processing of information. Examples would include genetic information, sexual history or other related issues, religious affiliation, information regarding communicable diseases, various forms of health information, personal information regarding children under certain ages (particularly if it is gathered via the Internet), and highly proprietary or confidential business information. While most collection and processing would be done only with notice and consent, there are examples where even the government can collect information without consent. This includes genetic information in certain circumstances, as well as mandatory disclosures of certain communicable diseases and medical information in connection with electronic health records. If consent by the consumer is given, additional collection and processing can be conducted of Tier I data. This type
of data would typically be subject to high levels of data security and violations of laws governing Tier I data could give rise to both civil and criminal sanctions [86].

Based on this idea, cognitive recognition systems may be in the Tier I (highly sensitive information). I would argue that a cognitive processing recognition system should be managed at the same level as health information, because some health-related data could be used to diagnose conditions from analyzing the pupillogram (similar to biofeedback systems like heart rate monitoring [21, 78]). In addition, the pupillogram and gaze data can infer other sensitive information even though a system might not be designed to. For example, an advertiser might want to test user interest for his/her new marketing campaign; however, sensitive personal information (such as sexual orientation) can also be extracted from such data via monitoring what an individual is looking at (i.e., gaze location) and the corresponding degree of enlargement in pupil size. This is a well studied phenomenon where homosexual individual’s pupils are more likely to enlarge in size when looking at photos of same gendered individuals [98]. Thus, the appropriate way to address concern is to ask users’ consent, and give control to users to manage and delete their data effortlessly. For our specific application, gaze data could be disabled, so when the pupil reacts to visible content, it is harder to identify what exactly a user was paying attention to. Beyond that, any application that implements pupillometry monitoring must provide essential data protection such as data encryption and protect user’s identity linked with the data. I suggest some examples of how applications can implement privacy safeguards with a cognitive sensing system. Advertisers must ask for user consent for testing their campaign with such a system. Intelligent learning system must be used only locally for the individual student and results not made available to others. Teachers might only access students’ data through a report at the same level as other academic assessments. The medical application must only allow for authorized agents such as a user’s doctor. Researchers must pass the review from an institutional review board (IRB). Ultimately, it would come down to two words: choice and transparency. Users can have choices to give or protect their data and be able to monitor where data will be used. Business should inform users about ‘privacy policy’ and ‘terms and conditions’ before opt-in
to the service. This creates trust between customers and service provider, and protects both parties from potential damages.

Cognitive recognition systems can be potentially harmful to users and society, if businesses, governments, and engineers are not careful and respectful of what their monitoring can uncover. It does not mean that we should prevent the development of science and technology. Technology is inevitably moving forward and unlocking new capabilities that we might not think of. The more that technology can do, the greater the benefit it can bring to humanity. Instead, we must regularly be aware that such great power comes with great responsibility. The prevention of potential harm, such as privacy issues, must continuously be reviewed, revised, and discussed to suit the current technology.
BIBLIOGRAPHY


[34] GOODFELLOW, I. J., WARDE-FARLEY, D., MIRZA, M., COURVILLE, A., AND BENGIO, Y. Maxout Networks.


[41] **Hilal, A., Daya, B., and Beuseroy, P.** Hough Transform and Active Contour for Enhanced Iris Segmentation.


Own, S. Y., and Me, N. Privacy 3 . 0–The Principle of Proportionality. 0–53.


Patel, S. Infrastructure Mediated Sensing. 274.


[114] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to Sequence Learning with Neural Networks.


[143] Zoph, B., and Le, Q. V. Neural Architecture Search with Reinforcement Learning.