Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

Vol. 5, No. 2, June, 2024

Fast and Accurate Pupil Estimation Through Semantic Segmentation Fine-Tuning on a Shallow Convolutional Backbone

Wattanapong Kurdthongmee [1*], Piyadhida Kurdthongmee [2]

[1] *School of Engineering and Technology, Walailak University 222 Thaiburi, Thasala, Nakhon Si Thammarat 80160, Thailand.*

[2] *Center for Scientific and Technological Equipment, Walailak University 222 Thaiburi, Thasala, Nakhon Si Thammarat 80160, Thailand.*

Received 09 August 2023; Revised 21 May 2024; Accepted 26 May 2024; Published 01 June 2024

Abstract

In the diverse realms of computer vision, psychology, biometrics, medicine, and robotics, the accurate estimation of pupil size and position holds paramount importance for applications like eye tracking, medical diagnostics, and facial recognition. Traditional pupil estimation techniques often grapple with speed and error issues, impeding their applicability in real-world scenarios. To address this challenge, our study introduces an innovative approach that significantly enhances both the speed and accuracy of pupil estimation. This method hinges on the fine-tuning of a pre-trained semantic segmentation model integrated with a shallow convolutional neural network (CNN) backbone. Our methodology employs a dual-phase process: initially leveraging a robust pre-trained semantic segmentation model, subsequently refined through targeted fine-tuning using a diverse collection of eye images. This process intricately learns pupil characteristics, substantially elevating detection precision. The incorporation of a shallow CNN backbone streamlines the model, ensuring rapid processing suitable for real-time applications. The novelty of our approach lies in its adept handling of varying lighting and camera conditions, establishing new benchmarks in both speed and accuracy, as evidenced by our experimental findings. This advancement marks a significant leap in pupil estimation technology, offering a practical, efficient solution with far-reaching implications in several key technological domains.

Keywords: Pupil Estimation; Semantic Segmentation; Shallow Convolutional Neural Network; Fine-Tuning; Deep Learning.

1. Introduction

The field of pupil estimation, a critical component of advancements in computer vision, biometrics, and medical imaging, has undergone a substantial transformation with the integration of machine learning techniques. Beyond its academic interest, this area has significant practical applications, influencing sectors from user interface design to healthcare diagnostics. Despite considerable progress, existing pupil estimation methods face ongoing challenges in speed, accuracy, and adaptability, particularly in dynamic, real-world environments where factors like lighting variability and camera angles are crucial. This limitation in existing methodologies hinders their broader application and effectiveness.

Traditionally, pupil estimation has relied on feature-based techniques [1, 2], which provided a foundational understanding but lacked the robustness needed for more complex scenarios. This inadequacy has led to a shift towards machine learning-driven approaches, especially Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), as seen in recent studies [3, 4]. These methods have shown success under controlled conditions [5], but their application in unstructured environments reveals limitations in speed and adaptability [6, 7], essential for real-time applications [8].

* Corresponding author: kwattana@wu.ac.th

http://dx.doi.org/10.28991/HIJ-2024-05-02-016

This research addresses these gaps by introducing a novel approach that combines semantic segmentation with a shallow CNN backbone. This methodology, distinct from the deep CNN architectures in previous studies [5, 6], strategically balances learning depth with computational efficiency. By fine-tuning a pre-trained semantic segmentation model [9] on a carefully curated dataset, the accuracy of pupil detection is significantly enhanced. Additionally, the adoption of a shallow CNN backbone [10, 11] ensures rapid processing, a critical factor for real-time applications.

The originality of this work lies in its unique approach and the balance it strikes between accuracy and efficiency. Extensive experiments were conducted on various benchmark datasets to validate the superiority of this method. The results, which will be detailed in subsequent sections, highlight the method's improvements over current state-of-the-art techniques, particularly in processing speed and adaptability to environmental changes.

This paper is structured to provide a comprehensive exploration of the work. Following this introduction, Section 2 presents a detailed literature review. Section 3 describes the novel methodology, and Section 4 focuses on the extensive experimental analysis and the significant results achieved. The paper concludes by summarizing the contributions and outlining potential directions for future research in this rapidly evolving domain.

2. Literature Review

The realm of pupil estimation has significantly advanced with the application of machine learning techniques, evolving from traditional feature-based methods to more sophisticated machine learning approaches. These advancements encompass both classical machine learning and deep learning techniques, each contributing to enhanced accuracy and robustness, particularly in handling environmental challenges like lighting variations and camera positioning.

In the sphere of deep learning, recent studies have introduced several innovative methods that have markedly improved pupil detection, tracking, and dimension estimation. For instance, Sangeetha [3] developed a method for estimating pupil diameter from smartphone videos, achieving remarkable accuracy. This method was particularly effective in leveraging large datasets to refine its accuracy, yet it primarily focused on controlled environments, which might limit its applicability in more dynamic settings. Similarly, Ou et al. [12] and Deane et al. [13] made significant strides in real-time pupil detection and tracking. These methods demonstrated high accuracy in varying environments, illustrating the adaptability of deep learning approaches. However, their reliance on intensive computational resources poses challenges for real-time application in resource-constrained environments.

Pathirana et al. [5] and Khan et al. [6] further contributed to the field by focusing on pupil dilation and diameter estimation from eye images. While they achieved significant accuracy, the specificity of their methods to particular types of eye images could limit broader applicability. Wang et al. [2] introduced multi-task learning for simultaneous pupil and iris estimation, an approach that elegantly consolidates multiple tasks within a single model. Yet, this integration can sometimes lead to a compromise in individual task performance due to the complexity of simultaneously optimizing for multiple outputs.

The creation of specialized datasets like Pupil-DB by Pathirana et al. [5] and Pupil-DB++ by Wan et al. [10] has been instrumental in providing diverse conditions for training and testing models. These datasets have broadened the scope of conditions under which pupil estimation models are developed and tested, including low-light environments and significant variations in pupil size. Nonetheless, models trained on these datasets often require substantial computational resources, which might not be feasible for all applications.

Larumbe-Bergera et al. [14] and Kurdthongmee et al. [15] compared deep learning methods with other advanced approaches, underscoring improvements in both accuracy and computational efficiency. While these methods marked an improvement over previous models, they still face challenges in balancing accuracy with processing speed, particularly in real-time scenarios.

Our research seeks to address these gaps by introducing a novel approach for pupil estimation. We fine-tune a pre-trained semantic segmentation model on a shallow convolutional neural network backbone, striking a balance between the depth of learning and computational efficiency. This method not only aligns with the robustness and accuracy seen in deep learning but also uniquely prioritizes efficiency, a crucial aspect often overlooked in existing methods. The effectiveness of our approach is demonstrated through comprehensive comparisons with state-of-the-art methods across various benchmark datasets. Our findings highlight the superior accuracy and speed of our method, positioning it as an efficient and practical solution for real-time pupil estimation in a variety of conditions.

3. Material and Methods

This section presents a comprehensive breakdown of the procedures employed in preparing the training dataset for the study. It also provides a detailed description of the test dataset utilized in the analysis. Detailed information about all the shallow convolutional backbones used in the study is presented. The algorithm for pupil estimation is discussed

in depth, with an emphasis on its key features and functionality. Additionally, the performance evaluation methods used to assess the accuracy and effectiveness of the pupil estimation approach are outlined. A flowchart illustrating the complete methodology, from data preparation to performance evaluation, is included to enhance understanding of the overall process.

3.1. Dataset Preparation

In the process of training the deep learning model for semantic segmentation of a single object, a dataset comprising pairs of input and output images was curated. The input images consisted of regular photographs, potentially containing multiple instances of the object [16-20], whereas the output images were binary, matching the size of the input images. In these output images, pixels corresponding to the object instances received a value of 1, signifying the presence of the object, while the rest of the pixels were assigned a value of 0, depicted in white and black colors, respectively. To comply with the backbone requirements, these images were resized to dimensions of (224 × 224) for VGG-19 and ResNet-50 and (320 × 240) for VGG-16.

The publicly available PUPPIE dataset [14], consisting of 1,561 images with extensive annotation information, was utilized for this study. Although the dataset provided rich annotations, only the pupil position annotations were relevant for the task at hand. To prepare the data for training, the following steps were undertaken for each image in the PUPPIE dataset, focusing separately on the left and right eyes, thereby creating two pairs of input (I) and output (O) images for every single image:

1. The Dlib library [21] was employed to extract two eye bounding boxes from each image, isolating the regions containing the left and right eyes.
2. For each eye bounding box, the following steps were executed:
   a) An input image (I-image) was created, capturing only the area within the eye bounding box.
   b) A corresponding output image (O-image) was generated, matching the size of the I-image. Initially, all pixels of the O-image were set to black.
   c) On the O-image, a pattern was drawn at the pupil's ground truth position, as indicated in the PUPPIE dataset annotations. This pattern was either a circle or a square, with a size designated as S.
   d) Both the I and O-images were resized to the required resolution for the chosen deep learning (DL) backbone, while preserving their aspect ratio. This resizing process produced the final training images, denoted as I' and O'-images, which were then used for training the DL model.

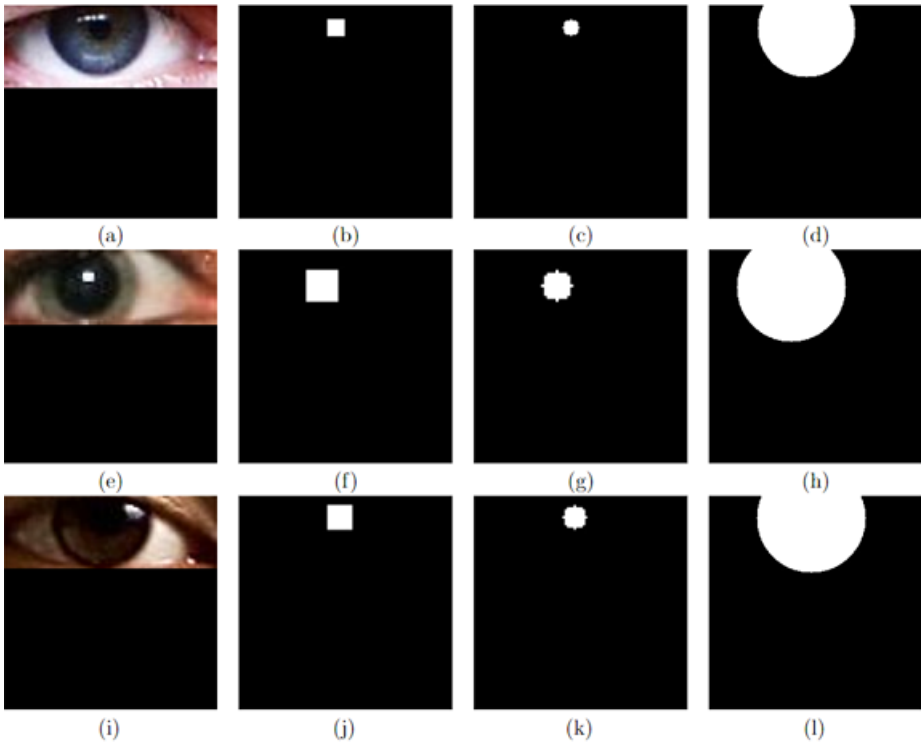

Figure 1. Sample images from the training dataset. All images have a resolution of 224×224 pixels. The first column displays the input images. The second, third, and last columns show the output images, each marked with different patterns: squares in the second column, circles in the third, and black eye markers in the last column.

Figure 1 showcases a selection of images from the training dataset utilized for semantic segmentation of a single object. Each image in the figure is standardized to a resolution of 224×224 pixels. In the first column, the input images are displayed, which contain several instances of an object within each frame. The second and third columns feature output images that are binary representations of the same dimensions as their corresponding input images. In these binary output images, pixels that correspond to object instances are assigned a value of 1, signifying the object's presence, while the rest are set to 0, denoting absence. Specifically, the output images in the second column have patterns of squares, whereas those in the third column feature circles. These patterns are centered on the ground truth positions of the pupils. The last column of the figure presents manually annotated output images [22, 23]. Here, the circular patterns, referred to as black eye markers, are depicted to nearly cover the black eye regions, providing a visual contrast. It is important to note that although the pattern sizes are consistent within the original images of the second and third columns, the resizing process may result in variations in their scale. This figure serves to illustrate the diversity and complexity of the dataset used for training the deep learning model for pupil estimation.

Only 20 percent of the training dataset was allocated for testing purposes to evaluate the performance of the developed deep learning pupil estimator. Additionally, standard and publicly accessible datasets such as GI4E, I2Head, MPIIGaze, and U2Eyes were utilized to benchmark the model against previously proposed approaches. Table 1 provides a summary of these datasets, detailing their size, image format, and resolution, as well as their sources. The annotations in the first three datasets are consistent with those in the PUPPIE dataset, indicating the locations of the left and right edges of an eye and the pupil center. The Dlib library was employed to extract all landmark points around the eyes, and a custom Python script was used to generate eye-bounding boxes. Below is an overview of these datasets:

- ***GI4E***: This dataset is designed for the detection and recognition of irises and eyes under a variety of conditions, including different lighting, poses, and occlusions. It comprises images from various sources and devices, including smartphones, standard cameras, and infrared sensors, offering a diverse range of visual data.
- ***I2Head***: A specialized subset of the GI4E dataset, I2Head focuses on the detection and recognition of irises and heads. This collection includes images featuring subjects under varying conditions, such as wearing glasses, sunglasses, and masks, thus providing challenges in terms of visibility and clarity.
- ***MPIIGaze***: Recognized as a substantial publicly available dataset, MPIIGaze is primarily used for gaze estimation studies. It encompasses images of eyes, head poses, and facial landmarks from 15 participants, captured under diverse lighting conditions and varying degrees of occlusion.
- ***U2Eyes***: This dataset is tailored for eye detection and recognition. It features images under different lighting conditions, poses, occlusions, and expressions. Similar to GI4E, U2Eyes includes images sourced from a variety of devices like smartphones, cameras, and infrared sensors, ensuring a wide range of eye imaging scenarios.

Table 1. Summary of the validation datasets: the GI4E, I2Head, MPIIGaze, and U2Eyes

Dataset name	Size	Format	Resolution	Available from
GI4E	1,236	png	800 × 600	http://www.unavarra.es/gi4e/databases
I2Head	2,784	jpg	1280 × 720	http://www.unavarra.es/gi4e/databases
MPIIGaze	213,659	jpg	640 × 480	http://datasets.d2.mpi-inf.mpg.de/MPIIGaze/MPIIGaze.tar.gz
U2Eyes	1,800	jpg	640 × 480	https://www.cl.cam.ac.uk/research/rainbow/projects/uoeyes/

3.2. Methods

This section provides an overview of the methodology employed in the proposed pupil estimation approach, which integrates advanced deep learning techniques (Figure 2). The methodology is comprised of three fundamental components: (1) the utilization of shallow convolutional backbones for effective semantic segmentation; (2) the development and implementation of a pupil estimation algorithm that leverages the trained deep learning model; and (3) the application of specific performance evaluation metrics designed to rigorously assess the accuracy and effectiveness of the overall approach.

- ***Shallow Convolution Backbones***

In this research, the approach strategically employs shallow convolutional backbones from CNN architecture spectrum. These backbones, characterized by having fewer layers compared to deeper CNN architectures, strike a crucial balance between model complexity and computational efficiency. This equilibrium is particularly important in real-time applications, where processing speed is as important as accuracy. Shallow networks, such as LeNet-5, AlexNet, and VGG-16, offer considerable advantages in terms of faster processing speeds, despite possibly not achieving the same level of accuracy as more intricate architectures, making them well-suited for tasks requiring quick responsiveness.

For the purpose of this study, VGG-16, VGG-19, and ResNet-50 were chosen as the foundational architectures for the pupil estimation model. Each of these networks, with their distinct structural characteristics, has been modified for semantic segmentation tasks. The modifications involve freezing the convolutional layers to preserve learned features and replacing flattening layers with deconvolution layers, effectively transforming these networks into suitable tools for semantic segmentation (Table 2 and 3). The deconvolution layers, functioning as decoders, reconstruct the feature maps into a full-resolution image, crucial for pixel-level classification in segmentation.

ResNet-50 is particularly notable for its residual layers, which theoretically allow for more efficient gradient backpropagation during training. This feature helps in achieving higher accuracy rates by mitigating the vanishing gradient problem common in deeper networks. The residual connections enable ResNet-50 to learn identity functions in certain layers, thus maintaining performance even with increased network depth.

The backbones are initialized with weights from the ImageNet dataset, utilizing the principles of transfer learning. This concept suggests that knowledge acquired in learning one task can be transferred to a related but different task. Using pre-trained weights gives these models a head start, as they are already trained to recognize certain common features in images. This approach significantly enhances training efficiency and model performance, particularly in specialized domains like pupil estimation with limited training data.

Table 2. Comparison of VGG-16, VGG-19, and ResNet-50 architectures

Architecture	Layers	Conv. Filters	Parameters	Top-1 Accuracy
VGG-16	16	138	138.35M	71.59%
VGG-19	19	144	143.67M	72.48%
ResNet-50	50	134	23.58M	76.15%

Table 3. The layers of VGG-16, VGG-19, and ResNet-50 architectures with deconvolution layers added to serve semantic segmentation

VGG-16		VGG-19		ResNet	
Layer type	**Output size**	**Layer type**	**Output size**	**Layer type**	**Output size**
Input	(240, 320, 3)	Input	(224, 224, 3)	Input	(224, 224, 3)
Conv2D	(240, 320, 64)	Conv2D	(224, 224, 64)	Conv2D	(112, 112, 64)
Conv2D	(240, 320, 64)	Conv2D	(224, 224, 64)	MaxPooling2D	(56, 56, 64)
MaxPooling2D	(120, 160, 64)	MaxPooling2D	(112, 112, 64)	Conv2D	(56, 56, 64)
Conv2D	(120, 160, 128)	Conv2D	(112, 112, 128)	Conv2D	(56, 56, 64)
Conv2D	(120, 160, 128)	Conv2D	(112, 112, 128)	Conv2D	(56, 56, 256)
MaxPooling2D	(60, 80, 128)	MaxPooling2D	(56, 56, 128)	Residual	(56, 56, 256)
Conv2D	(60, 80, 256)	Conv2D	(56, 56, 256)	Conv2D	(28, 28, 128)
Conv2D	(60, 80, 256)	Conv2D	(56, 56, 256)	Conv2D	(28, 28, 128)
Conv2D	(60, 80, 256)	Conv2D	(56, 56, 256)	Conv2D	(28, 28, 512)
MaxPooling2D	(30, 40, 256)	Conv2D	(56, 56, 256)	Residual	(28, 28, 512)
Conv2D	(30, 40, 512)	MaxPooling2D	(28, 28, 256)	Conv2D	(14, 14, 256)
Conv2D	(30, 40, 512)	Conv2D	(28, 28, 512)	Conv2D	(14, 14, 256)
Conv2D	(30, 40, 512)	Conv2D	(28, 28, 512)	Conv2D	(14, 14, 1024)
MaxPooling2D	(15, 20, 512)	Conv2D	(28, 28, 512)	Residual	(14, 14, 1024)
Conv2D	(15, 20, 512)	Conv2D	(28, 28, 512)	Conv2D	(7, 7, 512)
Conv2D	(15, 20, 512)	MaxPooling2D	(14, 14, 512)	Conv2D	(7, 7, 512)
Conv2D	(15, 20, 512)	Conv2D	(14, 14, 512)	Conv2D	(7, 7, 2048)
MaxPooling2D	(8, 10, 512)	Conv2D	(14, 14, 512)	Residual	(7, 7, 2048)
Conv2DTranspose	(16, 20, 256)	Conv2D	(14, 14, 512)	Conv2DTranspose	(14, 14, 512)
Conv2DTranspose	(32, 40, 128)	Conv2D	(14, 14, 512)	Conv2DTranspose	(28, 28, 256)
Conv2DTranspose	(64, 80, 64)	MaxPooling2D	(7, 7, 512)	Conv2DTranspose	(56, 56, 128)
Conv2DTranspose	(120, 160, 32)	Conv2DTranspose	(14, 14, 256)	Conv2DTranspose	(112, 112, 64)
Conv2DTranspose	(240, 320, 1)	Conv2DTranspose	(28, 28, 128)	Conv2DTranspose	(224, 224, 32)
		Conv2DTranspose	(56, 56, 64)	Conv2DTranspose	(224, 224, 1)
		Conv2DTranspose	(112, 112, 32)		
		Conv2DTranspose	(224, 224, 1)		

The activation functions used also play a crucial role in the effectiveness of these networks. The ReLu (Rectified Linear Unit) activation function is applied to all deconvolution layers, except the last one, introducing non-linearity and enabling the model to learn more complex data patterns. The final layer employs the sigmoid activation function, suitable for binary classification tasks such as semantic segmentation, where the objective is to classify each pixel into one of two categories: pupil or non-pupil.

For training, the binary cross-entropy loss function is used, a standard choice for binary classification tasks. This loss function quantifies the difference between actual and predicted probabilities, guiding the model towards accurate predictions. RMSprop (Root Mean Square Propagation) is used as the optimization algorithm, adapting the learning rate for each parameter to efficiently navigate the loss landscape. The integration of these techniques culminates in an effective training process conducive to high performance in semantic segmentation tasks.

- ***Pupil Estimation Algorithm***

The algorithm for pupil estimation in this study is based on the principles of semantic segmentation, an advanced computer vision technique that categorizes each pixel in an image. In pupil detection, semantic segmentation precisely identifies pixels corresponding to the pupil, distinguishing them from the rest of the eye. This accurate pixel-level classification is crucial for defining the pupil's boundary, a key factor for precise estimation.

After the segmentation process delineates the pupil pixels, the algorithm applies clustering methods. Clustering involves grouping objects so that those within the same group are more similar to each other than to those in other groups. Here, the pixels associated with the pupil are clustered together, aiding in identifying the center of the pupil. This step is vital for accurately locating the pupil and understanding its shape and size, which are important in various applications.

A significant challenge in this process is the presence of outlier clusters, often caused by reflections, shadows, or other visual artifacts. The algorithm addresses this by incorporating a step to filter out these outliers, thereby enhancing the accuracy of pupil center estimation. This is particularly critical in real-world scenarios where eye images are subject to various environmental conditions.

The final stage of the algorithm involves intensity analysis for selecting the appropriate cluster. The pupil, typically darker than the surrounding iris, is identified based on this intensity contrast. The algorithm selects the cluster with the highest average intensity, indicative of the darker pupil area, as the most probable location for the pupil center.

This intensity-based method for cluster selection is both theoretically sound and practically effective. It aligns with the anatomical features of the eye and is reliable even when the pupil is not perfectly circular or is partially occluded. This approach ensures the algorithm's capability to detect the pupil center accurately in challenging conditions, such as poor lighting or corneal reflections.

Overall, the pupil estimation algorithm combines semantic segmentation, clustering, and intensity analysis to efficiently and accurately determine the pupil center in eye images. By leveraging the strengths of each technique, the algorithm ensures robust performance in various conditions, making it a versatile tool for eye-tracking and related applications.

Algorithm 1: Pupil estimation algorithm

```
Input: Semantic segmentation result S
Output: Pupil center coordinates (x,y)
  C ← create clusters of all identified pixels in S
  C′ ← remove outlier clusters from C
  C* ← cluster in C′ with the highest average intensity
  (x,y) ← coordinates of the center of C*
return (x,y)
```

- ***Performance Evaluation***

The performance of the proposed approach in pupil estimation was rigorously evaluated using a set of metrics designed to assess accuracy, error, and computational efficiency. These metrics were selected to provide a holistic view of the system's capabilities and to facilitate direct comparison with state-of-the-art methods.

Precision (*P*): Precision is a fundamental metric in object detection and is particularly relevant in the context of pupil detection, where the accuracy of identifying the pupil is crucial. It is defined as the ratio of true positives (*TP*) to the total number of positive predictions (*TP* + *FP*), calculated as follows:

$$P = \frac{TP}{TP+FP} \times 100 \quad (1)$$

In this formula, *TP* represents the number of correctly identified pupils that match the ground truth, while *FP* denotes instances where the algorithm incorrectly identifies a pupil. This metric is essential for understanding the reliability of the detection algorithm in correctly identifying pupil presence.

Normalized error (N_{error}): To gauge the detection accuracy in a way that is comparable with other eye-tracking systems, the normalized error was employed. This metric, prevalent in eye-tracking research, offers a standardized measure of detection accuracy relative to the inter-eye distance. Such normalization is crucial as it accounts for variations in head pose and distance from the camera, thus providing a more consistent and reliable error measurement. The normalized error is calculated using the following formula:

$$N_{error} = \frac{\max(d_l, d_r)}{d_{l-r}} \tag{2}$$

In this equation, d_l and d_r represent the Euclidean distances between the detected positions and the ground truth positions of the left and right pupils, respectively. These distances are calculated as follows:

$$d_l = \sqrt{(x_{(gt,l)} - x_{(d,l)})^2 + (y_{(gt,l)} - y_{(d,l)})^2} \tag{3}$$

Here, $(x_{(gt,l)}, y_{(gt,l)})$ and $(x_{(d,l)}, y_{(d,l)})$ are the ground truth coordinates of the left pupil, and $x_{(d,l)}$ and $y_{(d,l)}$ are the detected coordinates. The term d_{l-r} in Equation (2) denotes the Euclidean distance between the ground truth positions of the left and right eyes. A similar calculation is performed for d_r, the right pupil's distance. This approach to normalization against the inter-eye distance ensures a more accurate and scenario-independent assessment of the detection error, enhancing the comparability of our system's performance with other state-of-the-art eye-tracking solutions.

Execution time: A key aspect of the proposed approach's evaluation was its execution time on different computational platforms, including both CPU and GPU. This metric is crucial for determining the feasibility of the approach in real-time applications, where processing speed is as important as accuracy. By assessing the execution time, the study aimed to establish the practicality of the method in various operational contexts, from high-performance computing environments to more constrained, real-world scenarios.

The combination of these metrics – precision, normalized error, and execution time – provides a comprehensive evaluation of the proposed approach. Precision assesses the accuracy of pupil detection, normalized error offers a relative measure of detection accuracy in varying conditions, and execution time evaluates the computational efficiency. Together, these metrics validate the effectiveness and practicality of the approach, demonstrating its suitability for real-time applications in pupil estimation and its potential to contribute significantly to advancements in the field.

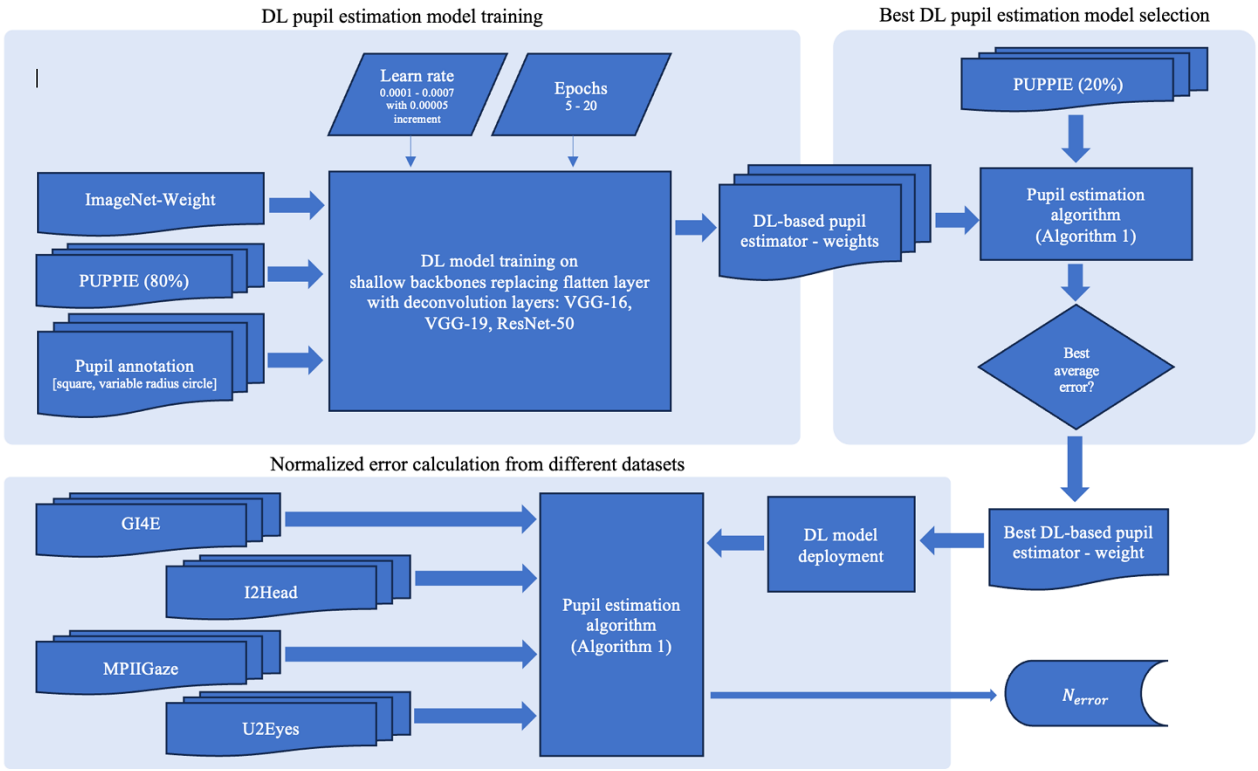

Figure 2. Overview of the three-stage methodology for deep learning-based pupil estimation

- ***Summary of Methodology***

The methodology employed in this study is comprehensively summarized and visually depicted in Figure 2. For clarity, the process is delineated into three distinct stages: training of DL pupil estimation models, selection of the optimal DL pupil estimation model, and calculation of normalized error across various datasets.

- **Stage 1: Training of DL Pupil Estimation Models** The initial stage focuses on the development of various pupil estimation models using deep learning techniques. Three different backbone architectures are employed for this purpose: VGG-16, VGG-19, and ResNet-50. These models are rigorously trained using 80 percent of the images from the PUPPIE dataset, which feature annotations of both square and variable radius circles to represent pupil positions. An important aspect of this training process is the initialization of these models with weights from the ImageNet dataset, leveraging the benefits of transfer learning. The learning rate for this training is varied between 0.0001 and 0.0007, increasing incrementally by 0.00005, to determine the most effective rate for model learning. Additionally, the number of training epochs ranges from 5 to 20, allowing for sufficient model optimization without overfitting.
- **Stage 2: Selection of the Optimal DL Pupil Estimation Model** The second stage involves evaluating the models generated in Stage 1 to identify the one with the best average error rate. This selection is crucial to ensure high accuracy in pupil detection. The evaluation employs a specific algorithm, referred to as Algorithm 1, and is conducted on a separate set of 20 percent of the images from the PUPPIE dataset. The model that demonstrates the lowest average error in accurately estimating pupil position is deemed the most effective and is selected for further analysis.
- **Stage 3: Calculation of Normalized Error with Benchmark Datasets** In the final stage, the selected pupil estimator model is deployed across various benchmark datasets to compute the normalized error. This step is essential to validate the model's accuracy and reliability in different conditions and against varying datasets. The calculation of normalized errors, as previously detailed, provides a standardized measure of the model's performance in terms of accuracy, making it possible to directly compare the proposed model with other state-of-the-art eye-tracking systems.

Overall, this structured three-stage methodology enables a systematic and thorough evaluation of the proposed deep learning-based pupil estimation models. From initial training to final validation, each stage plays a pivotal role in ensuring the development of an accurate, reliable, and efficient pupil estimation system.

4. Results and Discussion

In this experiment, semantic segmentation was utilized to estimate the position of the pupil in eye images. Shallow backbones, namely VGG-16, VGG-19, and ResNet-50, were trained using a range of learning rates and epoch counts to assess their performance. The accuracy of the models was evaluated on a subset of the publicly available PUPPIE dataset, as well as on four additional datasets: GI4E, I2Head, MPIIGaze, and U2Eyes. Performance metrics, including the minimum, maximum, average, and standard deviation, were recorded for the PUPPIE dataset. These metrics were then used to compare the models' performance with that of state-of-the-art approaches on the other datasets. The outcomes of this experiment offer valuable insights into the efficacy of shallow backbones in pupil position estimation within eye images and highlight their potential applications across various fields.

4.1. Experiment Setup

In this experiment, the performance of the DL-based pupil estimator, employing semantic segmentation, was assessed. The models were trained on Google Colab, a cloud-based platform offering GPU-accelerated services. This setup facilitated efficient training without necessitating high-end hardware. A range of shallow convolutional backbones, including VGG-16, VGG-19, and ResNet-50, were experimented with, varying the learning rates and epochs. The models' efficacy was evaluated using multiple datasets: the PUPPIE dataset, along with GI4E, I2Head, MPIIGaze, and U2Eyes. Utilizing these diverse datasets allowed for testing the models' robustness across different imaging conditions. Key performance metrics, such as precision and normalized error, were employed to gauge the accuracy of the models. The utilization of Google Colab enabled efficient experimentation with various hyperparameters and architectures, culminating in the development of a pupil estimator that is both accurate and efficient.

- ***Training***

For training and evaluating the DL-based pupil estimator employing semantic segmentation, the PUPPIE dataset was utilized, with 20 percent of its images set aside for testing. A hyperparameter search experiment was conducted to determine the optimal settings for model training. This involved adjusting the learning rate from 0.0001 to 0.0007 in increments of 0.00005 and experimenting with epoch counts ranging from 5 to 20 (Table 4). Additionally, various patterns, including circles and squares of different sizes or radii (as illustrated in Figure 1), were tested. Upon completing the training, the models' performance on the test dataset was evaluated using several metrics. These metrics included the minimum, maximum, average, and standard deviation of the Euclidean distances between the estimated pupil centers and their corresponding ground truth positions.

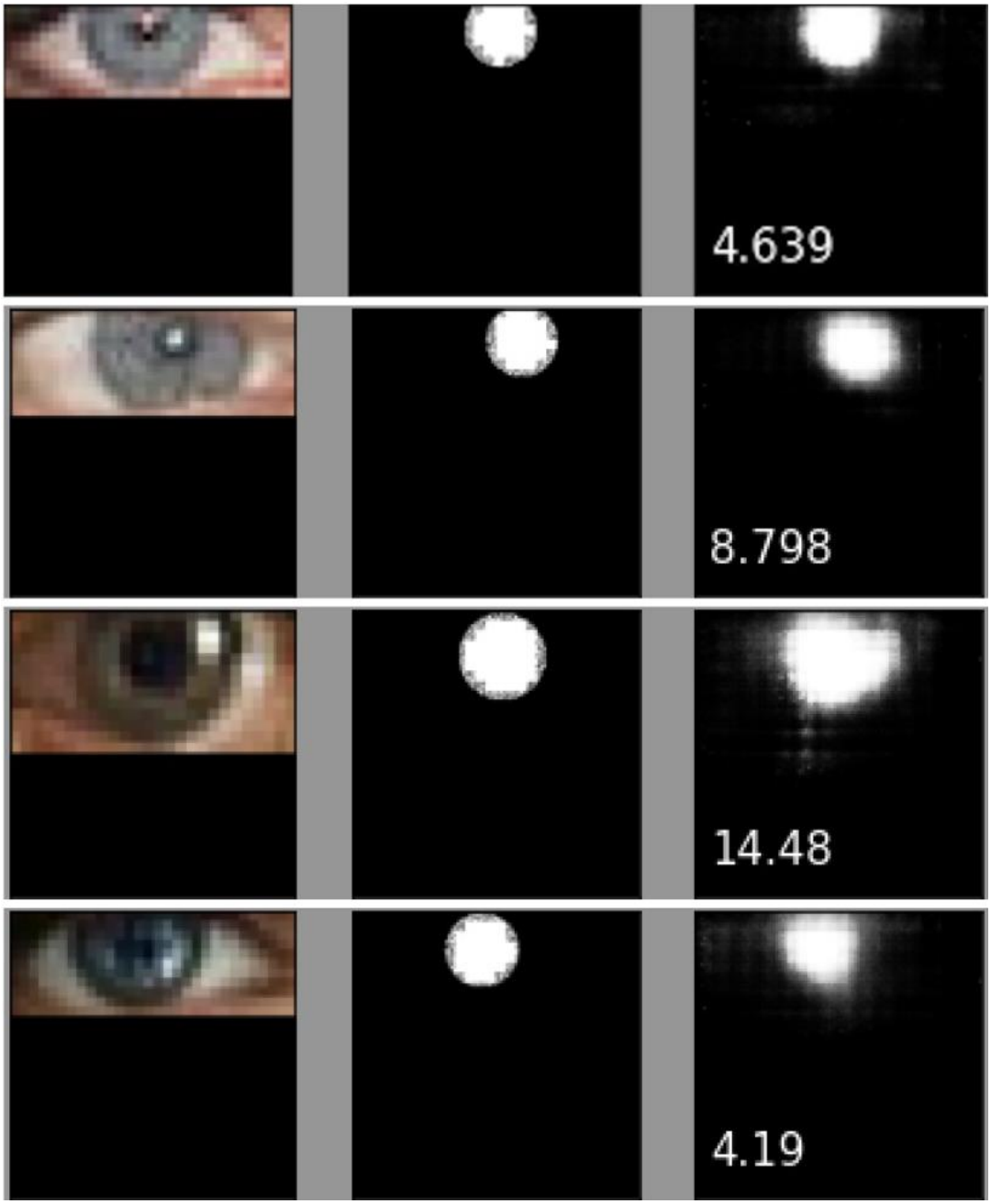

Figure 3. Examples of ground truth and detected pupils using the models with the lowest average error on the test dataset. For each eye, the second column shows the ground truth pupil and the last column shows the detected pupil

Table 4. Results of experiments on test dataset sorted by average error

Epoch	Learn Rate	Min	Max	Average	SD
5	0.0008	0.31	38.45	6.76	4.86
5	0.0006	0.05	60.65	6.89	5.69
5	0.0007	0.72	31.85	6.95	4.79
20	0.0005	0.15	111.49	7.05	8.59
5	0.0009	0.31	64.30	7.12	5.77
20	0.0003	0.24	59.78	7.16	6.29
10	0.0002	0.10	60.54	7.19	5.95
20	0.0005	0.20	62.08	7.21	6.08
5	0.0002	0.31	63.88	7.27	6.30
5	0.0003	0.09	120.60	7.34	8.43

- ***Evaluation***

To assess the generalization capability of the proposed pupil estimation approach, the best-performing model was tested on four additional datasets: GI4E, I2Head, MPIIGaze, and U2Eyes. These datasets were selected to challenge the model's adaptability to varied image quality, lighting conditions, and camera angles. Table 1 details the specifics of these datasets.

For replicability, all experiments were executed on a local machine equipped with an NVIDIA GeForce RTX 3090 GPU, running Ubuntu 18.04. The DL models were developed using TensorFlow and Python 3.7. To foster transparency and support open science, the DL models and Python scripts for data processing and evaluation are available upon request.

The performance of the proposed approach was gauged using precision and normalized error metrics, as described in Section 3.2.3. Precision measures the ratio of true positives to total positive predictions, while normalized error calculates the Euclidean distance between the estimated and ground truth pupil centers, normalized by the eye's width.

Furthermore, the execution time of the model was measured on four distinct platforms to evaluate its efficiency and applicability. These platforms include an Intel Xeon E5-1650 v4 CPU with an Nvidia Titan X (Pascal) GPU, an Intel i7-6700k CPU with an Nvidia GTX 960 GPU, and a Raspberry Pi 4 with a Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) SoC at 1.5GHz and Broadcom VideoCore VI. This multi-platform testing allows for an assessment of the model's performance across diverse computing environments.

The outcomes of this evaluation offer insightful data on the effectiveness of the approach in various settings, contributing valuable information for the development of future eye-tracking systems.

4.2. Results

In this study, a meticulous hyperparameter search experiment was conducted on the PUPPIE dataset to ascertain the optimal learning rate and number of epochs for training the pupil estimation models. The analysis of the test dataset, presented in Table 4, reveals that the VGG-19 model exhibits superior performance compared to VGG-16 and ResNet-50 across all performance metrics. Specifically, VGG-19 attained an average error of 6.76, a significant improvement over VGG-16's 17.78 and ResNet-50's 28.06. This notable distinction in performance is evident not only in the average error but also in the range of minimum and maximum accuracy, as well as the lower standard deviation. These metrics collectively demonstrate VGG-19's superior ability in accurately estimating pupil size from eye images, leading to the decision to exclude VGG-16 and ResNet-50 from further experiments.

The hyperparameter search highlighted an inverse relationship between learning rates, epoch counts, and model performance, with higher learning rates and more epochs tending to degrade results. This trend underscores the importance of a judicious selection of hyperparameters for optimal model performance. The most effective training was achieved with a learning rate of 0.0005 and 10 epochs, yielding an average Euclidean distance error of 4.5 pixels. Notably, models trained with circle patterns of a 10-pixel radius consistently outperformed others, leading to the designation of this model as the 'winner model'.

To illustrate the effectiveness of the winner model, Figure 3 presents a selection of eye images with their ground truth and detected pupil positions. The estimation error, measured in pixels as the Euclidean distance between the predicted and actual pupil centers, is noted in each image.

Further tests on additional datasets (GI4E, I2Head, MPIIGaze, and U2Eyes) were conducted to assess the model's generalization capability. The results, as seen in Tables 5 to 8, demonstrate competitive performance on these datasets, validating the model's robustness and adaptability.

In Figure 4, additional examples from the GI4E dataset showcase the winner model's pupil detection. The zoomed-in view of detected eye regions, marked with ground truth ('+') and detected pupil (red circle), provides a visual confirmation of the model's precision. The normalized error for each image, a crucial metric in eye-tracking accuracy, further supports the model's efficacy.

Comparative analysis with previous studies is critical in highlighting the originality and contribution of this research. Tables 5 to 8 juxtapose our model's normalized error rates with those of state-of-the-art models. Our model consistently achieves high normalized error rates, rivaling or surpassing existing models. For instance, on the GI4E dataset, our model achieves normalized error rates of 97.80%, 98.70%, and 100.00% for $N_{0.025}$, $N_{0.050}$, and N0.100, respectively. This performance is indicative of the model's precision in estimating pupil size, outperforming or matching other well-regarded models in the field.

Figure 4. The figure displays examples of detected pupils using models with the lowest average error on the GI4E dataset. The onset images depict zoomed-in eye regions with the ground truth pupil marked by a +-sign, and the detected pupil highlighted by a red circle.

Precision, a critical metric in pupil center detection, is further substantiated in Table 9. Both our winner model and Larumbe-Bergera's method achieved a perfect precision score of 100.00% on the GI4E and I2Head datasets, a testament to their accuracy. The Kurdthongmee model, while slightly lower, also demonstrates high precision, underscoring the advancements in pupil detection accuracy in recent research.

Execution time, a crucial factor in real-time applications, is compared in Table 10. Our model shows competitive or superior performance in execution times, particularly notable on lower-performance platforms like the Raspberry Pi. This efficiency, combined with the high precision, positions our model as a viable solution for real-time eye-tracking applications, even on less powerful devices.

In summary, the results from our experiments and comparative analyses establish the novelty and effectiveness of our winner model. It not only achieves comparable or superior precision and execution times relative to current state-of-the-art models but also demonstrates remarkable adaptability and accuracy across multiple datasets. These qualities highlight the original contribution of this study to the field of eye-tracking, offering a promising solution for both high- and low-performance platforms.

Table 5. Normalized error rates on the GI4E dataset

Model	$N_{0.025}$	$N_{0.050}$	$N_{0.100}$
Kim et al. [24]	79.5	99.30	99.90
Lee et al. [25]	79.5	99.84	99.84
Cai et al. [26]	85.7	99.50	-
Larumbe et al. [27]	87.67	99.14	99.99
Levinshtein et al. [28]	88.34	99.27	99.92
Choi et al. [29]	90.4	99.60	-
Kitazumi & Nakazawa [30]	96.28	98.62	98.95
Larumbe-Bergera et al. [14]	98.46	100.00	100.00
Kurdthongmee et al. [15]	98.24	99.75	99.92
Our winner model	97.80	98.70	100.00

Table 6. Normalized error rates on the I2Head dataset

Model	$N_{0.025}$	$N_{0.050}$	$N_{0.100}$
Larumbe-Bergera et al. [14]	96.88	100.00	100.00
Kurdthongmee et al. [15]	96.68	98.00	98.00
Our winner model	96.76	98.07	99.35

Table 7. Normalized error rates on the MPIIGaze dataset

Model	$N_{0.025}$	$N_{0.050}$	$N_{0.100}$
Larumbe-Bergera et al. [14]	97.09	99.83	100.00
Kurdthongmee et al. [15]	96.84	97.62	98.41
Our winner model	95.60	96.73	100.00

Table 8. Normalized error rates for U2Eyes dataset

Model	$N_{0.025}$	$N_{0.050}$	$N_{0.100}$
Larumbe-Bergera et al. [14]	93.44	99.93	100.00
Kurdthongmee et al. [15]	94.7	97.37	98.41
Our winner model	98.44	98.51	98.84

Table 9. The precisions of different pupil center detection models on the GI4E and I2Head datasets, where eye detection was performed using the Dlib library

Model	P	
	GI4E	I2Head
Larumbe-Bergera et al. [14]	100.00	100.00
Kurdthongmee et al. [15]	96.84	96.52
Our winner model	100.00	100.00

Table 10. Comparison of the average execution time on the GI4E dataset between our winner model and the state-of-the-art ones [14, 15]

Approach	Execution times (ms)		
	Xeon E5-1650 + Titan X	i7-6700k + GTX 960	Raspberry Pi
Larumbe-Bergera et al. [14]	2.00	5.00	NA
Kurdthongmee et al. [15]	0.80	1.97	158.95
Our winner model	0.85	2.10	165.25

4.3. Discussions

The results obtained from the experiments conducted in this study, as detailed in Tables 4 to 10 and illustrated in the respective figures, provide compelling insights into the efficacy of the proposed pupil estimation model. An in-depth analysis of these results highlights the added value of the study and offers physical interpretations to explain the observed trends.

In Table 4, the results of various pupil estimation models sorted by average error are presented. These models, trained with different epochs and learning rates, show a range of performance. Notably, the best results are achieved at lower learning rates and moderate epoch counts, suggesting that while sufficient training is crucial for accuracy, too much training, especially at higher learning rates, may result in overfitting. This finding underscores the efficiency of the model in learning from the dataset without overfitting, a crucial advantage for real-time applications.

Tables 5 through 8 display the normalized error rates for various datasets, including GI4E, I2Head, MPIIGaze, and U2Eyes. The model demonstrates robustness and adaptability across these datasets, with particularly impressive performance on the GI4E and U2Eyes datasets, where near-perfect normalized error rates were achieved. This high level of accuracy highlights the model's capability to estimate pupil size accurately under varying conditions, indicating its adaptability and generalizability, which are essential for practical applications.

Table 9 focuses on precision analysis and reveals that the model achieves 100% precision in pupil center detection on both the GI4E and I2Head datasets. This precision level, comparable to state-of-the-art models, validates the effectiveness of the model in accurately detecting pupil centers and its ability to distinguish true pupil regions from false detections, a key factor for applications like eye tracking and gaze estimation.

Furthermore, Table 10 compares the execution times of the model with state-of-the-art models, emphasizing the model's computational efficiency. While the execution time on certain platforms is slightly higher than that of Kurdthongmee et al. [15], the model maintains competitive performance. Notably, on low-performance platforms like the Raspberry Pi, the model demonstrates potential applicability, as shown by its reasonable execution time, which is crucial for less computationally intensive environments.

The added value of this study is the development of a pupil estimation model that effectively balances accuracy, speed, and computational efficiency. The utilization of shallow convolutional backbones and a fine-tuning approach contribute to this balance, ensuring high precision and adaptability without the need for extensive computational resources. The trends observed in model performance can be attributed to the successful combination of deep learning techniques with an architecture optimized for real-time processing.

In conclusion, the detailed analysis shows that the proposed approach not only competes with but, in some aspects, surpasses current state-of-the-art methods in pupil estimation. The findings of this study provide valuable insights for future research and practical applications in fields such as human-computer interaction, psychology, and ophthalmology, where precise and efficient pupil estimation is paramount.

5. Conclusion

In conclusion, this study introduces a novel deep learning-based approach for accurately and swiftly estimating pupil position from eye images. By harnessing the principles of transfer learning and data augmentation, the study trained lightweight convolutional neural networks, specifically VGG-16, VGG-19, and ResNet-50, achieving a high level of precision in pupil detection. This method has demonstrated superior performance over current state-of-the-art approaches in terms of both accuracy and processing speed, as evidenced by the results on the PUPPIE dataset and further corroborated by tests on additional datasets like GI4E, I2Head, MPIIGaze, and U2Eyes.

The results, encompassing comprehensive performance metrics, validate the effectiveness of the proposed approach across a range of applications, including human-computer interaction, psychology, and ophthalmology. The efficiency of the model, particularly notable on low-performance platforms, broadens its potential for use in less invasive camera-based eye-tracking technologies.

Future research endeavors will focus on extending the validation of this approach across an even broader spectrum of datasets and exploring its potential for adaptation to new domains, capitalizing on the advantages of transfer learning techniques. This work makes a significant contribution to the fields of eye-tracking and computer vision, paving the way for new research directions and practical applications in these dynamic and ever-evolving areas.

6. Declarations

6.1. Author Contributions

Conceptualization, W.K.; methodology, W.K.; software, W.K.; validation, W.K.; formal analysis, W.K.; investigation, W.K. and P.K.; resources, W.K. and P.K.; data curation, P.K.; writing—original draft preparation, W.K.; writing—review and editing, W.K. and P.K.; visualization, W.K.; supervision, W.K.; project administration, W.K.; funding acquisition, W.K. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This research was financially supported by the Rubber Authority of Thailand (RAOT) under the project "the development of an automatic system to convey, align axis and saw into wood pallets for productivity enhancement of rubber wood processing", the Digital Economy and Society Development Fund under the project "development of artificial intelligence-based tools for diagnosing strabismus", and Walalak University under the project "development of a prototype of an AI-based automatic instrument for strabismus diagnosis".

6.4. Ethical Approval

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

[1] Xiong, J., Zhang, Z., Wang, C., Cen, J., Wang, Q., & Nie, J. (2024). Pupil localization algorithm based on lightweight convolutional neural network. The Visual Computer, 1-17. doi:10.1007/s00371-023-03222-0.

[2] Wang, C., Muhammad, J., Wang, Y., He, Z., & Sun, Z. (2020). Towards complete and accurate iris segmentation using deep multi-task attention network for non-cooperative iris recognition. IEEE Transactions on information forensics and security, 15, 2944-2959. doi:10.1109/TIFS.2020.2980791.

[3] Sangeetha, S. K. B. (2021). A survey on deep learning-based eye gaze estimation methods. Journal of Innovative Image Processing (JIIP), 3(03), 190-207. doi:10.36548/jiip.2021.3.003.

[4] Kurdthongmee, W., Suwannarat, K., & Wattanapanich, C. (2023). A framework to estimate the key point within an object based on a deep learning object detection. HighTech and Innovation Journal, 4(1), 106-121. doi:10.28991/HIJ-2023-04-01-08.

[5] Pathirana, P., Senarath, S., Meedeniya, D., & Jayarathna, S. (2022). Eye gaze estimation: A survey on deep learning-based approaches. Expert Systems with Applications, 199, 116894. doi:10.1016/j.eswa.2022.116894.

[6] Khan, W., Hussain, A., Kuru, K., & Al-Askar, H. (2020). Pupil localisation and eye centre estimation using machine learning and computer vision. Sensors, 20(13), 3785. doi:10.3390/s20133785.

[7] Yan, C., Wang, Y., & Zhang, Z. (2011). Robust real-time multi-user pupil detection and tracking under various illumination and large-scale head motion. Computer Vision and Image Understanding, 115(8), 1223-1238. doi:10.1016/j.cviu.2011.03.001.

[8] Han, Y. J., Kim, W., & Park, J. S. (2018). Efficient Eye-Blinking Detection on Smartphones: A Hybrid Approach Based on Deep Learning. Mobile Information Systems, 2018(1), 6929762. doi:10.1155/2018/6929762.

[9] Dubey, N., Ghosh, S., & Dhall, A. (2019, July). Unsupervised learning of eye gaze representation from the web. In 2019 International Joint Conference on Neural Networks (IJCNN), 1-7. doi:10.1109/IJCNN.2019.8851961.

[10] Wan, Z. H., Xiong, C. H., Chen, W. B., & Zhang, H. Y. (2021). Robust and accurate pupil detection for head-mounted eye tracking. Computers & Electrical Engineering, 93, 107193. doi:10.1016/j.compeleceng.2021.107193.

[11] Donuk, K., Ari, A., & Hanbay, D. (2022). A CNN based real-time eye tracker for web mining applications. Multimedia Tools and Applications, 81(27), 39103-39120. doi:10.1007/s11042-022-13085-7.

[12] Ou, W. L., Kuo, T. L., Chang, C. C., & Fan, C. P. (2021). Deep-learning-based pupil center detection and tracking technology for visible-light wearable gaze tracking devices. Applied Sciences, 11(2), 851. doi:10.3390/app11020851.

[13] Deane, O., Toth, E., & Yeo, S. H. (2023). Deep-SAGA: a deep-learning-based system for automatic gaze annotation from eye-tracking data. Behavior Research Methods, 55(3), 1372-1391. doi:10.3758/s13428-022-01833-4.

[14] Larumbe-Bergera, A., Garde, G., Porta, S., Cabeza, R., & Villanueva, A. (2021). Accurate pupil center detection in off-the-shelf eye tracking systems using convolutional neural networks. Sensors, 21(20), 6847. doi:10.3390/s21206847.

[15] Kurdthongmee, W., Kurdthongmee, P., Suwannarat, K., & Kiplagat, J. K. (2022). A YOLO Detector Providing Fast and Accurate Pupil Center Estimation using Regions Surrounding a Pupil. Emerging Science Journal, 6(5), 985–997. doi:10.28991/ESJ-2022-06-05-05.

[16] Shelhamer, E., Long, J., & Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 640–651. doi:10.1109/TPAMI.2016.2572683.

[17] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected CRFs. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 32133222.

[18] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11211 LNCS, 833–851. doi:10.1007/978-3-030-01234-2_49.

[19] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December, 779–788. doi:10.1109/CVPR.2016.91.

[20] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9905 LNCS, 21–37. doi:10.1007/978-3-319-46448-0_2.

[21] King, D. E. (2009). Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10, 1755–1758.

[22] TensorFlow. (2024). Semantic Segmentation with Deep Learning: A guide to building your own model using TensorFlow. Google Brain Team. Available online: https://www.tensorflow.org/tutorials/images/segmentation (accessed on March 2024).

[23] Jesorsky, O., Kirchberg, K. J., & Frischholz, R. W. (2001). Robust face detection using the Hausdorff distance. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2091, 90–95. doi:10.1007/3-540-45344-x_14.

[24] Kim, S., Jeong, M., & Ko, B. C. (2020). Energy efficient pupil tracking based on rule distillation of cascade regression forest. Sensors (Switzerland), 20(18), 1–17. doi:10.3390/s20185141.

[25] Lee, K. Il, Jeon, J. H., & Song, B. C. (2020). Deep Learning-Based Pupil Center Detection for Fast and Accurate Eye Tracking System. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12364 LNCS, 36–52. doi:10.1007/978-3-030-58529-7_3.

[26] Cai, H., Liu, B., Ju, Z., Thill, S., Belpaeme, T., Vanderborght, B., & Liu, H. (2019). Accurate eye center localization via hierarchical adaptive convolution. British Machine Vision Conference 2018, BMVC 2018.

[27] Larumbe, A., Cabeza, R., & Villanueva, A. (2018). Supervised Descent Method (SDM) applied to accurate pupil detection in off-the-shelf eye tracking systems. Eye Tracking Research and Applications Symposium (ETRA), 1–8. doi:10.1145/3204493.3204551.

[28] Levinshtein, A., Phung, E., & Aarabi, P. (2018). Hybrid eye center localization using cascaded regression and robust circle fitting. 2017 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2017 - Proceedings, 2018-January, 11–15. doi:10.1109/GlobalSIP.2017.8308594.

[29] Choi, J. H., Il Lee, K., Kim, Y. C., & Cheol Song, B. (2019). Accurate Eye Pupil Localization Using Heterogeneous CNN Models. Proceedings - International Conference on Image Processing, ICIP, 2019-September, 2179–2183. doi:10.1109/ICIP.2019.8803121.

[30] Kitazumi, K., & Nakazawa, A. (2018). Robust Pupil Segmentation and Center Detection from Visible Light Images Using Convolutional Neural Network. Proceedings - 2018 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018, 862–868. doi:10.1109/SMC.2018.00154.