Lightweight Pupil Tracking with Semantic Segmentation and Confidence Estimation

Po-Chen Ko

Abstract

We study the task of predicting pupil segmentation masks and corresponding confidence values from short image sequences of an eye. We investigate how far a lightweight modeling strategy can go by combining a compact autoencoder for segmentation with classical image processing and temporal reasoning for confidence estimation. Our approach emphasizes efficiency, interpretability, and robustness, while maintaining strong performance on an internal evaluation split. We analyze the motivations behind our design decisions, discuss limitations, and outline several directions for future extension.

1 Introduction

Pupil tracking is an essential component in many gaze-estimation and human–computer interaction systems. The problem considered here involves two subtasks: (i) producing a pixel-level pupil segmentation mask for each frame in an eye-image sequence, and (ii) generating a binary confidence value indicating whether the pupil is visible.

Rather than adopting heavier segmentation architectures, we intentionally explore a lightweight modeling direction. This choice is motivated by practical deployment scenarios such as embedded or real-time systems, where computational efficiency, latency, and interpretability are critical. We pair a compact autoencoder with a rule-based temporal confidence estimator, enriched by classical image-processing techniques. Despite its simplicity, the resulting system performs reliably on our internal benchmark.

2 Pipeline Overview

Our pipeline consists of the following stages, we illustrate the pipiline in Figure 1:

1. Concatenate five consecutive grayscale frames into a 5-channel input.
2. Use an autoencoder to predict the pupil mask for the center frame.
3. Refine the predicted mask using morphological operations and a weighted median filter.
4. Compute per-frame predicted pupil areas.
5. Convert areas into binary confidence values via a tailored temporal algorithm.

We treat segmentation and confidence estimation as complementary but distinct tasks, allowing each component to be optimized independently while maintaining clear interpretability.

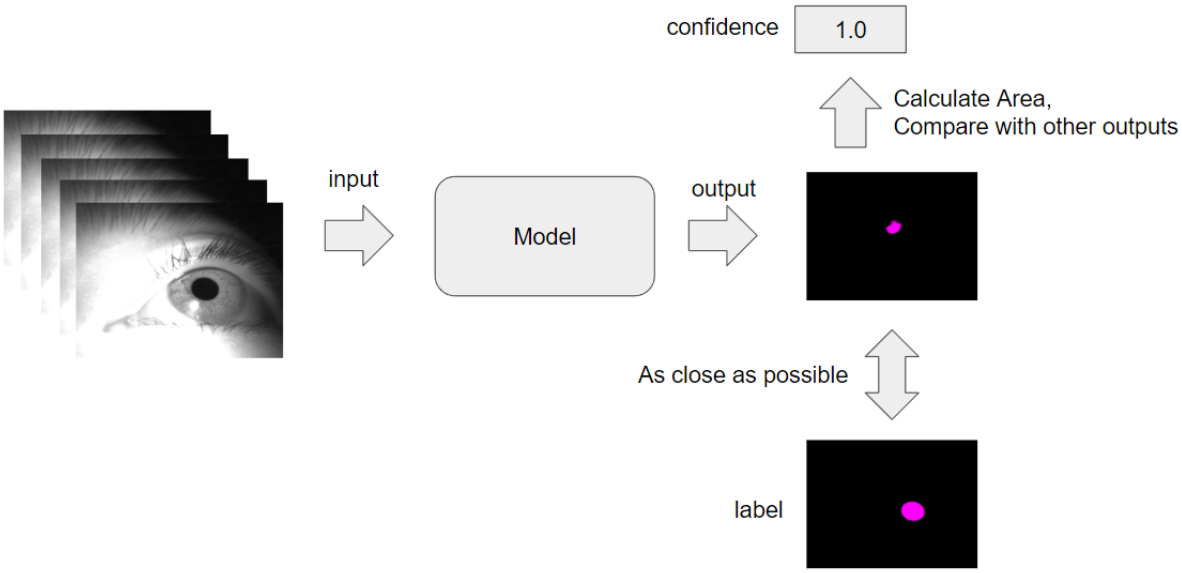

Figure 1: **Overall framework of our pupil-tracking system.** Consecutive frames are fed into the model to produce a segmentation mask; pupil area is then analyzed across neighboring outputs to produce a confidence value.

3 Pupil Segmentation Network

3.1 Autoencoder Architecture

We use a convolutional autoencoder to perform dense pupil segmentation. The encoder captures image features, and the decoder reconstructs the segmentation mask at the original resolution. Because the task requires pixel-level outputs aligned with the input image, the autoencoder architecture provides a natural and efficient structure.

We initially explored attaching a binary classifier to the encoder for joint training, but found that the classifier tended to overfit. The segmentation branch, however, was stable and accurate, so we retain the autoencoder as the core model.

3.2 Five-Frame Input Representation

Blinking introduces ambiguity when models rely on single frames. Partial occlusions can make the pupil appear artificially small, leading to unstable predictions. Since blinking is temporally continuous, we concatenate five consecutive frames to provide sufficient temporal context, as shown in Figure 2. The network predicts the mask for the middle frame, benefiting from frames where the pupil is more visible.

4 Confidence Estimation

Our final confidence estimator is a rule-based algorithm that operates on predicted pupil areas. This design emphasizes interpretability and robustness.

Let A_i denote the predicted pupil area for frame i. The algorithm consists of three components.

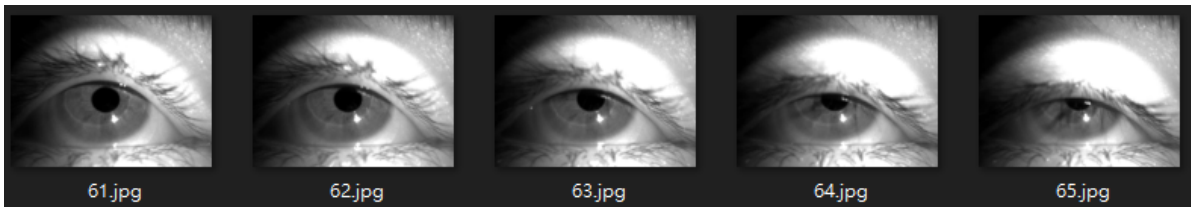

Figure 2: **Five-Frame Input Representation** We concatenate frames as input to improve robustness while blinking.

4.1 Stage 1: Sequence-Level Area Normalization

For each sequence, we compute:

1. the average predicted area $\bar{A}$,
2. and suppress frames with $A_i < 0.5\bar{A}$ by setting their area to zero.

This step filters out frames in which the pupil is barely visible or absent.

4.2 Stage 2: Local Window Filtering

To compensate for sequences whose average area is underestimated, we use a sliding window:

1. For each index i, compute the maximum pupil area in a 5-frame neighborhood.
2. Assign a preliminary label of 1 if A_i exceeds half of this local maximum, and 0 otherwise.

4.3 Stage 3: Temporal Median Smoothing

Pupil visibility evolves smoothly over time. We therefore apply a 5-frame median filter (with reflection padding) to remove isolated misclassifications and enforce temporal consistency.

5 Mask Post-Processing

5.1 Morphological Closing

Predicted masks may exhibit hollow regions in the pupil interior. Morphological closing (dilation followed by erosion) fills holes and restores shape continuity. Although it slightly softens edges, it significantly improves mask completeness.

5.2 Weighted Median Filtering

Weighted median filtering reduces local noise while preserving edges. Applying WMF to predicted masks sharpens boundaries and decreases segmentation error, as shown in Table 1. We visualize the effect of WMF postprocessing in Figure 3

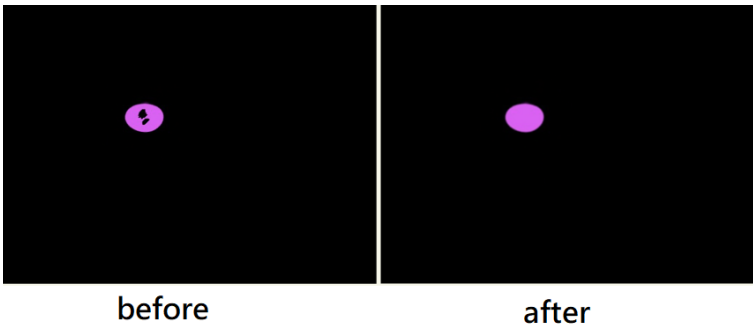

Figure 3: **Weighted Median Filtering.**

Data	Error before WMF (%)	Error after WMF (%)
1	10.6	6.8
2	3.1	1.9
3	3.6	2.8
4	3.7	2.4
5	3.6	2.3

Table 1: Area error before and after weighted median filtering.

6 Experiments

6.1 Training Details

We train on full-resolution images to preserve boundary integrity. Only frames with visible pupils are used for segmentation training. Data augmentation for eyelash-occluded samples (flips, small rotations) was explored, though its impact was modest.

6.2 Effect of Receptive Field

Early models often left the pupil interior unfilled. We increased the kernel size in a key convolutional layer to enlarge the receptive field, improving information flow from edges to interior pixels and reducing hollow predictions.

6.3 Evaluation on Internal Test Split

Table 2 reports accuracy on our internal evaluation split. The rule-based method achieves the strongest performance among all strategies we tested.

Method	Accuracy
Segmentation + learned confidence (two-stage)	0.767
Segmentation + learned confidence + smoothing	0.786
Above + augmentation	0.837
Ours	**0.886**

Table 2: Confidence prediction accuracy on the internal evaluation split.

7 Discussion and Limitations

Lightweight design. Our system intentionally prioritizes efficiency and interpretability. However, stronger segmentation architectures could further improve performance.

Dependence on segmentation quality. Confidence prediction heavily relies on the predicted pupil areas; errors in segmentation propagate downstream.

Rule-based constraints. While interpretable, rule-based decision boundaries may struggle when sequence statistics deviate from typical patterns.

Augmentation coverage. Only geometry-preserving augmentations were used; illumination and contrast augmentations remain underexplored.

Post-processing diversity. Beyond morphological closing and WMF, more advanced refinement techniques could be applied.

8 Future Work

Promising directions include:

- applying gamma correction and edge detectors for improved preprocessing,
- exploring perspective augmentations for viewpoint robustness,
- refining segmentation via contour-based ellipse fitting,
- trying segmentation architectures such as U-Net or DeepLab,
- revisiting neural confidence predictors with stronger regularization.

9 Conclusion

We presented a lightweight and interpretable pupil-tracking pipeline combining autoencoder-based segmentation with a rule-based confidence estimator. The approach is simple, efficient, and performs strongly on our internal evaluation. Our analysis highlights opportunities in preprocessing, model design, and post-processing that can further improve system robustness.

Acknowledgements

We thank the course staff for valuable discussions and feedback throughout this project.