Gaze Estimator Report — Version 2

Asifur Rahman | University of Arizona | VIP AI-Driven Healthcare Applications

Abstract Overview

This updated report presents a complete, refined pipeline for analyzing eye gaze behavior from a recorded mock perimetry test video. The goal is to estimate gaze direction (yaw/pitch), measure stimulus detection accuracy, and produce clinical-style metrics and visual field heatmaps, all using a fully offline approach.

In contrast to Version 1, this version corrects the assumed stimulus timing, adds detection metrics based on directional accuracy, and introduces multiple intuitive visualizations.

Corrections from Version 1

In the original version, I assumed that visual stimuli appeared every 2 seconds. After manually reviewing the video, I discovered:

- Stimuli actually appear every 3 seconds, approximately.
- Each stimulus position corresponds to a unique direction on the visual field grid.

This required:
- Manually mapping stimulus times and positions.
- Updating the internal logic of gaze matching and accuracy calculations.
- Changing peak matching and reaction time calculations from 2s intervals to 3s intervals.

Final Offline Approach (Google MediaPipe Face Mesh)

gaze_estimate.py: This script loads the video, extracts per-frame gaze data using *MediaPipe Face Mesh*, and computes normalized yaw/pitch based on iris and eye-corner landmarks.

Output: mediapipe_gaze.csv - time series of yaw and pitch per frame.

Gaze_analysis.py: Smooths yaw/pitch signals, detects peaks (gaze shifts), and matches them to the updated 3s stimulus schedule.

Outputs,

gaze_peaks.csv — matched peaks and stimulus timing
srt_results.csv — yaw/pitch reaction time per stimulus
Plots: yaw_plot.png, pitch_plot.png

gaze_metrics.py: This new script introduces:

Deviation calculation between the actual and ideal gaze for each direction.

Detection matching using error thresholds instead of peak proximity.

Heatmap visualizations for deviation, detection success, and yaw SRT.

All results are grouped by spatial direction on a 3×3 visual field..

Why MediaPipe Instead of Roboflow

The original script required an HTTP-hosted Roboflow model, which was unavailable. MediaPipe Face Mesh offered:

- Reliable, real-time facial landmark tracking
- Offline capability (no API or server is required).
- Transparent computation based on iris landmarks.

Results & Plots

Metrics are as follows,

Saccadic Reaction Time (**SRT**)**:**

Yaw: −0.075 ± 0.313 seconds

Pitch: −0.053 ± 0.240 seconds

Directionally Matched Detections:

10 / 12 stimuli (83.3%)

Based on actual vs. expected yaw/pitch per direction

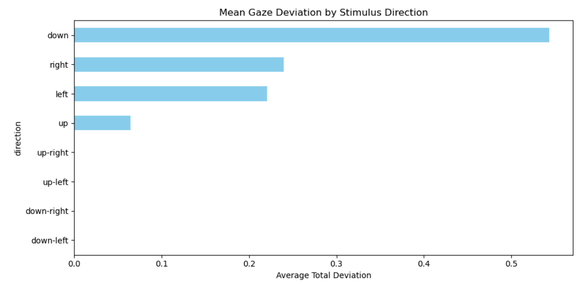
Missed SRTs:

Yaw: 1 stimulus

Pitch: 2 stimuli

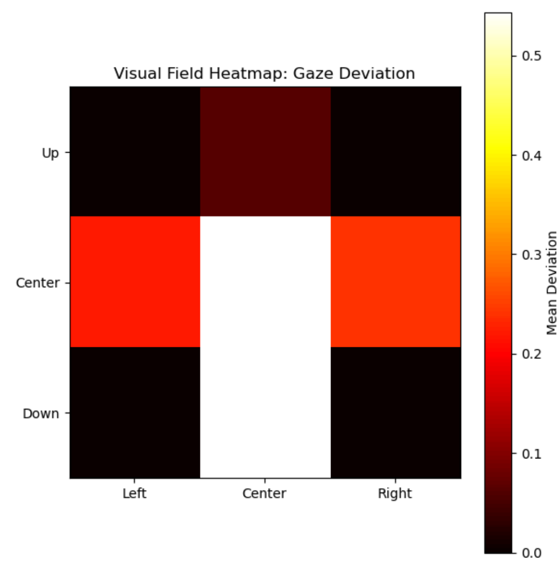
Visualizations

Figure 1 – Mean Deviation by Direction



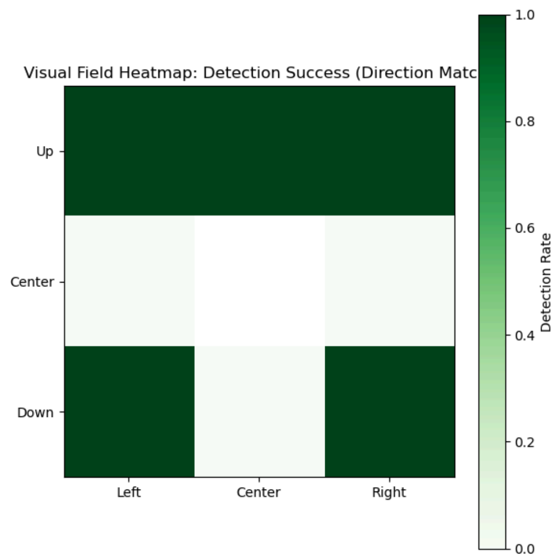
A bar plot showing total deviation in yaw/pitch for each labeled direction.

Figure 2 – Gaze Deviation Heatmap



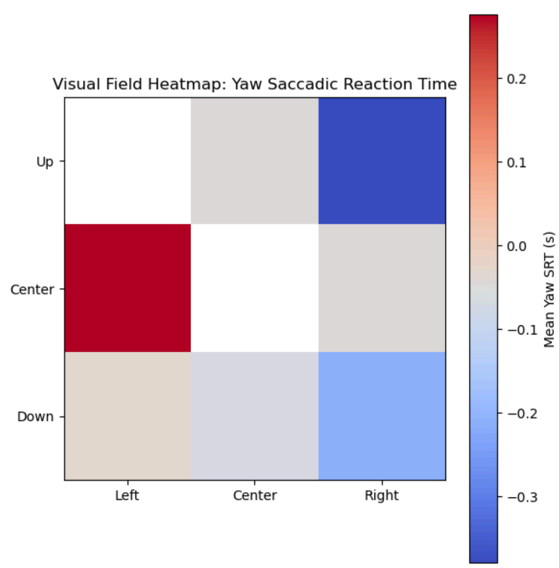
A 3×3 spatial heatmap showing average deviation per visual zone (brighter = less accurate).

Figure 3 – Detection Success Map



A Green-scale map showing which zones had directionally matched gaze within thresholds.

Figure 4 – SRT Heatmap



It displays average yaw reaction time per stimulus zone.

Conclusion

This version of the pipeline achieves a fully interpretable, spatially grounded eye-tracking analysis that matches the expectations of modern clinical eye tests. Key upgrades include directionally-aware detection metrics, stimulus mapping based on manual review, and the addition of visual field heatmaps for accuracy, timing, and success.

This refined foundation supports future integration with transformer models and mobile SDKs for broader deployment.