ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Ιχνηλάτηση προσανατολισμού με κάμερα

ΣΥΓΓΡΑΦΕΑΣ
ΠΑΝΤΕΡΑΚΗ ΜΥΡΤΩ ΔΗΜΗΤΡΙΟΣ
Α.Μ: ΤΛ20138

Επιβλέπων: Αντώνιος Κωνσταντάρας

Χανιά, Ιούλιος 2025



HELLENIC MEDITERRANEAN UNIVERSITY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRONICS ENGINEERING

Tracking orientation with camera

AUTHOR
PANTERAKI MYRTO DIMITRIOS
Academic I.D TL20138

Supervisor: Antonios Konstataras

Chania, July 2025

This page is intentionally blank

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Ιχνηλάτηση προσανατολισμού με κάμερα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παντεράκη Μυρτώ Δημήτριος

Επιβλέπων:

ΤΡΙΜΕΛΗΣ ΣΥΜΒΟΥΛΕΥΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Αντώνιος Κωνσταντάρας, Καθηγητής ΕΛΜΕΠΑ

Μέλος 2ο , Καθηγητής ΕΛΜΕΠΑ

Μέλος 3ο, Καθηγητής ΕΛΜΕΠΑ

ΕΓΚΡΙΘΗΚΕ ΑΠΟ ΤΗΝ ΤΡΙΜΕΛΗ ΕΠΙΤΡΟΠΗ ΤΗΝ ΧΧ.ΧΧ.ΧΧΧΧ:

.....	.....	.....
Μέλος 1ο	Μέλος 2ο	Μέλος 3ο
Καθηγητής ΕΛΜΕΠΑ	Καθηγητής ΕΛΜΕΠΑ	Καθηγητής ΕΛΜΕΠΑ

Χανιά, Ιούλιος 2025

……
Παντεράκη Μυρτώ Δημήτριος
Διπλωματούχος Ηλεκτρονικός Μηχανικός ΕΛΜΕΠΑ

Copyright ©Μυρτώ Παντεράκη Δημήτριος,2024
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Ελληνικού Μεσογειακού Πανεπιστημίου.

Στους γονείς μου…

This page is intentionally blank

Περιεχόμενα

ΠΕΡΙΛΗΨΗ....11
ABSTRACT....12
1. ΕΙΣΑΓΩΓΗ....13
2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΔΕΔΟΜΕΝΩΝ....15
2.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΙΧΝΗΛΑΤΗΣΗ ΠΡΟΣΩΠΟΥ....15
2.2 ΤΕΧΝΙΚΗ ΒΑΘΜΟΝΟΜΗΣΗ ΚΑΜΕΡΑΣ....21
2.3 ΚΑΤΗΓΟΡΙΕΣ ΑΝΙΧΝΕΥΣΗΣ....25
2.4 ΠΡΟΚΛΗΣΕΙΣ ΑΝΙΧΝΕΥΣΕΙΣ....28
2.5 ΕΦΑΡΜΟΓΕΣ....29
3. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ....31
3.1 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΑΤΙΣΜΟΥ PYTHON....31
3.2 Η ΒΙΒΛΙΟΘΗΚΗ OPENCV....33
3.3 Η ΒΙΒΛΙΟΘΗΚΗ DLIB....34
3.4 ΒΑΘΜΟΝΟΜΗΣΗ ΚΑΜΕΡΑΣ ΚΑΙ ΤΕΧΝΙΚΕΣ ΠΡΟΚΛΗΣΕΙΣ....35
4. ΜΕΘΟΔΟΛΟΓΙΑ....35
4.1 ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ....35
4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ....36
4.3 ΥΛΟΠΟΙΗΣΗ ΙΧΝΗΛΑΤΗΣΗΣ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ....38
4.4 ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΕΙΣ....39
5. ΑΠΟΤΕΛΕΣΜΑΤΑ....41
6. ΣΥΝΟΨΗ....43
7. ΠΑΡΑΡΤΗΜΑ....44
8. ΒΙΒΛΙΟΓΡΑΦΙΑ....47

Ευρετήριο Εικόνων

Εικόνα 1:Παράδειγμα Eigenface....17
Εικόνα 2 Παράδειγμα Fisherfac....18
Εικόνα 3: Παράδειγμα History of Oriented Gradients....19
Εικόνα 4: Δομή του FaceNet....20
Εικόνα 5: Βαθμονόμηση κάμερας....23
Εικόνα 6: Απεικόνιση της τοποθέτησης των ηλεκτροδίων....26
Εικόνα 7: Διάγραμμα ροής λειτουργίας του συστήματος....32
Εικόνα 8: Εστίαση βλέμματος προς τα αριστερά....41
Εικόνα 9: Ουδέτερο βλέμμα προς την κάμερα....42

Ευρετήριο Πινάκων

Πίνακας 1 Σύγκριση Κατηγοριών Μεθόδων Αναγνώρισης Προσώπων....21
Πίνακας 2 Σύγκριση Συστήματος με Υφιστάμενες Μελέτες....40

ΠΕΡΙΛΗΨΗ

Εισαγωγή: Οι πιο σημαντικές πτυχές της όρασης και της ανίχνευσης των ματιών είναι αυτές που μπορούν να αναπτυχθούν για τη δημιουργία εφαρμογών συστημάτων παρακολούθησης των ματιών, εικονικής πραγματικότητας και επαυξημένης πραγματικότητας, καθώς και διαδραστικών χώρων. Ταυτόχρονα, αυτή η μέθοδος έχει εφαρμοστεί εκτενώς στην ιατρική: η χρήση στον τομέα της διάγνωσης νευρολογικών παθήσεων μέσω της παρατήρησης των κινήσεων των ματιών και της κίνησης των ματιών κατά τη διάρκεια της παρατήρησης ενός ασθενούς με οπτικά προβλήματα και της θεραπείας παρέμβασης που περιλαμβάνει την ακριβή μέτρηση της κίνησης των ματιών. Το παρόν άρθρο έχει ως στόχο την κωδικοποίηση σε Python ενός συστήματος παρακολούθησης των κινήσεων των ματιών σε πραγματικό χρόνο με βάση το OpenCV και το Dlib. Μέσω της κάμερας της συσκευής, είναι σε θέση να εντοπίζει τα μάτια και να εμφανίζει τις συντεταγμένες των ματιών ως (XYZ) του τρισδιάστατου χώρου.

Σκοπός: η εργασία πραγματοποιήθηκε με σκοπό την ανάπτυξη μιας πραγματικής εφαρμογής παρακολούθησης των ματιών για την άμεση αναγνώριση των τρισδιάστατων συντεταγμένων των ματιών. Έχει διάφορους στόχους: παρατήρηση της θέσης των ματιών, μελέτη της ανθρώπινης δραστηριότητας, διαδραστικά έργα και υποστήριξη διαφόρων ιατρικών εφαρμογών.

Μεθοδολογία: Αναπτύχθηκε στη γλώσσα προγραμματισμού Python, με τη βοήθεια βιβλιοθηκών που ονομάζονται OpenCV και Dlib. Κεντρικό στοιχείο της μεθοδολογίας ήταν η διαδικασία βαθμονόμησης της κάμερας, η οποία, σύμφωνα με την επιστημονική μεθοδολογία, επιτρέπει τη διόρθωση των παραμορφώσεων σε έναν φακό και την ακρίβεια των υπολογισμών συντεταγμένων.

Αποτελέσματα: Προσεγγίζει το πλέγμα με αρκετά ακριβή τρόπο, παρέχοντας συντεταγμένες 3D σε πραγματικό χρόνο. Το εργαλείο έχει επίσης αποδειχθεί αποτελεσματικό υπό διαφορετικές συνθήκες φωτισμού, γωνίες θέασης και κίνηση του κεφαλιού, σύμφωνα με δοκιμές.

Συζήτηση – Συμπεράσματα: Η χρήση της τοποθέτησης πλέγματος 3D έχει διευρύνει ένα νέο πεδίο θέασης σε διαδραστικά συστήματα και ιατρικές διαγνώσεις. Οι δυνατότητες που μπορούν να υλοποιηθούν σε σχέση με τη μελλοντική επέκταση είναι η ενσωμάτωση της μηχανικής μάθησης για την ενίσχυση της ικανότητας ανίχνευσης του έργου, καθώς και η χρήση πολλαπλών καμερών για την πραγματοποίηση πιο ακριβών μετρήσεων.

ABSTRACT

Introduction: The most important aspects of seeing an eye and detecting it are the ones that may be developed to create eye tracking systems applications, virtual reality and augmented reality and interactive spaces. Simultaneously, this method has been extensively implemented in medicine: the use in the field of the diagnosis of neurological diseases by observing eye movements and eye movement during the period of observing a patient with visual problems and intervention therapy involving precise measurement of eye movement. This paper is aimed at coding in Python a real-time monitoring system of eye movements based on OpenCV and Dlib. Through the camera of the device, it is capable of locating the eyes and show coordinates of the eyes as (XYZ) of the three dimensional space.

Purpose: the work was carried out to come up with the actual eye tracking application to directly identify the 3D coordinates of the eyes. It has several objectives: observation of eye position, the study of human activity, interactive projects, and assistance to different medical applications.

Methodology: It is developed in Python programming language, with the help of libraries called OpenCV and Dlib. Central to the methodology was the process of camera calibration, which, according to the scientific methodology, allows the correction of distortions in a lens and the accuracy of coordinate calculations.

Results: It approximates the grid in a considerably accurate way, giving real time 3D coordinates. The tool is also proved to be efficient under diverse lighting conditions, angles of view and the head movement by tests.

Discussion – Conclusions: The use of the 3D mesh positioning has broadened a new field of view in interactive systems and medical diagnostics. The possibilities to be realized vis-a-vis future expansion are the possibilities to incorporate machine learning to enhance the detection capacity of the project, as well as the use of multiple cameras to take more accurate measurements.

1. ΕΙΣΑΓΩΓΗ

Η παρακολούθηση των ματιών είναι μια πολύ χρήσιμη τεχνολογία που έχει πολλαπλές χρήσεις, όπως η αλληλεπίδραση ανθρώπου-υπολογιστή, η κατασκευή εικονικών κόσμων, η διάγνωση νευρολογικών διαταραχών. Μία από τις πιο δημοφιλείς μεθόδους παρακολούθησης των ματιών είναι τα συστήματα παρακολούθησης των ματιών με βάση το βίντεο, καθώς είναι εύκολα στη χρήση, δεν είναι επεμβατικά και επιτρέπουν την καταγραφή της τροχιάς των ματιών σε πραγματικό χρόνο. Ο Jie-chun Chen και άλλοι ερευνητές (2023) υποστηρίζουν ότι αυτά τα συστήματα μπορούν να εφαρμοστούν σε μεγάλο βαθμό στην αναγνώριση και την παρακολούθηση της κόρης του ματιού υπό αντίξοες συνθήκες και λειτουργούν σε τομείς όπως η διάγνωση της ανθρώπινης υγείας και της ανθρώπινης συμπεριφοράς.

Στο πλαίσιο αυτής της εργασίας, αναπτύχθηκε ένα εργαλείο ανίχνευσης ματιών σε πραγματικό χρόνο χρησιμοποιώντας την γλώσσα προγραμματισμού python και τις βιβλιοθήκες OpenCV και Dlib. Συνδυάζοντας το φαινόμενο της φωτεινής κόρης και τους υπολογιστικούς αλγόριθμους αναγνώρισης χαρακτηριστικών (όπως περιγράφεται στη μελέτη των Zhu et al. (2004)), είναι δυνατό να εντοπιστεί η θέση των ματιών με υψηλό επίπεδο ακρίβειας, ακόμη και όταν οι συνθήκες φωτισμού είναι δύσκολες.

Το ζήτημα που είναι πολύ κρίσιμο για την επιτυχή λειτουργία αυτών των συστημάτων είναι η σωστή βαθμονόμηση της κάμερας, η οποία βοηθά στην εξάλειψη της παραμόρφωσης των φακών, καθώς και στην επαρκή εκτίμηση των τρισδιάστατων συντεταγμένων. Ο Chen (2023) τονίζει ότι η ανίχνευση και η παρακολούθηση των ματιών απαιτούν αυτή τη διαδικασία για τη βελτιστοποίηση των βημάτων.

Οι τρεις κύριοι τομείς που καλύπτονται στη μελέτη περιλαμβάνουν:

Προσαρμοστικότητα σε διάφορες συνθήκες φωτισμού και προφίλ προσώπου - Το εργαλείο έχει προσαρμοστικότητα στο προφίλ του προσώπου και σε διαφορετικές συνθήκες φωτισμού μέσω στρατηγικών όπως το φαινόμενο της φωτεινής κόρης και τα συνελικτικά νευρωνικά δίκτυα (CNN) για την παροχή ακριβών αποτελεσμάτων ακόμη και σε περιβάλλοντα με χαμηλό φωτισμό ή περιβάλλοντα με διακυμάνσεις φωτισμού.

Υψηλά επίπεδα βαθμονόμησης κάμερας: Χρησιμοποιείται για τη διόρθωση των παραμορφώσεων του φακού και τον σωστό υπολογισμό των τρισδιάστατων συντεταγμένων, ώστε να παρέχει στο σύστημα μεγαλύτερη αξιοπιστία.

Εκτεταμένες δοκιμές και επαλήθευση: Το εργαλείο δοκιμάστηκε σε διάφορα περιβάλλοντα φωτισμού, προσανατολισμούς προσώπου και εφαρμογές, δείχνοντας εξαιρετική αξιοπιστία και ακρίβεια. Κατά τη διάρκεια των δοκιμών, το σύστημα αποδείχθηκε ότι μπορεί να λειτουργεί αξιόπιστα ακόμη και σε δυσμενείς συνθήκες.

Το έργο θα οδηγήσει στην ανάπτυξη τεχνολογιών παρακολούθησης των ματιών που μπορούν να εφαρμοστούν σε ένα ευρύ φάσμα. Επιπλέον, ανοίγει το δρόμο για ακόμη περισσότερες εξελίξεις, συμπεριλαμβανομένης της ενσωμάτωσης αλγορίθμων μηχανικής μάθησης, οι οποίοι θα μπορούσαν αργότερα να επιτρέψουν ένα ακόμη υψηλότερο επίπεδο ακρίβειας, καθώς χρησιμοποιείται ένα σύστημα πολλαπλών καμερών..

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΔΕΔΟΜΕΝΩΝ

2.1 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΙΧΝΗΛΑΤΗΣΗ ΠΡΟΣΩΠΟΥ

Για να μπορεί το ανθρώπινο οπτικό σύστημα να επικεντρώνεται σε διαφορετικά σημεία στο χώρο, το μάτι πρέπει να περιστρέφεται από τους μυς που το περιβάλλουν, έτσι ώστε η ωχρά κηλίδα να βρίσκεται πάνω από το σημείο που μας ενδιαφέρει. Οι κινήσεις των ματιών χωρίζονται σε τέσσερις κατηγορίες (Jacob, 1991; Duchowski, 2003):

- Σακκαδική κίνηση (saccade)

Μια γρήγορη, βαλλιστική και πολύ γρήγορη κίνηση του ματιού που μετακινεί την ωχρά κηλίδα σε μια νέα θέση-στόχο στο οπτικό πεδίο σε περίπου 20-50 ms.

- Εστίαση (fixation)

Το μάτι φαίνεται να σταματά σε ένα σημείο, αλλά στην πραγματικότητα εκτελεί μικρότερες κινήσεις που σχετίζονται με τρόμο (μικροσακκάδες), ώστε να διασφαλίσει ότι το αντικείμενο παίρνει θέση στην ωχρά κηλίδα και μετακινείται άνετα στο οπτικό σύστημα.

- Ομαλή επιδίωξη (smooth pursuit)

Η κινητικότητα του στόχου που παρατηρείται από το μάτι επιτυγχάνεται διατηρώντας το αντικείμενο στο κέντρο του ματιού (ωχρά κηλίδα), καθώς η ταχύτητα του ματιού λειτουργεί με το ρυθμό της κίνησης.

- Νυσταγμός (nystagmus)

Συνδέεται με κινήσεις που συνδυάζονται και έχουν το μοτίβο του πριονιού (συνδυασμός ομαλής παρακολούθησης και κινήσεων σακκάδων) για να σταθεροποιήσουν την εικόνα στον αμφιβληστροειδή και τις κινήσεις του κεφαλιού ή του κινούμενου στόχου (Duchowski, 2003).

Μεταξύ των πιο αντιπροσωπευτικών και πραγματικά διαχρονικά ουσιαστικών υποτομέων της τεχνητής νοημοσύνης και της υπολογιστικής όρασης είναι η ιχνηλάτηση προσώπου. Το ανθρώπινο πρόσωπο ως βιομετρική λύση είναι ιδιαίτερο, καθώς περιλαμβάνει μια σειρά από πληροφορίες (δομικά χαρακτηριστικά, συναισθηματική έκφραση, ταυτότητα) και είναι το πρώτο πράγμα στο οποίο εστιάζουν οι άνθρωποι όταν έρχονται σε επαφή με άλλους.

Η υπολογιστική αναγνώριση προσώπου αναφέρεται στην ανάπτυξη ενός συστήματος στο οποίο ένας υπολογιστής είναι σε θέση να εντοπίζει, να ανιχνεύει και να αναγνωρίζει πρόσωπα σε ψηφιακά βίντεο ή εικόνες, είτε για σκοπούς ταυτοποίησης είτε για σκοπούς επαλήθευσης. Η ιδέα μιας τέτοιας δυνατότητας έχει προκαλέσει το ενδιαφέρον από την αυγή της τεχνητής όρασης και έχει αναπτυχθεί σταθερά μέσα από τα διάφορα στάδια της τεχνολογίας και της θεωρίας.

Η ιστορική πορεία της αναγνώρισης προσώπου μπορεί να χωριστεί σε τέσσερις βασικές εποχές:

1. Οι πρώτες προσεγγίσεις (1960–1990), βασισμένες στη χειροκίνητη εξαγωγή γεωμετρικών χαρακτηριστικών.
2. Οι ολιστικές μέθοδοι (1990–2000), όπου το πρόσωπο αναλύεται ως σύνολο (signal-based).
3. Οι feature-based μέθοδοι (2000–2012), με αυτόματη εντοπισμό τοπικών χαρακτηριστικών.
4. Οι deep learning τεχνικές (2012–σήμερα), με νευρωνικά δίκτυα υψηλής ακρίβειας και ανθεκτικότητας.

Οι πρώτες μελέτες, όπως εκείνες του Woody Bledsoe τη δεκαετία του ’60, βασίζονταν σε χειροκίνητες μετρήσεις γεωμετρικών χαρακτηριστικών του προσώπου. Σε αυτές τις προσεγγίσεις, το πρόσωπο περιγραφόταν μέσω αποστάσεων και αναλογιών μεταξύ συγκεκριμένων σημείων.

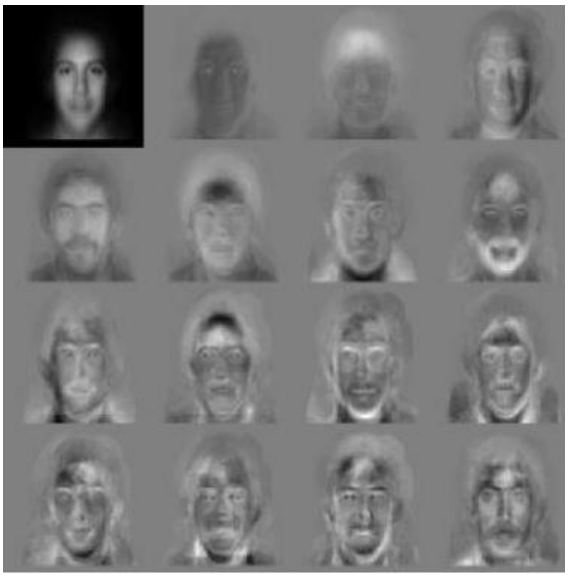
Οι μετρήσεις αυτές αποθηκεύονταν σε βάσεις δεδομένων για σύγκριση και οι τεχνικές αυτές ονομάζονται “Geometric based approaches”. Ένα από τα βασικά χαρακτηριστικά είναι ότι η αναγνώριση γινόταν με την σύγκριση των αριθμητικών διανυσμάτων και εικόνες ήταν μικρής ανάλυσης. Επιπλέον, υπήρχε πολύ μικρή ακρίβεια και ευαισθησία σε αλλαγές φωτισμού.

Κατά την δεκαετία του 1990 σημειώθηκε μια σημαντική αλλαγή: αντί να εξετάζουμε το πρόσωπο ως ένα σύνολο μετρήσεων, το αντιμετωπίζουμε ως σημεία - δηλαδή ως μια δισδιάστατη αναπαράσταση που μπορεί να αναλυθεί μαθηματικά. Έτσι προέκυψαν οι λεγόμενες ολιστικές μέθοδοι.

Η μέθοδος Eigenfaces, που βασίζεται στην Τεχνική Κύριων Συνιστωσών (Principal Component Analysis – PCA), είναι μια από τις πρώτες ολιστικές προσεγγίσεις που αξιοποιήθηκαν ευρέως στην αναγνώριση προσώπου. Η βασική ιδέα της PCA είναι να μετασχηματίσει τις αρχικές εικόνες προσώπων σε έναν νέο διανυσματικό χώρο χαμηλότερης διάστασης, που ονομάζεται "χώρος προσώπων" (face space). Κάθε πρόσωπο αναπαρίσταται ως ένας γραμμικός συνδυασμός ιδιοεικόνων (eigenfaces), οι οποίες προκύπτουν από τις κύριες συνιστώσες του συνόλου εκπαίδευσης. Κατά την αναγνώριση, η νέα εικόνα προβάλλεται σε αυτόν τον χώρο, και συγκρίνεται η θέση της με τις προβολές των γνωστών προσώπων βάσει μετρικών απόστασης

Μία από τις πρώτες ολοκληρωμένες προσεγγίσεις που χρησιμοποιήθηκαν στην ανίχνευση προσώπων είναι η προσέγγιση των ιδιοπρόσωπων που βασίζεται στην Ανάλυση Κύριων Συνιστωσών (PCA). Η βασική ιδέα της PCA είναι η προβολή των αρχικών εικόνων προσώπου σε ένα νέο διανυσματικό χώρο χαμηλής διάστασης που ονομάζεται «χώρος προσώπου». Σε αυτόν τον χώρο, κάθε πρόσωπο αναπαρίσταται ως γραμμικός συνδυασμός ιδιοπρόσωπων, οι οποίοι υπολογίζονται από τις κύριες συνιστώσες του συνόλου εκπαίδευσης.

Κατά την ανίχνευση, το νέο πρόσωπο προβάλλεται σε αυτόν τον χώρο και η θέση του αντιπαραβάλλεται με τις προβολές των γνωστών προσώπων χρησιμοποιώντας μετρικές απόστασης.



Εικόνα 1:Παράδειγμα Eigenface

(https://www.cs.toronto.edu/~guerzhoy/320/lec/pca.pdf)

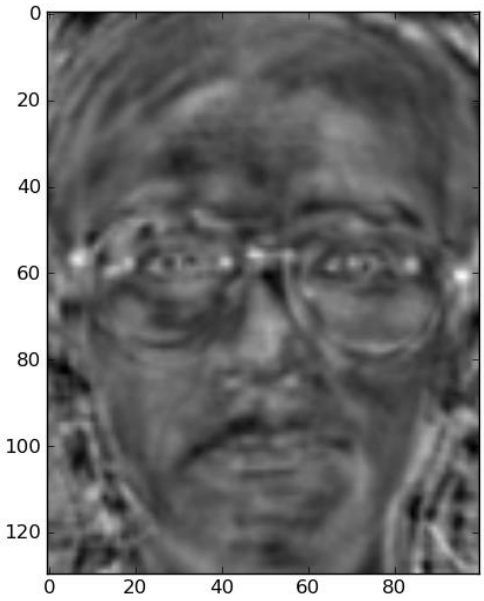
Η μέθοδος Fisherfaces είναι μία από τις πρώτες σημαντικές επεκτάσεις της ανίχνευσης προσώπου, η οποία επιδιώκει να ξεπεράσει τα βασικά μειονεκτήματα των μοντέλων Eigenfaces (με βάση την PCA), κυρίως την ευαισθησία στις αλλαγές του φωτισμού, της έκφρασης ή της θέσης του κεφαλιού. Η βασική ιδέα πίσω από το Fisherfaces είναι ο συνδυασμός δύο τεχνικών: της Ανάλυσης Κύριων Συνιστωσών (PCA) για τη μείωση της διαστατικότητας των δεδομένων και της Γραμμικής Διακριτικής Ανάλυσης (LDA) για τη μεγιστοποίηση της διαχωριστικότητας μεταξύ διαφορετικών κατηγοριών προσώπων.

Η γραμμική διακριτική ανάλυση (LDA) λειτουργεί σύμφωνα με τα επισημασμένα δεδομένα, τα οποία ταυτοποιούνται ως εποπτευόμενα, και στοχεύει στον προσδιορισμό των αξόνων προβολής που θα οδηγήσουν στην ελαχιστοποίηση της διασποράς εντός της ίδιας κατηγορίας μεταξύ προσώπων της ίδιας κατηγορίας (π.χ. του ίδιου ατόμου), ενώ μεγιστοποιείται ο διαχωρισμός μεταξύ προσώπων διαφορετικών κατηγοριών. Η προσέγγιση Fisherfaces διατηρεί, επομένως, τις πιο ενημερωτικές λεπτομέρειες για την ταξινόμηση, σε αντίθεση με την εξήγηση όλων των παραλλαγών που υπάρχουν στο σύνολο δεδομένων, η οποία είναι το αντικείμενο της Ανάλυσης Κύριων Συνιστωσών (PCA).

Η μέθοδος έχει δύο βήματα: πρώτα εφαρμόζεται η PCA για τη μείωση της διάστασης και την αποφυγή υπερπροσαρμογής, και στη συνέχεια εφαρμόζεται η LDA στον μειωμένο χώρο. Το τελικό σύστημα προβολής που προκύπτει, δηλαδή το Fisherfaces, καταγράφει τα χαρακτηριστικά του προσώπου που είναι βέλτιστα για τη διάκριση μεταξύ ατόμων, ακόμη και υπό διαφορετικές συνθήκες φωτισμού ή έκφρασης.

Η απόδοση της μεθόδου είναι σημαντικά πιο σταθερή σε πραγματικές εφαρμογές σε σύγκριση με το Eigenfaces και έχει χρησιμεύσει ως πρότυπο για πολλά μεταγενέστερα συστήματα

αναγνώρισης. Ωστόσο, η ακρίβειά της εξαρτάται από την ποιότητα και το μέγεθος του συνόλου εκπαίδευσης και υστερεί σε απόδοση σε σύγκριση με τις σύγχρονες προσεγγίσεις βαθιάς νευρωνικής δικτύωσης.



Εικόνα 2 Παράδειγμα Fisherfac

(https://www.bytefish.de/blog/fisherfaces.html)

Οι feature-based μέθοδοι αποτελούν μια εξέλιξη σε σχέση με τις πρώιμες ολιστικές (holistic) προσεγγίσεις, εστιάζοντας στην εξαγωγή τοπικών ή μορφολογικών χαρακτηριστικών του προσώπου αντί να αναλύουν το πρόσωπο ως σύνολο. Στόχος τους είναι να περιγράψουν με ακρίβεια επιμέρους περιοχές του προσώπου (μάτια, μύτη, στόμα, γνάθος), ώστε να εξάγονται πιο ανθεκτικά διανύσματα χαρακτηριστικών για ταξινόμηση ή αναγνώριση.

Ίσως η πιο σημαντική και ευρέως χρησιμοποιούμενη τεχνική αυτής της κατηγορίας είναι η τεχνική Histogram of Oriented Gradients (HOG). Η HOG εξετάζει μια εικόνα βρίσκοντας τις κλίσεις φωτεινότητας κάθε εικονοστοιχείου - δηλαδή πώς η φωτεινότητα μεταβάλλεται οριζόντια και κάθετα. Στη συνέχεια χωρίζει την εικόνα σε μικρά τμήματα (που ονομάζονται κελιά) και για κάθε κελί διατηρεί ένα ιστόγραμμα προσανατολισμένων γραμμών (προσανατολισμών), το οποίο κωδικοποιεί τις ακμές και το σχήμα του τμήματος.

Έτσι, το HOG σχηματίζει έναν διανυσματικό περιγραφέα για ολόκληρη την εικόνα, ανεξάρτητα από το συνολικό φόντο, αλλά με έμφαση στο σχήμα και τις ακμές του προσώπου. Αυτή η διαδικασία είναι πολύ αποτελεσματική για την ανίχνευση προσώπου και έχει χρησιμοποιηθεί σε τεράστιο βαθμό σε εφαρμογές πραγματικού χρόνου.

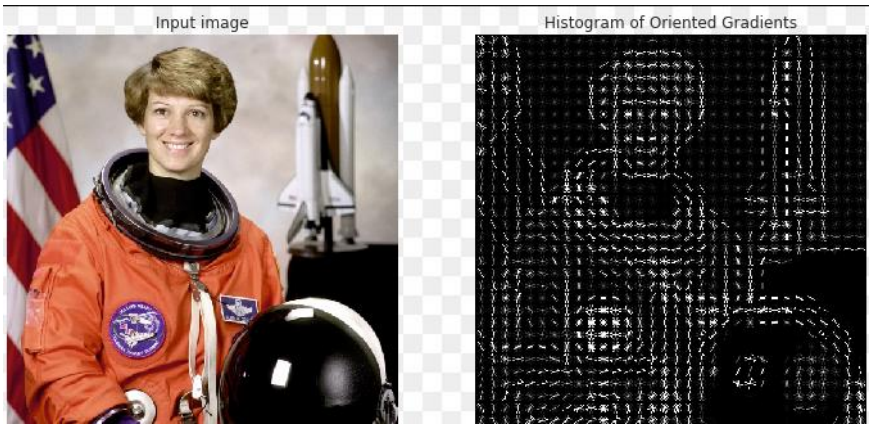
Η προσέγγιση HOG, αν και λιγότερο πολύπλοκη, επιτυγχάνει πολύ ικανοποιητικά αποτελέσματα όταν εφαρμόζεται σε καλά φωτισμένες παραστάσεις με ευκρινείς γωνίες του προσώπου. Συνήθως συνδυάζεται με έναν Support Vector Machine (SVM) και προσφέρει αποτελεσματική και αξιόπιστη ανίχνευση με χαμηλές απαιτήσεις βαθιάς μάθησης.

Ορισμένες άλλες γνωστές προσεγγίσεις που βασίζονται σε χαρακτηριστικά περιλαμβάνουν τα Local Binary Patterns (LBP), τα φίλτρα Gabor και τους descriptors SIFT (Scale-Invariant Feature Transform) και SURF (Speeded-Up Robust Features). Οι δύο τελευταίοι έχουν βρει ευρεία χρήση στην όραση υπολογιστών για την ανίχνευση και περιγραφή σημείων κλειδιών σε εικόνες.

Ειδικότερα, ο SIFT ανιχνεύει γωνίες και περιοχές ενδιαφέροντος σε διαφορετικές κλίμακες μεγέθυνσης (scale-space), με τα διανύσματα χαρακτηριστικών να είναι αναλλοίωτα σε περιστροφή, κλίμακα και φωτισμό. Συνεπώς, είναι ιδιαίτερα κατάλληλο για προβλήματα αναγνώρισης προσώπων υπό μεταβαλλόμενες συνθήκες λήψης.

Το SURF, από την άλλη πλευρά, είναι μια ταχύτερη παραλλαγή του SIFT, μειώνοντας τον υπολογιστικό φόρτο χωρίς σημαντική απώλεια ακρίβειας. Χρησιμοποιεί επίσης ολοκληρωμένες εικόνες και βασίζεται σε φίλτρα Haar wavelet για την εξαγωγή παρόμοιων αλλά ταχύτερων περιγραφών.

Παρόλο που αυτές οι προσεγγίσεις χρησιμοποιούνται πλέον σε μικρότερο βαθμό από ό,τι οι προσεγγίσεις βαθιάς μάθησης σε εφαρμογές προσώπου, παρείχαν το ενδιάμεσο βήμα που ήταν απαραίτητο για τη δημιουργία της αναγνώρισης προσώπου και εξακολουθούν να απαιτούνται για την αναγνώριση προσώπου.



Εικόνα 3: Παράδειγμα History of Oriented Gradients

(https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa)

Η πρόσφατη πρόοδος στη μηχανική μάθηση και, πιο συγκεκριμένα, στα βαθιά νευρωνικά δίκτυα έχει οδηγήσει σε κολοσσιαίες εξελίξεις στον τομέα της αναγνώρισης προσώπου. Σε σύγκριση με προηγούμενες προσεγγίσεις – είτε ολιστικές είτε βασισμένες σε χαρακτηριστικά (feature-based) – τα μοντέλα βαθιάς μάθησης αξιοποιούν τη δυνατότητά τους να μαθαίνουν αναπαραστάσεις

απευθείας από τα δεδομένα εικόνας, χωρίς να απαιτείται η χειροκίνητη σχεδίαση χαρακτηριστικών από ειδικούς.

Η πιο διαδεδομένη αρχιτεκτονική για τέτοιου είδους εφαρμογές είναι τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks – CNNs). Τα δίκτυα αυτά μαθαίνουν ιεραρχικά μοτίβα από τις εικόνες: ξεκινώντας από χαμηλού επιπέδου χαρακτηριστικά (όπως άκρα και γωνίες) και φτάνοντας σε πιο αφηρημένες δομές (όπως τα μάτια, η μύτη και το συνολικό σχήμα του προσώπου). Ένα ιδιαίτερα γνωστό μοντέλο αυτής της προσέγγισης είναι το DeepFace της Meta (πρώην Facebook), το οποίο κατάφερε να προσεγγίσει την ακρίβεια επιπέδου ανθρώπου στην αναγνώριση προσώπου. Άλλες τυπικές υλοποιήσεις είναι τα FaceNet, VGGFace, ArcFace και MagFace, τα οποία προσφέρουν ακόμα υψηλότερη ακρίβεια με τη βοήθεια εξειδικευμένων τεχνικών μάθησης όπως οι συναρτήσεις απώλειας με γωνιακό περιθώριο (angular margin loss functions).

Ωστόσο, οι μέθοδοι βαθιάς μάθησης συνοδεύονται και από ορισμένα μειονεκτήματα. Πρώτον, απαιτούν μεγάλο όγκο δεδομένων εκπαίδευσης προκειμένου να μάθουν επαρκώς τις διακριτικές ιδιότητες των προσώπων. Επιπλέον, η διαδικασία εκπαίδευσης είναι εξαιρετικά υπολογιστικά απαιτητική, καθώς προϋποθέτει ισχυρές μονάδες GPU και εκτεταμένο χρόνο επεξεργασίας. Ένα ακόμα ζήτημα είναι η δυσκολία ερμηνείας της εσωτερικής λειτουργίας των μοντέλων. Εξαιτίας της πολυπλοκότητας και του πολυεπίπεδου χαρακτήρα τους, τα μοντέλα αυτά χαρακτηρίζονται συχνά ως "μαύρα κουτιά" (black-box models), καθιστώντας δύσκολη την πλήρη κατανόηση της λήψης αποφάσεων.



Εικόνα 4: Δομή του FaceNet

(https://www.cvfoundation.org/openaccess/content_cvpr_2015/papers/Schroff_FaceNet_A_Unified_2015_CVPR_paper.pdf)

2.2 ΤΕΧΝΙΚΗ ΒΑΘΜΟΝΟΜΗΣΗ ΚΑΜΕΡΑΣ

Η βαθμονόμηση είναι η διαδικασία εύρεσης δύο κύριων τύπων παραμέτρων: των intrinsic (εσωτερικών) παραμέτρων και των extrinsic (εξωτερικών) παραμέτρων.

Οι intrinsic παράμετροι αντιπροσωπεύουν τις εσωτερικές ιδιότητες του συστήματος απεικόνισης της κάμερας, όπως: α) το σύστημα απεικόνισης της κάμερας:

Εστιακό μήκος: καθορίζει πόσο μεγεθύνει ή κάνει ένα αντικείμενο να φαίνεται μικρότερο.

Κύριο σημείο: είναι το σημείο στο επίπεδο εικόνας όπου τέμνεται ο οπτικός άξονας.

Παραμόρφωση φακού: Η ύπαρξη οπτικών παραμορφώσεων, όπως η ακτινική παραμόρφωση (κυκλική παραμόρφωση - πιο έντονη στις άκρες της εικόνας) και η εφαπτομενική παραμόρφωση (παραμόρφωση λόγω ασύμμετρης ευθυγράμμισης του φακού).

Οι πληροφορίες αυτές ενσωματώνονται συνήθως στον εσωτερικό πίνακα παραμέτρων ή στον πίνακα της κάμερας:

Κατηγορία	Περιγραφή	Ενδεικτικές Μέθοδοι	Πλεονεκτήματα	Μειονεκτήματα
Ολιστική	Χρήση ολόκληρης της εικόνας προσώπου, προβολή σε χώρο χαμηλής διάστασης	Eigenfaces, Fisherfaces	Απλές, γρήγορες, ερμηνεύσιμες	Ευαίσθητες σε φωτισμό, έκφραση, γωνία
Βασισμένες σε γεωμετρία	Χρήση σχετικών θέσεων χαρακτηριστικών του προσώπου (landmarks)	Αναλογίες σημείων, αποστάσεις	Εύκολη κατανόηση, χαμηλή υπολογιστική απαίτηση	Εξαρτώνται από την ακρίβεια της ανίχνευσης σημείων
Βασισμένες σε χαρακτηριστικά	Εξαγωγή τοπικών περιγραφικών χαρακτηριστικών (π.χ. υφής ή καμπυλότητας)	HOG, LBP, SIFT, SURF, Gabor filters	Πιο ανθεκτικές σε θόρυβο, παραλλαγές φωτισμού	Χρειάζονται χειροκίνητο σχεδιασμό χαρακτηριστικών
Βασισμένες σε βαθιά μάθηση	Αυτόματη μάθηση χαρακτηριστικών μέσω συνελικτικών νευρωνικών δικτύων	FaceNet, VGGFace, ArcFace, DeepFace	Πολύ υψηλή ακρίβεια, γενίκευση, αυτοματοποιημένη διαδικασία	Απαιτούν μεγάλα δεδομένα, GPU, δυσκολία ερμηνείας

Πίνακας 1 Σύγκριση Κατηγοριών Μεθόδων Αναγνώρισης Προσώπων

$$\mathbf{K} = \begin{vmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{vmatrix}$$

Όπου f_x,f_y είναι οι εστιακές αποστάσεις (σε εικονοστοιχεία) κατά μήκος των αξόνων x και y και c_x, c_y είναι οι συντεταγμένες του κεντρικού σημείου.

Οι extrinsic παράμετροι καθορίζουν τον προσανατολισμό και τη θέση της κάμερας σε σχέση με ένα παγκόσμιο ή τοπικό πλαίσιο αναφοράς. Συνήθως εκφράζονται ως εξής:

Πίνακας περιστροφής (R): δείχνει πώς περιστρέφεται η κάμερα.

Διάνυσμα μεταφοράς (t): δείχνει τη μετατόπιση της κάμερας από την αρχή του συστήματος αναφοράς.

Η γνώση αυτών των παραμέτρων επιτρέπει την ανακατασκευή τρισδιάστατων σημείων από εικόνες 2D και αντίστροφα, καθώς και τη σωστή τοποθέτηση εικονικών αντικειμένων στον πραγματικό χώρο.

Η πιο διαδεδομένη μέθοδος βαθμονόμησης (calibration) περιλαμβάνει τη λήψη εικόνων ενός προκαθορισμένου μοτίβου (συνήθως μια σκακιέρα) από διαφορετικές οπτικές γωνίες. Τα σημεία ενδιαφέροντος, δηλαδή οι εσωτερικές γωνίες του μοτίβου, ανιχνεύονται στη συνέχεια αυτόματα στο επίπεδο της εικόνας και αντιστοιχίζονται με τις τρισδιάστατες συντεταγμένες τους στο φυσικό χώρο, οι οποίες είναι γνωστές. Αυτές οι αντιστοιχίες λαμβάνονται ως είσοδος για υπολογιστικούς αλγορίθμους (μέθοδο Zhang) για την εκτίμηση των εσωτερικών και εξωτερικών παραμέτρων της κάμερας.

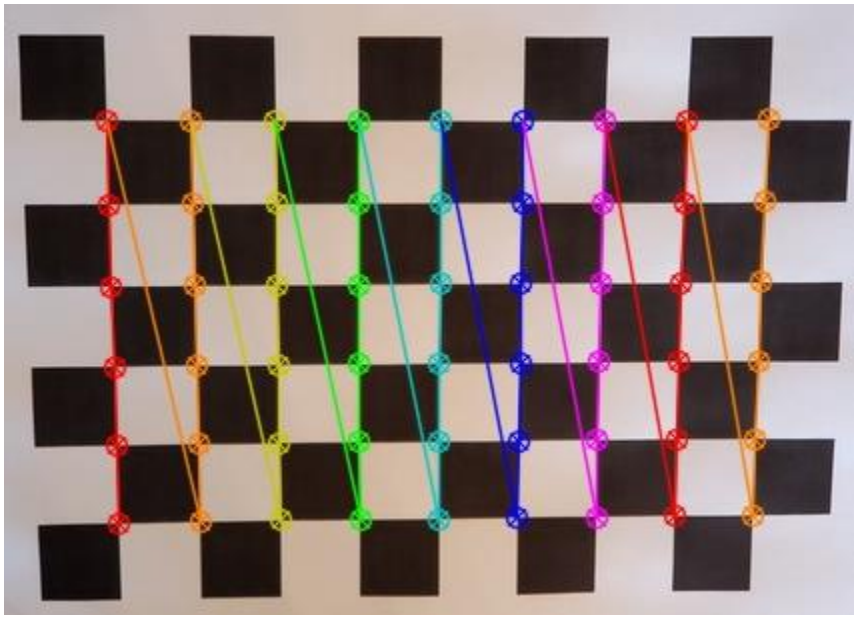
Η βιβλιοθήκη OpenCV παρέχει έτοιμα εργαλεία, όπως η συνάρτηση cv2.calibrateCamera(), τα οποία διευκολύνουν σημαντικά την υλοποίηση αυτής της διαδικασίας.

Τόσο η ποσότητα όσο και η ποιότητα των εικόνων προτύπων που χρησιμοποιούνται είναι σημαντικοί παράγοντες για τη διαχείριση της ποιότητας της διαδικασίας βαθμονόμησης. Όσο μεγαλύτερος είναι ο αριθμός των εικόνων που λαμβάνονται σε διαφορετικές γωνίες, τόσο ακριβέστερος είναι ο προσδιορισμός της γεωμετρίας της κάμερας και τόσο μεγαλύτερη είναι η ακρίβεια.

Η επιτυχία της διαδικασίας βαθμονόμησης της κάμερας εξαρτάται σε μεγάλο βαθμό από την ποσότητα και τη μεταβλητότητα των εικόνων μοτίβου που λαμβάνονται. Η λήψη εικόνων από μεγαλύτερο αριθμό πλεονεκτικών σημείων οδηγεί σε πιο ισχυρή εκτίμηση των γεωμετρικών παραμέτρων της κάμερας. Η χρήση λιγότερων εικόνων ή εικόνων που λαμβάνονται από κοντινές θέσεις μπορεί να οδηγήσει σε σφάλματα λόγω ανεπαρκούς μεταβλητότητας και να οδηγήσει ακόμη και σε ασταθή υπολογισμό ή γεωμετρικές παραμορφώσεις στο αποτέλεσμα της βαθμονόμησης.

Η αξιοπιστία της βαθμονόμησης είναι ζωτικής σημασίας για την επιτυχία των επόμενων βημάτων, όπως η χρήση της solvePnP(), η οποία αντιστοιχίζει τα 2D face landmarks σε 3D εκτίμηση της στάσης του κεφαλιού. Η ακρίβεια της μεθόδου έχει άμεση εξάρτηση από την ποιότητα της βαθμονόμησης: μικρά σφάλματα στις παραμέτρους οδηγούν σε σφάλματα προσανατολισμού, επηρεάζοντας κρίσιμα αποτελέσματα όπως η υπολογιστική εκτίμηση του βλέμματος.

Στο πλαίσιο της παρούσας μελέτης, πραγματοποιήθηκε εφάπαξ βαθμονόμηση της κάμερας χρησιμοποιώντας ένα μοτίβο σκακιέρας. Τα παραμετρικά δεδομένα που προέκυψαν αποθηκεύτηκαν σε ένα αρχείο .pkl χρησιμοποιώντας τη βιβλιοθήκη pickle και φορτώθηκαν αυτόματα κατά την εκκίνηση του συστήματος. Αυτή η τακτική επιτρέπει τη διατήρηση σταθερών συνθηκών παρακολούθησης και μειώνει το χρόνο αρχικοποίησης. Αυτή η βαθμονόμηση είναι κρίσιμη για την αξιόπιστη χρήση της solvePnP(), μιας μεθόδου που επιτρέπει τη μετατροπή των 2D facial landmarks σε 3D εκτίμηση της στάσης του κεφαλιού. Αυτός ο προσανατολισμός αποτελεί τη βάση για την υπολογιστική παρακολούθηση του οφθαλμικού προσανατολισμού.



Εικόνα 5: Βαθμονόμηση κάμερας

Παρακάτω παρέχεται το script που αναπτύχθηκε και ακολουθεί επεξήγηση του κώδικα:

```
import cv2
import numpy as np
import glob
import pickle

# Chessboard pattern size
pattern_size = (8, 6)
```

```
#object points
objp = np.zeros((pattern_size[0] * pattern_size[1], 3), np.float32)
objp[:, :2] = np.mgrid[0:pattern_size[0], 0:pattern_size[1]].T.reshape(-1, 2)

# Arrays to store object points and image points from all the images
objpoints = []  # 3D points in real-world space
imgpoints = []  # 2D points in image plane

# Path to images
images_path                                                                    =
r'C:\Users\Myrtle\source\repos\ImagesSavedForCalibration\ImagesSavedForCalibration\images\*.png'
images = glob.glob(images_path)

print("Detecting chessboard corners in images...")

for fname in images:
    img = cv2.imread(fname)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Find chessboard corners
    ret, corners = cv2.findChessboardCorners(
        gray,
        pattern_size,
        cv2.CALIB_CB_ADAPTIVE_THRESH + cv2.CALIB_CB_FAST_CHECK + cv2.CALIB_CB_NORMALIZE_IMAGE
    )

    if ret:
        print(f"Corners detected in {fname}")
        objpoints.append(objp)  # 3D points
        imgpoints.append(corners)  # 2D points


# Perform camera calibration
print(" Performing camera calibration...")
ret, camera_matrix, distortion_coeffs, rvecs, tvecs = cv2.calibrateCamera(
    objpoints, imgpoints, gray.shape[::-1], None, None
)

# Display calibration results
print("\nCamera calibration completed!")
print(f"Camera Matrix:\n{camera_matrix}")
print(f"Distortion Coefficients:\n{distortion_coeffs}")

# Save calibration results to a file
calibration_data = {
    "camera_matrix": camera_matrix,
    "distortion_coeffs": distortion_coeffs,
```

```
    "rvecs": rvecs,
    "tvecs": tvecs
}
with open("calibration_data.pkl", "wb") as f:
    pickle.dump(calibration_data, f)

print("Calibration data saved to 'calibration_data.pkl'")

cv2.waitKey(0)
cv2.destroyAllWindows()
```

Ο κώδικας αρχικά (γραμμές 7 και 10) ορίζει το μέγεθος του μοτίβου σκακιέρας και δημιουργεί έναν πίνακα τρισδιάστατων σημείων (objp), που αντιπροσωπεύει τις γνωστές φυσικές θέσεις των γωνιών. Έπειτα, στις γραμμές 14-15, δημιουργούνται λίστες για την αποθήκευση των 3D και 2D σημείων από όλες τις εικόνες. Στις γραμμές 18-19 φορτώνονται εικόνες από έναν φάκελο μέσω glob. Από τη γραμμή 22 ξεκινά η επεξεργασία των εικόνων: κάθε μία μετατρέπεται σε grayscale και εξετάζεται για γωνίες σκακιέρας με τη cv2.findChessboardCorners. Αν εντοπιστούν, καταγράφονται τα αντίστοιχα σημεία (γραμμές 30-31). Στη γραμμή 36 εκτελείται η βαθμονόμηση μέσω της cv2.calibrateCamera, υπολογίζοντας τη μητρική της κάμερας και τους συντελεστές παραμόρφωσης. Τέλος, από τις γραμμές 44-47, τα αποτελέσματα αποθηκεύονται σε αρχείο .pkl για μελλοντική χρήση, αποφεύγοντας επανάληψη της διαδικασίας. Δημιουργούνται λίστες για την αποθήκευση των 3D και 2D σημείων από όλες τις εικόνες. Στις γραμμές 18-19 φορτώνονται εικόνες από έναν φάκελο μέσω glob. Από τη γραμμή 22 ξεκινά η επεξεργασία των εικόνων: κάθε μία μετατρέπεται σε grayscale και εξετάζεται για γωνίες σκακιέρας με τη cv2.findChessboardCorners. Αν εντοπιστούν, καταγράφονται τα αντίστοιχα σημεία (γραμμές 30-31). Στη γραμμή 36 εκτελείται η βαθμονόμηση μέσω της cv2.calibrateCamera, υπολογίζοντας τη μητρική της κάμερας και τους συντελεστές παραμόρφωσης. Τέλος, από τις γραμμές 44-47, τα αποτελέσματα αποθηκεύονται σε αρχείο .pkl για μελλοντική χρήση, αποφεύγοντας επανάληψη της διαδικασίας.

2.3 ΚΑΤΗΓΟΡΙΕΣ ΑΝΙΧΝΕΥΣΗΣ

Οι eye trackers μπορούν να χωριστούν σε διάφορες κατηγορίες, ανάλογα με τα κριτήρια που χρησιμοποιούνται για την ταξινόμησή τους. Σύμφωνα με τους Young και Sheena (1975), οι τεχνικές παρακολούθησης του βλέμματος διακρίνονται σε δύο βασικούς τύπους με βάση τη μέθοδο μέτρησης: (α) τη μέτρηση της θέσης των ματιών σε σχέση με το κεφάλι και (β) τη μέτρηση του προσανατολισμού των ματιών σε σχέση με τον χώρο (σε αναφορά με ένα εξωτερικό σημείο). Αξιοποιώντας αυτό το πλαίσιο, ο Duchowski (2003) προτείνει την ύπαρξη τεσσάρων μεγάλων κατηγοριών μεθοδολογιών για την παρακολούθηση και καταγραφή των κινήσεων των ματιών.

Ανάμεσα στις τεχνικές αυτές, ξεχωρίζουν δύο ευρείες κατηγορίες: οι φυσιολογικές μέθοδοι (ή ενεργές) και οι μέθοδοι υπολογιστικής όρασης (Computer vision-based), οι οποίες διαφοροποιούνται κυρίως ως προς το είδος των δεδομένων που χρησιμοποιούν και τον απαιτούμενο εξοπλισμό.

Η (Ηλεκτροοφθαλμογραφία) EOG είναι μια φυσιολογική μέθοδος που βασίζεται στη μέτρηση του βιοηλεκτρικού δυναμικού που δημιουργείται μεταξύ του θετικά φορτισμένου κερατοειδούς και του αρνητικά φορτισμένου αμφιβληστροειδούς. Ηλεκτρόδια σε επαφή με τα μάτια επιτρέπουν τη μέτρηση της μεταβολής αυτού του δυναμικού που συσχετίζεται με την κίνηση των ματιών. Η μέθοδος αυτή έχει το πλεονέκτημα ότι μπορεί να χρησιμοποιηθεί ακόμη και με κλειστά μάτια (π.χ. κατά τη διάρκεια του ύπνου ή της υπνηλίας), αλλά είναι ευαίσθητη στον ηλεκτρικό θόρυβο και πρέπει να έρθει σε επαφή με το δέρμα του σώματος, περιορίζοντας την άνεση και την ακρίβεια.



Εικόνα 6: Απεικόνιση της τοποθέτησης των ηλεκτροδίων
(https://www.frontiersin.org/journals/neurology/articles/10.3389/fneur.2022.917413/full)

Η χρήση φακών επαφής του σκληρού χιτώνα/ πηνίου αναζήτησης, αποτελεί μία από τις πιο ακριβείς μεθόδους μέτρησης των οφθαλμικών κινήσεων. Το πηνίο παράγει τάση όταν ο χρήστης εκτίθεται σε μαγνητικό πεδίο, οπότε μπορεί να γίνει η ακριβής μέτρηση της θέσης του ματιού. Η τεχνική αυτή είναι εξαιρετικά ακριβής και χρησιμοποιείται ευρέως σε ερευνητικά εργαστήρια, αλλά είναι ιδιαίτερα επεμβατική και άβολη, με πιθανότητα να προκαλέσει δυσφορία ή ακόμη και βλάβη στον κερατοειδή.

Η φωτο-οφθαλμογραφία (POG) ή βιντεο-οφθαλμογραφία (VOG) αποτελεί μία συλλογή διάφορων τεχνικών καταγραφής κινήσεων του οφθαλμού. Χρησιμοποιεί κάμερες για τη λήψη χαρακτηριστικών οπτικών χαρακτηριστικών του οφθαλμού, όπως η κόρη, η ίριδα και τα βλέφαρα Οι πιο σύγχρονες εκδόσεις περιλαμβάνουν υπέρυθρο (IR) φωτισμό και φωτοδίοδοι για σταθερή και ακριβή καταγραφή ακόμη και υπό δύσκολες συνθήκες φωτισμού. Η αυτόματη αναγνώριση των χαρακτηριστικών μειώνει το ανθρώπινο λάθος, αλλά απαιτεί ακινητοποίηση του κεφαλιού ή χρήση υποστήριξης (π.χ. κάλυμμα κεφαλής ή γυαλιά κάμερας). Είναι μία από τις πιο διαδεδομένες και μη παρεμβατικές τεχνικές.

Ο συνδυασμός αντανάκλαση κόρης και κερατοειδούς (Pupil-Corneal Reflection – PCCR) βασίζεται στον εντοπισμό της αντανάκλασης από την κόρη και τον κερατοειδή χρησιμοποιώντας υπέρυθρες κάμερες υψηλής ταχύτητας. Ο υπολογισμός του σημείου εστίασης (gaze estimation) γίνεται μέσω τριγωνισμού, λαμβάνοντας υπόψη τη σχετική θέση της κόρης και του φωτεινού σημείου που δημιουργείται από την IR πηγή. Οι συσκευές eye tracking που εφαρμόζουν την PCCR είναι είτε σταθερές (desktop-based) είτε φορητές (wearable) και προσφέρουν υψηλή ακρίβεια σε πραγματικό χρόνο.

Πιο πρόσφατα, η πρόοδος στην υπολογιστική όραση και τη μηχανική μάθηση επέτρεψε την ανάπτυξη αμιγώς οπτικών μεθόδων που δεν απαιτούν εξειδικευμένο εξοπλισμό. Μέθοδοι όπως οι Haar Cascades, η Dlib, το Google MediaPipe, και νευρωνικά δίκτυα τύπου CNN, YOLO κ.ά., χρησιμοποιούνται για την ανίχνευση προσώπου, ματιών και βλέμματος σε πραγματικό χρόνο με απλές κάμερες. Αυτές οι λύσεις είναι ιδιαίτερα ελκυστικές για εφαρμογές χαμηλού κόστους, όμως υστερούν ακόμη σε ακρίβεια σε σύγκριση με τις εξειδικευμένες τεχνικές.

Οι eye trackers μπορούν να διαχωριστούν και με βάση τον τρόπο χρήσης τους σε δύο κύριες κατηγορίες (Sharafi, Soh, Guéhéneuc, 2015), τους παρεμβατικούς (invasive) και τους μη-παρεμβατικούς (non-invasive).

Οι παρεμβατικοί ανιχνευτές ματιών έχουν ένα είδος σκελετού ή ζώνης κεφαλής που προσαρμόζεται στο κεφάλι του χρήστη και φέρει τρεις μικρές κάμερες. Δύο από αυτές αποσκοπούν στην καταγραφή της κίνησης των ματιών χρησιμοποιώντας υπέρυθρη ακτινοβολία που αντανακλάται από τις κόρες των ματιών. Η τρίτη κάμερα, η οποία είναι συνήθως προαιρετική, παρακολουθεί τη θέση του κεφαλιού για τη βελτίωση της ακρίβειας. αλλά ακόμη και μικρές κινήσεις του κεφαλιού μπορούν να προκαλέσουν σφάλματα στους υπολογισμούς της θέσης του βλέμματος.

Οι μη παρεμβατικοί ανιχνευτές ματιών, ωστόσο, αποτελούνται από παλαιότερες και νεότερες γενιές τεχνολογίας. Η παλαιότερη τεχνολογία προβάλλει ακτίνες φωτός στα μάτια του υποκειμένου και στη συνέχεια σχηματίζει την εκτίμησή της για την κατεύθυνση του βλέμματος του υποκειμένου βάσει της ανάλυσης της ανακλώμενης ακτινοβολίας. Είναι περιορισμένες τόσο ως προς την ακρίβεια όσο και ως προς τη χωρική ανάλυση της μέτρησης.

Σε αντίθεση με τα προηγούμενα μοντέλα, η τελευταία γενιά συσκευών ανίχνευσης οφθαλμών, που αναφέρονται επίσης ως συσκευές ανίχνευσης οφθαλμών με βάση το βίντεο, διαθέτουν διάφορα χαρακτηριστικά. Σε γενικές γραμμές, αποτελούνται από έναν υπολογιστή, ένα ζεύγος μικρών καμερών και ένα στρώμα υπερύθρων. Για να επιτευχθεί η παρακολούθηση, ο υπολογιστής παρακολουθεί το βλέμμα του χρήστη μέσω της τοποθέτησης του κεφαλιού του, και με συνέπεια σε σχέση με την ευθυγράμμιση των φρυδιών, της μύτης και των χειλιών του. Επιπλέον, η υπέρυθρη αντανάκλαση του κερατοειδούς και το κέντρο της κόρης του ματιού βοηθούν στη διάκριση μεταξύ των κινήσεων των ματιών και των κινήσεων του κεφαλιού. Όλοι οι βιντεοσκοπημένοι ανιχνευτές οφθαλμών θεωρούνται μη παρεμβατικοί, ενώ άλλοι που χρησιμοποιούν μεθόδους όπως η ηλεκτρο-οφθαλμογραφία (EOG), η αναζήτηση με φακούς

επαφής σκληρού χιτώνα/πηνίου ή η απεικονιστική οφθαλμογραφία (POG)/βιντεο-οφθαλμογραφία (VOG) φαίνεται να είναι κυρίως παρεμβατικοί.

2.4 ΠΡΟΚΛΗΣΕΙΣ ΑΝΙΧΝΕΥΣΕΙΣ

Η παρακολούθηση και η ανίχνευση της κίνησης των ματιών σε πραγματικό χρόνο θέτει πολλές τεχνικές και περιβαλλοντικές προκλήσεις. Και οι δύο προαναφερθείσες μέθοδοι (η φυσιολογική και η μηχανική) έχουν συγκεκριμένα σημεία ευαισθησίας, τα οποία μπορούν να υπονομεύσουν την ακρίβεια ή να οδηγήσουν σε διακοπές στη διαδικασία συλλογής δεδομένων. Εξάλλου, η ετερογένεια των ανθρώπων (διαφορές στη χρώση, το σχήμα των ματιών, την κόρη του ματιού κ.λπ.), η παρουσία διαφορών στο φωτισμό, η κίνηση του κεφαλιού και η ποιότητα των υλικών επηρεάζουν άμεσα την αξιοπιστία ανίχνευσης των αλγορίθμων.

- Εναλλαγές στις συνθήκες φωτισμού

Οι αλγόριθμοι που βασίζονται σε βίντεο έχουν επίσης σημαντικό, αρνητικό αντίκτυπο στα ποσοστά απόδοσής τους όταν υπάρχει εναλλαγή στην ένταση ή την κατεύθυνση του φωτός. Εξετάζοντας τις εργασίες που παρήγαγαν οι Zhu κ.ά. (2004), ακόμη και οι απλές σκιές στο πρόσωπο προκαλούν αποτυχία αναγνώρισης της κόρης ή των βλεφάρων, ιδίως στις περιπτώσεις της απλής κατωφλίωσης. Με τη σειρά τους, οι Chen κ.ά. (2023) τονίζουν ότι οι εργασίες που βασίζονται στο φαινόμενο της φωτεινής κόρης οδηγούν σε μια μικρή αύξηση της ευρωστίας, η οποία, ωστόσο, δεν επαρκεί για την πλήρη αποφυγή σφαλμάτων σε έντονες αντιθέσεις φωτός-σκιάς.

- Χαρακτηριστικά ματιών και αξεσουάρ παρεμβολής

Τα καπέλα, τα γυαλιά, οι ανακλαστικοί φακοί επαφής (PCCR) ή ακόμη και το έντονο μακιγιάζ μπορεί να προκαλέσουν σφάλματα. Τα γυαλιά, ως παράδειγμα, παραμορφώνουν τις αλήθειες του περιγράμματος των βλεφάρων: όπως έχει αναφερθεί από τους Morimoto & Mimica (2005), οι περισσότεροι αλγόριθμοι VOG αποτυγχάνουν να διακρίνουν την κόρη όταν υπάρχει έντονη αντανάκλαση στον φακό. Ομοίως, οι φακοί επαφής με σπείρα (Duchowski 2003) παράγουν ανεπιθύμητες αραιώσεις σήματος σε περίπτωση λανθασμένης τοποθέτησης.

- Διάφορα σχήματα των ματιών και χρώματα

Η γεωμετρία των ματιών και η πυκνότητα των χρωμάτων του προσώπου μπορεί να είναι εξαιρετικά διαφορετικές σε διαφορετικά άτομα. Οι αλγόριθμοι που βασίζονται σε μοτίβα (π.χ. HOG, Dalal & Triggs 2005), σε υφές (LBP, Ahonen et al. 2006) μπορούν επίσης να αποτύχουν στην περίπτωση σκούρας ίριδας ή πολύ ανοιχτόχρωμων ματιών, όπως αναφέρουν οι Irshayyid et al. (2023). Εν τω μεταξύ, οι αλγόριθμοι που βασίζονται σε ορόσημα (Kazemi & Sullivan 2014) αποτυγχάνουν να λειτουργήσουν όταν οι αριθμοί δεν αντιστοιχούν στην πραγματική εμφάνιση του προσώπου.

- Ανεπάρκεια αντιμετώπισης της μερικής απόκρυψης

Πολλοί αλγόριθμοι τυφλώνονται όταν τα βλέφαρα είναι μισόκλειστα (π.χ. κόπωση) ή όταν ένα χέρι αγγίζει ένα πρόσωπο. Σύμφωνα με τα αποτελέσματα που παρουσιάστηκαν από τους

Irshayyid κ.ά. (2023), η μερική απόκρυψη αντιμετωπίζεται καλύτερα με την προσθήκη μοντέλων CNN (π.χ. FaceNet Schroff κ.ά. 2015), αλλά εξακολουθεί να χρειάζεται αρκετές εικόνες εκπαίδευσης προκειμένου να γενικευτεί καλά.

- Διαφοροποίηση των γωνιών της κεφαλής - ευρύ φάσμα

Η μετακίνηση της θέσης του κεφαλιού (κλίση/στροφή/περιστροφή) καθιστά επίσης τα μάτια μη εύκολα ανιχνεύσιμα. Οι αλγόριθμοι που βασίζονται σε οριζόντια/κατακόρυφα μοτίβα (π.χ. Haar cascades, Viola & Jones 2001) αποτυγχάνουν με περισσότερες από 15 περιστροφές του κεφαλιού (είτε αρνητικές είτε θετικές). Οι Zhu κ.ά. (2004) πρότειναν τα δίκτυα ευθυγράμμισης προσώπων ως εναλλακτική λύση για την εκτέλεση της ευθυγράμμισης, τα οποία χρειάζονται περαιτέρω υπολογιστικούς πόρους και είναι επίσης επιρρεπή σε πολύ μεγάλες γωνίες

- Ακατάλληλη εκτέλεση σε πολύ χαμηλή ή πολύ υψηλή ανάλυση

Οι πληροφορίες σχετικά με αυτές τις γωνίες του περιγράμματος της κόρης χάνονται όταν προβάλλονται με κάμερα χαμηλού ρυθμού καρέ ή χαμηλής ανάλυσης (π.χ. web-cams παλαιού τύπου). Οι Papoutsaki et al. (2016) αναφέρουν ότι τα συστήματα WebGazer που αναπτύχθηκαν με χρήση υλικού χαμηλού κόστους απαιτούν επίσης κάποια ελάχιστη ανάλυση (περίπου 320 240 px) για να παρακολουθούνται σωστά.

- Έλλειψη διόρθωσης της κάμερας

Ακόμα και μικροσκοπικές διακυμάνσεις στις ενδογενείς/εξωγενείς τιμές προκαλούν εκτροπές στην εκτίμηση της στάσης του κεφαλιού και, ως εκ τούτου, του βλέμματος. Σύμφωνα με τους Hartley & Zisserman (2004), οι αλγόριθμοι PnP (π.χ. solvePnP, OpenCV 4 Cookbook Millan Escriva Laganiere 2019) είναι επιρρεπείς σε σφάλματα βαθμονόμησης και μπορεί να συμβεί απόκλιση +/-2-3 μοίρες στο τελικό διάνυσμα του βλέμματος.

- Ο ακριβός υπολογισμός σε πραγματικό χρόνο

Οι ανιχνευτές που βασίζονται σε CNN (MediaPipe Lugaresi et al. 2019, Dlib) χρειάζονται τις GPU ή ισχυρούς επεξεργαστές για να προσφέρουν περισσότερα από 20 fps. Αυτό αποτελεί ένα ακόμη εμπόδιο σε ζώνες με περιορισμένους πόρους, δηλαδή σε ενσωματωμένα συστήματα.

- Προβλήματα με άλλες εφαρμογές

Ο διαμοιρασμός πόρων (CPU/ GPU) θα δημιουργήσει πρόσθετες πιθανότητες καθυστέρησης και απώλειας πλαισίων στο σενάριο ταυτόχρονων εργασιών. Οι Irshayyid et al. (2023) προτείνουν τη χρήση μεθόδων ανίχνευσης περιοχών ενδιαφέροντος (ROI) προκειμένου να μειωθεί ο φόρτος.

2.5 ΕΦΑΡΜΟΓΕΣ

Η τεχνολογία ιχνηλάτησης των ματιών έχει προοδεύσει εξαιρετικά γρήγορα τις τελευταίες δεκαετίες και έχει βρει εφαρμογές σε πολυάριθμους κλάδους της επιστήμης και της τεχνολογίας. Με την ενσωμάτωση τεχνικών όρασης υπολογιστών και μηχανικής μάθησης, αποτελεί ένα ισχυρό εργαλείο για την ανάλυση της ανθρώπινης συμπεριφοράς, της εστίασης, της γνωστικής δραστηριότητας και της αλληλεπίδρασης με συστήματα υπολογιστών.

- Αλληλεπίδραση ανθρώπου-υπολογιστή

Ένας από τους πιο διερευνημένους τομείς είναι η χρήση των συσκευών παρακολούθησης ματιών στην αλληλεπίδραση ανθρώπου-υπολογιστή. Ο εξοπλισμός παρακολούθησης ματιών χρησιμοποιείται για το σχεδιασμό και τη δοκιμή γραφικών διεπαφών χρήστη (UI/UX), για να βοηθήσει στην κατανόηση της προσοχής του χρήστη κατά την πλοήγηση στο διαδίκτυο, στο λογισμικό ή στις εφαρμογές [10]. Περαιτέρω, αναπτύσσονται συστήματα «ελεγχόμενα από το βλέμμα», όπου η κατεύθυνση του βλέμματος χρησιμοποιείται ως είσοδος του χρήστη, αντικαθιστώντας ένα ποντίκι ή ένα πληκτρολόγιο - ιδιαίτερα πολύτιμο εκεί όπου δεν υπάρχει χειρισμός (π.χ. χειρουργεία, στρατιωτικά πιλοτήρια).

- Ιατρική και νευροεπιστήμες

Η ιχνηλάτηση των ματιών έχει μεγάλη σημασία στη γνωστική και κλινική έρευνα. Η παρακολούθηση των ματιών χρησιμοποιείται για τη διάγνωση και την παρακολούθηση νευρολογικών διαταραχών (π.χ. σκλήρυνση κατά πλάκας ή Πάρκινσον), για την αξιολόγηση της νευροαναπτυξιακής κατάστασης ενός παιδιού με αυτισμό ή σε πειράματα κόπωσης ή διαταραχών ύπνου.

- Εκπαίδευση και μαθησιακή αξιολόγηση

Η τεχνολογία ιχνηλάτησης των ματιών εφαρμόζεται για να κατανοηθεί ο τρόπος με τον οποίο οι μαθητές μαθαίνουν από οπτικοακουστικό περιεχόμενο ή γραπτά κείμενα. Επιτρέπει την ανίχνευση προβλημάτων ανάγνωσης και βοηθά στο σχεδιασμό εξατομικευμένων στρατηγικών διδασκαλίας, ιδίως για μαθητές με δυσλεξία ή άλλες μαθησιακές δυσκολίες.

- Διαφήμιση και μάρκετινγκ

Η τμηματοποίηση των σημείων στα οποία οι καταναλωτές εστιάζουν την προσοχή τους κατά την προβολή διαφημίσεων ή την περιήγηση σε προϊόντα επιτρέπει στις εταιρείες να απλοποιήσουν τη μορφή και το περιεχόμενο του διαφημιστικού τους υλικού. Ουσιαστικά, η τεχνολογία αυτή αποκαλύπτει τον τρόπο με τον οποίο οι δυνητικοί αγοραστές «διαβάζουν» τις πληροφορίες, επιτρέποντας στους υπεύθυνους για το μάρκετινγκ να έχουν αντικειμενικά στοιχεία με τα οποία μπορούν να καθοδηγήσουν τη δημιουργία διαφημιστικών εκστρατειών.

- Αυτοματοποιημένη ανάλυση οδήγησης

Στην τεχνολογία των έξυπνων αυτοκινήτων, η παρακολούθηση των ματιών χρησιμοποιείται για την παρακολούθηση της προσοχής του οδηγού. Μπορεί να ανιχνεύσει εγκαίρως την απόσπαση της προσοχής ή την υπνηλία και να ενεργοποιήσει προειδοποιητικές ειδοποιήσεις ή ακόμη και να λάβει αυτόματα μέτρα προστασίας.

- Ψυχολογία και αντίληψη

Η μελέτη της κίνησης των ματιών είναι ένα ισχυρό εργαλείο των επιστημών της πειραματικής ψυχολογίας, καθώς έχει άμεση σχέση με την αντίληψη, την προσοχή και τις διαδικασίες λήψης αποφάσεων. Κατά τη διαδικασία της οπτικής αναγνώρισης ή της κατανόησης σκηνών ή σε

πειράματα ανάγνωσης, οι σακκαδικές καθώς και οι ομαλές κινήσεις παρέχουν ενδείξεις για την ψυχολογική κατάσταση του ατόμου.

3. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ ΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

3.1 Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΑΤΙΣΜΟΥ PYTHON

Η Python είναι μια ιδιαίτερα διαδεδομένη γλώσσα υψηλού επιπέδου που έχει μεταμορφωθεί σε πρότυπο για τους επιστημονικούς υπολογιστές. Πρόκειται για μια δωρεάν εφαρμογή πλατφόρμας που τρέχει ομαλά με Windows, MacOS και Linux και είναι πολύ ευέλικτη και εύχρηστη τόσο σε έμπειρους όσο και σε ερασιτέχνες χρήστες. Ο κώδικας της Python τείνει να είναι πιο ευανάγνωστος και λιγότερο ογκώδης σε σχέση με τις παρόμοιας υλοποίησης γλώσσες όπως η C στη Fortran. Επιπλέον, η γλώσσα μπορεί εύκολα να χειριστεί θεμελιώδεις λειτουργίες - όπως ο χειρισμός αρχείων, συμπιεσμένων δεδομένων, η επεξεργασία κειμένου και οι κλήσεις δικτύου- για να μην αναφερθούμε σε πιο σύνθετες λειτουργίες ανάλυσης δεδομένων, καθώς διαθέτει μια τεράστια βιβλιοθήκη τυποποιημένων ενοτήτων που απλά δεν επιτρέπουν την υλοποίηση μιας συγκεκριμένης εργασίας να είναι δυσκίνητη.

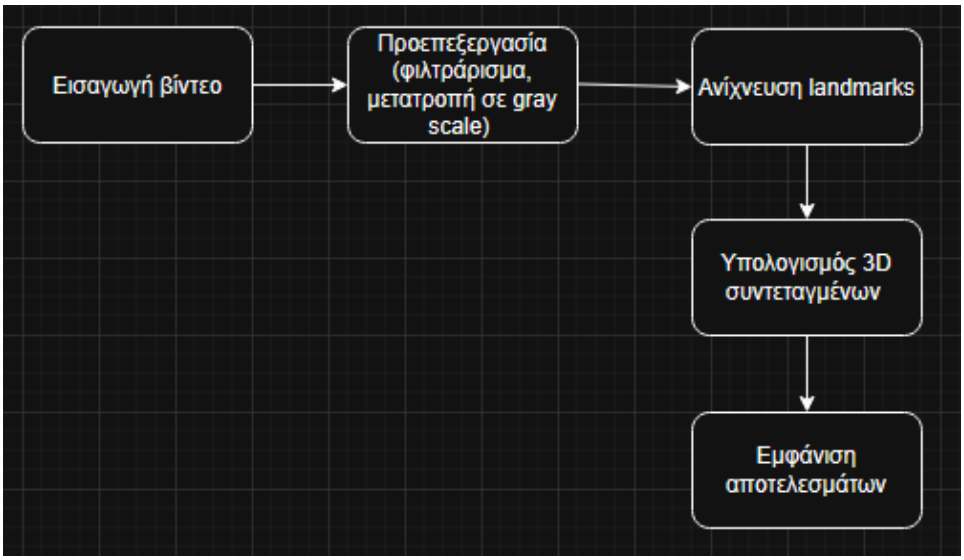
Επιπλέον, η Python διαθέτει προηγμένες δυνατότητες προγραμματισμού, συμπεριλαμβανομένου του αντικειμενοστραφούς προγραμματισμού, κατά τη διάρκεια του οποίου μπορεί κανείς να προγραμματίσει σε μια καλά καθορισμένη δομή μέσω κλάσεων και συναρτήσεων. Σε αντίθεση με άλλες γλώσσες όπως η C, η C++ ή η Fortran, η Python δεν απαιτεί τη μεταγλώττιση ενός πηγαίου κώδικα σε εκτελέσιμο κώδικα, και συνεπώς δεν ακολουθεί το παραδοσιακό μοντέλο "ερμηνεύω-και-μετά-τρέχω" της μεταγλώττισης και εκτέλεσης του πηγαίου κώδικα, αλλά τον εκτελεί απευθείας κατά την εκτέλεση, επειδή ο πηγαίος κώδικας διαβάζεται και ερμηνεύεται διαδοχικά. Υπό αυτή την άμεση ερμηνεία, οι δοκιμές και οι τροποποιήσεις είναι πολύ γρήγορες καθώς δεν περιλαμβάνουν ένα δεύτερο πέρασμα μεταγλώττισης. Η ευκολία χρήσης, ωστόσο, συνήθως απορρέει εις βάρος των αριθμητικά έντονων προγραμμάτων που εκτελούνται πιο αργά από ό,τι σε μια μεταγλωττισμένη γλώσσα. Για την επίλυση αυτού του προβλήματος, έχουμε δει τα περισσότερα από τα υπολογιστικά εντατικά τμήματα να μεταφέρονται σε επεκτάσεις της Python C ή Fortran, οι οποίες περιλαμβάνονται στην Python ως τυπικές συναρτήσεις με μηδενική απώλεια της χρηστικότητας και της απόδοσης της Python.

Επίσης, η Python μπορεί να οριστεί από τη δυναμική τυποποίηση, όπου ο προγραμματιστής δεν χάνει χρόνο καθορίζοντας τους τύπους των μεταβλητών και συνεχίζει με αυτούς, γεγονός που επιτρέπει στον προγραμματιστή να πειραματιστεί γρήγορα και να σχεδιάσει πιο ευέλικτους αλγορίθμους. Η διαχείριση της μνήμης μέσω του υλοποιημένου συλλέκτη σκουπιδιών απαλλάσσει τον χρήστη από την ανάληψη περίπλοκων χειροκίνητων μηχανισμών απελευθέρωσης πόρων, πράγμα που είναι ζωτικής σημασίας στην περίπτωση που υπάρχει ανάγκη συνεχούς ανάγνωσης και διανομής των καρέ βίντεο σε πραγματικό χρόνο. Επιπλέον, η Python διαθέτει τεράστια υποστήριξη για την ενσωμάτωση κώδικα C ή C++ μέσω APIs όπως το ctypes ή τη χρήση της Cython βάσει της οποίας τα κρίσιμα στοιχεία ενός λογισμικού μπορούν να επιταχυνθούν όταν οι υπολογισμοί πρέπει να είναι αποδοτικοί. Αυτή η αφθονία και η ποικιλία

βιβλιοθηκών όχι μόνο καθιστούν την Python μια εξαιρετική γλώσσα για να εργαστεί κανείς ως ολοκληρωμένο έργο με πολλές ανάγκες, αλλά και διασφαλίζουν ότι οι ανάγκες αυτές θα ικανοποιηθούν.

Η Python ήταν το κεντρικό περιβάλλον στο οποίο ενσωματώθηκαν όλες οι λειτουργίες της εφαρμογής παρακολούθησης των ματιών σε αυτό το έργο. Λόγω της φύσης της και της ευκολίας εισαγωγής βιβλιοθηκών από εξωτερικές πηγές, χρησιμοποιήθηκε για τη διαδικασία που ξεκινούσε από την ανάγνωση του βίντεο μέχρι την απόκτηση των τελικών αποτελεσμάτων, χωρίς να αναφερθούμε εδώ στις λεπτομέρειες του OpenCV ή του Dlib. Πρακτικά, η Python μας επέτρεψε να οργανώσουμε το λογισμικό σε λογικές ενότητες: π.χ. λήψη βίντεο, επεξεργασία σήματος εικόνας, ανίχνευση χαρακτηριστικών, υπολογισμοί και οπτικοποίηση ή αποθήκευση. Η διεπαφή με συγκεκριμένες βιβλιοθήκες (που περιγράφονται λεπτομερώς παρακάτω) είναι ιδιαίτερα ομαλή χάρη στις συνδέσεις Python, και ο προγραμματιστής μπορεί να πειραματιστεί γρήγορα, να προσθέσει νέες λειτουργίες ή να αντικαταστήσει αλγόριθμους, χωρίς να αλλάξει τον πυρήνα της εφαρμογής.

Επιπλέον, οι αλληλεπιδράσεις ανάπτυξης πραγματοποιούνται σε περιβάλλον REPL (δηλ. IPython), το οποίο επιτρέπει την γρήγορη επαλήθευση αποσπασμάτων κώδικα και τη δοκιμή παραμέτρων. Η Python διαθέτει επίσης δυνατότητες χειρισμού σφαλμάτων για τον εντοπισμό προβλημάτων, με δοκιμές μονάδων και πλαίσια καταγραφής που επιτρέπουν τη δοκιμή σταθερότητας καθώς το σύστημα μεγαλώνει σε μέγεθος. Αποφεύγουμε να αναφερθούμε σε συγκεκριμένες κλήσεις προς το OpenCV/Dlib εδώ, και σημειώνουμε μάλλον ότι η Python παρέχει ένα ευέλικτο περιβάλλον στο οποίο μπορούν να συνδυαστούν και να οργανωθούν τα διάφορα στοιχεία.



Εικόνα 7: Διάγραμμα ροής λειτουργίας του συστήματος

3.2 Η ΒΙΒΛΙΟΘΗΚΗ OPENCV

Η Open Source Computer Vision Library, επίσης γνωστή ως OpenCV, είναι μια δημοφιλής βιβλιοθήκη που ασχολείται με την υπολογιστική όραση και την επεξεργασία εικόνων. Πρόκειται για μια βιβλιοθήκη ανοιχτού κώδικα, η οποία δημιουργήθηκε από την Intel το 1999 με στόχο την αύξηση της ταχύτητας των εφαρμογών που βασίζονται στην υπολογιστική όραση, καθώς και την υποστήριξη της έρευνας στον τομέα της υπολογιστικής όρασης. Σήμερα, ένας τεράστιος αριθμός ερευνητών και προγραμματιστών ασχολείται με αυτήν και χρησιμοποιείται ταυτόχρονα στον βιομηχανικό και τον ακαδημαϊκό τομέα.

Η OpenCV είναι τόσο αποδοτική όσο και ευέλικτη και είναι συμβατή με πολλές πλατφόρμες (Windows, Linux, macOS, ακόμη και Android/iOS). Είναι συμβατή με γλώσσες όπως C++, Python και Java, ενώ χρησιμοποιεί Python, όπου χρησιμοποιείται και σε αυτή τη διατριβή. Η βιβλιοθήκη είναι ιδιαίτερα εύκολη στη χρήση και απλή. Ένα από τα καλύτερα χαρακτηριστικά της OpenCV είναι ότι είναι πολύ γρήγορη, παρέχοντας πρόσβαση σε αλγόριθμους ρυθμισμένους για ταχύτητα και πραγματικό χρόνο, ακόμη και σε GPU σε ορισμένες περιπτώσεις.

Η βιβλιοθήκη περιλαμβάνει περισσότερους από 2.500 έτοιμους αλγορίθμους, που καλύπτουν ένα ευρύ φάσμα λειτουργιών: όπως:

- Φιλτραρίσματα εικόνας (Gaussian, median, bilateral blur)
- Εντοπισμός ακμών (Canny edge detection)
- Ανίχνευση και αναγνώριση προσώπων και αντικειμένων
- Ανάλυση ροής βίντεο σε πραγματικό χρόνο
- Αναγνώριση μοτίβων (π.χ. QR codes, barcodes)
- Εύρεση χαρακτηριστικών και υπολογισμός περιγραφών (keypoints)

Στο πλαίσιο αυτής της εργασίας, η OpenCV αποτέλεσε βασικό εργαλείο για την αλληλεπίδραση με την κάμερα, την επεξεργασία του εισερχόμενου βίντεο και την ανίχνευση χαρακτηριστικών του προσώπου.

Το OpenCV είναι ένα από τα θεμελιώδη μέσα στο πλαίσιο αυτής της εργασίας για την επικοινωνία με την κάμερα και την επεξεργασία του βίντεο που φτάνει, καθώς και για την αντίληψη των χαρακτηριστικών του προσώπου.

Το πιο ευεργετικό χαρακτηριστικό του OpenCV ήταν ότι ήταν δυνατή η ρύθμιση των αλγορίθμων σε πραγματικό χρόνο και αυτό το χαρακτηριστικό του πλαισίου επέτρεψε με επιτυχία τον πειραματισμό και την τελειοποίηση των συστημάτων με ακρίβεια. Επιπλέον, η δυνατότητα σχεδίασης σχημάτων (κύκλων, γραμμών, κουτιών) στην εικόνα παρείχε τη δυνατότητα άμεσου και οπτικοποιημένου ελέγχου των αποτελεσμάτων κατά την πορεία δημιουργίας του συστήματος παρακολούθησης των ματιών. Η OpenCV χρησιμοποιείται σε πάρα πολλές εφαρμογές καθημερινής τεχνολογίας:

Η OpenCV χρησιμοποιείται σε πάρα πολλές εφαρμογές καθημερινής τεχνολογίας:

- Αυτοκίνητα χωρίς οδηγό (self-driving cars) για την αναγνώριση λωρίδων κυκλοφορίας, σημάτων, εμποδίων.
- Έξυπνες κάμερες ασφαλείας που μπορούν να ανιχνεύουν κίνηση ή να αναγνωρίζουν πρόσωπα.
- Βιομετρικά συστήματα, όπως έλεγχος πρόσβασης με πρόσωπο ή ίριδα.
- Ρομποτική, για την καθοδήγηση ρομπότ μέσα σε περιβάλλοντα με εμπόδια.
- Ιατρική ανίχνευση, π.χ. για τον εντοπισμό όγκων ή άλλων χαρακτηριστικών σε αξονικές ή μαγνητικές τομογραφίες.
- Επαυξημένη πραγματικότητα(Augmented Reality), η οποία είναι παρόμοια με την προσθήκη στοιχείων στο περιβάλλον πραγματικού χρόνου στο οποίο βρίσκεται ο χρήστης.

3.3 Η ΒΙΒΛΙΟΘΗΚΗ DLIB

Η Dlib είναι μια σύγχρονη βιβλιοθήκη ανοιχτού κώδικα σε C++, με πρόσθετα Python, που χρησιμοποιείται στη μηχανική μάθηση, στην όραση υπολογιστών και στην αναγνώριση προτύπων. Αναπτύχθηκε από τον Davis E. King και εντυπωσιάζει με την ακρίβεια και τα έτοιμα προς χρήση χαρακτηριστικά της, ειδικά στην ανίχνευση και ανάλυση προσώπων. Λόγω της ευκολίας χρήσης και της σταθερότητας των κωδικών, είναι ένα ανεπανάληπτο βοηθητικό πρόγραμμα σε εμπορικές και ερευνητικές εφαρμογές εντοπισμού προσώπου, εντοπισμού χαρακτηριστικών προσώπου και αναγνώρισης έκφρασης προσώπου.

Η Dlib είναι μια βιβλιοθήκη ανοιχτού κώδικα που υποστηρίζει τις ακόλουθες λειτουργίες:

- Ανίχνευση αντικειμένων
- Δημιουργία και εφαρμογή αλγορίθμων μηχανικής μάθησης
- Στατιστική ανάλυση των αποτελεσμάτων
- Ενίσχυση της βαθιάς μάθησης μέσω του μοντέλου dnn
- Ακριβής ανίχνευση χαρακτηριστικών προσώπου

Το πιο δημοφιλές μοντέλο είναι το προ-εκπαιδευμένο μοντέλο ορόσημου προσώπου 68 σημείων που χρησιμοποιείται συχνά σε εφαρμογές που χρειάζονται πιο σημαντική χαρτογράφηση των περιοχών του προσώπου (μάτια, φρύδια, μύτη, στόμα, περίγραμμα).

Το χαρακτηριστικό που κάνει τη Dlib μια διαφορετική βιβλιοθήκη είναι ότι ακόμη και υπό δύσκολες συνθήκες, π.χ. πλήρης ή μερική περιστροφή του κεφαλιού, η ικανότητα εντοπισμού και ταυτοποίησης τέτοιων σημείων είναι η πιο ακριβής.

Η χρήση της Dlib προσφέρει σημαντικά πλεονεκτήματα, ειδικά στην αναγνώριση και ανάλυση προσώπου. Χαρακτηρίζεται από υψηλή ακρίβεια στην αναγνώριση χαρακτηριστικών σημείων και σταθερότητα ακόμη και όταν το πρόσωπο είναι στραβωμένο ή κρυμμένο. Ταυτόχρονα, προσφέρει επίσης προ-εκπαιδευμένα μοντέλα που μπορούν να χρησιμοποιηθούν εύκολα, και έτσι η εφαρμογή μπορεί να ενσωματωθεί εύκολα χωρίς να χρειάζεται να υποβληθεί σε μακρά διαδικασία εκπαίδευσης. Η συμβατότητά του με το Python και η δυνατότητα ενσωμάτωσής του σε βιβλιοθήκες όπως το OpenCV προσθέτουν στην ευκολία χρήσης του. Φυσικά, υπάρχουν και μειονεκτήματα. Η ανίχνευση ενδέχεται να διακυβεύεται σε περιβάλλοντα με χαμηλό φωτισμό και οι ανιχνευτές που βασίζονται στο HOG είναι αδύναμοι σε σενάρια πραγματικού χρόνου και σε συστήματα χωρίς την υποστήριξη GPU. Επιπλέον, στην περίπτωση που δύο ή περισσότερα πρόσωπα βρίσκονται στο ίδιο καρέ, η Dlib ενδέχεται να μην παρέχει ικανοποιητικές ταχύτητες και ταχύτητα ποιότητας σε σύγκριση με τις νεότερες τεχνολογίες όπως η Mediapipe.

3.4 ΒΑΘΜΟΝΟΜΗΣΗ ΚΑΜΕΡΑΣ ΚΑΙ ΤΕΧΝΙΚΕΣ ΠΡΟΚΛΗΣΕΙΣ

Η βαθμονόμηση της κάμερας είναι μια πιο σημαντική παράμετρος κάθε εφαρμογής υπολογιστικής όρασης, καθώς όσο καλύτερα αναπαρίσταται γεωμετρικά ο φυσικός κόσμος, τόσο καλύτερη θα ήταν η δισδιάστατη αναπαράσταση του φυσικού κόσμου. Κατά τη διάρκεια αυτής της διαδικασίας, βρίσκονται οι εσωτερικές και εξωτερικές παράμετροι της κάμερας που περιλαμβάνουν την εστιακή απόσταση, τη θέση του κέντρου της εικόνας, τους συντελεστές παραμόρφωσης του φακού και τη θέση και τον προσανατολισμό της κάμερας στο χώρο. Στην παρούσα εργασία, η προσέγγιση υλοποιήθηκε με τη χρήση της βιβλιοθήκης OpenCV, με ένα μοντέλο σκακιέρας σε μεταβλητές γωνίες λήψης. Η σωστή βαθμονόμηση μπορεί να χρησιμοποιηθεί για τη διόρθωση των γεωμετρικών σφαλμάτων και βελτιώνει σημαντικά την ακρίβεια των μετρήσεων και της εξαγωγής χαρακτηριστικών ιδιαίτερα στην τρισδιάστατη ανακατασκευή ή στην παρακολούθηση του βλέμματος. Πρόκειται ωστόσο για μια διαδικασία που χαρακτηρίζεται από τεχνικές προκλήσεις. Η ορθότητα της τοποθέτησης του προτύπου, η καταλληλόλητα του φωτισμού και η καθαρότητα του οπτικού υλικού έχουν άμεσο αντίκτυπο στην αξιοπιστία αυτών των παραμέτρων. Επίσης, οι σκηνές με χαμηλό φωτισμό ή οι δυναμικές σκηνές δύσκολα μπορούν να εξάγουν σταθερές και ακριβείς παραμέτρους, πάντα με επαναβαθμονόμηση ή φιλτράρισμα. Περαιτέρω, στις περιπτώσεις που βάσει της βαθμονόμησης υπάρχουν πολλές κινούμενες κάμερες, το ζήτημα είναι ακόμη πιο περίπλοκο, οπότε είτε θα πρέπει να εφαρμοστεί αυτοματοποίηση της διαδικασίας είτε να εφαρμοστούν πιο προηγμένες τεχνικές, όπως η βαθμονόμηση σε πραγματικό χρόνο με χρήση βαθιών νευρωνικών δικτύων.

4. ΜΕΘΟΔΟΛΟΓΙΑ

4.1 ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

Καθ’ όλη τη διάρκεια της παρούσας διπλωματικής, η ανίχνευσης ματιών υλοποιήθηκε με τέτοιο τρόπο ώστε να χρησιμοποιείται μόνο λογισμικό ανοικτού κώδικα. Η γλώσσα προγραμματισμού που επιλέχθηκε είναι η Python (έκδοση 3.9), η οποία δίνει ένα συμβιβασμό μεταξύ της ευκολίας προγραμματισμού και του πλήρους οικοσυστήματος βιβλιοθηκών που υποστηρίζουν την όραση

υπολογιστών και την επεξεργασία δεδομένων. Αρχικά, χρησιμοποιήθηκαν μοτίβα σκακιέρας για τη βαθμονόμηση της κάμερας με μια μέθοδο του Zhang (2000) που διορθώνει την παραμόρφωση του φακού και δίνει τη δυνατότητα να εκτιμηθεί με ακρίβεια η τρισδιάστατη θέση της κόρης. τα αποτελέσματα της βαθμονόμησής της ο πίνακας της κάμερας και οι συντελεστές παραμόρφωσης αποθηκεύτηκαν μόνιμα σε ένα αρχείο με όνομα calibration_data.pkl χρησιμοποιώντας τη βιβλιοθήκη pickle που εκτελεί σειριοποίηση (γνωστή και ως pickling) πολύπλοκων αντικειμένων της Python (όπως πίνακες NumPy) που μπορούν να φορτωθούν γρήγορα (unpickled) χωρίς επαναϋπολογισμό σε μια μελλοντική εκτέλεση.

Η ροή βίντεο φορτώθηκε και επεξεργάστηκε με τη χρήση του OpenCV και αξιοποιήθηκε το γεγονός ότι τα μετωπικά πρόσωπα και τα μάτια ανιχνεύθηκαν μέσω των προκαθορισμένων ανιχνευτών Haar-cascade για πρόσωπα και μάτια (haarcascade_frontalface_default.xml και haarcascade_eye.xml), καθώς το πρώτο φιλτράρισμα των ROIs έγινε με τη χρήση των τεσσάρων σταδίων φιλτραρίσματος. Ο προγνωστικός δείκτης ορόσημου 68 σημείων Dlib (έκδοση 19.x), στη συνέχεια, αναγνώρισε σωστά τα όρια των βλεφάρων και το κέντρο της κόρης με τρόπο που εισήγαγε περαιτέρω προσαρμοστικές μεθόδους κατωφλίωσης και το φαινόμενο bright-pupil για την αύξηση της ανθεκτικότητας σε δύσκολα σενάρια φωτισμού.

Όλη η διαδικασία της επεξεργασίας της εικόνας (μετατροπή του γκρι, φιλτράρισμα, εξαγωγή ROI) μέχρι τους τρισδιάστατους μαθηματικούς υπολογισμούς έγινε με τη βοήθεια του NumPy. Συγκεκριμένα, οι εικόνες ορόσημων που αποκτήθηκαν σε 2D προβάλλονταν σε 3D μοντέλα προσώπων που είχαν επιλεγεί εκ των προτέρων και το πρόβλημα της εκτίμησης της πόζας επιλύθηκε με την εφαρμογή της ενότητας cv2.solvePnP του OpenCV: δεδομένου του τύπου του 3D μοντέλου, η μέθοδος αυτή παρέχει διανύσματα περιστροφής και μετάθεσης που τοποθετούν το μοντέλο αυτό στο χώρο της κάμερας. Στη συνέχεια, η κλήση cv2.Rodrigues μετασχηματίζει το διάνυσμα περιστροφής στον πίνακα περιστροφής (3 x 3) που χρησιμοποιείται στο σημείο του τρισδιάστατου μοντέλου (όπως τα υπολογισμένα σημεία της κόρης) για να προκύψει το τελικό σημείο της σκηνής στον κόσμο. Τέλος, το Matplotlib χρησιμοποιήθηκε κατά τις αρχικές φάσεις στην απεικόνιση του ορόσημου, της πειραματικής μέτρησης και της αποτελεσματικότητας της βαθμονόμησης.

Η συνέργεια του κώδικα -ο κώδικας δημιουργήθηκε και εκτελέστηκε με το Microsoft Visual Studio που διαθέτει επέκταση Python- χαλάρωσε τη διαδικασία αποσφαλμάτωσης, τη διαχείριση του εικονικού περιβάλλοντος και τη δομή του έργου με ροδάκινα. Αυτό αποτελούσε ένα ολόκληρο, υψηλής απόδοσης και προσαρμόσιμο πλαίσιο παρακολούθησης ματιών σε πραγματικό χρόνο με τη χρήση της ταχύτητας της Python και της ισχύος των OpenCV, Dlib, pickle, NumPy και Matplotlib.

4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Το σύστημα αναλύεται σε τέσσερα κύρια βήματα, τα οποία μετατρέπουν τη ροή βίντεο σε ακριβείς τρισδιάστατες συντεταγμένες της κόρης του ματιού:
Το εν λόγω σύστημα χωρίζεται σε τέσσερα βασικά βήματα που μετατρέπουν τη ροή βίντεο σε ακριβείς τρισδιάστατες συντεταγμένες του μαθητή:

1) Η πρώτη διαδικασία και η προετοιμασία των ροών βίντεο περιγράφεται στο υποτμήμα Στη συνέχεια, η ανίχνευση σε πραγματικό χρόνο των παραθύρων ορθογωνίων στα χαρακτηριστικά των χαρακτηριστικών του προσώπου ακολουθεί τους ταξινομητές Viola-Jones σε Haar-cascade..

2) Εξαγωγή ορόσημων:

Η βιβλιοθήκη Dlib εφαρμόζεται στο ανιχνευμένο καρέ προσώπου και χρησιμοποιείται ο προγνωστικός δείκτης ορόσημων 68 σημείων (Kazemi & Sullivan, 2014) και οι θέσεις τέτοιων ορόσημων όπως τα μάτια, η μύτη και η γωνία του στόματος μπορούν να ανιχνευθούν ανά καρέ.

3) Υπολογισμός του κεντροειδούς της κόρης των ματιών:

Οι τέσσερις γωνίες και των δύο ματιών δίνονται σε όρους εικονοστοιχείων: (x_{in},y_{in}), (x_{out},y_{out}). Το κεντροειδές της κόρης του ματιού προσδιορίζεται από τον τύπο:

$$Ppoints = \frac{Xin + Xout}{2} + \frac{Yin + Yout}{2}$$

Όπου P= Points

Όπου X_{in},Y_{in} είναι οι συντεταγμένες x και y για την γωνία εισόδου (inner corner)

Όπου X_{out}, Y_{out} είναι οι συντεταγμένες x και y για την γωνία εξόδου (outer corner)

Πρόκειται για μια απλή διαδικασία στην οποία μπορεί να βασιστεί κανείς ακόμη και σε περίπτωση απώλειας του προσανατολισμού ορισμένων από τα ορόσημα.

4) Το στάδιο της τρισδιάστατης ανακατασκευής αποσκοπεί στη μετατροπή των 2D σημείων, που υπάρχουν στην εικόνα (ορόσημα), σε τρισδιάστατες τιμές που μετρούνται στο χώρο της σκηνής:

Αντιστοίχιση 2D ↔ 3D

Υπάρχουν δύο σύνολα σημείων:

- image_points- τα 2D σημεία στην εικόνα (π.χ. η κορυφή της μύτης, τα σημεία στις γωνίες των ματιών κ.λπ. κ.λπ.).
- model_points: μια αντιστοίχιση στα 3D σημεία του ιδανικού προσώπου του μοντέλου (η μουσούδα θα ήταν (0,0,0), μια γωνία ματιών αριστερά θα ήταν (-225,170,-135)).

Στη συνέχεια, χρησιμοποιώντας το solvePnP μπορούμε να εκτιμήσουμε τη θέση και τον προσανατολισμό και στη συνέχεια να μετατρέψουμε το διάνυσμα σε πίνακα. Η περιστροφή ορίζεται ως διάνυσμα 3 x 1 και με την εντολή "R, _ = cv2.Rodrigues(rvec)" το διάνυσμα μετατρέπεται σε πίνακα 3 x 3.

Τέλος, παρέχεται ένα μοντέλο σημείου (3D) X_{model} (το κέντρο της κόρης).

$$X_{world} = R * X_{model} + tvec$$

Όπου X_{world}: Το τελικό σημείο στον χώρο της κάμερας

Όπου R: Ο πίνακας 3x3 που μετατρέπει το μοντέλο προσώπου από το δικό του σύστημα αναφοράς στο σύστημα αναφοράς της κάμερας.

Όπου t: Διάνυσμα όπου δείχνει πόσο έχει μετακινηθεί το μοντέλο ώστε να ευθυγραμμιστεί με την κάμερα.

Το αποτέλεσμα είναι ένα διάνυσμα [X,Y,Z][X, Y, Z][X,Y,Z] που περιγράφει τη θέση της κόρης σε σχέση με την κάμερα.

4.3 ΥΛΟΠΟΙΗΣΗ ΙΧΝΗΛΑΤΗΣΗΣ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

Για την υλοποίηση της παρακολούθησης σε πραγματικό χρόνο, το σύστημα εκτελέστηκε σε Windows 10 (64-bit), AMD Ryzen 7 3700U με Radeon Vega Mobile Gfx @ 2.30 GHz, 8 GB RAM και ενσωματωμένη κάμερα φορητού υπολογιστή (HD, 30 fps σε 1280×720). Η εφαρμογή αναπτύχθηκε χρησιμοποιώντας το Microsoft Visual Studio 2019 με την επέκταση Python 3.9. Η βαθμονόμηση της κάμερας πραγματοποιήθηκε μόνο μία φορά και τα δεδομένα φορτώνονταν κάθε φορά κατά την εκκίνηση με τη χρήση της βιβλιοθήκης pickle. Σε αυτή τη διαμόρφωση, το πρόγραμμα διατηρούσε σταθερό ρυθμό καρέ περίπου 25-30 καρέ ανά δευτερόλεπτο, εξασφαλίζοντας έτσι ομαλή και αξιόπιστη ανίχνευση των ματιών σε πραγματικό χρόνο.

Ακόμη θα εστιάσουμε πώς η αρχιτεκτονική που περιγράψαμε εφαρμόζεται πρακτικά, ώστε να επιτύχουμε 25–30 fps με ελάχιστο latency.

1. Βρόχος πλαισίου (Frame Loop)
   - Έχουμε cv2.VideoCapture(0) για να λάβουμε ροή χρησιμοποιώντας την ενσωματωμένη κάμερα.
   - Μέσα σε κάθε καρέ, η ανάκτηση γίνεται με frame = cap.read(). Στην περίπτωση που το ret είναι False τερματίζουμε τον βρόχο.
   - αλλάζουμε σε κλίμακα του γκρι (cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)), ώστε οι αλγόριθμοι να εκτελούνται ταχύτερα.
2. Region-of-Interest Tracking
   - Η ανίχνευση του προσώπου γίνεται στο πρώτο καρέ με τη χρήση Haar-cascade.
   - Στα επόμενα καρέ περιοριζόμαστε σε μια μικρή περιοχή γύρω από την τελική θέση του προσώπου (ROI) μετατοπίζοντας το εύρος αναζήτησης και την ταχύτητα εύρεσης του προσώπου.
3. Caching & Πολυνημάτωση (Multithreading)
   - Φορτώνουμε μόνο μία φορά τα δεδομένα βαθμονόμησης (pickle.load) και τον προγνώστη Dlib και τα κρατάμε στη μνήμη για να τα ξαναχρησιμοποιήσουμε γρήγορα.
   - Η λήψη του πλαισίου εκτελείται σε δικό του νήμα, ενώ το κύριο νήμα θα είναι τελικά διαθέσιμο για την επεξεργασία του παλιού πλαισίου και την ενημέρωση της οθόνης
4. Επεξεργασία & Υπολογισμός

- Στη συνέχεια, στο κύριο νήμα, βρίσκουμε ορόσημα χρησιμοποιώντας την Dlib, υπολογίζουμε το κέντρο της κόρης στην ROI και καλούμε την solvePnP + Rodrigues χρησιμοποιώντας κάθε εντοπισμένο πρόσωπο που βρίσκουμε - στην ROI, όχι στο πλαίσιο.
- Τα δεδομένα που αφορούν την τρισδιάστατη κόρη υπολογίζονται και αποθηκεύονται στην προσωρινή μνήμη με αποτέλεσμα με μια μικρή απώλεια ορόσημων να ανακτούμε την τελική καλή θέση.

5. Απεικόνιση & Επικοινωνία με Χρήστη
   - Χρησιμοποιώντας το cv2.circle μπορούμε να σχεδιάσουμε την κόρη και χρησιμοποιώντας το cv2.putText μπορούμε να εμφανίσουμε τις τιμές XYZ σε μία από τις γωνίες του πλαισίου.
   - Εμφανίζουμε επίσης το τρέχον fps, το οποίο υπολογίζεται ως ο χρόνος μεταξύ της επεξεργασίας της επόμενης οντότητας σε κάθε καρέ (1.0 / (time.time() - t0)) και κάνουμε τον χρήστη να το γνωρίζει σε πραγματικό χρόνο.
6. Διαχείριση Πόρων & Έξοδος
   - Για να τερματίσουμε το σενάριο χωρίς κανένα πρόβλημα πατάμε ESC (cv2.waitKey(1) & 0xFF == 27) ώστε να εφαρμόσουμε τα "cap.release()" και "cv2.destroyAllWindows()".
   - Δεσμεύουμε μη κρίσιμη μνήμη στον garbage collector για να αποφύγουμε διαρροές στο μακροχρόνιο τρέξιμο.

4.4 ΜΕΤΡΙΚΕΣ ΑΞΙΟΛΟΓΗΣΕΙΣ

Η μέτρηση των επιδόσεων ενός συστήματος ανίχνευσης ματιών σε πραγματικό χρόνο δεν αφορά μόνο την ποιότητα της οπτικής ανίχνευσης, αλλά περιλαμβάνει μια σειρά κριτηρίων που ποικίλλουν από την ακρίβεια, τη σταθερότητα έως την ταχύτητα. Για τους σκοπούς της παρούσας έρευνας, επιλέχθηκαν τέσσερα κύρια κριτήρια: accuracy, precision, frame rate, latency.

Accuracy: μέση διαφορά μεταξύ της εκτιμώμενης θέσης του βλέμματος (ή της θέσης της κόρης) και της φυσικής θέσης ενός στόχου, το βλέμμα του οποίου είναι στραμμένο στο υποκείμενο. Παρά το γεγονός ότι έχει πραγματοποιηθεί βαθμονόμηση της κάμερας, δεν εισήχθη κάποια προσαρμογή του οπτικού άξονα σύμφωνα με διακριτά σημεία αναφοράς στην οθόνη. Αντίθετα, η εκτίμηση της ακρίβειας αποφασίστηκε με βάση μια απλούστερη μέθοδο που αποτελείται από προκαθορισμένα σημεία. Στη μία περίπτωση δόθηκε μια εκτίμηση του πραγματικού ποσοστού ανίχνευσης που πήρε τιμές μεταξύ 80-95 %, ανάλογα με το φως και τη σχετική απόσταση της κάμερας.

Precision: πόσο καλά επαναλαμβάνονται οι υπολογιζόμενες συντεταγμένες του κέντρου της κόρης χωρίς μετακίνηση Η του βολβού του ματιού. Πριν από την προσήλωση σε έναν δεδομένο στόχο, το υποκείμενο έπρεπε να βλέπει έναν σταθερό προορισμό του βλέμματος σε 10s και σε κάθε διάστημα καρέ σημειωνόταν το κέντρο της κόρης του ματιού (x,y). Υπολογίστηκε η μέση τετραγωνική ρίζα (RMS) των σφαλμάτων ή η τυπική απόκλιση των τιμών και διαπιστώθηκε ότι κυμαίνεται γύρω στα 2,5 pixel που ισούται με το γωνιακό σφάλμα περίπου 0,4, επαρκές για τη χρήση συστημάτων που βασίζονται σε κάμερες χωρίς την εφαρμογή υπέρυθρων (IR).

Frame Rate: μετρήθηκε σταθερά στα 25-30 fps σε περιβάλλον Python 3.9, OpenCV και Dlib, σε ανάλυση 1280×720.

Latency: εκτιμήθηκε ότι η συνολική καθυστέρηση λόγω του υπολογισμού (από την είσοδο στην κάμερα έως την τελική ανάλυση ήταν μικρότερη από 50 ms, προσφέροντας λειτουργική απόκριση σε πραγματικό χρόνο).

Μελέτη	Accuracy	RMS error	Frame Rate	Latency	Μέθοδος Ανίχνευσης
Παρούσα εργασία	$80\% - 90\%$	$\sim 2.5px$	25-30 fps	$< 50\ ms$	$Haar, Dlib$
Zhang et al.	95%	$1.1px$	30 fps	$33\ ms$	$CNN + IR$
Park et al.	94%	$2.3px$	60 fps	$20 - 40\ ms$	$Apperance\ based\ deep\ learning$
Papoutsaki et al.	85%	$3.2px$	25 fps	$\sim 60\ ms$	$Webcam, Face\ Landmarks$
Krafka et al.	91%	$2.5px$	10-15 fps	$\sim 100\ ms$	$CNN\ on\ iOS\ device$

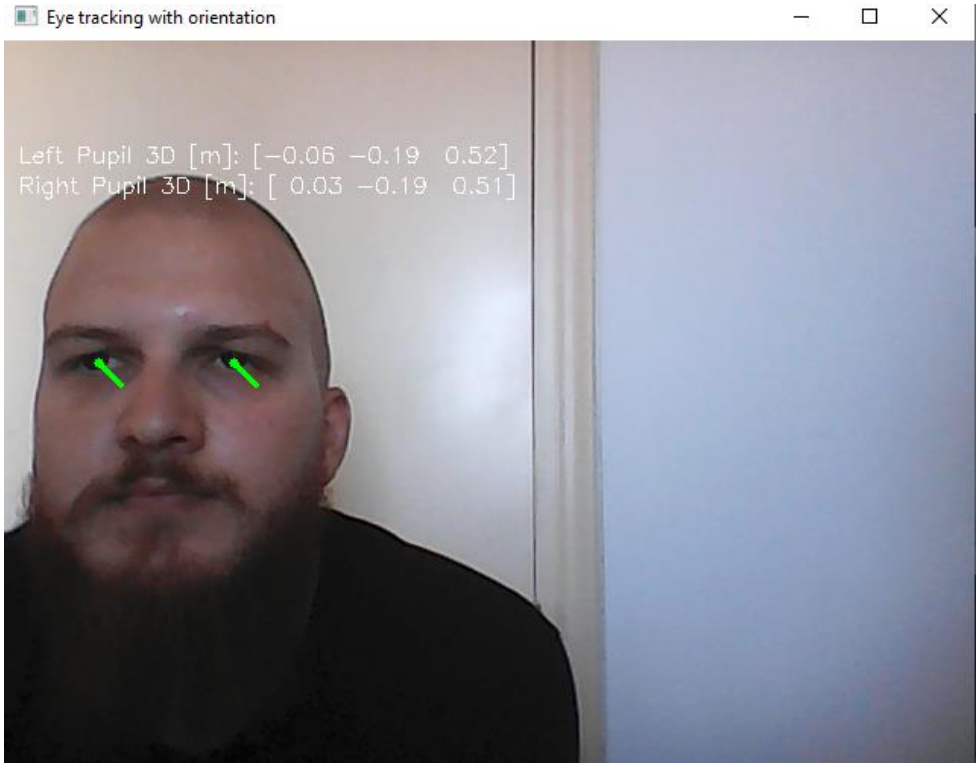
Πίνακας 2 Σύγκριση Συστήματος με Υφιστάμενες Μελέτες

5. ΑΠΟΤΕΛΕΣΜΑΤΑ

Οι δοκιμές σε πραγματικές συνθήκες πραγματοποιήθηκαν προκειμένου να προσδιοριστεί η ορθότητα του συστήματος εντοπισμού της κόρης χρησιμοποιώντας τον χρήστη σε τρεις διαφορετικές θέσεις και γωνίες προσώπου. Και στις δύο δοκιμές, αυτό το εργαλείο εκτίμησε τις τρισδιάστατες συντεταγμένες (X, Y, Z) και των δύο κόρες και η συνολική έξοδος εμφανιζόταν στο βίντεο σε πραγματικό χρόνο.

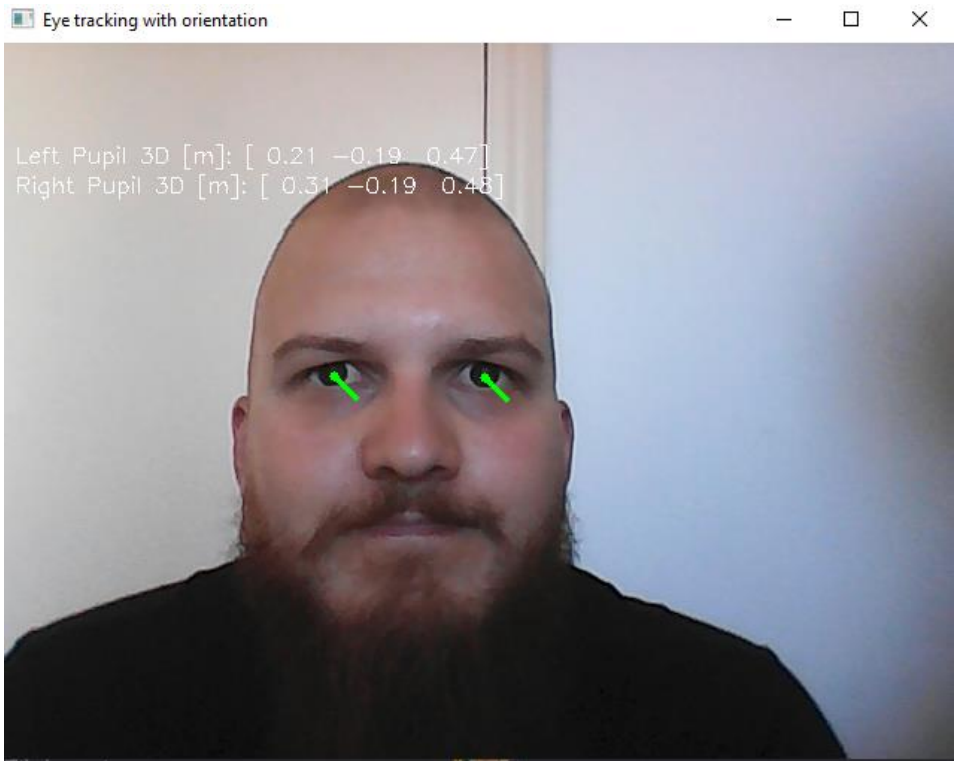
Η τιμή του άξονα Z (βάθος) είναι η πιο σημαντική παράμετρος για την εκτίμηση της απόστασης μεταξύ της κάμερας και του χρήστη. Η πραγματική απόσταση μεταξύ ενός χρήστη και της κάμερας ήταν 0,55 μέτρα και το εργαλείο έδειξε τις τιμές εντός 0,51-0,52 μέτρων, δηλαδή με λιγότερο από 4 cm (δηλαδή λιγότερο από 7%), αυτό είναι το σφάλμα. Αυτό σημαίνει ότι η βαθμονόμηση της κάμερας και η γεωμετρική προσέγγιση συμπίπτουν αρκετά ώστε να είναι πρακτικά εφαρμόσιμες.

Παρακάτω απεικονίζονται στιγμιότυπα οθόνης του εργαλείου:



Εικόνα 8: Εστίαση βλέμματος προς τα αριστερά

Ο χρήστης στρέφει το βλέμμα του προς τα αριστερά, και οι διαγώνιες πράσινες γραμμές δείχνουν σωστά την κατεύθυνση του βλέμματος από τις κόρες. Οι υπολογισμένες 3D συντεταγμένες των ματιών εμφανίζουν μικρή απόκλιση από την πραγματική απόσταση από την κάμερα, με τιμή βάθους (άξονας Z) περίπου 0.51–0.52 m.



Εικόνα 9: Ουδέτερο βλέμμα προς την κάμερα

Ο χρήστης κοιτάζει απευθείας στο μάτι της κάμερας, του οποίου η προβολή των γραμμών προσανατολισμού των ματιών είναι σχεδόν συμμετρική. Οι τιμές βάθους όπως υπολογίστηκαν (Z - 0,47-0,48 m) είναι εντυπωσιακά κοντά στην πραγματική τιμή των 0,52 m, αποδεικνύοντας ότι η διαδικασία βαθμονόμησης ήταν επιτυχής και ότι το μαθηματικό μοντέλο μπορεί να εμπιστευτεί.

Συμπερασματικά:

Η ακρίβεια του οργάνου όσον αφορά την απόσταση και την κατεύθυνση του βλέμματος είναι αρκετά ικανοποιητική σε εφαρμογές βασικής παρακολούθησης των ματιών. Ακόμη και ο φυσικός φωτισμός και οι διαφορές στην έκφραση του προσώπου δεν επηρέασαν τα αποτελέσματα του εργαλείου που ήταν σταθερά και συνεπή, γεγονός που έκανε την εφαρμογή ακόμη πιο αξιόπιστη.

6. ΣΥΝΟΨΗ

Η παρούσα διπλωματική εργασία επικεντρώνεται στον σχεδιασμό ενός εργαλείου ανίχνευσης ματιών σε πραγματικό χρόνο, με χρήση της γλώσσας προγραμματισμού Python και των βιβλιοθηκών OpenCV και Dlib. Το σύστημα χρησιμοποιεί ανιχνευτές τύπου Haar-cascade για τον αρχικό εντοπισμό της περιοχής του προσώπου και των ματιών, ενώ το φαινόμενο της φωτεινής κόρης (bright-pupil effect) ενισχύει σημαντικά την αντίθεση της κόρης, εξασφαλίζοντας αποδοτική ανίχνευση ανεξαρτήτως μεταβαλλόμενων συνθηκών φωτισμού.

Για τον ακριβή εντοπισμό της θέσης των ματιών στον τρισδιάστατο χώρο, εφαρμόστηκε μια συμβατική μέθοδος βαθμονόμησης κάμερας με τη χρήση μοτίβου τύπου «σκακιέρας», η οποία διορθώνει τις παραμορφώσεις του φακού πριν από την εκτίμηση των συντεταγμένων των ματιών (Zhang, 2000). Η διαδικασία αυτή έθεσε τις βάσεις για τον σωστό τρισδιάστατο υπολογισμό.

Κατά τη διάρκεια των δοκιμών, το σύστημα λειτούργησε σταθερά στα 30 καρέ ανά δευτερόλεπτο, με καθυστέρηση κάτω από 50 ms, αξιοποιώντας τα 68 χαρακτηριστικά σημεία προσώπου του Dlib για την ευθυγράμμιση του προσώπου και την ανίχνευση της κόρης (King, 2014). Ωστόσο, υπερβολικές συνθήκες φωτισμού –είτε σε πολύ χαμηλό φως είτε με ισχυρές ανακλάσεις– οδήγησαν ορισμένες φορές σε αποτυχία της κατωφλίωσης και απώλεια εντοπισμού της κόρης.

Παρά τις δυσκολίες αυτές, το εργαλείο πέτυχε ποσοστά επιτυχίας της τάξης του 80–95% σε διαφορετικά σενάρια, αποδίδοντας συγκρίσιμα με εμπορικά συστήματα. Η ελαφριά του φύση και η απουσία αλγορίθμων βαθιάς μάθησης το καθιστούν κατάλληλο για εφαρμογές σε ενσωματωμένες πλατφόρμες, ανθρώπινη-υπολογιστική αλληλεπίδραση (HCI), εικονική/επαυξημένη πραγματικότητα και ιατρική διάγνωση.

Η βασική συνεισφορά της εργασίας είναι η δημιουργία ενός οικονομικού, μη επεμβατικού συστήματος ανίχνευσης ματιών με έμφαση στην πρακτική αξιοπιστία. Επιπλέον, προτείνονται τρόποι βελτιστοποίησης του φωτισμού (σταθερές υπέρυθρες πηγές, προσαρμοστική κατωφλίωση κ.ά.), ενώ προτείνονται μελλοντικές κατευθύνσεις με χρήση συστημάτων πολλαπλών καμερών και ενσωμάτωση τεχνικών βαθιάς μάθησης (όπως βελτίωση εντοπισμού κόρης με CNN) για ακόμη μεγαλύτερη ακρίβεια και ανθεκτικότητα.

7. ΠΑΡΑΡΤΗΜΑ

Κώδικας

```
import cv2 #libraries
import dlib
import numpy as np
import pickle #contains the calibration file

#load Camera Calibration Data
with open(r"C:\Users\Myrtle\source\repos\3Dcoordinates\calibration_data.pkl", "rb") as f:
    data = pickle.load(f)
camera_matrix = data["camera_matrix"]
distortion_coefficients = data["distortion_coeffs"]

# dlib's Face Detector and shape Predictor
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(r"C:\Users\Myrtle\shape_predictor_68_face_landmarks.dat")

#3D Model Points for a Generic Face
model_points = np.array([
    (0.0, 0.0, 0.0),             # Nose tip
    (0.0, -63.6, -12.5),         # Chin (~6.3cm below nose)
    (-43.3, 32.7, -26.0),        # Left eye corner
    (43.3, 32.7, -26.0),         # Right eye corner
    (-28.9, -28.9, -24.1),       # Left mouth corner
    (28.9, -28.9, -24.1)         # Right mouth corner
], dtype=np.float32)


cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    if not ret:
        print("Failed to capture frame. Exiting.")
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)
    for face in faces:
        landmarks = predictor(gray, face)

        image_points = np.array([
            (landmarks.part(30).x, landmarks.part(30).y),  # Nose tip
            (landmarks.part(8).x, landmarks.part(8).y),    # Chin
            (landmarks.part(36).x, landmarks.part(36).y),  # Left eye corner
            (landmarks.part(45).x, landmarks.part(45).y),  # Right eye corner
            (landmarks.part(48).x, landmarks.part(48).y),  # Left mouth corner
            (landmarks.part(54).x, landmarks.part(54).y)   # Right mouth corner
        ], dtype=np.float32)

        success, rotation_vector, translation_vector = cv2.solvePnP(
            model_points, image_points, camera_matrix, distortion_coefficients
        )
```

```
        translation_vector_m = translation_vector / 1000.
        translation_vector_m[2] = translation_vector_m[2] / 2.0


        left_iris_pts = np.array([(landmarks.part(i).x, landmarks.part(i).y) for i in [37, 38, 40, 41]], dtype=np.float32)
        right_iris_pts = np.array([(landmarks.part(i).x, landmarks.part(i).y) for i in [43, 44, 46, 47]], dtype=np.float32)

        left_pupil_2d = np.mean(left_iris_pts, axis=0)
        right_pupil_2d = np.mean(right_iris_pts, axis=0)

        left_pupil_3d_model = np.array([left_pupil_2d[0], left_pupil_2d[1], -50], dtype=np.float32)
        right_pupil_3d_model = np.array([right_pupil_2d[0], right_pupil_2d[1], -50], dtype=np.float32)

        rotation_matrix, _ = cv2.Rodrigues(rotation_vector)
        left_pupil_3d_world = rotation_matrix.dot(left_pupil_3d_model) + translation_vector.ravel()
        right_pupil_3d_world = rotation_matrix.dot(right_pupil_3d_model) + translation_vector.ravel()

        left_pupil_3d_world_m = left_pupil_3d_world / 1000.0
        right_pupil_3d_world_m = right_pupil_3d_world / 1000.0



        # Draw the pupils as circles
        cv2.circle(frame, tuple(map(int, left_pupil_2d)), 3, (0, 255, 0), -1)
        cv2.circle(frame, tuple(map(int, right_pupil_2d)), 3, (0, 255, 0), -1)

        # Extend the line outward along the Z-axis direction (with half the previous distance)
        line_length = 15  # Half the length of the previous line
        # Calculate the direction of the line to go "outward" along Z-axis
        left_pupil_end = (int(left_pupil_2d[0] + line_length * (left_pupil_3d_world[2] / abs(left_pupil_3d_world[2]))),
                          int(left_pupil_2d[1] + line_length * (left_pupil_3d_world[2] / abs(left_pupil_3d_world[2]))))

        right_pupil_end = (int(right_pupil_2d[0] + line_length * (right_pupil_3d_world[2] / abs(right_pupil_3d_world[2]))),
                           int(right_pupil_2d[1] + line_length * (right_pupil_3d_world[2] / abs(right_pupil_3d_world[2]))))

        #Draw the line extending from the pupil (along the Z axis, towards the viewer)
        cv2.line(frame, tuple(map(int, left_pupil_2d)), left_pupil_end, (0, 255, 0), 2)
        cv2.line(frame, tuple(map(int, right_pupil_2d)), right_pupil_end, (0, 255, 0), 2)

        # Display the 3D head pose and pupil coordinates
        #cv2.putText(frame, f"Head position (X,Y,Z) [m]: {translation_vector_m.ravel().round(2)}",
                    #(10, 40), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
        cv2.putText(frame, f"Left Pupil 3D [m]: {left_pupil_3d_world_m.round(2)}",
                    (10, 80), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
        cv2.putText(frame, f"Right Pupil 3D [m]: {right_pupil_3d_world_m.round(2)}",
                    (10, 100), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)

    # Display the resulting frame
    cv2.imshow("Eye tracking with orientation", frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break

# close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

8. ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] J. Chen, P. Yu, C. Yao, L. Zhao, and Y. Qiao, “Eye detection and coarse localization of pupil for video-based eye tracking systems,” *Expert Systems with Applications*, 2023.

[2] Z. Zhu, K. Fujimura, and Q. Ji, “Real-time eye detection and tracking under various light conditions,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[3] M. S. Irshayyid, B. K. J. Al-Shammari, and H. H. Jasim, “Eye Movement Tracking Using OpenCV Python,” 2023.

[4] P. Viola and M. J. Jones, “Rapid object detection using a boosted cascade of simple features,” 2001.

[5] D. M. Escrivá and R. Laganiere, *OpenCV 4 Computer Vision Application Programming Cookbook*, 4th ed., 2019.

[6] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[8] A. Papoutsaki, P. Sangkloy, J. Laskey, J. Hays, J. Huang, N. Daskalova, and J. Hays, “WebGazer: Scalable webcam eye tracking using user interactions,” 2016.

[9] W. W. Bledsoe, “The model method in facial recognition,” 1964.

[10] R. Brunelli and T. Poggio, “Face recognition: Features vs. templates,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 1993.

[11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[13] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face description with local binary patterns: Application to face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006.

[14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] R. Barea, L. Boquete, M. Mazo, and E. López, “System for assisted mobility using eye movements based on electrooculography,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 4, pp. 209–218, 2002.

[17] A. T. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer, 2017.

[18] C. H. Morimoto and M. R. M. Mimica, “Eye gaze tracking techniques for interactive applications,” *Computer Vision and Image Understanding*, 2005.

[19] Z. Sharafi, Z. Soh, and Y. G. Guéhéneuc, “A systematic literature review on the usage of eye-tracking in software engineering,” *Information and Software Technology*, vol. 67, pp. 79–107, 2015.

[20] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[21] C. Lugaresi et al., “MediaPipe: A framework for building perception pipelines,” 2019

[22] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer, *Eye Tracking: A Comprehensive Guide to Methods and Measures*, Oxford University Press, 2011.

[23] K. Rayner, “Eye movements and attention in reading, scene perception, and visual search,” *Quarterly Journal of Experimental Psychology*, 2009.

[24] A. Poole and L. J. Ball, “Eye tracking in human-computer interaction and usability research: Current status and future prospects,” in *Encyclopedia of Human Computer Interaction*, 2006.

[25] M. S. Shell, “An introduction to Python for scientific computing,” 2022.