Πανεπιστήμιο Θεσσαλίας
Βόλος, Φεβρουάριος, 2020

Gaze Estimation Algorithms using low cost Cameras

Αλγόριθμοι Αναγνώρισης Κατεύθυνσης Βλέμματος με Χρήση Καμερών Χαμηλού Κόστους

Χρήστος Αξελός

Επιβλέπων: Γεράσιμος Ποταμιάνος

Μέλη επιτροπής: Νικόλαος Μπέλλας, Αντώνιος Αργυρίου

Διπλωματική εργασία

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πανεπιστήμιο Θεσσαλίας

Βόλος, Ελλάδα

Μια διατριβή που υποβλήθηκε για την εκπλήρωση των απαιτήσεων της Διπλωματικής Εργασίας για τη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Δήλωση Συγγραφής

Εγώ, ο Χρήστος Αξελός, δηλώνω πως αυτή η διπλωματική εργασία, με τίτλο "Αλγόριθμοι αναγνώρισης κατεύθυνσης βλέμματος με χρήση καμερών χαμηλού κόστους" και το έργο που παρουσιάζεται σε αυτήν είναι δικό μου. Βεβαιώνω ότι:

- Αυτό το έργο έγινε εξ ολοκλήρου ή κυρίως κατά την υποψηφιότητα για πτυχίο σε αυτό το Πανεπιστήμιο.
- Όπου οποιοδήποτε μέρος της παρούσας διπλωματικής έχει προηγουμένως υποβληθεί για πτυχίο ή οποιοδήποτε άλλο προσόν σε αυτό το Πανεπιστήμιο ή σε οποιοδήποτε άλλο ίδρυμα, αυτό έχει δηλωθεί με σαφήνεια.
- Οπου έχω συμβουλευθεί τη δημοσιευμένη δουλειά των άλλων, αυτό πάντα αποδίδεται σαφώς.
- Οπου έχω αναφέρει το έργο άλλων, η πηγή δίνεται πάντοτε. Με εξαίρεση τέτοιες παραθέσεις, αυτή η διπλωματική είναι εξ ολοκλήρου δική μου δουλειά.
- Εχω αναγνωρίσει όλες τις κύριες πηγές βοήθειας.
- Οπου η εργασία βασίζεται σε δουλειά που πραγματοποίησα από κοινού με άλλους, έχω καταστήσει σαφές τι ακριβώς έπραξαν οι άλλοι και αυτό που έχω συνεισφέρει ο ίδιος.

Πανεπιστήμιο Θεσσαλίας

Ελλάδα, Φεβρουάριος 2020

Χρήστος Αξελός

Περίληψη

Η *βαθιά μηχανική μάθηση* και συγκεκριμένα τα νευρωνικά δίκτυα έχουν προσφέρει μια καινούργια οπτική γωνία στην επίλυση προβλημάτων. Οι τεχνικές αυτές έχουν υιοθετηθεί από διάφορες εφαρμογές της καθημερινής ζωής ή της βιομηχανίας. *Η αναγνώριση της κατεύθυνσης του βλέμματος* αποτελεί μία από αυτές τις εφαρμογές. Από τότε που τα νευρωνικά δίκτυα ξεκίνησαν να χρησιμοποιούνται για την επίλυση προβλημάτων που έχουν να κάνουν με την εκτίμηση της κατεύθυνσης του βλέμματος, παρέχουν συνεχώς λύσεις που υπερισχύουν σε σχέση με τις προηγούμενες.

Η συγκεκριμένη εργασία προτείνει μια λύση βασισμένη στα δημοφιλή *residual networks* (ResNets), μια καινοφανή αρχιτεκτονική συνελικτικού νευρωνικού δικτύου που προτάθηκε στο ILSVRC [1] το 2015. Συγκεκριμένα, το προτεινόμενο δίκτυο είναι το ResNet-20, το οποίο επιτυγχάνει ανταγωνιστικές επιδόσεις σε σχέση με τη βιβλιογραφία. Το δίκτυο αυτό επιτυγχάνει επιθυμητές επιδόσεις ανεξαρτήτως των εξωτερικών συνθηκών του περιβάλλοντος (in-the-wild), μπορεί να λειτουργεί χωρίς τη χρήση τεχνικών ρύθμισης της κάμερας (camera calibration) και ανεξάρτητα από τα χαρακτηριστικά προσώπου του κάθε χρήστη. Τέλος, η προτεινόμενη λύση μπορεί να υποκαθιστά τη χρήση ακριβών συσκευών ειδικού σκοπού, όταν η επίδοση με αρκετά μεγάλη ακρίβεια δεν είναι αναγκαία. Ως εκ τούτου, μπορεί να μειωθεί το κόστος κατασκευής αυτών των εφαρμογών και να αποτελούν αυτές προσβάσιμες όχι μόνο στους εξειδικευμένους χρήστες, αλλά σε όποιον διαθέτει υπολογιστή ή άλλη συσκευή με κάμερα.

Abstract

Deep learning and particularly neural networks have offered a new point of view in problem solving. These techniques have been heavily adopted by many applications used in daily life or industry. *Gaze estimation* belongs to this category of applications. Since neural networks have been used in order to solve problems related to gaze estimation, they continuously provide solutions that outperform the previous ones.

This thesis proposes a neural network architecture based on the popular *residual networks* (ResNets), a novel convolutional network introduced in ILSVRC [1] (2015). Specifically, the proposed method is a ResNet-20 network, which achieves competitive performance compared to the literature. This architecture achieves desirable performance regardless of the environmental conditions (in-the-wild) or the facial characteristics and it can also operate well without any calibration techniques.

Finally, this solution can substitute the use of expensive, special hardware when high accuracy is not necessary. As a result, reducing the production cost can make these applications accessible not only to specialized users, but to everyone with a laptop and a web camera.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Γεράσιμο Ποταμιάνο για την πλήρη καθοδήγηση κατά τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας. Οι υποδείξεις του και η στάση που έδειξε με βοήθησαν να ολοκληρώσω επιτυχώς τη διπλωματική μου εργασία, καθώς και να αποκτήσω ένα καλύτερο ακαδημαϊκό υπόβαθρο. Θέλω να ευχαριστήσω επίσης τους συνεπιβλέποντες καθηγητές κ. Νικόλαο Μπέλλα και κ. Αντώνιο Αργυρίου, καθώς και τους υπόλοιπους καθηγητές του τμήματος που με βοήθησαν όλα αυτά τα χρόνια της φοίτησής μου.

Επιπλέον, είμαι ευγνώμων στους γονείς μου και στην αδερφή μου που ήταν δίπλα μου και με υποστήριξαν με κάθε τρόπο κατά τη διάρκεια των σπουδών μου.

Τέλος, οφείλω να πω ένα μεγάλο ευχαριστώ στους φίλους και συμφοιτητές μου, οι οποίοι μου παρείχαν χρήσιμες τεχνικές συμβουλές και ιδέες ώστε να καταφέρω να ολοκληρώσω τη διπλωματική αυτήν εργασία.

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Λίστα Αλγορίθμων

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Εκτίμηση της κατεύθυνυσης του βλέμματος

Ως *εκτίμηση της κατεύθυνσης του βλέμματος* ορίζουμε τη διαδικασία όπου προσπαθούμε να εκτιμήσουμε το **προς τα πού** κοιτάει κάποιος. Η διαδικασία αυτή χρησιμοποιείται από διάφορες εφαρμογές που έχουν να κάνουν με την αλληλεπίδραση ανθρώπου-μηχανής [2], την εικονική πραγματικότητα, την βιομηχανία των βιντεοπαιχνιδιών και την ανάλυση προσοχής [3]. Για να καταγράψουμε τη δραστηριότητα του ανθρώπινου ματιού, απαιτείται μια εφαρμογή με χρήση κάμερας.

Όπως περιγράφεται στο [4], οι εκτιμητές αυτοί μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες, ανάλογα με το **πού είναι τοποθετημένη** η κάμερα στο χώρο. Η πρώτη κατηγορία εκτιμητών περιέχει τις *κοντά στο μάτι* κάμερες, οι οποίες είναι προσδεμένες σε συσκευές που τις φοράμε στο κεφάλι. Για παράδειγμα, οι διάφορες συσκευές εικονικής πραγματικότητας. Η δεύτερη κατηγορία περιέχει τους εκτιμητές *απομακρυσμένης εικόνας*, όπου οι εικόνες λαμβάνονται από μια κάμερα που βρίσκεται σε συγκεκριμένο, σταθερό σημείο στο χώρο. Για παράδειγμα, η κάμερα του υπολογιστή μπορεί να θεωρηθεί ως τέτοια κάμερα. Οι εικόνες που λαμβάνονται σε αυτήν την κατηγορία εκτιμητών έχουν συνήθως χαμηλότερη ανάλυση, λόγω της αυξημένης απόστασης ανάμεσα στο μάτι και στο φακό της κάμερας. Ωστόσο, ο χρήστης σε αυτήν την περίπτωση δεν είναι υποχρεωμένος να φοράει στο κεφάλι του κάποια συσκευή. Στη διπλωματική αυτήν εργασία, θα χρησιμοποιήσουμε εκτιμητές

απομακρυσμένης εικόνας, κάνοντας χρήση της **κάμερας του υπολογιστή**.

Σημαντικό κριτήριο διαχωρισμού των εκτιμητών κατεύθυνσης βλέματος είναι επίσης η χρήση ή όχι των *εσωτερικών παραμέτρων της κάμερας*, κάθε φορά που ένας νέος χρήστης χρησιμοποιεί την εφαρμογή (person-specific camera calibration). Ρυθμίζοντας τα εσωτερικά χαρακτηριστικά της κάμερας για κάθε χρήστη μπορεί να αυξηθεί η ευστοχία του συστήματος, ωστόσο η διαδικασία αυτή απαιτεί αρκετό χρόνο. Επιπλέον, το σύστημα θα πρέπει να διαχωρίζει τα πρόσωπα των χρηστών, πράγμα που αυξάνει την πολυπλοκότητα. Για αυτόν το λόγο, σε αυτήν τη διπλωματική εργασία αποφεύγουμε να ρυθμίζουμε τις παραμέτρους κάθε φορά, θέτοντας εξαρχής σε αυτές σταθερές **προσεγγιστικές** τιμές.

Τέλος, διαχωρίζουμε τους εκτιμητές κατεύθυνσης βλέμματος σε άλλες δύο κατηγορίες. Στους εκτιμητές που χρησιμοποιούν *υλικό ειδικού σκοπού* και σε αυτούς που δεν χρησιμοποιούν. Στην πρώτη κατηγορία ανήκουν οι εκτιμητές που περιγράψαμε στην πρώτη παράγραφο, δηλαδή αυτοί που διαθέτουν *κοντά στο μάτι* κάμερες και οι εκτιμητές *απομακρυσμένης εικόνας*. Οι εκτιμητές αυτοί μπορούν να πετύχουν μεγάλη ευστοχία όταν χρησιμοποιούν ειδικό υλικό (hardware) για την επεξεργασία και κατέχουν κάμερες υψηλής ευκρίνειας ή είναι τοποθετημένες πολύ κοντά στα μάτια. Ωστόσο, το υψηλό κόστος αυτών των συσκευών δεν τις κάνει προσιτές για μεγάλο αριθμό ανθρώπων. Στη δεύτερη κατηγορία ανήκουν οι εκτιμητές που αποφεύγουν τη χρήση ειδικού υλικού και χρησιμοποιούν τους διαθέσιμους πόρους ενός υπολογιστή γενικής χρήσης. Προφανώς σε αυτό το σενάριο η ευστοχία θα είναι χαμηλότερη. Ωστόσο, η συνεχής πρόοδος στον τομέα της όρασης υπολογιστών και της τεχνητής νοημοσύνης μας παρέχει όλο και πιο αξιόπιστες λύσεις που ανήκουν σε αυτήν την κατηγορία. Στα πλαίσια της διπλωματικής εργασίας, **δεν** θα χρησιμοποιηθεί ειδικό υλικό και θα ακολουθήσουμε την προσέγγιση της δεύτερης κατηγορίας εκτιμητών.

1.2 Η προσέγγισή μας

Προηγούμενες εργασίες πάνω στην αναγνώριση κατεύθυνσης βλέμματος δείξανε πως η χρήση νευρωνικών δικτύων μπορεί να επιφέρει αρκετά μεγάλη αύξηση στην επίδοση σε σχέση με αντίστοιχες εργασίες που χρησιμοποιούσαν πιο παραδοσιακές μεθόδους, όπως οι μέθοδοι Adaptive Linear Regression (ALR) στο [5], Support Vector Regression (SVR) στο

[6], k Nearest Neighbors (kNN) και τα Regression Forests (RF) στο [7]. Για παράδειγμα, στο [8] έγινε χρήση ενός πολυτροπικού συνελικτικού δικτύου (Multimodal Convolutional Network), βελτιώνοντας την επίδοση κατά **6%** σε σχέση με την καλύτερη επίδοση της μέχρι τότε βιβλιογραφίας (Regression Forests [7]), ενώ στο [9] έγινε χρήση ενός VGG-16 δικτύου, αυξάνοντας την επίδοση κατά **18%** σε πραγματικά δεδομένα και **19%** σε συνθετικά (GazeNet+) σε σχέση με τα Regression Forests [7]. Μια παρόμοια προσέγγιση με τη δική μας υπάρχει στο [10], ωστόσο η λύση αυτή χρησιμοποιεί την εκδοχή των *πλήρως προενεργοποιημένων ResNets* (fully preactivated ResNets), ενώ η εφαρμογή τους εστιάζει στις κινητές συσκευές.

Παρατηρώντας τα αποτελέσματα αυτά της βιβλιογραφίας αλλά και το γεγονός ότι οι συνεχώς βελτιωμένες αρχιτεκτονικές νευρωνικών δικτύων φαίνονται αρκετά υποσχόμενες, θεωρήθηκε καλό να στραφούμε προς αυτήν την κατεύθυνση. Για το λόγο αυτό, στη διπλωματική αυτήν εργασία έγινε χρήση της αρχιτεκτονικής **ResNet** [11], η οποία διακρίθηκε στο ILSVRC [1] το 2015 και αποτελεί μία από τις πιο πρόσφατες και αξιόλογες αρχιτεκτονικές συνελικτικού δικτύου.

1.3 Η συνεισφορά της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία παρουσιάζει την αρχιτεκτονική **ResNet-20**, μια αρχιτεκτονική συνελικτικού δικτύου που βασίζεται στο δίκτυο *Resnet* (Residual Network) [11]. Το δίκτυο ResNet διαθέτει μονοπάτια παράκαμψης συνελικτικών στρωμάτων (shortcut paths), προσέγγιση που μέχρι τώρα δεν έχει ακολουθηθεί στη βιβλιογραφία της ανίχνευσης κατεύθυνσης βλέμματος (gaze tracking) κατά κόρον. Η επίδοση της μεθόδου αυτής είναι πολύ κοντά στην αντίστοιχη των πιο πρόσφατων εργασιών της βιβλιογραφίας.

Επιπλέον, παρέχουμε μια υλοποίηση σχετικά με το πώς μπορούμε να αξιοποιήσουμε την πληροφορία της κατεύθυνσης του βλέμματος, ώστε να μπορεί ο υπολογιστής να εντοπίζει το σημείο που κοιτάει ένας χρήστης στην οθόνη (gaze tracking).

1.4 Δομή διπλωματικής εργασίας

Το υπόλοιπο μέρος της διπλωματικής εργασίας είναι οργανωμένο ως εξής:

- Στο κεφάλαιο 2 περιγράφουμε αναλυτικά τη διαδικασία που παράγει τις εισόδους του δικτύου ResNet-20.
- Στο κεφάλαιο 3 περιγράφουμε το προτεινόμενο δίκτυο ResNet-20.
- Στο κεφάλαιο 4 περιγράφουμε μια εφαρμογή που αξιοποιεί την πληροφορία της κατεύθυνσης του βλέμματος.
- Στο κεφάλαιο 5 αξιολογούμε την επίδοση του ResNet-20 και την συγκρίνουμε με τη βιβλιογραφία.
- Τέλος, στο κεφάλαιο 6 συνοψίζουμε και παραθέτουμε σημειώσεις που προεκτείνουν τη μεθοδολογία μας.

Κεφάλαιο 2

Προεπεξεργασία δεδομένων εισόδου

Στο κεφάλαιο αυτό θα γίνει αναλυτική εξήγηση της προεπεξεργασίας των δεδομένων που θα γίνουν είσοδος στο προτεινόμενο δίκτυο ResNet-20. Η διαδικασία ξεκινά με τη λήψη μιας εικόνας σε πραγματικό χρόνο από την κάμερα του υπολογιστή και τελειώνει με έναν τελικό μετασχηματισμό.

Πριν μπούμε όμως στην προεπεξεργασία των δεδομένων, θα πρέπει να ορίσουμε τα 3 συστήματα συντεταγμένων που θα χρησιμοποιήσουμε για την εξαγωγή των χαρακτηριστικών (features) και να δείξουμε πώς γίνονται οι μετασχηματισμοί από το ένα σύστημα στο άλλο. Ορίζουμε τα συστήματα συντεταγμένων όπως ακριβώς ορίζονται στο [7].

2.1 Ορισμός Συστημάτων συστεταγμένων

2.1.1 Συντεταγμένες χώρου

Το πρώτο σύστημα συντεταγμένων αποτελεί το τρισδιάστο σύστημα συντεταγμένων του χώρου (3d world coordinates). Ως αρχή των αξόνων ορίζεται ένα τρισδιάστατο σημείο το οποίο βρίσκεται στο αντικείμενο που μελετάμε. Στην περίπτωσή μας, τέτοιο αντικείμενο θεωρείται το ανθρώπινο πρόσωπο και ως σημείο αναφοράς το μέσο σημείο ανάμεσα στα δύο μάτια του προσώπου.

2.1.2 Συντεταγμένες κάμερας

Το δεύτερο σύστημα συντεταγμένων αποτελεί το τρισδιάστατο σύστημα συντεταγμένων της κάμερας (3d camera coordinates). Το σύστημα αυτό είναι όμοιο με το προηγούμενο, αλλά χρησιμοποιεί ως σημείο αναφοράς την κάμερα. Μπορούμε εύκολα να μετατρέψουμε ένα σημείο από τις συντεταγμένες χώρου στις συντεταγμένες κάμερας, εφόσον γνωρίζουμε τις εξωτερικές (extrinsic) παραμέτρους της κάμερας, δηλαδή τους πίνακες *περιστροφής* (rotation) και *μετατόπισης* (translation) του αντικειμένου ως προς την κάμερα.

Για παράδειγμα, αν υποθέσουμε πως οι συντεταγμένες χώρου ενός σημείου είναι (U, V, W) και γνωρίζουμε τον πίνακα περιστροφής $\boldsymbol{R}$ (3x3 πίνακας) και το διάνυσμα μετατόπισης $\boldsymbol{t}$ (3x1 διάνυσμα), τότε το ίδιο σημείο στο σύστημα συντεταγμένων της κάμερας (X, Y, Z) ορίζεται από τον τύπο (2.1).

$$\begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} \mathrm{U} \\ \mathrm{V} \\ \mathrm{W} \end{bmatrix} + \boldsymbol{t} \tag{2.1}$$

2.1.3 Συντεταγμένες οθόνης

Τέλος, το τρίτο σύστημα συντεταγμένων είναι το δισδιάστατο σύστημα συντεταγμένων της οθόνης (2d screen coordinates). Είναι εύκολο να αποκτήσουμε τις δισδιάστατες συντεταγμένες τις οθόνης από τις τρισδιάστατες συντεταγμένες της κάμερας, εφόσον ξέρουμε τις εσωτερικές (intrinsic) παραμέτρους της κάμερας, δηλαδή τα εστιακά μήκη (focal length) και τα οπτικά κέντρα (optical centers).

Για παράδειγμα, έχοντας αποκτήσει από την (2.1) τις συντεταγμένες κάμερας (X, Y, Z) και εφόσον γνωρίζουμε τα μήκη εστίασης f_x, f_y και τα οπτικά κέντρα (c_x, c_y), τότε το ίδιο σημείο στο σύστημα συντεταγμένων της οθόνης (x, y) προκύπτει από τον τύπο (2.2).

$$\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ 1 \end{bmatrix} = \mathbf{s} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{bmatrix} \tag{2.2}$$

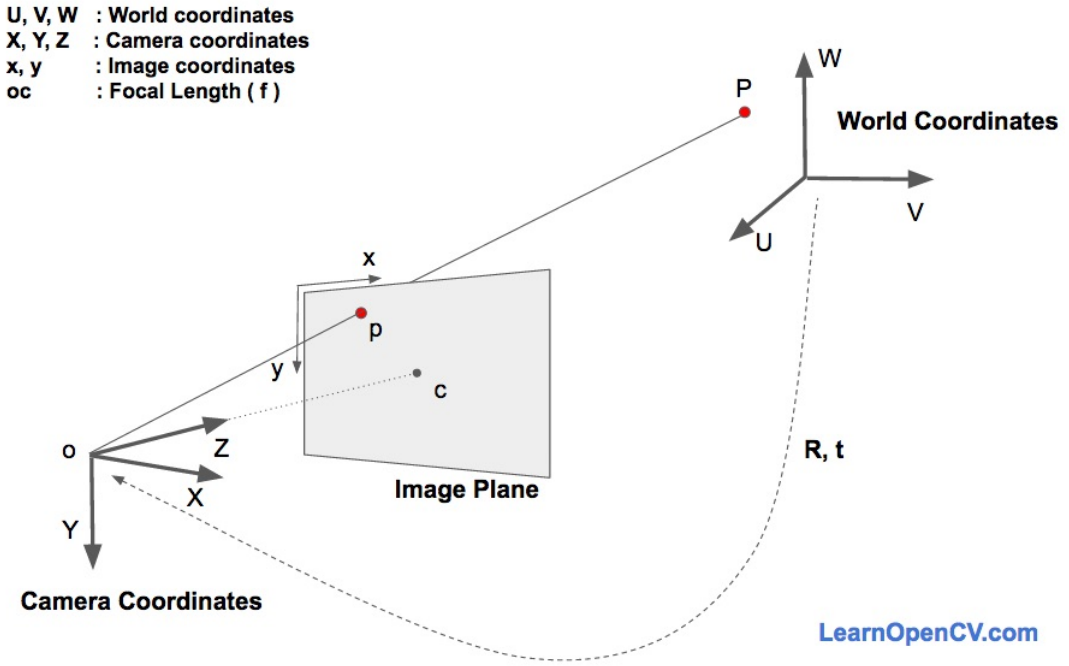
Το **s** (scale) δηλώνει έναν συντελεστή κλίμακας που εκφράζει το βάθος μιας εικόνας. Η εξίσωση (2.3) μας δείχνει πως μπορούμε να μετατρέψουμε τις συντεταγμένες χώρου (δεξί μέλος) στις συντεταγμένες κάμερας (αριστερό μέλος), εφόσον γνωρίζουμε τον πίνακα περιστροφής και το διάνυσμα μετατόπισης.

$$\mathbf{s} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} \mathrm{U} \\ \mathrm{V} \\ \mathrm{W} \\ 1 \end{bmatrix} \tag{2.3}$$

Το σχήμα 2.1 μας δείχνει την σχέση ανάμεσα σε αυτά τα 3 συστήματα συντεταγμένων.

2.2 Εξαγωγή χαρακτηριστικών

Για την εξαγωγή χαρακτηριστικών ακολουθούμε την εξής διαδικασία. Αρχικά ανιχνεύουμε αν υπάρχει πρόσωπο στην εικόνα. Αν υπάρχει, προσπαθούμε να εντοπίσουμε 6 σημεία στο πρόσωπο χρησιμοποιώντας κατάλληλα εκπαιδευμένα μοντέλα. Έπειτα, προσπαθούμε να εξάγουμε την πόζα του κεφαλιού, κάνοντας αντιστοίχιση των 6 αυτών σημείων με 6 σημεία ενός γνωστού εκ των προτέρων μοντέλου προσώπου που βρίσκονται στις συντεταγμένες χώρου. Τέλος, μετασχηματίζουμε την εικόνα εισόδου και την πόζα του κεφαλιού, ώστε να έρθουν αυτά σε συμβατή μορφή με τις βάσεις δεδομένων που χρησιμοποιούμε.



Σχήμα 2.1: Τα 3 συστήματα συντεταγμένων που συζητήθηκαν στην ενότητα 2.1. (Σχήμα από το [12]).

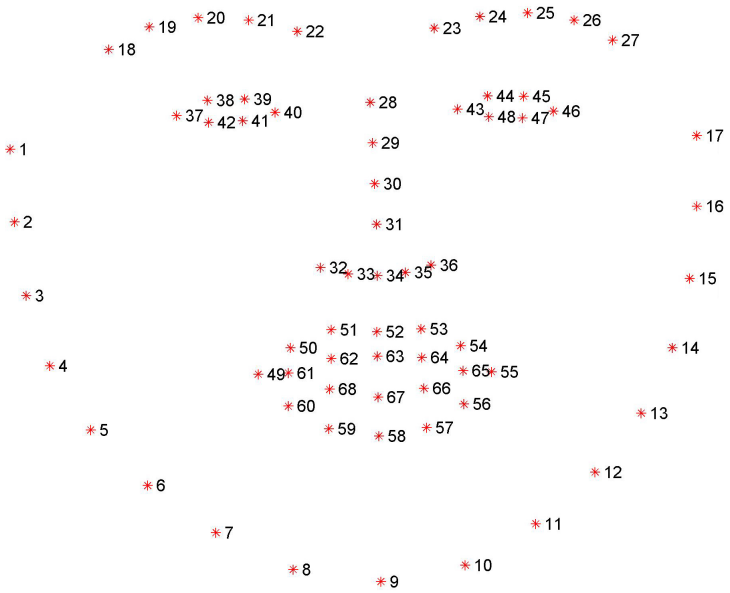
2.2.1 Ανίχνευση προσώπου

Αρχικά λαμβάνουμε μια εικόνα από την κάμερα του υπολογιστή. Έπειτα, χρησιμοποιούμε έναν προεκπαιδευμένο ανιχνευτή προσώπου του εργαλείου Dlib [13]. Ο ανιχνευτής προσώπου αυτός είναι βασισμένος στον αλγόριθμο HoG [14].

Εφόσον βεβαιωθούμε πως υπάρχει όντως κάποιο πρόσωπο στην εικόνα, προσπαθούμε να εξάγουμε πληροφορίες σχετικά με την πόζα του κεφαλιού στον τρισδιάστατο χώρο.

2.2.2 Εντοπισμός σημείων στην εικόνα προσώπου

Το επόμενο βήμα είναι να εντοπίσουμε 6 χαρακτηριστικά σημεία που φαίνονται στην δισδιάστατη εικόνα προσώπου. Ακολουθούμε όμοια διαδικασία με το [7] και υπολογίζουμε 6 ακριβώς σημεία, διότι μετρήσεις δείξανε πως αυτός ο αριθμός είναι ο μικρότερος που απαιτείται για την ικανοποιητική εξαγωγή της πόζας του κεφαλιού. Τα σημεία αυτά είναι οι 2 άκρες του κάθε ματιού και οι 2 άκρες του στόματος, τα οποία και φαίνονται στο σχήμα 2.2. Η ανίχνευση των σημείων αυτών έγινε με χρήση ενός εκπαιδευμένου μοντέλου 68 σημείων προσώπου (landmark detector) που παρέχεται στο [15] και βασίζεται στα ενεργά μοντέλα



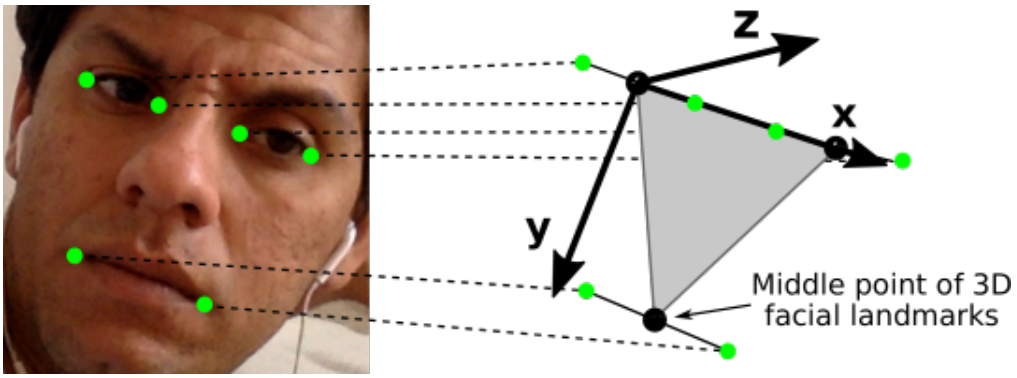
Σχήμα 2.2: Στην εικόνα φαίνονται τα 68 σημεία που εντοπίζει ο αλγόριθμος [15]. Θα χρειαστούμε 6 από αυτά, τα σημεία 37, 40, 43, 46, 49 και 55. (Εικόνα από το [15]).

προσανατολισμού ή αλλιώς AOMs (Active Orientation Models) [16].

2.2.3 Ανίχνευση πόζας κεφαλιού

Ο υπολογισμός του $\boldsymbol{R}$ γίνεται με τον εξής τρόπο. Αρχικά προσεγγίζουμε τις εσωτερικές παραμέτρους της κάμερας (f_x, f_y, c_x, c_y) όπως γίνεται στο [8]. Έπειτα, από την (2.3) παρατηρούμε πως αν γνωρίζουμε τα κατάληλλα $\boldsymbol{R}$ και $\boldsymbol{t}$ μπορούμε να *προβάλλουμε* το τρισδιάστατο μοντέλο προσώπου 6 σημείων από τις συντεταγμένες χώρου στις συντεταγμένες οθόνης. Στη συνέχεια, προσπαθούμε να ελαχιστοποιήσουμε το *σφάλμα προβολής* (reprojection error) που δημιουργείται ανάμεσα στα 6 σημεία που εντοπίζει ο αλγόριθμος στο [15] και στα 6 σημεία του μοντέλου προσώπου που έχει υποστεί προβολή. Στόχος μας είναι η εύρεση των κατάλληλων $\boldsymbol{R}$ και $\boldsymbol{t}$ ώστε να ελαχιστοποιείται το σφάλμα προβολής. Στο σημείο αυτό εφαρμόζουμε τη βελτιστοποίηση Levenberg-Marquardt [17] ώστε να υπολογιστούν τα βέλτιστα $\boldsymbol{R}$ και $\boldsymbol{t}$. Όλη αυτή η διαδικασία συνοψίζεται μέσω της συνάρτησης `solvepnp()` της OpenCV [18].

Μόλις υπολογίσουμε τα $\boldsymbol{R}$ και $\boldsymbol{t}$, προβάλλουμε το μοντέλο προσώπου 6 σημείων στις συντεταγμένες κάμερας σύμφωνα με τον τύπο (2.3). Στη συνέχεια, υπολογίζουμε τα μέσο σημείο του κάθε ματιού και το μέσο σημείο του στόματος στις συντεταγμένες κάμερας. Τέλος, μπορούμε να αντιληφθούμε την πόζα ως το δισδιάστατο διάνυσμα $(\hat{\boldsymbol{p}}_{\boldsymbol{hor}}, \hat{\boldsymbol{p}}_{\boldsymbol{vert}})$, όπου



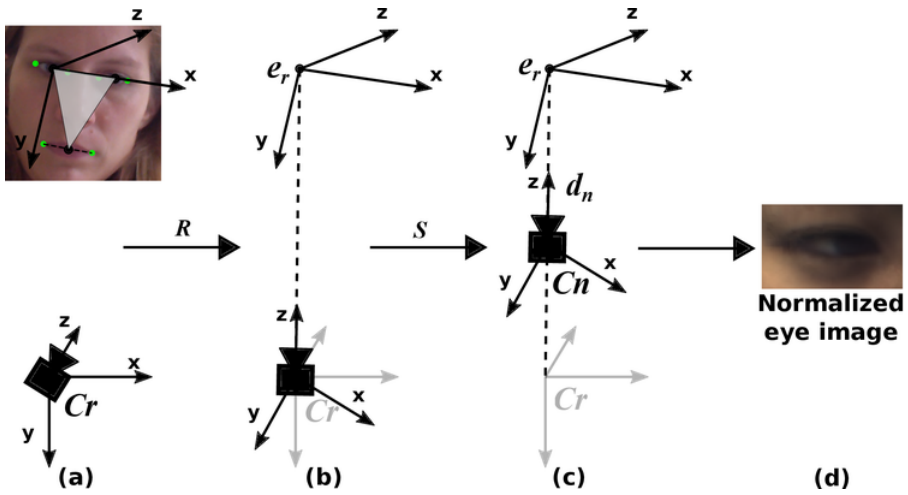
Σχήμα 2.3: Στα αριστερά φαίνονται τα 6 σημεία που εντόπισε ο αλγόριθμος εντοπισμού σημείων στην εικόνα. Στα δεξιά φαίνεται το μοντέλο προσώπου να έχει υποστεί μετατόπιση και περιστροφή με τέτοιο τρόπο, ώστε να ελαχιστοποιείται το σφάλμα προβολής (reprojection error). Το γκρι τρίγωνο εκφράζει την επιφάνεια που σχηματίζεται από τα μάτια και το στόμα, ενώ η κάθετη σ'αυτό ευθεία (δεν φαίνεται στο σχήμα) μας δίνει πληροφορίες για την πόζα. (Εικόνα από το [9]).

$\hat{\boldsymbol{p}}_{\boldsymbol{hor}}$, $\hat{\boldsymbol{p}}_{\boldsymbol{vert}}$ η οριζόντια και η κατακόρυφη αντίστοιχα γωνία που σχηματίζονται ανάμεσα στις εξής ευθείες. Στην κάθετη ευθεία στο επίπεδο που σχηματίζουν τα μέσα των ματιών και του στόματος και στην ευθεία που ενώνει το πρόσωπο με την κάμερα. Αυτό αποτυπώνεται καλύτερα στα δεξιά του σχήματος 2.3.

2.3 Μετασχηματισμός δεδομένων στο σύστημα κανονικοποιημένης κάμερας

Στο σημείο αυτό, μετασχηματίζουμε την εικόνα εισόδου και την πόζα, ακολουθώντας την ίδια διαδικασία με το [19]. Συνοπτικά, στόχος αυτού του μετασχηματισμού είναι η μετατροπή των δεδομένων από τις συντεταγμένες κάμερας στο σύστημα συντεταγμένων *κανονικοποιημένης κάμερας*, όπως το ονομάζουν οι δημιουργοί του [19]. Αποδεικνύεται πως με αυτήν τη μέθοδο εξαλείφονται καλύτερα οι διαφορές που οφείλονται στην πόζα απ'ότι αν χρησιμοποιούσαμε τα αυθεντικά δεδομένα. Εκτενέστερη αναφορά αυτού του μετασχηματισμού γίνεται στο [19].

Μετά το τέλος του μετασχηματισμού, έχει γίνει αποκοπή της αρχικής εικόνας ώστε να απομονώσουμε την εικόνα του δεξιού ή αριστερού ματιού. Το σχήμα 2.4 δείχνει πως γίνεται ο μετασχηματισμός αυτός. Εκτός από την εικόνα, βλέπουμε την επίδραση του μετασχηματισμού αυτού στην πόζα αλλά και στην κατεύθυνση του βλέμματος. Οι αλλαγές αυτές οφείλονται στους πίνακες $\boldsymbol{C_r, R, S, C_n}$ του σχήματος 2.4.



Σχήμα 2.4: Μετατροπή της εικόνας εισόδου (a) στην κανονικοποιημένη της μορφή (d). Ο μετασχηματισμός που εφαρμόζεται αφαιρεί τις περιστροφές και κρατάει μόνο την περιοχή του ματιού. Εφαρμόζεται περιστροφή (b) και κλιμάκωση (c), ενώ στο τέλος (d) εφαρμόζουμε την αποκοπή. Ο πίνακας $\boldsymbol{C_r}$ περιέχει τις εσωτερικές παραμέτρους της κάμερας, ο πίνακας $\boldsymbol{R}$ χρησιμοποιείται ως πίνακας μετασχηματισμού του πίνακα περιστροφής του αυθεντικού πίνακα περιστροφής στο σύστημα συντεταγμένων κάμερας. Αντίστοιχη λειτουργία έχει και ο πίνακας $\boldsymbol{C_n}$ αλλά για τον πίνακα εσωτερικών παραμέτρων της κάμερας $\boldsymbol{C_r}$, ενώ ο πίνακας $\boldsymbol{S}$ πραγματοποιεί μεγέθυνση της εικόνας. (Εικόνα από το [19]).

Μετά το τέλος του μετασχηματισμού, χρησιμοποιούμε τις μετασχηματισμένες εκδοχές της εικόνας του ματιού και τις πόζας του κεφαλιού ως είσοδο στο προτεινόμενο δίκτυο ResNet-20. Σαν έξοδο από το δίκτυο αυτό λαμβάνουμε την κανονικοποιημένη κατεύθυνση βλέμματος. Η διαδικασία αυτή περιγράφεται αναλυτικά στο κεφάλαιο 3.

Κεφάλαιο 3

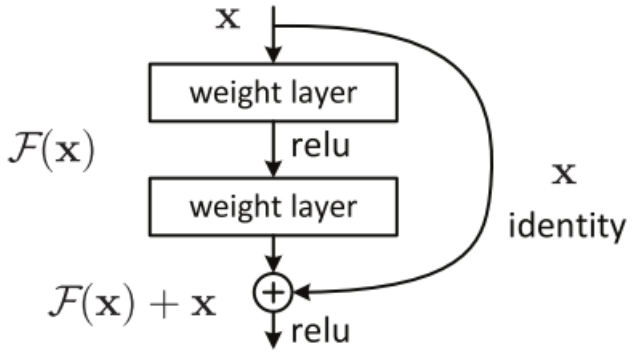
Το Προτεινόμενο Δίκτυο ResNet-20

Αφού έχουμε αποκτήσει από το κεφάλαιο 2 την κανονικοποιημένη πόζα και εικόνα, προσπαθούμε με αυτά τα δύο δεδομένα να προβλέψουμε την *κανονικοποιημένη κατεύθυνση* του βλέμματος, δηλαδή την κατεύθυνση του βλέμματος στο κανονικοποιημένο σύστημα της κάμερας που αναφέραμε στην ενότητα 2.3. Για να το πετύχουμε αυτό, χρησιμοποιούμε το συνελικτικό δίκτυο *ResNet* [11], εφαρμόζοντας όμως κάποιες αλλαγές σε αυτό. Συγκεκριμένα, η προτεινόμενη λύση βασίζεται στην αρχιτεκτονική **ResNet-20**. Ο αριθμός 20 προέρχεται από τον αριθμό των συνελίξεων στο κύριο μονοπάτι του δικτύου.

3.1 Σύντομο θεωρητικό υπόβαθρο

Το δίκτυο *ResNet* [11] αποτελεί μια αρχιτεκτονική συνελικτικού δικτύου που προτάθηκε στο ILSVRC [20] το 2015. Έγιναν γνωστά λόγω της ικανότητας τους να εξαλείφουν σε μεγάλο βαθμό το πρόβλημα της σχεδόν μηδενικής ανανέωσης των βαρών στα αρχικά συνελικτικά στρώματα (vanishing gradients problem). Επιπλέον, έχει αποδειχθεί πως πετυχαίνουνε παρόμοιες επιδόσεις με άλλα δίκτυα, χρησιμοποιώντας όμως πολύ λιγότερα συνελικτικά στρώματα (μικρότερο βάθος).

Οι διαφορές των δικτύων αυτών σε σχέση με τις γνωστές αρχιτεκτονικές είναι οι εξής. Αρχικά, αντικαθίσταται το συνελικτικό στρώμα με το *βασικό μπλοκ* (basic block). Το *βασικό μπλοκ* αποτελείται από δύο τριάδες πράξεων, όπου κάθε τριάδα αποτελείται από μία 3×3 συνέλιξη, μία κανονικοποίηση σακιδίου (batch normalization [21]) και από μία συνάρτηση ενεργοποίησης ReLU. Συχνά, αντί για δύο 3×3 συνελίξεις, επιλέγεται η εκδοχή



Σχήμα 3.1: Στην εικόνα κάθε κουτάκι (weight layer) αποτελείται από μία 3×3 συνέλιξη και από μία κανονικοποίηση σακιδίου (batch normalization [21]). Η καμπυλωτή γραμμή αποτελεί το *μονοπάτι συντόμευσης*. (Εικόνα από το [11]).

με τις 3 συνελίξεις (1×1, 3×3, 1×1) για καλύτερη χρονική επίδοση (εκδοχή bottleneck).

Επιπλέον, εισάγεται η έννοια του *μονοπατιού παράκαμψης* (shortcut path). Τα μονοπάτια αυτά παρακάμπτουν συνελικτικά στρώματα και εφαρμόζεται σε αυτά η *ταυτοτική συνάρτηση* (identity function). Έπειτα αθροίζονται με την έξοδο ενός *βασικού μπλοκ*. Αν ο αριθμός των καναλιών του μονοπατιού και του *βασικού μπλοκ* διαφέρει, τότε αντί της *ταυτοτικής συνάρτησης* εφαρμόζεται το *αθροιστικό μπλοκ* (residual block). Τα *αθροιστικά μπλοκ* αποτελούνται από μία 1×1 συνέλιξη και από μία κανονικοποίηση σακιδίου (batch normalization [21]).

Στην εικόνα 3.1 βλέπουμε το *βασικό μπλοκ* καθώς και το *μονοπάτι συντόμευσης*.

3.2 Υπολογισμός παραμέτρων του ResNet-20

Το δίκτυο **ResNet-20** προέκυψε έπειτα από πειραματισμούς πάνω στις βασικές παραμέτρους του δικτύου ResNet. Χρησιμοποιώντας κάποιες αρχικές τιμές στις παραμέτρους, δοκιμάσαμε να αλλάξουμε τις τιμές αυτές μία κάθε φορά. Μόλις λάβουμε την βέλτιστη τιμή για μία παράμετρο, δηλαδή την τιμή που παράγει το *μικρότερο μέσο σφάλμα*, τότε κρατάμε αυτήν την τιμή και δοκιμάζουμε να αλλάξουμε την επόμενη παράμετρο.

Προσπαθήσαμε να αλλάξουμε τις παραμέτρους με συγκεκριμένη σειρά. Προτεραιότητα δώσαμε στις παραμέτρους που επηρεάζουν τα αρχικά στάδια του αλγορίθμου, όπως οι

παράμετροι της αρχικής συνέλιξης του δικτύου (gate block). Στην πορεία ασχοληθήκαμε με τις παραμέτρους των υπόλοιπων συνελικτικών στρωμάτων του δικτύου και αφήσαμε τελευταίες τις παραμέτρους που έχουν σχέση με τα διασυνδεδεμένα στρώματα. Για κάθε συνελικτικό στρώμα που προσθέταμε, ελέγχαμε για ποιές τιμές των παραμέτρων του στρώματος παίρνουμε καλύτερη επίδοση. Αν για καμία τιμή των παραμέτρων του τρέχοντος στρώματος δεν παίρναμε καλύτερη επίδοση, σταματούσαμε να προσθέτουμε στρώματα. Αλλιώς κρατούσαμε το τρέχον στρώμα και τις βέλτιστες παραμέτρους του και κάναμε την ίδια διαδικασία για το επόμενο συνελικτικό στρώμα. Με παρόμοιο τρόπο ασχοληθήκαμε και με τις παραμέτρους των διασυνδεδεμένων δικτύων, όπως τον αριθμό των στρωμάτων και των νευρώνων.

Οι παράμετροι του δικτύου πάνω στις οποίες πειραματιστήκαμε είναι οι παρακάτω:

- **Εκδοχή του δικτύου ResNet**. Επιλέξαμε τη βασική εκδοχή του αλγορίθμου, ενώ δοκιμάσαμε και τις εκδοχές *ReLU before addition* και *full pre-activation* [3].
- **Μέγεθος φίλτρου της αρχικής συνέλιξης** (gate block convolution). Δοικιμάστηκαν οι τιμές 3, 5, 7, 9. Σαν βέλτιστη τιμή επιλέχθηκε το **7**.
- **Αριθμός καναλιών εξόδου αρχικής συνέλιξης** (number of output channels of gate block). Δοκιμάστηκαν οι τιμές 16, 32, 64, 128 και σαν βέλτιστη τιμή επιλέχθηκε το **64**.
- **Πλήθος συνελικτικών και διασυνδεδεμένων στρωμάτων**. Ο αριθμός αυτών των στρωμάτων υπολογίστηκε με τη μέθοδο που περιγράψαμε στη δεύτερη παράγραφο της ενότητας 3.2. Συγκεκριμένα, χρησιμοποιείται **μία** αρχική συνέλιξη, **δεκαέξι** συνελικτικά στρώματα και **τρία** διασυνδεδεμένα στρώματα στο κύριο μονοπάτι του δικτύου. Επιπλέον σε τέσσερα σημεία του δικτύου υπάρχουν **τέσσερεις** συνελίξεις που εφαρμόζονται στα μονοπάτια παράκαμψης (shortcuts).
- **Τιμές καναλιών εξόδου ανά στρώμα.** Πειραματιστήκαμε με διάφορες τιμές μεταξύ 16 και 1024 για τα συνελικτικά στρώματα και μεταξύ 128 και 2048 για τα διασυνδεδεμένα στρώματα. Ως βέλτιστες τιμές καναλιών εξόδο επιλέχθηκαν οι **64, 128, 256, 512** για τα συνελικτικά στρώματα και **512** για τα διασυνδεδεμένα.

3.3 Λειτουργία δικτύου ResNet-20

Σχηματικά, το δίκτυο φαίνεται στο σχήμα 3.2. Αρχικά η εικόνα διαστάσεων 60×36 περνάει από ένα συνελικτικό στρώμα (κίτρινο χρώμα στο σχήμα 3.2), ώστε να αυξήσουμε τον αριθμό των καναλιών εξόδου (output channels). Στη συνέχεια εφαρμόζουμε 16 συνελικτικά στρώματα (πράσινο χρώμα στο σχήμα 3.2). Ορισμένα από αυτά (σκούρο πράσινο χρώμα στο σχήμα 3.2) αυξάνουν σταδιακά τα κανάλια (channels) από 64 σε 512. Επιπλέον, σε τέσσερα σημεία του δικτύου υπάρχουν τέσσερεις συνελίξεις (λαδί χρώμα στο σχήμα 3.2) που εφαρμόζονται στα μονοπάτια παράκαμψης (shortcuts).

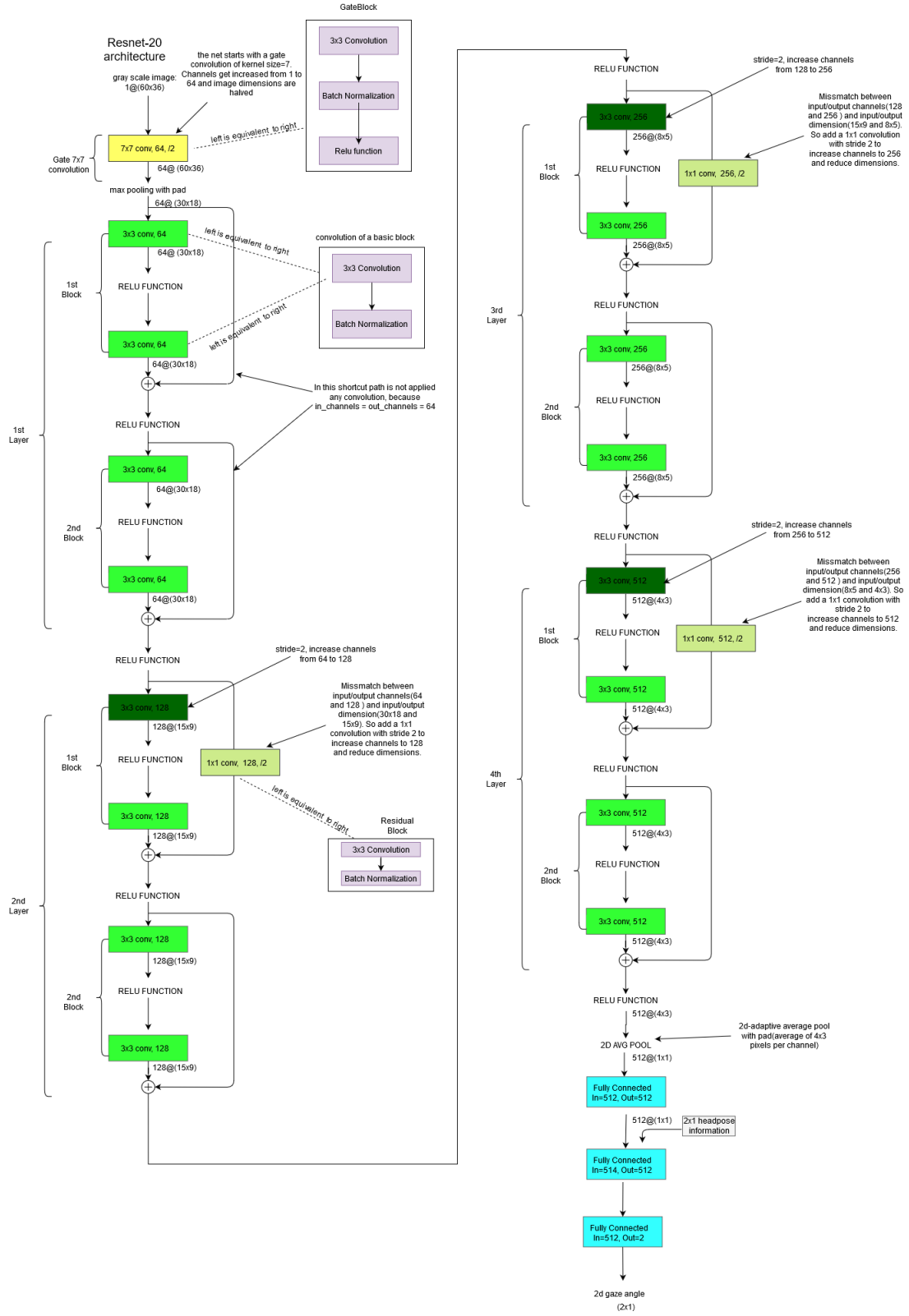
Έπειτα, εφαρμόζονται 3 διασυνδεδεμένα δίκτυα (τιρκουάζ χρώμα στο σχήμα 3.2), όπου ανάμεσα στο πρώτο και στο δεύτερο ενσωματώνουμε το διάνυσμα πόζας $(\hat{\boldsymbol{p}}_{\boldsymbol{hor}}, \hat{\boldsymbol{p}}_{\boldsymbol{vert}})$ που υπολογίσαμε στην ενότητα 2.2.3. Το τρίτο διασυνδεδεμένο δίκτυο παράγει ως έξοδο τη δισδιάστατη κατεύθυνση του βλέμματος στο σύστημα κανονικοποιημένης κάμερας.

Τέλος, εφαρμόζουμε το μετασχηματισμό που αναφέραμε στην ενότητα 2.3 αλλά αντίστροφα, δηλαδή από το κανονικοποιημένο σύστημα συντεταγμένων στο αυθεντικό σύστημα συντεταγμένων, όπως γίνεται στο [19]. Το αποτέλεσμα του μετασχηματισμού αυτού είναι η απόκτηση των γωνιών $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}})$, που εκφράζουν την οριζόντια και κατακόρυφη γωνία του βλέμματος.

3.4 Λογισμικό εκπαίδευσης δικτύου

Για την εκπαίδευση του προτεινόμενου δικτύου, κάναμε χρήση του λογισμικού PyTorch [22], το οποίο αποτελεί την ταχύτερα αναπτυσσόμενη πλατφόρμα βαθιάς μηχανικής μάθησης σήμερα. Ο λόγος που επιλέξαμε το PyTorch συγκεκριμένα είναι το γεγονός ότι συνδιάζει την απλότητα που παρέχει η πλατφόρμα Keras [23], ενώ ταυτόχρονα μας παρέχει την πληθώρα επιλογών του Tensorflow [24].

Πριν κάνουμε χρήση του PyTorch, προσπαθήσαμε να κατασκευάσουμε το δίκτυο χρησιμοποιώντας το Keras [23]. Ωστόσο, αποδείχθηκε πολύ δύσκολο να φτιάξουμε με χρήση αυτού το προτεινόμενο μας δίκτυο, καθώς χρειαζόταν αρκετά περίπλοκη διαδικασία για να κατασκευάσουμε ένα δίκτυο με παραπάνω από μία εισόδους (εικόνα και το διάνυσμα της πόζας του κεφαλιού). Το πρόβλημα αυτό λύθηκε με τη χρήση του PyTorch.



Σχήμα 3.2: Στα αριστερά φαίνονται τα πρώτα 9 συνελικτικά στρώματα του δικτύου ResNet-20. Στα δεξιά φαίνονται τα τελευταία 11 συνελικτικά στρώματα του δικτύου ResNet-20.

Τέλος, η εκπαίδευση του δικτύου έγινε με χρήση του Google Colab[1], το οποίο πάρεχει δωρεάν αρκετά υψηλή υπολογιστική ισχύ.

[1]https://colab.research.google.com/

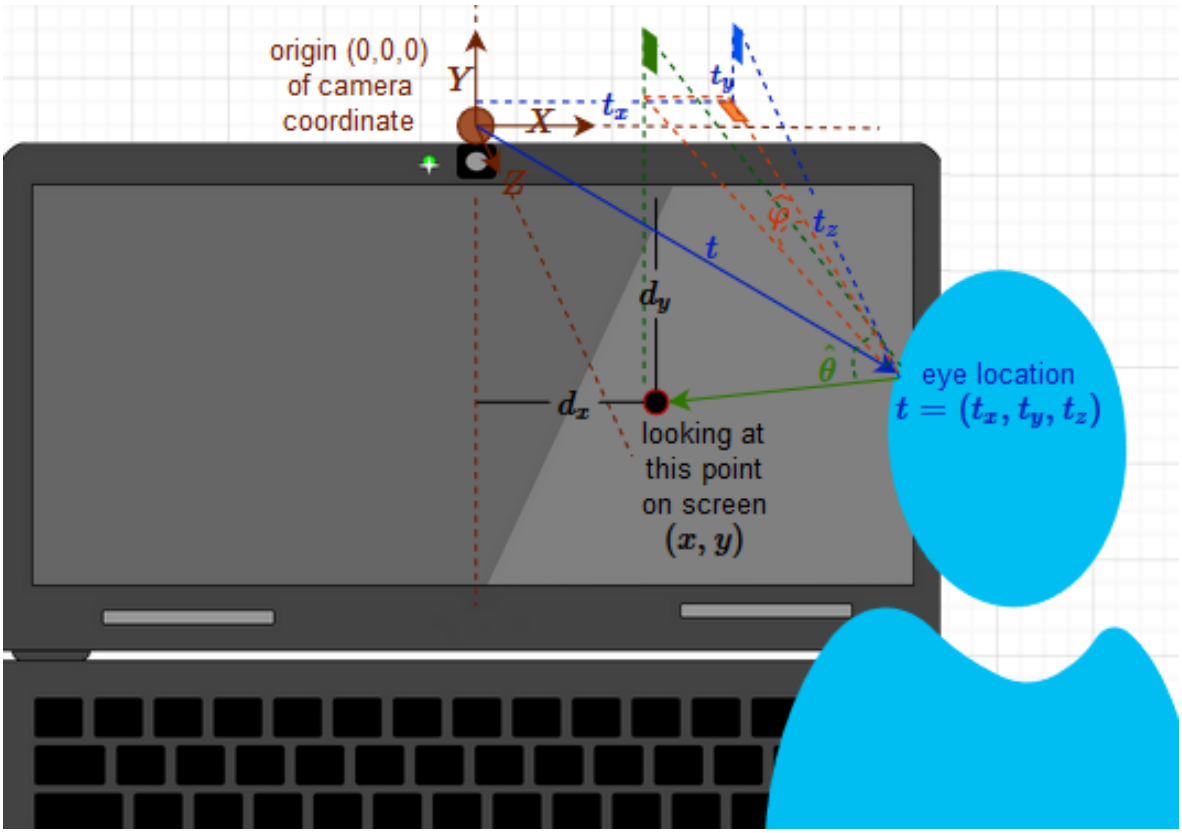
Κεφάλαιο 4

Αλληλεπίδραση Ανθρώπου και Υπολογιστή μέσω της Κατεύθυνσης Βλέμματος

Στο κεφάλαιο αυτό, θα περιγράψουμε αναλυτικά μια εφαρμογή *πραγματικού χρόνου*, όπου θα κάνουμε χρήση της κατεύθυνσης του βλέμματος που αποκτήσαμε από το κεφάλαιο 3. Συγκεκριμένα, θα περιγράψουμε μια μέθοδο σχετικά με το πώς μπορεί ο υπολογιστής να αναγνωρίσει το πού ακριβώς στην οθόνη κοιτάει ένας χρήστης. Για την υλοποίηση της εφαρμογής αυτής θα χρειαστούμε την θέση του κεφαλιού ως προς την οθόνη, την κατεύθυνση του βλέμματος, τις διαστάσεις της οθόνης σε χιλιοστά του μέτρου (millimetres), καθώς και την ανάλυση της οθόνης σε pixels. Τέλος, μέσα από την εφαρμογή αυτή διαπιστώσαμε πως ο προτεινόμενος αλγόριθμος του κεφαλαίου 3 πράγματι μπορεί να χρησιμοποιηθεί υπό πραγματικές συνθήκες. Οπότε θεωρήσαμε την εφαρμογή αυτήν ως έναν επιπρόσθετο τρόπο ελέγχου της εγκυρότητας του προτεινόμενου αλγορίθμου.

4.1 Σχηματική αναπαράσταση εφαρμογής

Στο σημείο αυτό, γνωρίζουμε το $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}})$, το δισδιάστατο διάνυσμα της κατεύθυνσης του βλέμματος που υπολογίσαμε στην ενότητα 3.3, καθώς και το διάνυσμα $\boldsymbol{t}$, δηλαδή το διάνυσμα της θέσης του κεφαλιού στον τρισδιάστατο χώρο με σημείο αναφοράς την κάμερα του υπολογιστή, το οποίο υπολογίσαμε στην ενότητα 2.2.3. Εφαρμόζοντας τριγωνομετρία



Σχήμα 4.1: Παράδειγμα ανίχνευσης σημείου στην οθόνη με δεδομένο το διάνυσμα μετατόπισης $\boldsymbol{t}$, τις γωνίες βλέμματος $\hat{\boldsymbol{\phi}}$, $\hat{\boldsymbol{\theta}}$ και ζητούμενο το σημείο $\boldsymbol{(x, y)}$ της οθόνης.

μπορούμε να υπολογίσουμε το σημείο στην οθόνη που κοιτάει ο χρήστης. Αυτό φαίνεται αναλυτικά στο σχήμα 4.1.

Το καφέ χρώμα της εικόνας δηλώνει το σύστημα συντεταγμένων κάμερας. Το μπλε διάνυσμα δηλώνει τη θέση του ματιού ως προς το σύστημα αυτό. Το πράσινο διάνυσμα δηλώνει την κατεύθυνση του βλέμματος. Η πορτοκαλί γωνία $\hat{\boldsymbol{\phi}}$ δηλώνει την οριζόντια γωνία της κατεύθυνσης του βλέμματος, η πράσινη γωνία $\hat{\boldsymbol{\theta}}$ δηλώνει την κατακόρυφη γωνία του βλέμματος, ενώ παράλληλα φαίνονται και οι διακεκομμένες ευθείες που ορίζουν τις γωνίες αυτές. Το d_x δηλώνει πόσο δεξιά ή πόσο αριστερά βρίσκεται το σημείο $\boldsymbol{(x, y)}$ στην οθόνη που κοιτάει ο χρήστης σε σχέση με την κάμερα, ενώ το d_y δηλώνει το πόσο πάνω ή κάτω από την κάμερα κοιτάει ο χρήστης.

4.2 Εύρεση σημείου ενδιαφέροντος στην οθόνη

Πριν υπολογίσουμε το σημείο $\boldsymbol{(x, y)}$ της οθόνης σε pixels, υπολογίζουμε τις μετατοπίσεις d_x, d_y (σε χιλιοστά του μέτρου), χρησιμοποιώντας την εφαπτομένη των $\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}}$ και το διάνυσμα $\boldsymbol{t}$ (σε χιλιοστά του μέτρου). Μόλις βρούμε τα (d_x, d_y), τα μετατρέπουμε από χιλιοστά σε pixels. Για να το πετύχουμε, κάνουμε μια προσέγγιση της αναλογίας pixels ανά χιλιοστό. Συγκεκριμένα, μετράμε με ένα χάρακα την οριζόντια και κάθετη διάσταση της οθόνης (σε χιλιοστά), ενώ ταυτόχρονα βρίσκουμε την ανάλυση της οθόνης σε pixels. Αν η οθόνη μας έχει διαστάσεις $m_x \times m_y$ χιλιοστά και $p_x \times p_y$ pixels, τότε οι αναλογίες που θέλουμε είναι

$$mmPerPixel_x = \frac{m_x}{p_x} \tag{4.1}$$

για την οριζόντια περίπτωση και

$$mmPerPixel_y = \frac{m_y}{p_y} \tag{4.2}$$

για την κατακόρυφη περίπτωση. Για παράδειγμα, εάν οι διαστάσεις είναι 3400mm×1900mm σε χιλιοστά και 1240×780 σε pixels, τότε οι αναλογίες χιλιοστών ανά pixel που προκύπτουν είναι 0.3647 οριζόντια και 0.4105 κατακόρυφα. Στη συνέχεια, από τους τύπους (4.3) και (4.4) υπολογίζουμε τις τιμές των (d_x, d_y) σε pixels χρησιμοποιώντας τους συντελεστές που υπολογίστηκαν από τους τύπους (4.1) και (4.2).

$$d_x(pixels) = \frac{d_x(mm)}{mmPerPixel_x} \tag{4.3}$$

$$d_y(pixels) = \frac{d_y(mm)}{mmPerPixel_y} \tag{4.4}$$

Τέλος, αφού υπολογίσουμε τα (d_x, d_y) σε pixels, ψάχνουμε τις συντεταγμένες $\boldsymbol{(x, y)}$ που δηλώνουν το κεντρικό pixel της περιοχής της οθόνης που κοιτάει ο χρήστης. Συγκεκριμένα,

για τα οριζόντια pixels έχουμε

$$x = \frac{MAX\_WIDTH}{2} + d_x \tag{4.5}$$

μιας και το d_x δηλώνει την μετατόπιση από τη μέση της οθόνης, ενώ για τα κατακόρυφα pixels έχουμε

$$y = d_y \tag{4.6}$$

αφού το d_y δηλώνει την κατακόρυφη μετατόπιση από το ύψος της κάμερας. Στους αλγορίθμους 1, 2 φαίνονται αναλυτικά οι υπολογισμοί των $\boldsymbol{(x, y)}$.

Algorithm 1: Υπολογισμός του x

```
Outputs: x ;
Inputs: φ, t_x, t_z ;
Parameter: MID_HOR_PIXEL=620 ; // Middle Horizontal Pixel
Parameter: MMs_PER_PIXEL=0.3647 ; // Milimeters per pixel (ratio)
if t_x < 0 and φ > 0 then
    d_x = −t_x + t_z tan(φ) ;
    x =MID_HOR_PIXEL − d_x/MMs_PER_PIXEL ;
else if t_x < 0 and φ < 0 then
    d_x = t_z tan(φ) ;
    x = MID_HOR_PIXEL + (t_x − d_x)/MMs_PER_PIXEL ;
else if t_x > 0 and φ > 0 then
    d_x = −t_z tan(φ) ;
    x = MID_HOR_PIXEL + (t_x + d_x)/MMs_PER_PIXEL ;
else if t_x > 0 and φ < 0 then
    d_x = t_x − t_z tan(φ); ;
    x = MID_HOR_PIXEL + d_x/MMs_PER_PIXEL
```

Algorithm 2: Υπολογισμός του y

Outputs: y ;
Inputs: θ, t_y, t_z ;
Parameter: MMs_PER_PIXEL=0.4105 ; `// Milimeters per pixel (ratio)`
if $t_y < 0$ *and* $\theta > 0$ **then**
    $d_y = t_z \tan(\theta) - t_y$;
    $y = d_y / MMs\_PER\_PIXEL$; ;
else if $t_y < 0$ *and* $\theta < 0$ **then**
    $d_y = -t_z \tan(-\theta)$;
    $y = d_y / MMs\_PER\_PIXEL$;
else if $t_y > 0$ *and* $\theta > 0$ **then**
    $d_y = -t_y - t_z \tan(-\theta)$; ;
    $y = d_y / MMs\_PER\_PIXEL$
else if $t_y > 0$ *and* $\theta < 0$ **then**
    $d_y = -t_y + t_z \tan(-\theta)$;
    $y = d_y / MMs\_PER\_PIXEL$;

4.3 Λογισμικό εφαρμογής

Όλα τα στάδια της παραπάνω εφαρμογής υλοποιήθηκαν στη γλώσσα προγραμματισμού C++, εκτός από την εκπαίδευση του προτενόμενου δικτύου που έγινε στη γλώσσα Python. Επιλέχθηκε η γλώσσα C++, καθώς πέραν από την ευστοχία των μετρήσεων, βασικό κριτήριο ήταν και η ταχύτητα εκτέλεσης.

Αρχικά, για την προεπεξεργασία των δεδομένων που αναφέραμε στο κεφάλαιο 2 χρησιμοποιήθηκαν τα εργαλεία OpenCV [18] και Dlib [13]. Στην πορεία, μέσω της διεπαφής προγραμματισμού[1] του PyTorch (C++ API), φορτώσαμε το εκπαιδευμένο μοντέλο στην εφαρμογή. Τέλος, χρησιμοποιήσαμε τη βιβλιοθήκη SDL[2] ώστε να φτιάξουμε ένα γραφικό περιβάλλον το οποίο θα αποτυπώνει το σημείο που κοιτάει ο χρήστης στην οθόνη.

Ο κώδικας της παραπάνω εφαρμογής βρίσκεται στο Github[3].

[1] https://pytorch.org/tutorials/advanced/cpp_export.html
[2] https://www.libsdl.org/download-2.0.php
[3] https://github.com/caxelos/MPIIGaze/blob/hpc-code/webcam_face_pose_ex.cpp

Κεφάλαιο 5

Πειράματα και Αποτελέσματα

Στο κεφάλαιο αυτό θα αξιολογήσουμε τον αλγόριθμο ResNet-20 που αναλύσαμε στο κεφάλαιο 3 και θα συγκρίνουμε την προτεινόμενη λύση μας με τις πιο πρόσφατες λύσεις της βιβλιογραφίας. Θα ακολουθήσουμε τον ίδιο τρόπο αξιολόγησης με το [9] και θα κάνουμε χρήση των βάσεων δεδομένων MPIIGaze [9] και UT Multiview [7]. Από το MPIIGaze λαμβάνουμε 45,000 τυχαία δείγματα, 3000 από κάθε συμμετέχοντα (15 συμμετέχοντες). Από το UT Multiview λαμβάνουμε 64,000 τυχαία δείγματα, 1280 ανά συμμετέχοντα (50 συμμετέχοντες).

5.1 Βάσεις δεδομένων MPIIGaze και UT Multiview

Παρακάτω περιγράφουμε αναλυτικά τον τρόπο συλλογής των δεδομένων που χρησιμοποιήσαμε για το στάδιο της εκπαίδευσης και της αξιολόγησης. Σημειώνουμε πως τα δεδομένα των βάσεων αυτών δεν είναι τα τελικά δεδομένα που χρησιμοποιήσαμε, καθώς εφαρμόσαμε πάνω σε αυτά προεπεξεργασία με χρήση του μετασχηματισμού που αναφέραμε στην ενότητα 2.3.

5.1.1 UT Multiview

Η βάση δεδομένων UT Multiview [7] αποτελεί μια βάση από δεδομένα που αποκτήθηκαν σε εργαστηριακές συνθήκες. Περιέχει δεδομένα τα οποία συλλέχθηκαν από 50 συμμετέχοντες

(35 άνδρες, 15 γυναίκες), ηλικίας 20 μέχρι 40 ετών. Κατά τη διάρκεια συλλογής των δεδομένων, τα κεφάλια των συμμετεχόντων ήταν στερεωμένα και απείχαν μια σταθερή απόσταση 60 εκατοστών από μια οθόνη LCD. Γύρω από την οθόνη υπήρχαν στερεωμένες οχτώ 1.3 megapixel, PointGrey Flea3 USB3.0 κάμερες, οι οποίες ήταν συγχρονισμένες από έναν υπολογιστή ώστε να λαμβάνουν ταυτόχρονα φωτογραφίες (εικόνες 1, 2 στο [7]). Πριν τη λήψη φωτογραφιών, έχει προηγηθεί ρύθμιση των εσωτερικών και εξωτερικών παραμέτρων για κάθε ένα συμμετέχοντα του UT Multiview (calibration of intrinsic and extrinsic parameters).

Η συλλογή των δεδομένων έγινε ως εξής. Οι συμμετέχοντες έπρεπε να κοιτάνε μια κουκκίδα στην οθόνη, η οποία ανά κάποιο χρονικό διάστημα άλλαζε θέση τυχαία. Οι λήψεις κάθε κάμερας λαμβάνονταν υπ΄όψην ως ξεχωριστό δείγμα. Τέλος, οι γωνίες κατεύθυνσης βλέμματος κυμαίνονταν στο εύρος $[-25°, +25°]$ οριζόντια και $[-15°, +15°]$ κατακόρυφα.

5.1.2 MPIIGaze

Η βάση δεδομένων MPIIGaze [9] αποτελεί μια βάση από δεδομένα που αποκτήθηκαν σε **μη** εργαστηριακές συνθήκες (in-the-wild). Αντίθετα με το UT Multiview, στόχος εδώ είναι η δημιουργία μιας βάσης από δεδομένα που συλλέχθηκαν υπό διαφορετικές συνθήκες μεταξύ τους. Για αυτόν το λόγο, η συλλογή δεδομένων διήρκησε αρκετό διάστημα, από εννιά μέρες ως τρεις μήνες για μερικούς συμμετέχοντες, ώστε να καλύπτονται οι διαφορετικές συνθήκες συλλογής δεδομένων, όπως η τοποθεσία, η ώρα, η φωτεινότητα, ακόμα και οι διαφοροποιήσεις του ίδιου του ματιού.

Τα δεδομένα σε αυτήν την περίπτωση συλλέχθηκαν από τις κάμερες των λάπτοπ των συμμετεχόντων κατά τη διάρκεια της μέρας. Στο λάπτοπ κάθε συμμετέχοντα υπήρχε εγκατεστημένη μια εφαρμογή που έτρεχε στο παρασκήνιο και ανά δέκα λεπτά ζητούσε από τους χρήστες να κοιτάξουν μια σειρά από είκοσι κουκκίδες σε τυχαία σημεία στην οθόνη. Οι γωνίες κατεύθυνσης βλέμματος κυμαίνονταν στο εύρος $[-18°, +18°]$ οριζόντια και $[-1.5°, +20°]$ κατακόρυφα.

5.2 Αξιολόγηση μεθόδων

Θα πραγματοποιήσουμε τρία είδη αξιολογήσεων. Στην πρώτη, η εκπαίδευση γίνεται με 64,000 τυχαία δείγματα της βάσης UT Multiview [7] και η αξιολόγηση με 45,000 τυχαία δείγματα της βάσης MPIIGaze [9] (cross-dataset evaluation). Στη δεύτερη, χρησιμοποιούμε μόνο τη βάση δεδομένων MPIIGaze και εξετάζουμε κάθε συμμετέχοντα ξεχωριστά στο δίκτυο που έχει εκπαιδευτεί με δεδομένα από τους υπόλοιπους δεκατέσσερεις συμμετέχοντες, με 3,000 δείγματα από τον καθένα (leave-one-person-out). Τέλος, πραγματοποιούμε αξιολόγηση παρόμοια με τη δεύτερη, μόνο που αυτήν τη φορά τα δεδομένα εκπαίδευσης δεν προέρχονται από τα υπόλοιπα δεκατέσσερα άτομα, αλλά από το ίδιο το άτομο που εξετάζεται κάθε φορά (person-specific validation). Επιπρόσθετα, κάνουμε μια αύξηση χιλίων δειγμάτων μέσο όρο ανά άτομο σε σχέση με το δεύτερο σενάριο (συνολικά ανά άτομο 4,000 δείγματα μέσο όρο).

Για την αξιολόγηση υπολογίζουμε το μέσο ευκλίδειο σφάλμα όλων των δειγμάτων, όπως φαίνεται στον τύπο (5.1)

$$mean\ test\ error = \frac{\sum_{i=1}^{N} \sqrt{(\hat{\phi}_{pred}^{(i)} - \hat{\phi}_{target}^{(i)})^2 + (\hat{\theta}_{pred}^{(i)} - \hat{\theta}_{target}^{(i)})^2}}{N}, \tag{5.1}$$

όπου $\hat{\boldsymbol{\phi}}_{\boldsymbol{pred}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{pred}}$ είναι οι γωνίες βλέμματος που προβλέψαμε και $\hat{\boldsymbol{\phi}}_{\boldsymbol{target}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{target}}$ οι πραγματικές γωνίες βλέμματος, ενώ το N εκφράζει τον αριθμό των δειγμάτων αξιολόγησης (test samples).

Τέλος, οι υπερπαράμετροι εκπαίδευσης που χρησιμοποιήθηκαν και στα τρία σενάρια αξιολόγησης που αναφέραμε φαίνονται στον πίνακα 5.1. Μετά από μετρήσεις, στην περίπτωση του *cross-dataset* evaluation η βέλτιστη τιμή του μεγέθους σακιδίου (batch size) είναι 32, διότι υπάρχει μεγάλο πλήθος δεδομένων εκπαίδευσης σε αυτήν την περίπτωση. Στην περίπτωση του *leave-one-person-out* evaluation, ως βέλτιστος αριθμός του μεγέθους σακιδίου υπολογίστηκε το 16, ενώ στην περίπτωση του *person-specific* evaluation αυτό τέθηκε ίσο με 4, μιας και στην περίπτωση αυτήν τα δεδομένα εκπαίδευσης ήταν πολύ λιγότερα σε σχέση με τις άλλες δύο αξιολογήσεις. Τέλος, ο αριθμός των εποχών ανά σενάριο αξιολόγησης διαφέρει. Σε όλες τις περιπτώσεις αξιολόγησης, συνεχίζαμε την εκπαίδευση, έως ότου παρατηρηθεί σύγκλιση στο μέσο σφάλμα (mean test error). Το μέσο σφάλμα υπολογίζεται από τις τελευταίες 5 εποχές από την στιγμή που ξεκινάει να υπάρχει σύγκλιση.

hyperparameters	Cross-dataset Evaluation	Leave-one-person-out Evaluation	Person-specific Evaluation
Batch size	32	16	4
Epochs	40	25	30
Initial learning rate	0.0001	0.0001	0.0001
Every 10 epochs:	learning rate/10	learning rate/10	learning rate/10
Momentum:	0.9	0.9	0.9
Optimizer:	Nesterov	Nesterov	Nesterov

Πίνακας 5.1: Οι υπερπαράμετροι εκπαίδευσης του προτεινόμενου δικτύου ResNet-20.

methods	mean error	standard deviation
Regression Forests [7] (2014)	15.4°	4.5°
Mnist Net [8] (2015)	13.9°	2.5°
GazeNet [9] (2018)	**9.8°**	2.4°
ResNet-20 (ours)	12.45°	**2.21°**

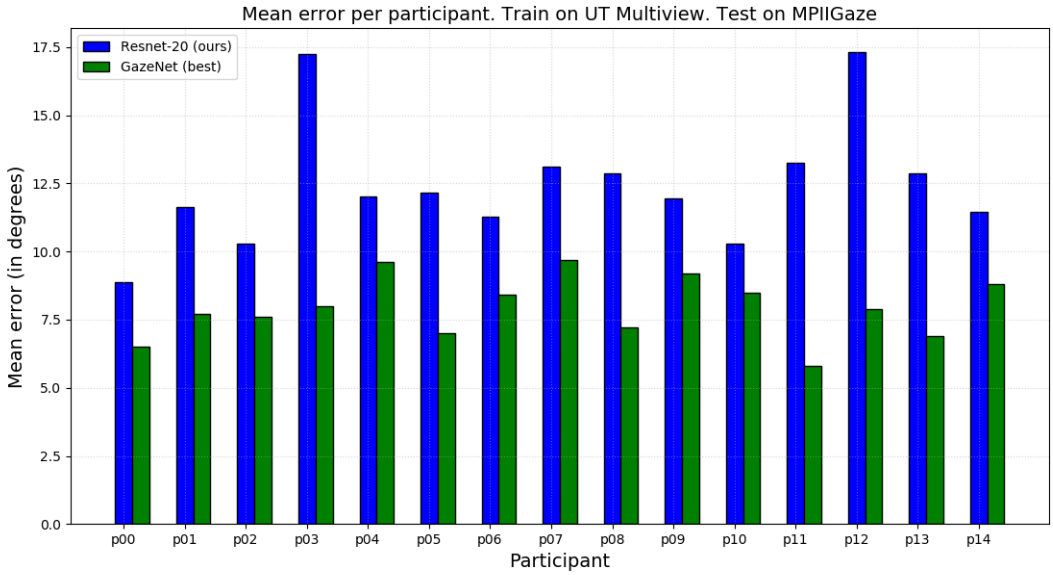
Πίνακας 5.2: Εκπαίδευση μοντέλων στο UT Multiview και αξιολόγηση στο MPIIGaze (cross-dataset evaluation). Οι τιμές εκφράζουν το μέσο σφάλμα και την τυπική απόκλιση ανάμεσα στους 15 συμμετέχοντες του MPIIGaze.

5.2.1 Εκπαίδευση στο UT Multiview και αξιολόγηση στο MPIIGaze

Αρχικά θα κάνουμε την εκπαίδευση του μοντέλου με 64,000 δείγματα από το UT Multiview και θα ελέγξουμε την ευστοχία του μοντέλου με 45,000 δείγματα από το MPIIGaze (cross-dataset evaluation). Η περίπτωση αυτή αποτελεί το δυσκολότερο σενάριο όσον αφορά την ευστοχία των προβλέψεων. Στον πίνακα 5.2 φαίνεται το μέσο σφάλμα όλων των συμμετεχόντων του MPIIGaze, ενώ στην εικόνα 5.1 φαίνεται το μέσο σφάλμα κάθε συμμετέχοντα του MPIIGaze. Ενώ το μέσο σφάλμα του αλγορίθμου μας (12.45°) είναι μικρότερο από το αντίστοιχο των περισσοτέρων μεθόδων της βιβλιογραφίας, είναι μεγαλύτερο κατά **21.3%** του GazeNet [9] (9.8°).

5.2.2 Αξιολόγηση ανά συμμετέχοντα στο MPIIGaze

Στη συνέχεια, θα πραγματοποιήσουμε μετρήσεις στο MPIIGaze, χρησιμοποιώντας την *leave-one-person-out* αξιολόγηση σε 45,000 δείγματα. Σε κάθε βήμα θα εξετάζεται

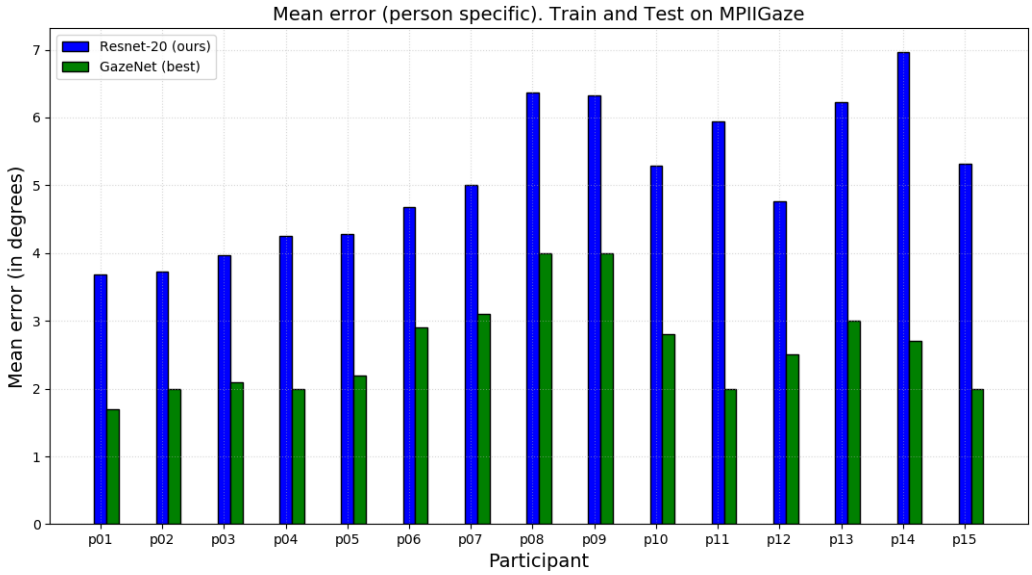


Σχήμα 5.1: Εκπαίδευση μοντέλων στο UT Multiview και αξιολόγηση στο MPIIGaze (cross-dataset evaluation). Οι τιμές εκφράζουν το μέσο σφάλμα καθενός από τους 15 συμμετέχοντες του MPIIGaze.

methods	mean error	standard deviation
Regression Forests [7] (2014)	6.7°	0.7°
Mnist Net [8] (2015)	6.3°	0.6°
GazeNet [9] (2018)	**5.5°**	**0.5°**
ResNet-20 (ours)	6.95°	0.65°

Πίνακας 5.3: Εκπαίδευση και αξιολόγηση μοντέλων στο MPIIGaze, όπου κάθε φορά αξιολογούμε ένα άτομο σε δίκτυο, που έχει εκπαιδευτεί με δεδομένα των υπόλοιπων δεκατεσσάρων ατόμων (leave-one-person-out evaluation). Οι τιμές εκφράζουν το μέσο σφάλμα και την τυπική απόκλιση ανάμεσα στους 15 συμμετέχοντες του MPIIGaze.

ένας συμμετέχοντας και το δίκτυο θα εκπαιδεύεται από τους υπόλοιπους δεκατέσσερεις. Τα συγκεντρωτικά αποτελέσματα φαίνονται στον πίνακα 5.3. Στην περίπτωση αυτή ο αλγόριθμός μας δεν καταφέρνει τα επιθυμητά αποτελέσματα, μιας και το μέσο σφάλμα είναι αρκετά υψηλό (6.95°) σε σχέση με τα αντίστοιχα σφάλματα των αλγορίθμων της βιβλιογραφίας.



Σχήμα 5.2: Εκπαίδευση και αξιολόγηση μοντέλων με δεδομένα από το ίδιο άτομο της βάσης MPIIGaze (person-specific evaluation). Οι τιμές εκφράζουν το μέσο σφάλμα καθενός από τους 15 συμμετέχοντες του MPIIGaze.

5.2.3 Αξιολόγηση και εκπαίδευση ανά συμμετέχοντα στο MPIIGaze

Τέλος, πραγματοποιούμε μετρήσεις στο MPIIGaze όπως στην ενότητα 5.2.2, με τη διαφορά ότι σε αυτήν την περίπτωση τα δεδομένα εκπαίδευσης προέρχονται από το ίδιο το άτομο που αξιολογούμε και όχι από τα δεκατέσσερα υπόλοιπα άτομα (*person-specific* evaluation). Συγκεκριμένα, χρησιμοποιούμε την αξιολόγηση *5-fold cross validation*, με το 80% των δεδομένων να αποτελούν τα δεδομένα εκπαίδευσης και το 20% τα δεδομένα αξιολόγησης. Το μέσο σφάλμα καθενός συμμετέχοντα φαίνεται στο σχήμα 5.2. Σε αυτήν την περίπτωση οι προβλέψεις είναι σαφώς πιο εύκολες, καθώς τόσο η εκπαίδευση όσο και ο έλεγχος γίνονται με δεδομένα από το ίδιο άτομο μόνο. Ωστόσο η επίδοση της μεθόδου μας απέχει αρκετά από την επίδοση της καλύτερης μεθόδου (GazeNet [9]) στο σενάριο αυτό.

5.3 Συμπεράσματα

Οι αρχικές εκτιμήσεις ήταν η απόκτήση μιας ελαφρώς καλύτερης επίδοσης σε σχέση με τις προτεινόμενες αρχιτεκτονικές της βιβλιογραφίας, λόγω των μονοπατιών παράκαμψης που

διαθέτουν τα δίκτυα ResNet, καθώς και της ικανότητας τους να εξαλείφουν σε μεγάλο βαθμό το πρόβλημα της σχεδόν μηδενικής ανανέωσης των βαρών στα αρχικά συνελικτικά στρώματα (vanishing problem). Ωστόσο, σε κανένα από τα τρία σενάρια αξιολόγησης δεν ξεπέρασε η προτεινόμενη μέθοδος την καλύτερη επίδοση της βιβλιογραφίας, παρόλο που στο σενάριο της *cross-dataset* αξιολόγησης είχαμε μια αρκετά καλή επίδοση.

Οι τρεις μέθοδοι αξιολόγησης στην ενότητα 5.2 μας έδειξαν ότι η επίδοση του προτεινόμενου δικτύου εξαρτάται από το είδος της αξιολόγησης που ακολουθούμε. Η προτεινόμενη μέθοδος ResNet-20 έχει καλύτερα αποτελέσματα στο πρώτο σενάριο (cross-dataset), όπου η εκπαίδευση και η αξιολόγηση γίνονται σε διαφορετικές βάσεις δεδομένων, ενώ στα σενάρια αξιολόγησης *leave-one-person-out* και *person-specific*, η επίδοση δεν ήταν η αναμενόμενη. Ένας από τους πιθανούς λόγους της καλύτερης επίδοσης σε αυτήν την περίπτωση είναι το μεγάλο πλήθος δεδομένων εκπαίδευσης που υπάρχει στο σενάριο αυτό (64,000 δείγματα) σε σχέση τα άλλα δύο σενάρια (42,000 δείγματα στο σενάριο *leave-one-person-out* και 1,250 δείγματα ανά άτομο στο σενάριο *person-specific*). Η αρχιτεκτονική που χρησιμοποιήσαμε διαθέτει μεγαλύτερο βάθος σε σχέση με τις υπόλοιπες αρχιτεκτονικές της βιβλιογραφίας. Επομένως, είναι πολύ πιθανό να χρειάζεται περισσότερα δεδομένα για να δώσει καλύτερα αποτελέσματα σε σχέση με τις μεθόδους της βιβλιογραφίας.

Κεφάλαιο 6

Επίλογος και Μελλοντική Δουλειά

Στην εργασία μας δοκιμάστηκε η μέθοδος ResNet-20, η οποία προέκυψε έπειτα από πειραματισμούς πάνω σε διάφορες παραμέτρους του δικτύου *ResNet*. Παρά το γεγονός ότι το προτεινόμενο δίκτυο δεν πέτυχε την επιθυμητή επίδοση στις αξιολογήσεις *leave-one-person-out* και *person-specific*, είχε αξιοσημείωτη επίδοση στην αξιολόγηση *cross-dataset*, όντας το δίκτυο με το δεύτερο μικρότερο μέσο σφάλμα στη βιβλιογραφία, ίσο με 12.45° (βλ. πίνακα 5.2). Φυσικά υπάρχουν αρκετοί ακόμη παράγοντες ή τεχνικές που δεν δοκιμάσαμε στα πλαίσια της διπλωματικής εργασίας, οι οποίοι όμως θα μπορούσαν να αυξήσουν την επίδοση. Στη συνέχεια παραθέτουμε μερικές από τις σκέψεις μας πάνω σε αυτό το κομμάτι.

Αρχικά μπορούμε να πειραματιστούμε με κάποια διαφορετική αρχιτεκτονική δικτύου ή κάποιον άλλο αλγόριθμο μηχανικής μάθησης. Αρκετά αξιόλογο φαίνεται το δίκτυο *ResNetXt* [25], το οποίο δανείζεται στοιχεία από τα δίκτυα ResNet και VGG [26]. Στο δίκτυο αυτό, ένα κύριο μονοπάτι μπορεί να δημιουργήσει *μονοπάτια-παρακλάδια* τα οποία είτε αθροίζονται είτε συγχωνεύονται μεταξύ τους (cardinality increase). Ακολουθεί την προσέγγιση *network-in-neuron*, όπου αντικαθίσταται το συμβατικό εσωτερικό γινόμενο μεταξύ εισόδων και βαρών ενός στρώματος με το γνωστό μπλοκ των δικτύων ResNet [11] (bottleneck εκδοχή), το οποίο εφαρμόζεται σε κάθε μονοπάτι-παρακλάδι. Τέλος, χρησιμοποιεί συνελίξεις με φίλτρα που έχουν ίδιες μεταξύ τους διαστάσεις, όπως γίνεται δηλαδή και στο δίκτυο VGG.

Άλλη μια αλλαγή που παρουσιάζει αρκετό ενδιαφέρον είναι η μετατροπή του προβλήματος από πρόβλημα *προσέγγισης* (regression) σε πρόβλημα *ταξινόμησης* (classification). Αυτό μπορεί να γίνει μέσω ενός νευρωνικού δικτύου που εκτελεί ταξινόμηση και στο τελευταίο του στρώμα

δεν καλεί απλώς την *softmax* που αποκρύπτει αρκετές πληροφορίες σχετικά με την τελική πρόβλεψη, αλλά χρησιμοποιεί την τεχνική *ordinal classification* [27], η οποία αντιμετωπίζει ένα πρόβλημα προσέγγισης ως πρόβλημα ταξινόμησης. Τέτοιο δίκτύο αποτελεί το [28]. Για παράδειγμα, αντί η εξόδος του δικτύου να είναι μία γωνία, μπορούμε να θεωρήσουμε την έξοδο ως ένα διάνυσμα από γειτονικά εύρη γωνιών. Στόχος του δικτύου είναι να προβλέψει σε ποιό εύρος γωνιών ανήκει η γωνία που ψάχνουμε.

Τέλος, όσον αφορά τα δεδομένα εισόδου, μπορούμε να πραγματοποιήσουμε επιπρόσθετες αλλαγές, οι οποίες πολύ συχνά έχουν καλύτερα αποτελέσματα. Για παράδειγμα, μπορεί να γίνει η επιλογή κάποιου άλλου μετασχηματισμού των δεδομένων αντί αυτού που έγινε στην ενότητα 2.3 και τα δεδομένα να είναι κατανεμημένα πιο ομοιόμορφα.

Βιβλιογραφία

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "ImageNet Large Scale Visual Recognition Challenge", *CoRR*, vol. abs/1409.0575, 2014. arXiv: 1409.0575. [Online]. Available: http://arxiv.org/abs/1409.0575.

[2] P. Majaranta and A. Bulling, "Eye Tracking and Eye-Based Human-Computer Interaction", in *Advances in Physiological Computing*, ser. Human-Computer Interaction Series, S. H. Fairclough and K. Gilleade, Eds., London: Springer, 2014, ch. 3, pp. 39–65, ISBN: 978-1-4471-6391-6. DOI: 10.1007/978-1-4471-6392-3_3.

[3] Y. Sugano, X. Zhang, and A. Bulling, "AggreGaze: Collective Estimation of Audience Attention on Public Displays", in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ser. UIST '16, Tokyo, Japan: Association for Computing Machinery, 2016, pp. 821–831, ISBN: 9781450341899. DOI: 10.1145/2984511.2984536. [Online]. Available: https://doi.org/10.1145/2984511.2984536.

[4] J. Kim, M. Stengel, A. Majercik, S. De Mello, D. Dunn, S. Laine, M. McGuire, and D. Luebke, "NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation", in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19, Glasgow, Scotland UK: Association for Computing Machinery, 2019, ISBN: 9781450359702. DOI: 10.1145/3290605.3300780. [Online]. Available: https://doi.org/10.1145/3290605.3300780.

[5] K. A. Funes Mora and J. Odobez, "Person independent 3D gaze estimation from remote RGB-D cameras", in *2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 2787–2791. DOI: 10.1109/ICIP.2013.6738574.

[6] T. Schneider, B. Schauerte, and R. Stiefelhagen, "Manifold Alignment for Person Independent Appearance-Based Gaze Estimation", in *2014 22nd International*

Conference on Pattern Recognition, Aug. 2014, pp. 1167–1172. DOI: 10.1109/ICPR.2014.210.

[7] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation", in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1821–1828. DOI: 10.1109/CVPR.2014.235.

[8] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 4511–4520. DOI: 10.1109/CVPR.2015.7299081.

[9] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 162–175, Jan. 2019, ISSN: 1939-3539. DOI: 10.1109/tpami.2017.2778103. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2017.2778103.

[10] E. T. Wong, S. Yean, Q. Hu, B. S. Lee, J. Liu, and R. Deepu, "Gaze Estimation Using Residual Neural Network", in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2019, pp. 411–414. DOI: 10.1109/PERCOMW.2019.8730846.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[12] S. Mallick, *Head Pose Estimation using OpenCV and Dlib*, https://www.learnopencv.com/head-pose-estimation-using-opencv-and-dlib/.

[13] D. E. King, "Dlib-Ml: A Machine Learning Toolkit", *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Dec. 2009, ISSN: 1532-4435.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, Jun. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[15] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge", in *2013 IEEE International Conference on Computer Vision Workshops*, Dec. 2013, pp. 397–403. DOI: 10.1109/ICCVW.2013.59.

[16] G. Tzimiropoulos, J. Alabort-i-Medina, S. P. Zafeiriou, and M. Pantic, "Active Orientation Models for Face Alignment In-the-Wild", *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2024–2034, Dec. 2014, ISSN: 1556-6021. DOI: 10.1109/TIFS.2014.2361018.

[17] M. Lourakis, "A Brief Description of the Levenberg-Marquardt Algorithm Implemened by Levmar", *A Brief Description of the Levenberg-Marquardt Algorithm Implemented by Levmar*, vol. 4, Jan. 2005.

[18] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, 2000.

[19] X. Zhang, Y. Sugano, and A. Bulling, "Revisiting Data Normalization for Appearance-Based Gaze Estimation", in *Proceedings of the 2018 ACM Symposium on Eye Tracking Research Applications*, ser. ETRA '18, Warsaw, Poland: Association for Computing Machinery, 2018, ISBN: 9781450357067. DOI: 10.1145/3204493.3204548. [Online]. Available: https://doi.org/10.1145/3204493.3204548.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *CoRR*, vol. abs/1502.03167, 2015.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[23] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning", in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[25] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 5987–5995. DOI: 10.1109/CVPR.2017.634.

[26] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size", in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov. 2015, pp. 730–734. DOI: 10.1109/ACPR.2015.7486599.

[27] E. Frank and M. Hall, "A Simple Approach to Ordinal Classification", vol. 2167, Aug. 2001, pp. 145–156. DOI: 10.1007/3-540-44795-4_13.

[28] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression", in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1279–1284. DOI: 10.1109/IJCNN.2008.4633963.