Gesture recognizition and haptic feedback: Rock Paper Scissors Game

Διαδραστικές Τεχνολογίες – Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Κοσμάς Αρχοντής
1084020

Μαρία Ασπιώτη
1083881

Βασιλική Κασουρίδου
1083790

Γεώργιος Στεργιόπουλος
1083861

Περίληψη

Αυτή η εργασία, έγινε στα πλαίσια της διδασκαλίας του μαθήματος Διαδραστικές Τεχνολογίες και έχει σκοπό την εξοικείωση μας με διάφορες μορφές αλληλεπίδρασης ανθρώπου-μηχανής. Συγκεκριμένα, μέσω της υλοποίησης ενός παιχνιδιού Πέτρα-Ψαλίδι-Χαρτί, αξιοποιούμε τεχνικές αναγνώρισης χειρονομιών μέσω Υπολογιστικής Όρασης και συνελικτικών νευρωνικών δικτύων και προσφέρομε στον χρήστη απτική ανάδραση, ενημερώνοντας τον για την εξέλιξη του παιχνιδιού.

GitHub Repository Rock Paper Scissors Game

Keywords—HCI, Gestures, Gesture Recognition, GUI, RPS, Artificial Intelligence

1. ΕΙΣΑΓΩΓΗ

Το πρόγραμμα του οποίου η μέθοδος ανάπτυξης περιγράφεται παρακάτω έχει σκοπό την ανάδειξη αναγνώρισης κοινών χειρονομιών από τον υπολογιστή μέσω ανάλυσης εικόνας και νευρωνικών δικτύων, οπτική παρουσίαση αυτής καθώς και απτική ανάδραση μέσω μιας συσκευής χεριού. Αναπτύχθηκε στη Python και με τις σχετικές βιβλιοθήκες της.

Χρησιμοποιήθηκε αρχιτεκτονική MVC (Model-View-Controller) για την οργάνωση του κώδικα, επαυξημένο στη λογική με Controller που επικοινωνεί με τη κάμερα, τη συσκευή απτικής αλληλεπίδρασης και το μοντέλο TN, κάτι που επέτρεψε στην ομάδα να διαχωρίσει αποτελεσματικά τις εργασίες και να αναπτύξει παράλληλα κώδικα.

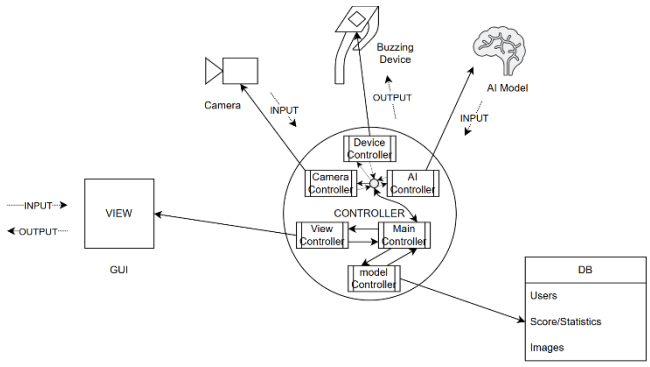

Figure 1: Αρχιτεκτονική MVC του προγράμματος

2. ΤΟ ΜΟΝΤΕΛΟ ΑΝΑΓΝΩΡΙΣΗΣ ΧΕΙΡΟΝΟΜΙΩΝ

Το σύστημα αναγνώρισης χειρονομιών για το παιχνίδι "Πέτρα, Ψαλίδι, Χαρτί" αναπτύχθηκε χρησιμοποιώντας το πακέτο MediaPipe Model Maker. Αυτή η λύση χαμηλού κώδικα επιτρέπει την αποδοτική προσαρμογή μοντέλων μηχανικής μάθησης για ενσωματωμένες συσκευές. Το πακέτο αυτό χρησιμοποιεί ένα συνελικτικό νευρωνικό δίκτυο (CNN) για την ανάλυση δεδομένων από σημεία αναφοράς στο χέρι, καθιστώντας το ιδιαίτερα κατάλληλο για εργασίες αναγνώρισης χειρονομιών.

Ο αρχικός μας στόχος ήταν να βελτιώσουμε την απόδοση του προεκπαιδευμένου μοντέλου MediaPipe εκπαιδεύοντάς το με επιπλέον δεδομένα χειρονομιών. Ακολουθήσαμε τη διαδικασία του MediaPipe Model Maker βήμα προς βήμα:

1. *Προετοιμασία Συνόλου Δεδομένων*
   Το σύνολο δεδομένων αντλήθηκε από το HaGRID dataset και εμπλουτίστηκε με συλλεγμένες εικόνες για να βελτιωθεί η ανθεκτικότητα του μοντέλου σε διάφορες συνθήκες, όπως διαφορετικούς προσανατολισμούς χεριών, μεγέθη και φωτισμούς.
   Η διαδικασία προεπεξεργασίας του MediaPipe εντόπιζε αυτόματα τα σημεία αναφοράς στο χέρι από τις εικόνες, παραλείποντας όσες δεν περιείχαν ανιχνεύσιμα χέρια.

2. *Διαχωρισμός Συνόλου Δεδομένων*
   Το σύνολο δεδομένων χωρίστηκε σε 80% για εκπαίδευση (training set), 10% για επικύρωση (validation set) και 10% για δοκιμή (test set), ώστε να διασφαλιστεί αξιόπιστη αξιολόγηση του μοντέλου.

3. *Εκπαίδευση Μοντέλου*
   Εκπαιδεύσαμε το μοντέλο με διάφορες διαμορφώσεις υπερπαραμέτρων (hyperparameters), όπως ρυθμό μάθησης (learning rate), μέγεθος παρτίδας (batch size) και αριθμό εποχών (epochs). Εξερευνήθηκαν οι εξής διαμορφώσεις:
   - Ρυθμοί μάθησης: 0.001, 0.002, 0.003
   - Ποσοστά διαγραφής (dropout): 0.05, 0.1, 0.2
   - Εποχές: 10, 20, 30

   Δοκιμάσαμε επίσης επιπλέον κρυφά στρώματα χρησιμοποιώντας την παράμετρο layer_widths για τη διερεύνηση πιο βαθιών αρχιτεκτονικών μοντέλων.

4. *Αξιολόγηση Μοντέλου*
   Μετά την εκπαίδευση, τα μοντέλα αξιολογήθηκαν στο σύνολο δεδομένων δοκιμής, ενώ καταγράφηκαν μετρικές απόδοσης, όπως ακρίβεια και απώλεια (loss**)**.

Παρά τις προσπάθειές μας να βελτιώσουμε το μοντέλο με επιπλέον δεδομένα, το προεκπαιδευμένο μοντέλο το παρεχόμενο σύνολο δεδομένων πέτρα, ψαλίδι, χαρτί

παρουσίασε σταθερά καλύτερες επιδόσεις. Το προεκπαιδευμένο μοντέλο πέτυχε μεγαλύτερη ακρίβεια και ήταν πιο σταθερό σε διαφορετικά σύνολα επικύρωσης.

Οι βασικές υπερπαράμετροι που εξετάστηκαν περιλάμβαναν τον **ρυθμό μάθησης**, ο οποίος ρυθμίστηκε για ισορροπία μεταξύ ταχύτητας εκπαίδευσης και σύγκλισης, και το **ποσοστό διαγραφής**, που βελτίωσε τη γενίκευση αλλά σε υψηλές τιμές προκάλεσε υποεκπαίδευση. Το **μέγεθος παρτίδας** (2-4) αποδείχθηκε αποτελεσματικό λόγω του μικρού συνόλου δεδομένων, ενώ ο αριθμός **εποχών** έδειξε φθίνουσα απόδοση μετά από συγκεκριμένο σημείο, υποδεικνύοντας την ανάγκη για έγκαιρη διακοπή της εκπαίδευσης. Αυτές οι ρυθμίσεις μας βοήθησαν να βελτιστοποιήσουμε την απόδοση του μοντέλου.

Τελικά, το προεκπαιδευμένο μοντέλο MediaPipe και το παρεχόμενο σύνολο δεδομένων απέδωσαν τα καλύτερα αποτελέσματα, δείχνοντας την ανθεκτικότητα του συστήματος. Παρόλο που η προσαρμοσμένη εκπαίδευση δεν ξεπέρασε το βασικό μοντέλο, η διερεύνηση παρείχε πολύτιμες γνώσεις για τη συμπεριφορά του μοντέλου και την επίδραση των υπερπαραμέτρων.

Το επιλεγμένο μοντέλο, εξαγόμενο ως αρχείο TensorFlow Lite, χρησιμοποιείται πλέον στο σύστημα του παιχνιδιού μας για την αναγνώριση χειρονομιών σε πραγματικό χρόνο. Αυτή η ενσωμάτωση διασφαλίζει την ακριβή ανίχνευση χειρονομιών, αποτελώντας τη βάση για τη λογική του παιχνιδιού και τους μηχανισμούς απτικής ανάδρασης.

3. Η ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΚΑΙ ΤΟ ΠΑΙΧΝΙΔΙ

Η γραφική διεπαφή της εφαρμογής πραγματοποιήθηκε με τη βιβλιοθήκη **PyQt5** που επιτρέπει ραγδαία και απλή ανάπτυξη γραφικών διεπαφών. Παράλληλα, η πρόσβαση στην κάμερα του συστήματος εξασφαλίζεται μέσω της βιβλιοθήκης **OpenCV**. Για την αποτροπή «παγώματος» ή καθυστερήσεων στη κύρια διεπαφή, οι λειτουργίες της κάμερας και η λογική του παιχνιδιού εκτελούνται σε ξεχωριστό *νήμα* (thread), μέσω της κλάσης **QThread** της **PyQt5**. Η επικοινωνία μεταξύ του νήματος και της διεπαφής υλοποιείται μέσω της μετάδοσης *σημάτων* (signals), για την πραγματικού χρόνου ενημέρωση της διεπαφής. Τα σήματα είναι υπεύθυνα για την ζωντανή μετάδοση της ροής της κάμερας, την προβολή των αποτελεσμάτων κάθε γύρου και τον καθορισμό του νικητή του παιχνιδιού.

Στην πρώτη σελίδα της εφαρμογής, ο χρήστης ορίζει το σκορ που χρειάζεται να φτάσει ένας παίκτης για να αναδειχθεί νικητής του παιχνιδιού. Ακόμη, εφόσον το επιθυμεί, μπορεί να συνδεθεί ή να δημιουργήσει λογαριασμό, ώστε να αποθηκεύονται τα παιχνίδια του στη βάση δεδομένων. Ένας συνδεδεμένος χρήστης έχει πρόσβαση σε μία λίστα με τα προηγούμενα παιχνίδια του, προσφέροντας μία προσωποποιημένη εμπειρία.

Στη δεύτερη σελίδα, πραγματοποιείται το παιχνίδι. Μία *αντίστροφη μέτρηση* ενημερώνει τον χρήστη για την έναρξη κάθε γύρου, ώστε να προετοιμάσει την κίνησή του. Όταν λήξει η αντίστροφη μέτρηση, το **μοντέλο TN** αναγνωρίζει την χειρονομία του χρήστη, η οποία μετέπειτα συγκρίνεται με την τυχαία επιλεγμένη χειρονομία του υπολογιστή και καθορίζεται ο νικητής του γύρου. Η επιλογή του υπολογιστή απεικονίζεται στην διεπαφή με μία αντίστοιχη εικόνα, για μεγαλύτερη αμεσότητα. Ο παίκτης που θα φτάσει πρώτος στο καθορισμένο νικητήριο σκορ κερδίζει το παιχνίδι. Αν ο χρήστης είναι συνδεδεμένος, το παιχνίδι αποθηκεύεται στη βάση δεδομένων. Μετά τη λήξη του παιχνιδιού, ο χρήστης μπορεί να ξεκινήσει νέο παιχνίδι με τον υπολογιστή.

4. ΑΠΤΙΚΗ ΑΛΛΗΛΕΠΙΔΡΑΣΗ

Μόλις ολοκληρωθεί το παιχνίδι, με βάση όσα περιγράψαμε παραπάνω και ο νικητής (Υπολογιστής ή χρήστης) έχει αποφασιστεί, πέρα από την ενημέρωση του χρήστη μέσω της γραφικής διεπαφής, η εφαρμογή μας του παρέχει και απτική ενημέρωση. Πιο συγκεκριμένα, ο χρήστης που φοράει Smartwatch, λαμβάνει μια ειδοποίηση (οπού συνεπάγεται και δόνηση) όπου τον ενημερώνει για το αποτέλεσμα του παιχνιδιού. Αυτή η λειτουργία υλοποιήθηκε βασιζόμενοι σε μια εφαρμογή δημιουργίας αυτοματισμών (IFTTT), η οποία παρέχει την δυνατότητα δημιουργίας Webhook, και καθορισμό συμβάντων που θα συμβούν όταν γίνει trigger το Webhook. Στον κώδικα της εφαρμογής μας λοιπόν, ο device Controller αναλαμβάνει τον ρόλο να στείλει ανάλογα το αποτέλεσμα του παιχνιδιού, ένα GET request (HTTP), το οποίο με την σειρά του δημιουργεί ειδοποίηση με το επιθυμητό μήνυμα τόσο στο κινητό, όσο και στο Smartwatch.

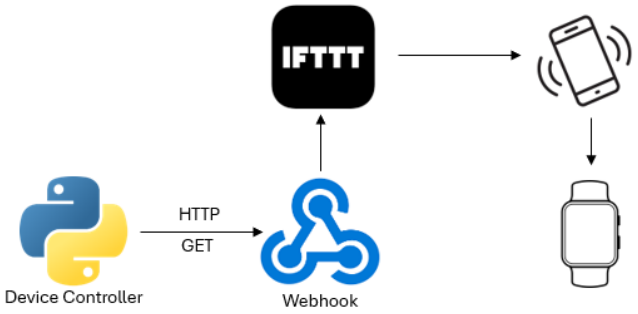

Figure : Αρχιτεκτονική Απτικής Αλληλεπίδρασης

Για την υλοποίηση του συγκεκριμένου σημείου έγιναν και άλλες προσεγγίσεις όπως ο απευθείας έλεγχος της wearable συσκευής μέσω Bluetooth, άλλα λόγο των περιορισμών από τις εταιρίες των κατασκευαστών, αυτό δεν κατέστη δυνατό και καταφύγαμε στην παραπάνω λύση.

Παρακάτω βλέπουμε τις ειδοποιήσεις σε κινητό και smartwatch, σε περίπτωση νίκης και ήττας.