

Μάιος 2021

Ατομική Διπλωματική Εργασία

**Καταγραφή κίνησης από εφαρμογή εικονικής
πραγματικότητας από υλικό Arduino**

Ανδρέας Λοΐζου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Καταγραφή κίνησης από εφαρμογή εικονικής
πραγματικότητας από υλικό *Arduino***

Ανδρέας Λοΐζου

Επιβλέπων Καθηγητής
Δρ. Κωνσταντίνος Σ. Παττίχης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάιος 2021

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω το μέντορά μου στην συγκεκριμένη πτυχιακή εργασία Δρ. Κωνσταντίνο Σ. Παττίχη, για την βοήθεια και την σωστή καθοδήγηση που μου πρόσφερε καθώς όλη την διάρκεια της πτυχιακής εργασίας τόσο για την υλοποίηση της εφαρμογής, όσο και για τη συγγραφή της παρούσας εργασίας. Επίσης μεγάλες ευχαριστίες και στον Δρ. Ανδρέας Αριστείδου για την άψογη συνεργασία και για την βοήθεια που μου έδωσε για να κατανοήσω τον αλγόριθμο του MotioNet και να μπορέσω να τον υλοποιήσω, επίσης τον ευχαριστώ πολύ που με έφερε σε επαφή μαζί με το άτομο που υλοποίησε τον αλγόριθμο για να μου απαντήσει περεταίρω απορίες. Ευχαριστίες στους 8 συμφοιτητές μου και φίλους που μόλις χρειάστηκα την βοήθεια τους για να εισπράξω αποτελέσματα ήρθαν αμέσως χωρίς κάποιο δισταγμό.

Τέλος μεγάλες ευχαριστίες θέλω να δώσω στους γονιούς μου Γιώργο και Ευτυχία Λοϊζου που με βοήθησαν και με στήριξαν σε όλη την προπτυχιακή μου φοιτητική προσπάθεια. Επίσης την συγκεκριμένη πτυχιακή την αφιερώνω στα δύο αυτά άτομα που δεν έπαψαν ποτέ να πιστεύουν σε εμένα και να με στηρίζουν.

Περίληψη

Τα τελευταία χρόνια η εικονική πραγματικότητα και κάποια από τα πεδία της βλέπουμε να εμφανίζονται και να αναπτύσσονται συνεχώς στην καθημερινότητα μας. Αυτό οφείλεται στην μεγάλη ανάγκη του ανθρώπου να δίνει ορισμένες εργασίες στο συγκεκριμένο τομέα και να συνεργάζεται και με άλλους τομείς της πληροφορικής όπως η τεχνίτη νοημοσύνη και τεχνικών της μηχανικής μάθησης για καλύτερη αξιολόγηση των αποτελεσμάτων και για περισσότερη αυτοματοποίηση των εφαρμογών τις εικονικής πραγματικότητας. Οι εφαρμογές των συγκεκριμένο δύο τεχνολογιών μαζί είναι αμέτρητες καθώς χρησιμοποιούνται στην ιατρική για ανίχνευση καρκινωμάτων και άλλων ασθενειών, για την ανάκτηση της κίνησης για να ελεγκτή εάν ο ασθενής ή ο ηλικιωμένος πραγματοποιεί σωστά μια άσκηση για φυσική αποκατάσταση. Τα τελευταία χρόνια γίνονται πολλές προσπάθειες για ανάπτυξη εφαρμογών που έχουν σαν κύριο σκοπό την ανάκτηση κίνησης και να αναγνωρίζουν εάν ο χρήστης κάνει σωστά την συγκεκριμένη άσκηση και να τον διορθώνει η εφαρμογή στο εάν πραγματοποιεί κάποιο λάθος.

Στην προσπάθεια μου να βοηθήσω στην υλοποίηση ενός παρόμοιου συστήματος, ήταν να ανακτήσω την κίνηση μέσω από ένα microcontroller και σε μια εφαρμογή για android με την χρήση του αλγορίθμου του MotioNet. Η εφαρμογή αναπτύχθηκε σε microcontroller αρχικά αλλά επειδή είχαμε προβλήματα με την έλλειψη μνήμης, αναπτύξαμε την εφαρμογή σε συσκευές android.

Στην παρούσα εργασία περιγράφονται τη ακολούθησε πριν την ανάπτυξη της εφαρμογής για εξοικείωση με αυτή, αποφάσεις που λήφθηκαν κατά την υλοποίηση της εφαρμογής, τη τεχνικές και αλγόριθμοι υλοποιήθηκαν για την ανάκτηση της κίνησης, τυχόν προβλήματα που εμφανίστηκαν και πως επιλύθηκαν, αποτελέσματα που πείρα μεσώ από την εφαρμογή που ανέπτυξα και πόσο αξιόπιστο είναι το σύστημα μου και μία συζήτηση που συγκρίνονται τα αποτελέσματα μου με το πρωτότυπο του αλγορίθμου με την χρήση βίντεο από το dataset Human36m και συμπεράσματα από την εφαρμογή μου και τυχόν μελλοντικές εργασίες που μπορούν να εφαρμοστούν βάση αυτού που υλοποίησα. Η συγκεκριμένη εφαρμογή δουλεύει μόνο σε κινητά τηλέφωνα τύπου “smartphone” με android λογισμικό και υπάρχει η δυνατότητα να αναγνωρίζει την ανθρώπινη κίνηση και να υπολογίζει της περιστροφές του σώματος.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή.....	1
1.1 Γενική εισαγωγή.....	1
1.2 Εισαγωγή στην Ανάκτηση Κίνησης.....	2
1.3 Εισαγωγή στο Arduino.....	3
1.4 Στόχος Πτυχιακής εργασίας.....	3
1.5 Ανασκόπηση πτυχιακής εργασίας.....	4
Κεφάλαιο 2 Ερευνητικό Υπόβαθρο Πτυχιακής εργασίας.....	5
2.1 Αρχιτεκτονική του SparkFun Edge.....	5
2.2 Αρχιτεκτονική του STM32F746G Discovery Kit.....	8
2.3 Εισαγωγή στο Tiny ML	9
2.4 Παράδειγμα του Hello World	10
2.4.1 Παραδείγματα εκτέλεσης στο STM32F746G Discovery Kit	10
2.5 Παράδειγμα του Wake-Word Detection	11
2.5.1 Παραδείγματα εκτέλεσης στο SparkFun Edge.....	12
2.5.2 Παραδείγματα εκτέλεσης στο STM32F746G Discovery Kit	13
2.6 Παράδειγμα του Person Detection	15
2.6.1 Παραδείγματα εκτέλεσης στο SparkFun Edge.....	15
Κεφάλαιο 3 Μεθοδολογία του αλγόριθμου MotioNet	17
3.1 Γενικές πληροφορίες αλγόριθμου	17
3.2 Ανακατασκευή της κίνησης	18
3.3 Αρχιτεκτονική του δικτύου	20
Κεφάλαιο 4 Υλοποίηση Εφαρμογής	24
4.1 Υλοποίηση σε microcontroller.....	24
4.1.1 Καταγραφή βίντεο στο microcontroller.....	24
4.1.2 Ανάκτηση των Confident Values	26
4.2 Υλοποίηση του αλγόριθμου σε Android εφαρμογή.....	26
4.2.1 Ανάκτηση των Confident Values	27
4.2.2 Υποδίκτυα E_S και E_Q	30
4.2.3 Forward Kinematic	32
Κεφάλαιο 5 ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ	36
5.1 Αξιολόγηση Confident values.....	36
5.2 Αξιολόγηση υποδικιών E_S και E_Q	41

5.2.1 Αξιολόγηση στο υποδίκτυο E_Q	41
5.2.2 Αξιολόγηση στο υποδίκτυο E_S	43
5.3 Αξιολόγηση του Forward Kinematics.....	46
Κεφάλαιο 6 ΣΥΖΗΤΗΣΗ.....	48
6.1 Συζήτηση αποτελεσμάτων	48
Κεφάλαιο 7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ	51
7.1 Συμπεράσματα	51
7.2 Μελλοντική εργασία	52
Βιβλιογραφία	54
Παράρτημα Α	A-1

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική εισαγωγή.....	1
1.2 Εισαγωγή στην Ανάκτηση Κίνησης.....	2
1.3 Εισαγωγή στο Arduino.....	3
1.4 Στόχος Πτυχιακής εργασίας.....	3
1.5 Ανασκόπηση πτυχιακής εργασίας.....	4

1.1 Γενική εισαγωγή

Τα τελευταία χρόνια το Virtual Reality είναι ένα πολύ σημαντικό πεδίο στο τομέα της πληροφορικής. Πολλή το χρησιμοποιούν για την υλοποίηση ηλεκτρονικών παιχνιδιών και πολύ άλλοι για την υλοποίηση θεμάτων όπου θα έχουν σχέση με τον ιατρικό, ή τον εκπαιδευτικό τομέα. Το Virtual Reality αρχικά είναι ένα ελεγχόμενο περιβάλλον, στο οποίο περιβάλλον μπορούν να πραγματοποιηθούν διάφορες εργασίες με την χρήση διαφόρων αισθητηριακών συσκευών. Με την βοήθεια αυτών των συσκευών μας δίνεται η δυνατότητα, να ανακτηθούν δεδομένα σε πραγματικό χρόνο και να αναλυθούν για να παρακολουθηθεί η απόδοση του χρήστη με το σύστημα που αλληλοεπιδρά.

Το Virtual reality στο τομέα της υγείας τα τελευταία χρόνια έχει αναπτυχθεί σε μεγάλο βαθμό και εκτιμάται ότι θα συνεχίζει να αναπτύσσεται και να βοηθά για πολλά χρόνια στον ιατρικό και παραϊατρικό τομέα. Πολλές εφαρμογές της εικονικής πραγματικότητας έχουν υλοποιηθεί, με κύριο σκοπό να ανακαλύψουν εάν ένας ασθενής έχει ασθενήσει από μία συγκεκριμένη ασθένεια και εάν είναι δυνατό από την εφαρμογή να αναγνώριση εάν βρίσκεται σε προχωρημένο στάδιο, ή σε αρχικό στάδιο. Άλλες εφαρμογές στη ιατρική με την βοήθεια στην εικονική πραγματικότητα είναι σε χειρουργικές επεμβάσεις, ή σε εκπαίδευση του ιατρικού προσωπικού όπου ο γιατρός μπορεί να βλέπει το εσωτερικό του ασθενή και να παρακολουθήσει διάφορες περιοχές του εσωτερικού σώματος που σε άλλες περιπτώσεις δεν θα μπορούσε να τις παρακολουθήσει.

Το virtual reality έχει αναπτυχθεί και σε παραϊατρικούς τομείς όπως στην περίπτωση του Mental Health & Psychological Therapy. Στο συγκεκριμένο παραϊατρικό τομέα μπορούν να μεταφέρουν εικονικά τον ασθενή σε κάποιο άλλο διαφορετικό μέρος. Μέσα από αυτό το εικονικό μέρος όπου ο ασθενής αλληλοεπιδρά μπορούν να δουν την ψυχολογική αντίδραση του σε διαφορετικές καταστάσεις. Ένας ακόμη παραϊατρικός τομέας που έχει συμβάλει το virtual reality είναι αυτός του Pain Management και Physical Therapy. Μέσα από το συγκεκριμένο πεδίο υπάρχει η δυνατότητα ο ασθενής να τοποθετείτε σε ένα μοντέλο εικονικής πραγματικότητας να υποδεικνύουν ποια άσκηση πρέπει να κάνει ο ασθενής για την φυσική του αποκατάσταση και ακόμη μπορεί να ελέγχει εάν την κάνει σωστά την άσκηση.

1.2 Εισαγωγή στην Ανάκτηση Κίνησης

Η ανάκτηση κίνησης στο virtual reality είναι από τα πιο σημαντικά πεδία του και είναι το πιο ευρέως διαδεδομένο τα τελευταία χρόνια. Η ανάκτηση και πρόβλεψη της κίνησης του ανθρώπινου σώματος μπορεί να πραγματοποιηθεί μέσα από χειριστήρια, από αισθητήρες της κίνησης, από accelerometers, ή το Azure Kinect και από άλλα διάφορα ιδικά εξαρτήματα για την ανάκτηση της κίνησης. Τα τελευταία όμως χρόνια στο Virtual Reality αναπτύσσονται και πάρα πολύ αλγόριθμοι όπου βασίζονται σε ένα βαθύ νευρονικό δίκτυο για την ανάκτηση της κίνησης μέσω ενός βίντεο αποθηκευμένο από πριν στο χώρο του υπολογιστή, ή από βίντεο που καταγράφεται σε πραγματικό χρόνο.

Η ανάκτηση της κίνησης μπορεί να εφαρμοστεί σε διάφορους τομείς όπως virtual reality, σε ειδικά εφέ για χρήση σε κινηματογραφικές ταινίες, σε ηλεκτρονικά παιχνίδια ψυχαγωγίας και εκμάθησης σε φοιτητές τις ιατρικής και σε άλλα πεδία όπου μπορούν να χρειαστούν της ανάκτηση κίνησης.

Στον τομέα της υγείας με την ανάκτηση της κίνησης βοηθά τον ασθενή με το να εφαρμόζει κάποιες ασκήσεις όπως του υποδεικνύει το σύστημα με σκοπό την βελτίωση της φυσικής του αποκατάστασης μετά από κάποια ασθένεια ή τραυματισμό , ή στην διόρθωση της εκτέλεσης των ασκήσεων εάν κάποιος από τους ασθενής κάνει κάποιο λάθος. Ακόμη μπορείς να υπάρχει ένας συνδυασμός με αυτά τα δύο όπου κατά την υπόδειξη της άσκησης για αποκατάσταση και το σύστημα να τον καθοδηγεί σε περίπτωση που κάνει κάτι λάθος ο χρήστης.

1.3 Εισαγωγή στο Arduino

Το Arduino είναι μια πλατφόρμα ανοικτού κώδικα όπου μπορείς να αναπτύξεις διάφορα project όπου έχουν επαφή με το πραγματικό κόσμο. Τα Arduino έχουν ένα μεγάλο αριθμό με διαφορετικά εξαρτήματα όπου το κάθε εξάρτημα κάνει κάποια διαφορετική δουλειά. Κάθε διαφορετικό είδος Arduino έχει ένα αριθμό από pins, η δική τους ευθνή είναι για να συνδεθούν με διάφορα εξαρτήματα όπου θέλουμε να τα χρησιμοποιήσουμε στην εφαρμογή που θέλουμε να αναπτύξουμε. Τα συγκεκριμένα pins χωρίζονται σε δύο κατηγορίες η πρώτη κατηγορία είναι τα digital pins όπου μπορούν να κάνουν διάβασμα και γράψιμο σε μία συγκεκριμένη κατάσταση και συνήθως υπάρχουν 14 συγκεκριμένα Pins. Η δεύτερη κατηγορία είναι τα analog pins που μπορούν να διαβάσουν και να γράψουν σε ένα εύρος από διάφορες τιμές και είναι χρήσιμο για έλεγχο του fine-grained, συνήθως τα περισσότερα Arduino έχουν 6 από τα συγκεκριμένα pins.

Όλα έχουν κάποια θήρα micro-usb, ή type-c για να μπορεί να τροφοδοτείται με το ρεύμα και συνήθως η κατανάλωση του ρεύματος είναι πολύ χαμηλή. Αυτό συμβαίνει για το λόγο ότι τροφοδοτεί LEDs, ή κάποιους αισθητήρες ή φωτογραφική μηχανή όπου οι ανάγκες τους από ηλεκτρική ενέργεια είναι σχετικά πολύ χαμηλές. Κάποια Arduino δέχονται και μπαταρία για να καλύψουν την ανάγκες τους από ηλεκτρική ενέργεια. Από την συγκεκριμένη θήρα, μπορείς ακόμη να ανεβάσεις κάποιο project επάνω στο Arduino. Ένα Arduino έχει ένα κεντρικό επεξεργαστή όπου εκτελεί το πρόγραμμα όπου θέλουμε να εφαρμόσει και μπορεί να λαμβάνει αποφάσεις σε σχέση με διάφορα δεδομένα εισόδου. Υπάρχουν διάφοροι κεντρικοί επεξεργαστές για τα Arduino όπως Atmel και ARM. Ο κάθε επεξεργαστής διαφέρει από την αρχιτεκτονική που έχει κτιστή και από τον συνολικό αριθμό μνήμης που έχει διαθέσιμο.

1.4 Στόχος Πτυχιακής εργασίας

Ο κύριος στόχος της πτυχιακής εργασίας ήταν μέσα από την καταγραφή βίντεο να αναγνωρίζει την κίνηση του χρήστη και να την μετατρέπει σε τρισδιάστατη κίνηση, για αυτό θα χρησιμοποιούσαμε τον αλγόριθμο του MotioNet. Για την εκπόνηση του συγκεκριμένου προβλήματος κύριος σκοπός ήταν να τρέχει επάνω σε microcontroller αλλά για προβλήματα μη επαρκούς μνήμης διαλέξαμε στο τελικό να κατασκευάσουμε μια εφαρμογή που θα τρέχει σε android συσκευές. Σαν στόχο και μαζί μία πρόκληση είχαμε τοποθέτηση μέσα στις προδιαγραφές της εφαρμογής κατά τον σχεδιασμό της να τρέχει ο αλγόριθμος του MotioNet σε πραγματικό χρόνο εκτέλεσης.

Ο δεύτερος κύριος στόχος ήταν ο αλγόριθμος να έτρεχε απευθείας επάνω στο κινητό τηλέφωνο χωρίς την χρήση κάποιου server και να χρησιμοποιεί HTTP εντολές. Ο λόγος είναι διότι με αυτό τον τρόπο παρέχουμε καλύτερη ασφάλεια στα δεδομένα του χρήστη και έχουμε καλύτερο χρόνο εκτέλεσης διότι δεν έχουμε καθυστέρηση.

1.5 Ανασκόπηση πτυχιακής εργασίας

Μέσα από την πτυχιακή μου εργασία μου αρχικά θα μελετήσουμε το ερευνητικό υπόβαθρό που είναι το Κεφάλαιο 2. Στο συγκεκριμένο κεφάλαιο αναλύονται οι αρχιτεκτονικές των microcontroller και κάποια παραδείγματα που εκτελέστηκαν απάνω στους microcontroller για μεγαλύτερη εξοικείωση στην ανάπτυξη κάποιων εφαρμογών με τα συγκεκριμένα μοντέλα.

Από το Κεφάλαιο 3, που είναι η Μεθοδολογία του αλγορίθμου του MotioNet, που είναι ο αλγόριθμος που χρησιμοποιήσαμε για την ολοκλήρωση της εφαρμογής μας. Μέσα από το συγκεκριμένο κεφάλαιο αναλύουμε πως λειτουργεί ο αλγόριθμος και τη χρησιμοποιεί για την ανακατασκευή της κίνησης. Μέσα από το Κεφάλαιο 4 στην συνέχεια αναφέρω την διαδικασία και τη προβλήματα αντιμετώπισα και τι έκανα να τα αντιμετωπίσω κατά την διάρκεια της ανάπτυξης της εφαρμογής μέσα από microcontroller και από την ανάπτυξη της σε android εφαρμογή.

Στην συνέχεια μέσα από το Κεφάλαιο 5 θα αξιολογηθεί κάθε τεχνική που χρησιμοποίησα μέσα στην εφαρμογή μου για την πλήρη ανάκτηση της κίνησης. Μέσα από το συγκεκριμένο κεφάλαιό περιέχονται και κάποια παραδείγματα από τα αποτελέσματα και κάποιες γραφικές παραστάσεις που εξηγούν κατά πόσο είναι καλή και χρήσιμη η συγκεκριμένη τεχνική. Μέσα από το Κεφάλαιο 6 αναπτύσσεται μία συζήτηση από τα αποτελέσματα της δικής μου υλοποίησης του αλγορίθμου με αυτή της αρχικής υλοποίησης του αλγορίθμου του MotioNet, που στην συγκεκριμένη συζήτηση αναλύονται τα αποτελέσματα από κάποια βίντεο του dataset Human 3.6m που χρησιμοποιήθηκαν για validation και στις δύο περιπτώσεις.

Στο τέλος από το Κεφάλαιο 7 υπάρχουν τα τελικά συμπεράσματα από την πτυχιακή εργασία και στο τέλος τη μπορεί να χρησιμοποιηθεί από την δική μου εφαρμογή για Μελλοντική εργασία και πως μπορεί να βελτιωθεί η δική μου εργασία.

Κεφάλαιο 2

Ερευνητικό Υπόβαθρο Πτυχιακής εργασίας

2.1 Αρχιτεκτονική του SparkFun Edge.....	5
2.2 Αρχιτεκτονική του STM32F746G Discovery Kit.....	8
2.3 Εισαγωγή στο Tiny ML	9
2.4 Παράδειγμα του Hello World	10
2.4.1 Παραδείγματα εκτέλεσης στο STM32F746G Discovery Kit.....	10
2.5 Παράδειγμα του Wake-Word Detection	11
2.5.1 Παραδείγματα εκτέλεσης στο SparkFun Edge.....	12
2.5.2 Παραδείγματα εκτέλεσης στο STM32F746G Discovery Kit	13
2.6 Παράδειγμα του Person Detection	15
2.6.1 Παραδείγματα εκτέλεσης στο SparkFun Edge.....	15

2.1 Αρχιτεκτονική του SparkFun Edge

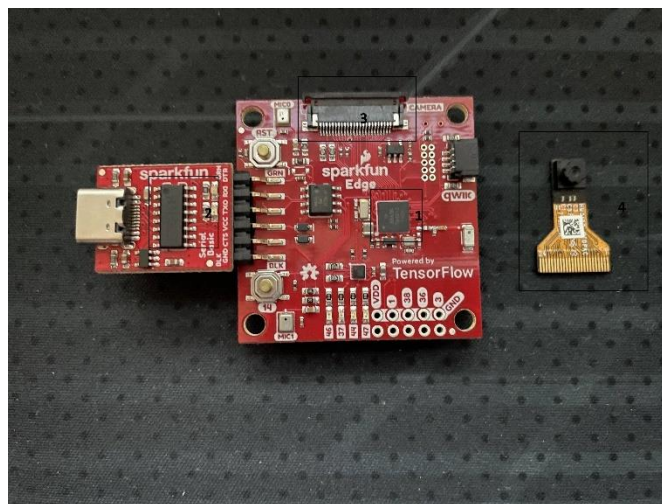


Figure 2.1.1

Το SparkFun Edge αναπτύχθηκε από την εταιρεία που κατασκευάζει microcontrollers την SparkFun μαζί με την βοήθεια της TensorFlow για να κατασκευαστή ένα microcontroller με την δυνατότητα να τρέξει άψογα το Tiny ML και να παράγει ικανοποιητικά αποτελέσματα. Σαν κεντρικό επεξεργαστή έχει τον ARM Cortex-M4F με

συχνότητα επεξεργαστή στα 48MHz και με 96 MHz εάν ο επεξεργαστής βρίσκεται σε λειτουργία burst mode.

Ο συγκεκριμένος επεξεργαστής έχει σαν αρχιτεκτονική την Armv7E-M, η συγκεκριμένη αρχιτεκτονική είναι βασισμένη για microcontrollers για να έχουν πιο αποδοτική απόδοση. Για να το επιτύχουν αυτό στην συγκεκριμένη αρχιτεκτονική χρησιμοποιούν μόνο τις εντολές Thumb, όπου το συνολικό μέγεθος και η ντετερμινιστική λειτουργία σε κάποια εφαρμογή για microcontroller είναι πιο σημαντική παρά την απόλυτη απόδοση της. Η συγκεκριμένη αρχιτεκτονική επεξεργαστών έχει αναπτυχθεί μέσω των σημαντικών αναθεωρήσεων όπου υποστηρίζει με σκοπό την καλύτερη απόδοση. Η έκδοση της ARMv7, αναγνωρίζει πλήρως την ποικιλομορφία που αναφέραμε προηγούμενος, καθορίζοντας ένα αρχιτεκτονικό προφίλ που προσαρμόζει την αρχιτεκτονική σε διάφορες και διαφορετικές απαιτήσεις της αγοράς.

Η συγκεκριμένη αρχιτεκτονική παρέχει ενεργοποίηση των υλοποιήσιμων για καλύτερη απόδοση και ταχύτητα μέσω από τον απλό σχεδιασμό του pipeline. Η εξαιρετικά ντετερμινιστική του λειτουργία παρέχει εκτέλεση μέσω από χαμηλό αριθμό κύκλων εκτέλεσης, ελάχιστη καθυστέρησης διακοπής με σύντομα pipelines και λειτουργία χωρίς cache μνήμης. Μπορεί να εκτελέσει άψογα προγράμματα που είναι υλοποιημένα με τις γλώσσες προγραμματισμού C και C++, για το λόγο ότι μπορούν να μεταφραστούν εύκολα σε γλώσσα μηχανής με την χρήση ενός gcc arm toolchain. Η συγκεκριμένη αρχιτεκτονική είναι ιδικά σχεδιασμένη για βαθιά ενσωματωμένα συστήματα. Και παρέχει υποστήριξη για debugging και στο λογισμικό για κινούμενα συστήματα.

Το συγκεκριμένο μοντέλο παρέχει έξι I²C/SPI masters όπου το SPI είναι ο host και το I²C είναι ένα serial bus. Το I²C είναι two-wired bus όπου είναι υπεύθυνο για την επικοινωνία δυο ή περισσότερων εξαρτημάτων που βρίσκονται στο ίδιο board. Το SPI υποστηρίζει πλήρης αμφίδρομοι επικοινωνία με ψηλότερο throughput από το I²C. Τα δύο αυτά χρησιμοποιούνται για καλύτερη επικοινωνία στους microcontrollers μεταξύ τους μέσω ενός πρωτοκόλλου. Παρέχει 2 UARTs όπου επικοινωνούν μεταξύ διαφορετικά από ότι τα I²C και SPI και μεταφέρουν δεδομένα από τον επεξεργαστή μέσω από το ένα UARTs προς το άλλο για να τα μετατρέψουν τα δεδομένα από παράλληλα δεδομένα σε σειριακά δεδομένα. Η συγκεκριμένη αρχιτεκτονική χρειάζεται 921160 bauds μέγιστη ταχύτητα. Έχει ένα I²C/SPI slave και 15 κανάλια ADC από 14 bit το καθένα. Το ADC είναι ένα πολύ χρήσιμο εργαλείο στους microcontrollers που μετατρέπει μία αναλογική

τάση σε κάποιο pin σε ένα ψηφιακό αριθμό. Ακόμη περιέχει ένα Bluetooth επεξεργαστή για ασύρματη σύνδεση όπου υποστηρίζει BLE5.

Το συγκεκριμένο μοντέλο microcontroller έχει 1MB Flash memory και 384KB από SRAM memory. Έχει δύο MEMS μικρόφωνα όπου μπορεί να υποστηρίξει αναγνώριση φωνής ή κάποια συγκεκριμένη λέξη. Ένα ενσωματωμένο 3-axis accelerometer αισθητήρα όπου μπορεί να υπολογίσει την κίνηση όταν κάποιος κρατά το microcontroller και το μετακινεί. Για τέλος έχει ένα interface για να μπορεί να συνδεθεί μια OV7670 κάμερα.

Η αρχιτεκτονική του συγκεκριμένου μοντέλου έχει την δυνατότητα να σύνθεση απευθείας με την κάμερα Himax H01B και είναι κάμερα με χαμηλή κατανάλωση ενέργειας. Η συγκεκριμένη κάμερα έχει 320pixels ανάλυση φωτογραφίας και το μέγιστο frame rate που μπορεί να αποδώσει είναι στα 60fps. Τα δεδομένα του βίντεο μεταφέρονται από 1bit, 4bit, ή 8bit interface με υποστήριξη σε κάθε frame με line synchronization. Το κανάλια χρώματος που παίρνει η συγκεκριμένη φωτογραφική είναι grayscale τιμές, δηλαδή μόνο αποχρώσεις του γκριζου. Έχει κύκλωμα ανίχνευσης της κίνησης με προγραμματισμένο ROI και όριο ανίχνευσης με ψηφιακή έξοδο για λειτουργία ως διακοπή.

Από το Figure 2.1.1 μπορούμε να παρατηρήσουμε το microcontroller καθώς και κάποια από τα κύρια εξαρτήματα, που είναι ενσωματωμένα απάνω του. Εκεί που αναγράφεται ο αριθμός 1 μπορούμε να παρατηρήσουμε τον επεξεργαστή που έχει τον, Armv7E-M. Στον αριθμό 2 μπορούμε να δούμε πως είναι το CH340C, που συνδέετε με τον υπολογιστή και είναι υπεύθυνο να μεταφέρει τα binary αρχεία που δημιουργούνται μέσα στον επεξεργαστή για να τρέξει το application που ανεβάζουμε επάνω στο microcontroller. Από τον αριθμό 3 μπορούμε να δούμε το pin όπου συνδέετε η κάμερα επάνω στο microcontroller. Ενώ από τον αριθμό 4 μπορούμε να δούμε την κάμερα Himax HB01B0, όπου μπορεί να συνδεθεί απευθείας επάνω στο συγκεκριμένο microcontroller.

2.2 Αρχιτεκτονική του STM32F746G Discovery Kit

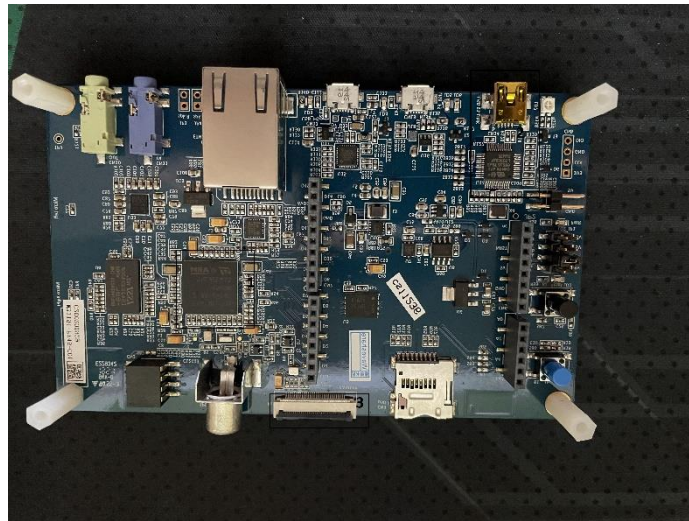


Figure 2.2.1

Το Discovery Kit είναι μια πλήρης πλατφόρμα όπου μπορεί να αναπτύξει και να τρέχει ένα project κατασκευασμένο για να τρέχει σε κάποιο microcontroller. Ο επεξεργαστής του είναι Arm Cortex M7 όπου είναι σχεδιασμένος για το συγκεκριμένο μοντέλο microcontroller.

Ο συγκεκριμένος επεξεργαστής βασίζεται στην αρχιτεκτονική επεξεργαστών Armv7E-M, όπου είναι βασισμένη για microcontrollers για καλύτερη απόδοση παρά το μικρό τους μέγεθος. Υποστηρίζει μόνο τα σετ εντολών Thumb για καλύτερη απόδοση και ταχύτητα. Έχει σαν χαρακτηριστικά ένα 6-stage superscalar pipeline με branch prediction και μία προαιρετική FPU, ικανή για λειτουργίες μονής ακρίβειας και προαιρετική διπλής ακρίβειας. Η συγκεκριμένη αρχιτεκτονική είναι ικανή για υψηλή απόδοση και δύναμη μέσω του απλού σχεδιασμού pipeline που έχει. Παρέχει υψηλή ντετερμινιστική λειτουργία λόγω του ότι παρέχει μονό η χαμηλό αριθμό κύκλων εκτέλεσης, χαμηλή καθυστέρηση λόγω των μικρών pipelines που έχει και δυνατότητα λειτουργία cache. Η ARM έχει την δυνατότητα να τρέχει c και c++ γρήγορα στους επεξεργαστές τις λόγω της αρχιτεκτονικής της. Είναι ιδικά σχεδιασμένη για βαθιά ενσωματωμένα συστήματα. Και παρέχει υποστήριξη για debugging και υποστήριξη στο λογισμικό για κινούμενα συστήματα.

Το συγκεκριμένο μοντέλο παρέχει 1 MB Flash memory και 34KB από RAM, όπου ενώνονται σε ένα πακέτο BGA216. Παρέχει ενσωματωμένο επάνω Quad-SPI 128Mbit όπου το συγκεκριμένο βρίσκετε στους σύγχρονους microcontrollers. Είναι γρηγορότερο

από το απλό SPI λόγω του ότι χρησιμοποιεί 4 data lines για διάβασμα και γράψιμο, σε σχέση με το απλό SPI που χρησιμοποιεί 2 data line. Έχει σχεδιαστεί για να παρέχουν βοήθεια στις flash memory όταν χρησιμοποιείται η περισσότερη μνήμη για multimedia και η μνήμη του microcontroller δεν είναι αρκετή. Έχει ενσωματωμένα 128Mbit SDRAM αλλά μόνο τα 64 μπορούμε να τα διαχειριστούμε.

Το συγκεκριμένο microcontrollers τρέχει ARM Mbed OS, είναι ανοικτού κώδικα λειτουργικού συστήματος που δίνει την δυνατότητα να κτίζεις και να κάνεις deploy IOT εφαρμογές με την χρήση Tiny ML για το συγκεκριμένο microcontroller και για άλλα όπου υποστηρίζουν και τρέχουν το συγκεκριμένο λειτουργικό σύστημα.

Από το Figure 2.2.1 μπορούμε να παρατηρήσουμε το συγκεκριμένο microcontroller καθώς και κάποια από τα κύρια εξαρτήματα, όπου είναι ενσωματωμένα επάνω. Στον αριθμό 1 μπορούμε να παρακολουθήσουμε τον επεξεργαστή Arm Cortex M7. Εκεί που αναγράφεται ο αριθμός 2 μπορούμε να δούμε την θήρα που συνδέετε ο microcontroller με τον υπολογιστή. Και εκεί που είναι ο αριθμός 3 το pin που συνδέετε η κάμερα με τον microcontroller.

2.3 Εισαγωγή στο Tiny ML

Το Tiny ML άρχισε να αναπτύσσεται τα τελευταία 5 χρόνια από την Google. Καταφέρνοντας να κατασκευάσουν και να τρέχουν μοντέλα νευρωνικών δικτύων, με το συνολικό τους μέγεθος να μην ξεπερνά τα μερικά KB. Τα μοντέλα Tiny ML αναπτύχθηκαν αρχικά για να λειτουργούν σε smartphone συσκευές με κατανάλωση μερικές δεκάδες από μνήμη RAM και από flash memory. Τα μικροσκοπικά αυτά μοντέλα έχουν και την ιδιαιτερότητα να τρέχουν τα συγκεκριμένα Tiny ML μοντέλα και να καταναλώνουν πάρα πολύ λίγη ηλεκτρική ενέργεια.

Στην συνέχεια άρχισε να αναπτύσσεται το Tiny ML και να εφαρμόζεται σε microcontrollers όπου έχουν την ιδιότητα να καταναλώνουν μικρή ηλεκτρική ενέργεια και να έχουν την δυνατότητα να τρέξουν αυτά τα μικρά neural network μοντέλα.

Η χρήση του Tiny ML έφερε μεγάλες και θετικές επιπτώσεις στο τομέα της πληροφορικής όπως χαμηλό χρόνο καθυστέρησης εφόσον το μοντέλο τρέχει κατευθείαν επάνω στο microcontroller και δεν χρειάζεται να στέλνει τα δεδομένα σε κάποιο server για να τρέχουν και με αυτό τον τρόπο ελαττώνει τον χρόνο καθυστέρησης του τελικού αποτελέσματος. Από την στιγμή που το μοντέλο τρέχει απευθείας επάνω στο microcontroller όπως ανέφερα προηγούμενος και δεν χρειάζεται να στέλνονται τα

δεδομένα σε κάποιο server έχουμε χαμηλότερο bandwidth και καλύτερη ασφάλεια στο να κλέψουν τα δεδομένα μας.

2.4 Παράδειγμα του Hello World

Σε αυτό το παράδειγμα εκπαιδεύουμε το μοντέλο μας με το να παίρνει σαν δεδομένα τιμές του x και αυτό να προβλέπει τις τιμές του y . Με αυτό το παράδειγμα προσπαθούμε να μάθουμε τις βασικές έννοιες του machine learning και του deep learning.

Στην συνέχεια μετά το training, validation και testing μέσω του Google Colab, δημιουργούμε το δικό μας μοντέλο του neural network. Όπου για την δημιουργία του μοντέλου αυτού χρησιμοποιούμε το Keras από την βιβλιοθήκη του TensorFlow. Για να δραστηριοποιήσουμε το neural network που κατασκευάζουμε χρησιμοποιώ μια μαθηματική function την ReLU όπου μας επιστρέφει όποιο αποτέλεσμα είναι μεγαλύτερο του 0.

Τα δεδομένα του μοντέλου μας που δημιουργούμε τα χωρίζουμε σε 60 τις εκατό για training, το 20 τις εκατό για validation και το υπόλοιπο 20 τις εκατό που απομένει για testing. Μόλις δημιουργήσουμε το μοντέλο μας τότε το κάνουμε compile χρησιμοποιώντας την standard optimizer μέθοδο και για αναγνώριση τα σφάλματα που υπάρχουν στο μοντέλο χρησιμοποιούμε την μέθοδο του mean square error.

Μόλις τελειώσουμε από την κατασκευή του μοντέλου τότε το μετατρέπουμε σε TensorFlow Lite μοντέλο χωρίς quantization. Στην συνέχεια μετατρέπουμε το TensorFlow Lite μοντέλο που μόλις δημιουργήσαμε σε αρχείο της c++. Με αυτό το τρόπο μπορούμε να ανεβάσουμε σαν binary αρχείο το μοντέλο μας επάνω στο microcontroller και να περιορίσουμε το χώρο που καταναλώνουμε επάνω στο microcontroller μας.

2.4.1 Παραδείγματα εκτέλεσης στο STM32F746G Discovery Kit

Στο Arduino η εκτέλεση του παραδείγματος θα έχει μια κόκκινη τελεία εκεί που θα έπρεπε να είναι η τιμή και θα μετακινητέ μεταξύ της οθόνης. Η τιμές που θα παίρνει η τέλια θα είναι από 0 έως 2π .

Για να κάνουμε deploy το παράδειγμα επάνω στο Discovery Kit αρχικά με την εντολή make δημιουργούμε ένα νέο χώρο όπου περιέχει μέσα το παράδειγμα του Hello World για το συγκεκριμένο μοντέλο Arduino. Στην συνέχεια πηγαίνουμε στο αρχείο που δημιουργήσαμε και με εντολή mbed config root. κάνουμε το συγκεκριμένο φάκελο το root του mbed project και με το mbed deploy προετοιμάζουμε το πρόγραμά με την

δημιουργία των dependencies που χρειάζεται μέσω της εντολής mbed. Με την χρησιμοποίηση του gcc arm toolchain δημιουργείτε ένα νέο binary εκτελέσιμο αρχείο και στην συνέχεια κάνουμε αντιγραφή το binary εκτελέσιμο αρχείο και το τοποθετούμε από τον χώρο όπου το δημιουργήσαμε από τον προσωπικό μου χώρο επάνω στο μοντέλο

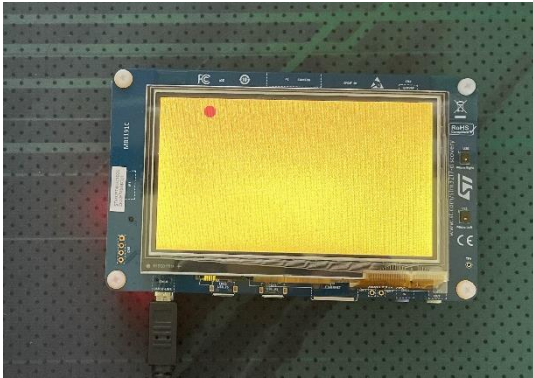


Figure 2.4.1

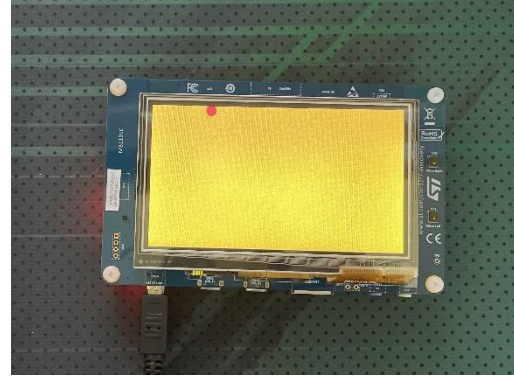


Figure 2.4.2

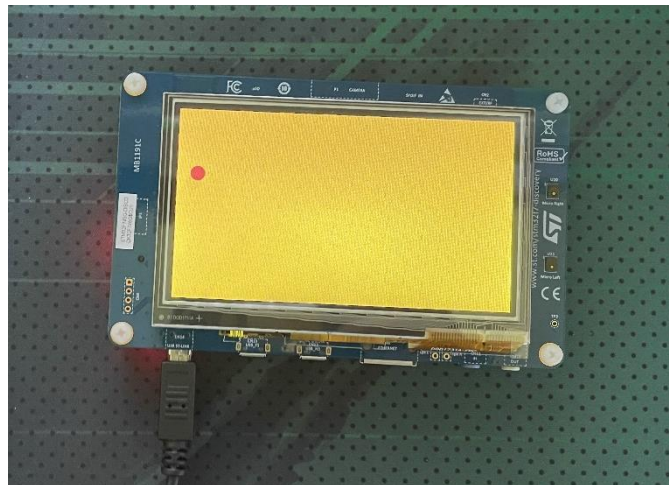


Figure 2.4.3

Figure 2.4.1 βλέπουμε να παίρνει μία τιμή λίγο πριν το $\pi/2$, ενώ στο Figure 2.4.2 βλέπουμε να παίρνει τιμή στο $\pi/2$ ακριβώς. Ενώ στο Figure 2.4.3 παρατηρούμε να παίρνει τιμή λίγο μετά το 0. Η γενική μου παρατήρηση από αυτό το παράδειγμα είναι πως με τις τιμές που παράγει το σύστημα σχεδιάζεται μία ημιτονοειδής συνάρτηση.

2.5 Παράδειγμα του Wake-Word Detection

Το μοντέλο αυτού του παραδείγματος είναι ένα TensorFlow Lite μοντέλο όπου μέσα του εμπεριέχει ένα dataset, που μέσα του υπάρχουν 65000 λέξεις όπου η κάθε μια διαρκεί το

πολύ ένα δευτερόλεπτο. Ο συνολικός αριθμός διαφορετικών λέξεων όπου έχει μέσα το Dataset μας είναι 30. Το μοντέλο μας εκπαιδευτικό για να μπορεί να αναγνωρίσει μόνο το yes, no και unknown και εάν υπάρχει ησυχία. Στο unknown σημαίνει πως συμπεριλαμβάνονται οι άλλες 28 διαφορετικές λέξεις του dataset και όταν ακούσει κάποια από αυτές τις λέξεις αναφέρει σαν άκουσε unknown.

Η εφαρμογή μέσα από το μικρόφωνο του όταν αναγνωρίσει κάποιο ήχο τότε το παίρνει και από ακατέργαστο ήχο τον μετατρέπει σε spectrogram. Στην συνέχεια ο TensorFlow Lite interpreter τρέχει το μοντέλο και προσπαθεί να αναγνωρίσει τη ήταν ο ήχος που μόλις άκουσε yes, no, unknown ή ησυχία. Στην συνέχεια μόλις αναγνωριστεί τη ήχος ακουστικό τότε το εμφανίζει το Arduino.

2.5.1 Παραδείγματα εκτέλεσης στο SparkFun Edge

Εάν ο microcontroller ακούσει την λέξη yes τότε εμφανίζει στο command line Heard yes και μαζί ανάβει το πορτοκαλί led. Εάν ακούσει τη λέξη no τότε εμφανίζει στο command line Heard no και μαζί ανάβει το κόκκινο led. Εάν ακούσει κάποια λέξη που δεν αναγνωρίζει τότε εμφανίζει στο command line Heard unknown και μαζί ανάβει το πράσινο led και εάν. είναι ησυχία τότε ανάβει μόνο το μπλε LED.

Για να κάνουμε deploy το παράδειγμα του βιβλίου στο Arduino SparkFun Edge χρησιμοποίησα το Arduino IDE και διάλεξα το BOARD αυτό του Apollo3 Edge, σαν bootloader SparkFun Variable Loader Artemis όπου είναι υπεύθυνο να βάζει το microcontroller σε μορφή να μπορεί να υποδεχθεί το binary αρχείο. Για SVL Baud Rate να είναι ίση με 921600, όπου το baud rate είναι ο ρυθμός διάδοσης των αρχείων στο Arduino. Διαλέγουμε το σωστό port όπου είναι το Arduino συνδεδεμένο με τον υπολογιστή μας. Στην συνέχεια επιλέγουμε από τα Tools το Burn Bootloader σαν έχουμε πατημένα τα κουμπιά 14 που βρίσκονται στο microcontroller, και πριν από αυτό πατήσαμε το κουμπί rst. Έχοντας συνεχώς πατημένο το κουμπί 14 μέχρι να τελειώσει επιτυχώς το Bootloader και στο τέλος κάνουμε upload το project επάνω στην πλατφόρμα.

```

COM4
-----
Apollis9 Buzzer Mode is Available
Apollis9 operating in Buzzer Mode (400Hz)

Heard unknown (201) 89560ms
Heard unknown (204) 89460ms
Heard yes (204) 89472ms
Heard unknown (209) 81235ms
Heard unknown (232) 81303ms
Heard unknown (220) 81746ms
Heard unknown (219) 81820ms
Heard unknown (201) 82156ms
Heard unknown (204) 82564ms
Heard unknown (209) 84310ms
Heard unknown (209) 84738ms
Heard unknown (210) 84672ms
Heard unknown (204) 87385ms
Heard unknown (203) 89377ms
Heard unknown (203) 89459ms
Heard unknown (209) 89260ms
Heard unknown (210) 810694ms
Heard unknown (219) 81248ms
Heard unknown (205) 81320ms
Heard unknown (204) 812403ms
Heard unknown (211) 813364ms
Heard unknown (204) 817184ms
Heard unknown (204) 827424ms
Heard unknown (202) 822454ms
Heard unknown (202) 82333ms
Heard unknown (203) 82440ms
Heard unknown (202) 833433ms
Heard unknown (211) 836333ms
Heard unknown (204) 837240ms
Heard unknown (209) 839390ms
Heard yes (205) 838590ms
Heard yes (222) 836374ms
Heard yes (201) 839238ms
Heard unknown (201) 840205ms
Heard unknown (209) 841493ms
Heard unknown (217) 841444ms
Heard unknown (204) 841943ms
Heard unknown (215) 842045ms
Heard unknown (209) 842370ms
Heard unknown (212) 842770ms
Heard no (207) 848720ms

```

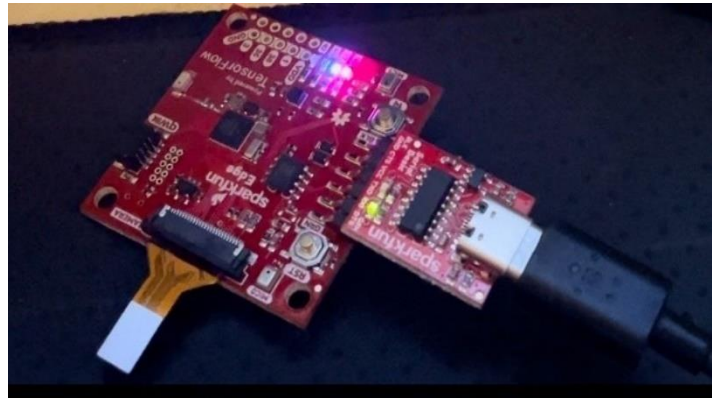


Figure 2.5.1

Figure 2.5.2

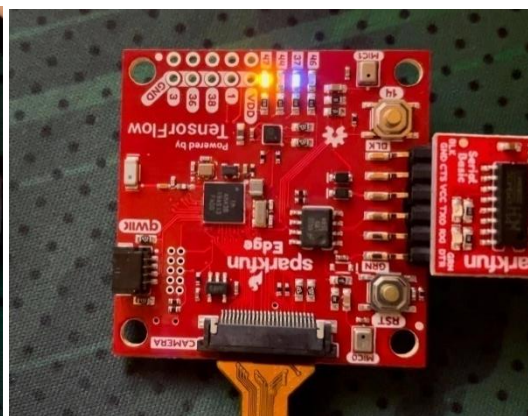
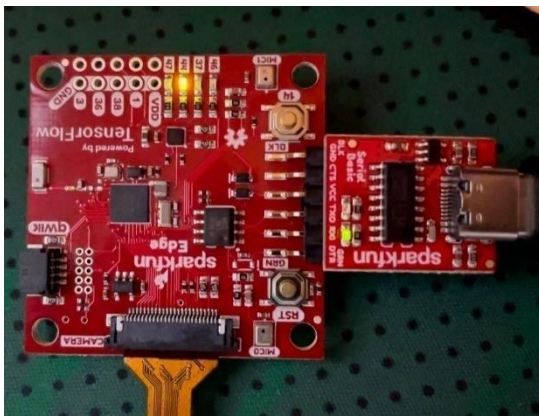


Figure 2.5.3

Figure 2.5.4

Στο Figure 2.5.1 βλέπουμε τα αποτελέσματα που παίρνει το Arduino όταν τίθεται σε λειτουργία με το συγκεκριμένο project. Στο Figure 2.5.2 βλέπουμε όταν ακούει την λέξη no ανάβει το κόκκινο led. Ενώ όταν ακούει την λέξη yes ανάβει το πράσινο led σύμφωνα με το Figure 2.5.4. Ενώ όταν ακούει κάποια λέξη που δεν ξέρει τότε ανάβει το πορτοκαλή, μαζί με το μπλέ led σύμφωνα με το Figure 2.5.3

2.5.2 Παραδείγματα εκτέλεσης στο STM32F746G Discovery Kit

Στην LCD οθόνη του θα μας εμφανίζει εάν άκουσε yes ή no, ή εάν δεν ακούει κάτι και τυπώνει silence, ή εάν αυτό που ακούει δεν το αναγνωρίζει τυπώνει unknown. Για να κάνουμε deploy το παράδειγμα επάνω στο Discovery Kit αρχικά με την εντολή make δημιουργούμε ένα νέο χώρο όπου περιέχει μέσα το παράδειγμα του Hello World για το συγκεκριμένο μοντέλο Arduino. Στην συνέχεια πηγαίνουμε στο αρχείο που

δημιουργήσαμε και με εντολή `mbd config root`. κάνουμε το συγκεκριμένο φάκελο το `root` του `mbd project` και με το `mbd deploy` προετοιμάζουμε το πρόγραμά με την δημιουργία των `dependencies` που χρειάζεται μέσω της εντολής `mbd`. Με την χρησιμοποίηση του `arm toolchain` δημιουργείτε ένα νέο `binary` εκτελέσιμο αρχείο και στην συνέχεια κάνουμε αντιγραφή το `binary` εκτελέσιμο αρχείο και το τοποθετούμε από τον χώρο όπου το δημιουργήσαμε από τον προσωπικό μου χώρο επάνω στο μοντέλο του Arduino.

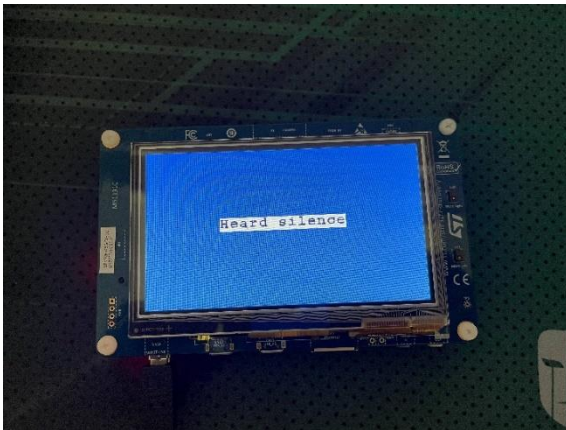


Figure 2.5.5



Figure 2.5.6

Στο Figure 2.5.5 βλέπουμε πως η εφαρμογή μας δεν ακούει τίποτε και τυπώνει `Heard silence` στην lcd οθόνη του Discovery Kit. Στο Figure 2.5.6 παρατηρούμε πως αναγνωρίζει ότι είπαμε την λέξη `no` και τυπώνει `Heard no` στην lcd οθόνη του Discovery Kit



Figure 2.5.7



Figure 2.5.8

Στο Figure 2.2.7 παρατηρούμε πως ακούει κάποιο ήχο το Arduino μας αλλά δεν αναγνωρίζει τι ήχος είναι. Στο Figure 2.2.8 βλέπουμε πως το Arduino αναγνωρίζει πως είπαμε την λέξη yes και μας τυπώνει Heard yes στην lcd οθόνη του Discovery Kit.

2.6 Παράδειγμα του Person Detection

Στο Person Detection το πρόγραμμα χρησιμοποιεί ένα pre trained convolutional neural network όπου εκπαιδεύτηκε επάνω στο Visual Wake Words dataset. Αυτό το dataset περιέχει μέσα 115000 διαφορετικές φωτογραφίες όπου η κάθε μία μας λέει κατά πόσο είναι η δεν είναι άνθρωπος. Το μοντέλο εκπαιδεύτηκε για να ξεχωρίζει κατά πόσο υπάρχει κάποιο άτομο μπροστά από την κάμερα και κατά πόσο δεν υπάρχει κάποιο άτομο.

Το πρόγραμμα αναλύει όλα τα frames που παίρνει από την camera που έχουμε συνδεδεμένη επάνω στο Arduino. Στην συνέχεια μέσω του TensorFlow Lite interpreter τρέχει το μοντέλο που αναφέραμε πιο πάνω και προσπαθεί να ξεχωρίσει κατά πόσο είναι άνθρωπος και κατά πόσο δεν είναι άνθρωπος. Στην συνέχεια μας επιβεβαιώνει εάν είναι άνθρωπος η εάν δεν είναι με το ποια τιμή είναι μεγαλύτερη.

2.6.1 Παραδείγματα εκτέλεσης στο SparkFun Edge

Όταν εντοπίζεται από την κάμερα του Arduino κάποιος άνθρωπος τότε θα ανάβει και το πράσινο LED του SparkFun Edge. Εάν όμως το αποτέλεσμα πως δεν είναι άνθρωπος είναι μεγαλύτερο από το αποτέλεσμα πως είναι τότε θα ανάβει το πορτοκαλή LED.

Για να κάνουμε deploy το παράδειγμα του βιβλίου στο Arduino SparkFun Edge χρησιμοποίησα το Arduino IDE και διάλεξα το BOARD αυτό του Apollo3 Edge, σαν bootloader SparkFun Variable Loader Artemis όπου είναι υπεύθυνο να βάζει το Arduino σε μορφή να μπορεί να υποδεχθεί το binary αρχείο. Για SVL Baud Rate να είναι ίση με 921600, όπου το baud rate είναι ο ρυθμός διάδοσης των αρχείων στο Arduino. Διαλέγουμε το σωστό port όπου είναι το Arduino συνδεδεμένο με τον υπολογιστή μας. Στην συνέχεια επιλέγουμε από τα Tools Burn Bootloader σαν έχουμε πατημένα τα κουμπί 14 που βρίσκονται στο microcontroller, και πριν από αυτό πατήσαμε το κουμπί rst. Έχουμε συνεχώς πατημένο το κουμπί 14 μέχρι και μόλις τελείωση επιτυχώς το Bootloader διαλέγουμε να γίνει το project upload στο Arduino.

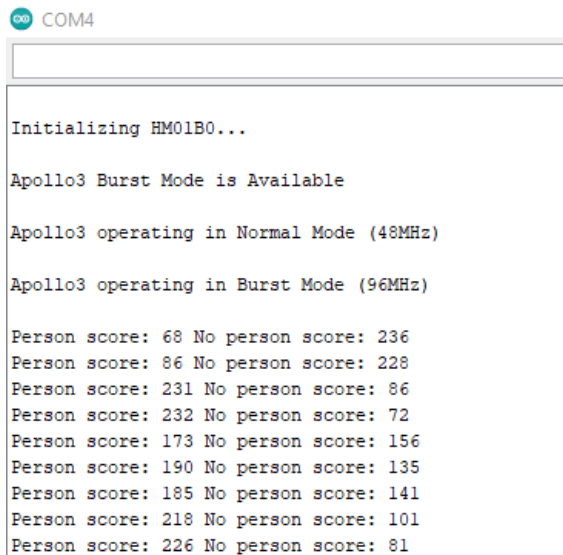


Figure 2.6.1



Figure 2.6.2

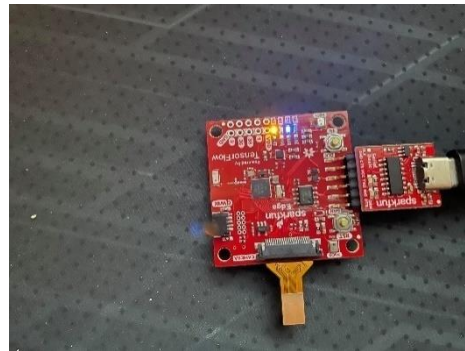


Figure 2.6.3

Στο Figure 2.6.1 παρατηρούμε τα αποτελέσματα που μας εμφανίζει το Arduino στο command line του όταν ανιχνεύσει κάποιο άτομο σε ένα από τα frame του βίντεο. Στο Figure 2.6.2 βλέπουμε πως ανάβει το πράσινο LED αυτό σημαίνει πως το αποτέλεσμα να είναι άνθρωπος είναι μεγαλύτερο με το αποτέλεσμα που να μην ήταν άνθρωπος. Στο Figure 2.6.3 παρατηρούμε πως ανάβει το πορτοκαλί LED αυτό σημαίνει πως το αποτέλεσμα να μην είναι άνθρωπος είναι μεγαλύτερο από αυτό που να είναι άνθρωπος.

Κεφάλαιο 3

Μεθοδολογία του αλγόριθμου MotioNet

3.1 Γενικές πληροφορίες αλγορίθμου	17
3.2 Ανακατασκευή της κίνησης	18
3.3 Αρχιτεκτονική του δικτύου	20

3.1 Γενικές πληροφορίες αλγορίθμου

Ο αλγόριθμος που θα χρησιμοποιήσω για την υλοποίηση του project είναι αυτός του MotioNet. Το MotioNet είναι ένα βαθύ νευρωνικό δίκτυο όπου ανακατασκευάζει την κίνηση του ανθρώπινου σκελετού από ένα monocular βίντεο. Άλλες υλοποιήσεις αλγορίθμων για την παραγωγή κίνησης του ανθρώπινου σκελετού βασίζονται στην πλοήγηση του σκελετού, ή σε αντίστροφη κινηματική κίνηση για να συσχετιστεί ένας συνεπής σκελετός με χρονικά συνεκτικές περιστροφές των αρθρώσεων του σώματος. Η μέθοδος υλοποίησης που χρησιμοποιεί ο αλγόριθμος του MotioNet έχει μια εντελώς διαφορετική προσέγγιση καθώς είναι η πρώτη μέθοδος, όπου βάσει των δεδομένων που εξάγει, δηλαδή τις γωνίες των αρθρώσεων οι οποίες δημιουργούν άμεσα ένα πλήρης κινηματικό σκελετό, όπου παρουσιάζει την πλήρη κίνηση του ανθρώπινου σώματος.

Η προσέγγιση του αλγορίθμου του MotioNet, βρίσκεται σε ένα deep neural network με ενσωματωμένη την προηγούμενη κίνηση που εκτελεί το ανθρώπινο σώματος, τις οποίες κινήσεις της αποσυνθέτει σε δύο διαφορετικές ακολουθίες όπου η κάθε μια αποθηκεύει διαφορετικά χαρακτηριστικά του ανθρώπινου σκελετού. Στην μία ακολουθία αποθηκεύει ένα μονό συμμετρικό σκελετό που κωδικοποιεί τα μήκη του οστού και σε μία άλλη ακολουθία με τις τρισδιάστατες περιστροφές των αρθρώσεων όπου σχετίζονται με τα global root position και τα foot contact labels. Αυτά τα δύο χαρακτηριστικά εισάγονται σε ένα ενσωματωμένο forward kinematic layer όπου μαζί εξάγουν σαν αποτέλεσμα την τρισδιάστατη θέση της κάθε άρθρωσης όπου πριν από την εξαγωγή του τελικού αποτελέσματος η κάθε άρθρωση συγκρίνεται μαζί με το ground truth για την ελαχιστοποίηση του σφάλματος. Το πλεονέκτημα του MotioNet σε σχέση με άλλους

αλγορίθμους ανακατασκευής της κίνησης είναι πως έχει την δυνατότητα να μαθαίνει να συνδέει την περιστροφή της φυσικής άρθρωσης απευθείας από τα δεδομένα όπου εκπαιδεύτηκαν, σε σχέση με άλλους αλγορίθμους όπου υποθέτουν ένα υποκείμενες μοντέλο, η βγάζουν συμπεράσματα από τις θέσεις της κάθε άρθρωσης ξεχωριστά με την βοήθεια κάποιου data agnostic inverse kinematic solver. Ο αλγόριθμος με την επιβολή ενός συνεπή ενιαίου σκελετού μαζί με τις χρονικά συνεκτικές περιστροφές των αρθρώσεων που περιορίζουν την μεγιστοποίηση του σφάλματος στο αποτέλεσμα που παράγει σε σχέση με άλλους παρόμοιους αλγορίθμους ανακατασκευής της ανθρώπινης κίνησης.

Ο αλγόριθμος σαν αρχείο εξόδου παράγει ένα αρχείο κίνησης όπου μέσα εμπεριέχει όλες τις πληροφορίες κίνησης που κάνει το ανθρώπινο σώμα μέσα από το βίντεο.

3.2 Ανακατασκευή της κίνησης

Όπως αναφέραμε και προηγούμενος το σύστημα του MotioNet βασίζεται σε ένα βαθύ νευρονικό δίκτυο, όπου η δομή για την ανακατασκευή της ανθρώπινης κίνησης είναι μοναδική και η πρώτη σε υλοποίηση και εφαρμογή.

Το δίκτυο του MotioNet έχει την ιδιότητα να μαθαίνει να χαρτογραφεί τις δισδιάστατες κοινές θέσεις. Οι κοινές αυτές θέσεις εξάγονται από το βίντεο που δίνουμε σαν παράμετρο για να αναλυθεί από τον αλγόριθμο και αυτός τις παράγει σε δύο ξεχωριστά συστατικά. Ένα από τα δύο συστατικά που παράγονται είναι ένας μονός συμμετρικός σκελετός όπου αντιπροσωπεύει το μήκος κάθε οστού που βρίσκεται στον συγκεκριμένο σκελετό. Το δεύτερο συστατικό είναι μία δυναμική ακολουθία όπου είναι ανεξάρτητη από το σκελετό. Η ακολουθία μέσα της περιέχει τις περιστροφές τις κάθε άρθρωσης, το global root position και τα foot contact labels. Αυτά τα τρία συστατικά αποτελούν μια πλήρης περιγραφή της κίνησης από ολόκληρο το ανθρώπινο σώμα χωρίς να χρησιμοποιήσουμε περισσότερη επεξεργασία η αντίστροφη κινηματική για να αποκτήσουμε την πλήρης τρισδιάστατη αναπαράσταση κίνησης.

Για να εκπαιδεύσουμε το δίκτυο μας αξιοποιούμε τα σύνολα δεδομένων κίνησης όπου περιλαμβάνουν διάφορες χρονικές ακολουθίες από διάφορες κινήσεις όπου η κάθε κίνηση είναι από διαφορετικό θέμα. Η κάθε κίνηση αντιπροσωπεύει μόνο ένα σκέλος του σκελετού μας και μία χρονική ακολουθία των τρισδιάστατων περιστροφών όπου μπορεί να εκτέλεση μια άρθρωση και διάφορες θέσεις που μπορεί να αναπαραστήσει μία άρθρωση.

Κατά την διάρκεια της εκπαίδευσης του δικτύου, η τρισδιάστατη κίνηση προβάλετε σε δισδιάστατη κίνηση από διάφορες αυθαίρετες οπτικές γωνίες και το δίκτυο εκπαιδεύεται για να έχει την δυνατότητα να μπορεί να ανακτήσει τα στοιχεία της δυναμικής ακολουθίας όπου την αναφέραμε προηγουμένως. Στην συνέχεια της εκπαίδευσης του δικτύου μας τα foot contact labels, εξάγονται από διάφορες καταγραμμένες κινήσεις από την εξέταση της απόστασης των αρθρώσεων που βρίσκονται στα πόδια από το έδαφος και επίσης από την ταχύτητα που έχουν.

Η κύρια τεχνική συμβολή του αλγορίθμου μας είναι στην κατασκευή ενός νέου 2-branch δικτύου όπου η συμβολή του στον αλγόριθμο είναι η ανακατασκευή των δυναμικών ιδιοτήτων των κινήσεων, των περιστροφών των αρθρώσεων, τις global root positions που έχει το ανθρώπινο σώμα κατά την κίνηση και από τα foot contact label που εμπεριέχει και πάλι το ανθρώπινο σώμα κατά την κίνηση ενώ ακόμη υπάρχει ξεχωριστά ένας στατικός κινητικός σκελετός.

Το δίκτυο του MotioNet αποτελείται από διάφορες τεχνικές όπου κάποιες από αυτές έχουν εφαρμοστή προηγούμενος από άλλους αλγόριθμους, ενώ κάποιες από αυτές τις τεχνικές έχουν εφαρμοστή πρώτη φορά από τον αλγόριθμο του MotioNet. Οι τεχνικές όπου χρησιμοποιήθηκαν πρώτα από τον αλγόριθμο του MotioNet είναι το Forward Kinematic layer όπου προτάθηκε το 2018 από το Villegas, αλλά ο MotioNet είναι ο πρώτος αλγόριθμος που τον εφάρμοσε για αλγόριθμο εκτίμησης της κίνησης του ανθρωπίνου σώματος. Ο αλγόριθμός του MotioNet είναι ο πρώτος στο είδος του που εφαρμόζει μια διευκρινιστική κριτική στην ρεαλιστική των χρονικών ακολουθιών για τον υπολογισμό των γωνιακών ταχυτήτων που κάνει κάθε άρθρωση. Το δίκτυο του MotioNet είναι εκπαιδευμένο με τρόπο τέτοιο ώστε να μπορεί μετατρέπει τις θέσεις που βρίσκονται οι αρθρώσεις σε περιστροφικές κινήσεις.

3.3 Αρχιτεκτονική του δικτύου

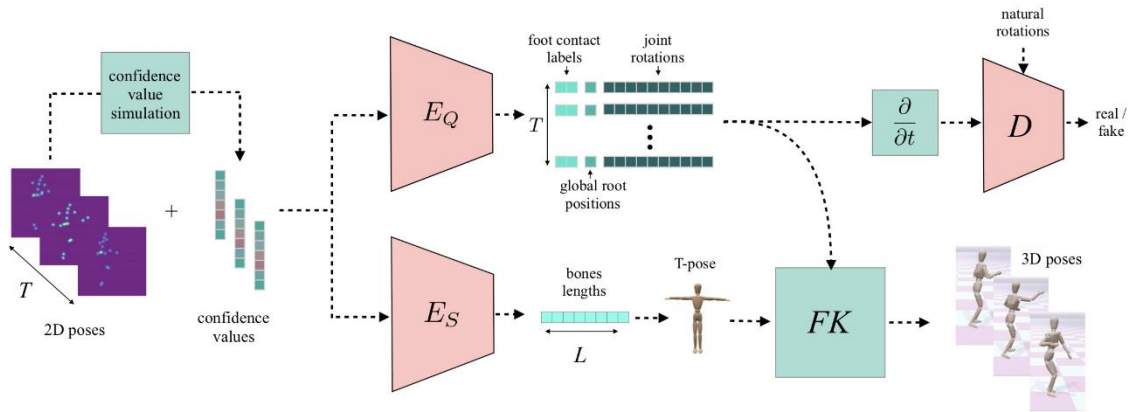


Figure 3.3.1

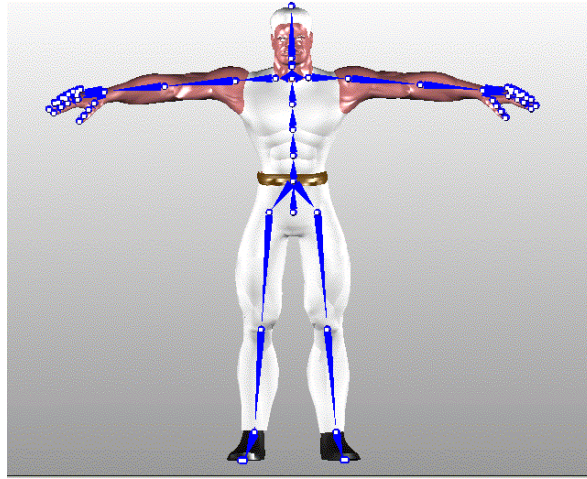


Figure 3.3.2

Η αρχιτεκτονική του δικτύου του αλγορίθμου του MotioNet ορίζεται από τον μαθηματικό τύπο $P_{s,q,r} \in R^{T \times 3J}$ όπου μας ορίζει μία ακολουθία των τρισδιάστατων συντεταγμένων των αρθρώσεων του σκελετού. Το $s \in R^L$ μας ορίζει τον σκελετό τις ακολουθίας, το $q \in R^{T \times QJ}$ μας διατυπώνει τις περιστροφές των αρθρώσεων και το $r \in R^{T \times 3}$ μας επισημάνει τα global root positions, όπου έχει L μας αντικατοπτρίζει τον συνολικό αριθμό των άκρων του σώματος, το T μας δείχνει το προσωρινό συνολικό μέγεθος τις τρισδιάστατης ακολουθίας και όπου παρουσιάζεται το Q μας επισημάνει το συνολικό μέγεθος των περιστροφών στο διάνυσμα αναπαράστασης της κίνησης. Όλα αυτά μαζί δημιουργούν την τρισδιάστατη ακολουθία που αναφέραμε προηγουμένως.

Ο σκοπός και ο κύριος στόχος του αλγορίθμου είναι να κατασκευάσει ένα βαθύ νευρονικό δίκτυο για την αποσύνθεση ακολουθιών όπου προβάλλονται οι τρισδιάστατες

συντεταγμένες των αρθρώσεων. Από τον μαθηματικό τύπο $C(P_{s,q,r}; c) \in R^{T \times 2J}$, όπου το C είναι προοπτική χειριστική προβολή όπου προβάλλει της τρισδιάστατες κοινές θέσεις σε δισδιάστατες χρησιμοποιώντας παραμέτρους κάμερας. Ο μαθηματικός τύπος που υποδείξαμε μόλις τώρα παράγεται σε 4 διαφορετικές μεταβλητές την $\tilde{q} \in R^{T \times QJ}$ που συλλέγει τις δυναμικές περιστροφικές πληροφορίες τις κίνησης, την $\tilde{s} \in R^L$ και μας δίνει την πλήρης περιγραφή του ενιαίου συνεπής σκελετού, με την μεταβλητή $\tilde{r} \in R^{T \times 3}$ που υπολογίζει τα global root positions με την πάροδο του χρόνου και με την μεταβλητή $\tilde{f} \in \{0,1\}^{T \times 2}$ που υπολογίζει εάν και τα δύο πόδια αγγίζουν στο έδαφος σε κάθε frame του βίντεο. Οι πρώτες τρεις μεταβλητές μπορούν να συνδυαστούν μαζί μέσω της forward kinematic, για να υπολογίσει την τρισδιάστατη ακολουθία της στάσης του σώματος και αντικατοπτρίζεται από την μαθηματική πράξη $\tilde{p}_{\tilde{s},\tilde{q},\tilde{r}} \in R^{T \times 3J}$, που καθορίζεται από τις συντεταγμένες των αρθρώσεων(x, y, z).

Από το Figure 3.3.1 μπορούμε να παρατηρήσουμε την πλήρη λειτουργία του δικτύου του αλγορίθμου και που εφαρμόζεται κάθε λειτουργία που έχει ο αλγόριθμος και τη συμβολή έχει σε αυτόν. Ακόμη μπορούμε να παρατηρήσουμε πως κατά την διάρκεια της εκπαίδευσης κάθε δεδομένο από τα frames του βίντεο, ορίζεται από τον μαθηματικό τύπο, που αναφέραμε και αναλύσαμε προηγουμένως τον, $C(P_{s,q,r}; c) \in R^{T \times 2J}$, ο οποίος τροφοδοτείται παράλληλα σε δύο υποδίκτυα του δικτύου μου το E_S και το E_Q . Το E_S είναι υποχρεωμένο στο να δημιουργήσει το σκελετό όπου ο σκελετός δημιουργείται από τον μαθηματικό τύπο $\tilde{s} = E_S(C(P_{s,q,r}; c))$. Ενώ το E_Q έχει σαν κύρια δουλειά στον υπολογισμό τριών μεταβλητών των περιστροφών των αρθρώσεων, των global root position και τα foot contact labels και μπορούμε να δούμε πως υπολογίζονται μέσω του τύπου $\tilde{q}, \tilde{r}, \tilde{f} = E_Q(C(P_{s,q,r}; c))$, κάθε μεταβλητή παρουσιάζεται αντιστοίχως στον τύπο. Ο κύριος σκοπός είναι ότι και οι τρεις μεταβλητές υπολογίζονται από ένα κοινό υποδίκτυο και ότι και οι τρεις μεταβλητές έχουν κάποια κοινή συσχέτιση μεταξύ τους. Όπως για παράδειγμα το global root position σε ένα άνθρωπο που τρέχει συσχετίζεται χρονικά με την περιστροφή των άκρων και τον χρόνο που έχουν επαφή με το έδαφος. Επίσης η τρεις αυτές μεταβλητές δηλαδή οι αρθρώσεις, τα global root position και τα foot contact labels είναι δυναμικές μεταβλητές, δηλαδή μπορούν να αλλάξουν σε κάθε frame.

Ο αλγόριθμος μέσω του σχεδιασμού του E_Q διατηρεί και αποθηκεύει όλες τις χρονικές πληροφορίες όπου μπορεί να κατέχει. Ο αλγόριθμος σε πρακτικό επίπεδο χρησιμοποιεί

μονοδιάστατα convolution layers στο άξονα που βρίσκεται ο χρόνος, με αποτέλεσμα να μας δίνουν ένα προσωρινό σύνολο με περιστροφές αρθρώσεων σε σχέση με το χρόνο. Όμως αντίθετος ο σκελετός είναι στατικός και το υποδίκτυο E_S χρησιμοποιεί προσαρμοστική ομαδοποίηση για να καταφέρει να καταρρεύσει ο άξονας του χρόνου, με αποτέλεσμα να έχω ένα σταθερό διάνυσμα στο μέγεθος του οστού ανεξάρτητα από τις διαστάσεις εισόδου.

Ο αλγόριθμος του MotioNet για τον υπολογισμό των μεταβλητών $\tilde{s}$, $\tilde{q}$ και $\tilde{r}$ που είναι τρισδιάστατες συντεταγμένες των αρθρώσεων(x, y, z) χρησιμοποιεί το Forward Kinematic. Το Forward Kinematic αναφέρεται στην διαδικασία για τον υπολογισμό των τρισδιάστατων συντεταγμένων των αρθρώσεων(x, y, z), δεδομένου ότι του δίνονται οι περιστροφές των αρθρώσεων και ο αρχικός σκελετός. Το Forward Kinematic πραγματοποιείται από επιτυχίες περιστροφή των άκρων του σώματος ξεκινώντας από το κέντρο του σώματος δηλαδή την λεκάνη, έως και τις άλλες άρθρωσης του σώματος στα χέρια και στα πόδια. Εφόσον οι συγκεκριμένες λειτουργίες στον αλγόριθμο είναι γραμμικές λειτουργίες, τότε μπορούμε να τις ενσωματώσουμε όλες μαζί συγκεντρωμένες σε ένα ξεχωριστό υποδίκτυο δηλαδή διαφορετικό Forward Kinematic layer όπου θα επιτρέπει την αναπαραγωγή προς τα πίσω. Ο αλγόριθμος στην πράξη εφόσον έχει ήδη στην κατοχή του τον υπολογισμό του εκτιμώμενου μήκους των οστών $\tilde{s}$, αρχικά ξαναχτίζουμε την T-pose πόζα $\tilde{s}_{init} \in R^{3J}$, μέσω της τοποθέτησης των τρισδιάστατων συντεταγμένων των αρθρώσεων του ανθρώπινου σκελετού. Η δημιουργία της T-pose πόζα γίνεται με την τοποθέτηση της λεκάνης στο κέντρο, ενώ οι άλλες συντεταγμένες τοποθετούνται μόνο στο x και y επίπεδο ενώ στο z επίπεδο ισούνται με 0. Η τοποθέτηση γίνεται βάση των καθορισμένων αντισταθμίσεων όπως είναι στο $\tilde{s}$ και η T-pose κατασκευάζεται όπως φαίνεται και στο Figure 3.3.2. Στην συνέχεια ο αλγόριθμος σε κάθε χρονική στιγμή t πραγματοποιεί το Forward Kinematic πάνω στο $\tilde{s}_{init}$ χρησιμοποιώντας τις περιστροφικές αρθρώσεις του t frame στο βίντεο, όπου διατυπώνεται από τον τύπο $\tilde{q} \in R^{4J}$ όπου παίρνει σαν δεδομένα το $\tilde{P}_{\tilde{s}, \tilde{q}_J \tilde{r}=0}^t = FK(\tilde{s}_{init}, \tilde{q}^t)$, όπου το $\tilde{P}_{\tilde{s}, \tilde{q}_J \tilde{r}=0}^t$ μας αντικατοπτρίζει την τοπική τρισδιάστατη πόζα στο συγκεκριμένο frame t, με συντεταγμένη την λεκάνη όπως αυτή της προέλευσης. Η περιστροφές των αρθρώσεων εφαρμόζονται διαδοχικά σε κάθε άρθρωση του σκελετού ξεκινώντας από την λεκάνη δηλαδή το κέντρο του σώματος και τελειώνοντας στις αρθρώσεις των άκρων του σώματος, και αυτό μπορεί να αντικατοπτριστεί και από το $\tilde{P}_n^t = \tilde{P}_{parent(n)}^t + R_n^t \tilde{s}_n$,

όπου το $\tilde{P}_n^t \in R^3$ είναι η συντεταγμένη της νιοστής άρθρωσης την χρονική στιγμή t , το $\tilde{P}_{parent(n)}^t$ είναι η θέση που βρίσκεται ο πατέρας της νιοστής άρθρωσης την χρονική στιγμή t και το $R^n \in R^{3 \times 3}$ είναι πίνακας που βρίσκονται όλες οι περιστροφές όλων των νιοστών αρθρώσεων και το $\tilde{s}_n \in R^3$ είναι η τρισδιάστατη αντιστάθμιση της νιοστής άρθρωσης σχετικά με τον πατέρα της το $\tilde{s}_{init}$. Στο τέλος ο αλγόριθμος δοσμένου της τοπικής τρισδιάστατης πόζας του ανθρώπινου σκελετού προσθέτει το global root position για να εκτίμηση πλήρως την τρισδιάστατη πόζα μέσω του μαθηματικού τύπου, $\tilde{P}_{\tilde{s}, \tilde{q}}^t \tilde{r}=0 = FK(\tilde{s}_{init}, \tilde{q}^t) + \tilde{r}^t$. Δεδομένου ότι το Forward Kinematic αποτελείται από διάφορους τελεστές, μπορεί να τοποθετηθεί ολόκληρο σαν ένα υποδίκτυο σε ολόκληρο το δίκτυο του αλγορίθμου και ως ένα layer μέσω του οποίου μπορεί να είναι εφαρμοσμένο η διάδοση προς τα πίσω. Η Forward Kinematic μας παρουσιάζεται και μέσω του Figure 3.3.1

Κεφάλαιο 4

Υλοποίηση Εφαρμογής

4.1 Υλοποίηση σε microcontroller.....	24
4.1.1 Καταγραφή βίντεο στο microcontroller.....	24
4.1.2 Ανάκτηση των Confident Values.....	26
4.2 Υλοποίηση του αλγορίθμου σε Android εφαρμογή.....	26
4.2.1 Ανάκτηση των Confident Values.....	27
4.2.2 Υποδίκτυα E_s και E_d	30
4.2.3 Forward Kinematic.....	32

4.1 Υλοποίηση σε microcontroller

Η υλοποίηση του αλγορίθμου αποφάνθηκε να υλοποιηθεί στο SparkFun Edge, για το λόγο ότι είναι ένα από τα πιο ισχυρά microcontroller για να υποστηρίξει TinyML. Με την υποστήριξη μοντέλων έως 3MB χωρητικότητας

4.1.1 Καταγραφή βίντεο στο microcontroller

Πρώτο βήμα κατά την υλοποίηση της εφαρμογής ήταν η καταγραφή βίντεο δηλαδή κάποιο συγκεκριμένο αριθμό από frames, για να χρησιμοποιηθεί μετά από τον αλγόριθμο και να ανάλυση την ανθρώπινη μέσα από τα συγκεκριμένα frames. Για την καταγραφή και ανάλυση βίντεο σε microcontroller χρειάστηκε η χρησιμοποίηση του SparkFun Edge και της φωτογραφικής μηχανής Himax HM01B01. Η φωτογραφική μηχανή είναι ρυθμισμένη να τραβάει frames, με κάθε frames να έχει διαστάσεις 96x96 και 8bit διαφορετικές αποχρώσεις του γκριζου για κάθε pixel, δηλαδή 256 διαφορετικά χρώματα του γκριζου μεταξύ του μαύρου και του άσπρου. Ο συνολικός χώρος που παίρνει κάθε frame στην μνήμη του Arduino είναι 12000bits δηλαδή 1,5KB.

Αρχικά ελέγχουμε ένα η φωτογραφική μηχανή έχει αρχικοποιηθεί επάνω στο microcontroller. Εάν δεν έχει αρχικοποιηθεί τότε αρχικοποιώ την φωτογραφική μηχανή

μέσω του pin που είναι συνδεδεμένη. Μαζί αρχικοποιώ και τα LEDs που βρίσκονται στις θέσεις 37, 44, 46 και 47 και βάζω το Arduino σε λειτουργία Burst Mode στα 96MHz. Η λειτουργία burst mode επιτρέπει τον επεξεργαστή να τρέχει με μεγαλύτερη συχνότητα από αυτή που είναι προγραμματισμένη. Εμείς το προγραμματίσαμε να τρέχει στην μέγιστη δυνατή συχνότητα που επιτρέπει ο συγκεκριμένος επεξεργαστής. Με αυτό τον τρόπο καταφέρνουμε να επεξεργαζόμαστε τα frames στο καλύτερο δυνατό χρόνο. Όταν τελειώσουν οι αρχικοποιήσεις τότε η φωτογραφική μηχανή τραβάει κάποιο μικρό αριθμό από frames, ώστε η φωτογραφική μηχανή να ρυθμιστεί κατάλληλα και να είναι σε ετοιμότητα να τραβήξει frames.

Όταν η φωτογραφική μηχανή είναι σε ετοιμότητα στο να διαβάσει frames, τότε σε ένα προσωρινό register καταγράφουμε ένα frame, όπου μέσα σε αυτό αναγράφεται η πληροφορία της φωτογραφίας που μόλις πάρθηκε από την φωτογραφική μηχανή. Έπειτα από αυτό το σημείο καταγράφουμε τα pixels του frame που παράχθηκε σε ένα πίνακα που παίρνει χώρο στην προσωρινή μνήμη του microcontroller.

Για να αποθηκεύσουμε την πληροφορία της φωτογραφίας που πήραμε αρχικά υπολογίζουμε τον αριθμό των pixel που μόλις αποκτήσαμε και στην συνέχεια υπολογίζουμε το κέντρο της εικόνας που βρίσκεται. Με την ανάκτηση του κέντρου της εικόνας μπορούμε να βάλουμε με περισσότερη ακρίβεια σε ποια τιμή αντιστοιχεί κάθε pixel από την πληροφορία που καταγράψαμε στο register από την φωτογραφική μηχανή. Στην συνέχεια για κάθε τιμή του frame τοποθετούμε ένα 8bit αριθμό σε κάθε pixel. Κάθε pixel παίρνει μια τιμή από 0 έως 255, που είναι η συνολικές αποχρώσεις του γκριζου, η τιμές 0 και 255 αντικατοπτρίζουν το μαύρο και το γκριζο αντίστοιχα και όσο πιο μεγάλος είναι η τιμή του pixel τόσο πιο ανοικτό χρώμα στην απόχρωση του γκριζου είναι το συγκεκριμένο pixel.

Μόλις καταγραφεί και επεξεργαστή κάποιο frame έχω ρυθμίσει το microcontroller να ανάβει το πράσινο led για debugging, και να αναγνωρίζουμε πως την συγκεκριμένη στιγμή έχει καταγραφή και επεξεργαστή κάποιο frame. Το microcontroller το ρύθμισα με τρόπο τέτοιο ώστε κάθε βίντεο να καταγράφει 60 frames και να τα αποθηκεύει για περαιτέρω επεξεργασία για να αναληθί η κίνηση που παράγει ο άνθρωπος μέσα από τον συγκεκριμένο αριθμό από frames που καταγράφηκαν και αποθηκεύτηκαν.

4.1.2 Ανάκτηση των Confident Values

Για την ανάκτηση των confident values δηλαδή των keypoints τα x και y που βρίσκεται κάθε joint rotation στον ανθρώπινο σκελετό. Για να το επιτύχω αυτό χρησιμοποίησαμε το PoseNet της TensorFlow, σε αντίθεση με την αρχική υλοποίηση που χρησιμοποίησαν το OpenPose της OpenCV. Διάλεξα το συγκεκριμένο τρόπο για ανάκτηση των confident values για το λόγο του ότι ένας microcontroller μπορεί να τρέξει μόνο TensorFlow Lite μοντέλα λόγω του ότι υποστηρίζετε από την TensorFlow με το να παρέχει κάποιες βιβλιοθήκες σε σχέση με το ότι δεν υπάρχει η OpenCV για χρήση από microcontroller.

Την πρώτη φορά προσπάθησα με ένα pre-trained TensorFlow Lite μοντέλο συνολικής 100KB χωρητικότητας. Μετάτρεψα το μοντέλο σε C++ κώδικα για να μπορέσει να παραχθεί εκτελέσιμο αρχείο και να εκτελεστεί στο microcontroller. Αρχικά φόρτωνα το μοντέλο στον interpreter, όπου ο συγκεκριμένος interpreter δηλαδή ένας διερμηνέας του μοντέλου, είναι σχεδιασμένο με τρόπο να είναι γρήγορος και λιτός και με την χρήση στατικών γραφημάτων και έναν προσαρμοσμένο κατανεμητή μνήμη για να εξασφάλιση όσο το ελάχιστο φόρτο, αρχικοποίηση και λιγότερη καθυστέρηση κατά την εκτέλεση. Κατά την ώρα που ο interpreter προσπαθούσε να κάνει load το μοντέλο του PoseNet, το πρόγραμμα επάνω στο microcontroller σταματούσε ανεπιτυχώς καθώς δεν είχε την επαρκή μνήμη για να υποστήριξη το συγκεκριμένο μοντέλο.

Την δεύτερη φορά δοκίμασα με ένα άλλο μοντέλο συνολικής χωρητικότητας 3MB. Αρχικά μετάτρεψα το μοντέλο σε C++ κώδικα για να μπορέσει ο gcc arm compiler να το κάνει εκτελέσιμο αρχείο για microcontroller. Όταν το έκανα upload το project επάνω στο microcontroller μου επέστρεφε μήνυμα buffer overflow, καθώς ο microcontroller δεν είχε την επαρκή μνήμη και δυνατότητα να υποστήριξη τέτοιου είδους μοντέλα για επιστροφή της κίνησης.

Επομένως μαζί με τον επιβλέπω καθηγητή μου της ατομικής διπλωματικής εργασίας κρίναμε πως το σύστημα ήταν μη αξιόπιστο υλοποίηση σε microcontroller, καθώς δεν υπάρχει κάποιο microcontroller με τους επαρκή πόρους σε μνήμη και σε δυναμικότητα του επεξεργαστή του.

4.2 Υλοποίηση του αλγορίθμου σε Android εφαρμογή

Εφόσον η υλοποίηση σε microcontroller ήταν μην εφικτή αποφανθήκαμε καθώς δεν παρείχε την επαρκή μνήμη, να αναπτύσσαμε την εφαρμογή android συσκευές διότι πλέον τα περισσότερα από αυτά σου δίνουν την υπολογιστή δύναμη όσο ένας υπολογιστής. Επίσης χρησιμοποιήσαμε android συσκευές διότι παρέχουν την ίδια αρχιτεκτονική

επεξεργαστών όπως και οι microcontroller, δηλαδή αρχιτεκτονική ARM, απλά είναι σε πιο μεγάλης δυναμικότητας επεξεργαστές.

4.2.1 Ανάκτηση των Confident Values

Όπως είπα και προηγουμένως, η αρχική υλοποίηση του αλγορίθμου είχε χρησιμοποιήσει το OpenPose της OpenCV για την ανάκτηση των αρθρώσεων του ανθρώπινου σώματος, εγώ αποφάσισα να χρησιμοποιήσω το OpenPose της TensorFlow Lite καθώς μας δίνει τη δυνατότητα η TensorFlow να το χρησιμοποιήσουμε σε Android εφαρμογές μέσω της χρήσης του με την TensorFlow Lite.

Το PoseNet για την εξακρίβωση της ανθρώπινης πόζας χρησιμοποιεί μία box-free bottom-up μέθοδο, με δύο διαφορετικά βήματα για την ανίχνευση των 17 2D αρθρώσεων του ανθρώπινου σώματος. Το δίκτυο για την εύρεση των 17 αρθρώσεων είναι εκπαιδευμένο με τρόπο να υπάρχει ένα ground truth για τα σημεία των αρθρώσεων μέσα από την εκπαίδευση που υπήρχε στο συγκεκριμένο δίκτυο με το COCO dataset.

Δίνοντας ένα αριθμό από κάποια frames μέσω αυτού του αλγορίθμου προσπαθώ να εντοπίσω τα ορατά άρθρωση που ανήκουν στο άνθρωπο που εμφανίζεται μέσα στο συγκεκριμένο i-οστό frame χωρίς να επηρεάζεται εάν κάποιο από τα μη ορατά keypoints. Κάθε frame πριν εισαχθεί μέσα στο μοντέλο πρέπει να έχει διαστάσεις ύψος και πλάτος 257x257.

Για την ανάκτηση των keypoints το PoseNet παράγει ένα αριθμό από heatmaps στο πρώτο κανάλι ανά keypoint και αλλά δύο κανάλια ανά keypoint για τις μετατόπισης με κατεύθυνση οριζόντια και κάθετος. Έστω ότι το x_i είναι η δισδιάστατη θέση(x, y) στην εικόνα όπου $i = 1, \dots, N$ και είναι ο δείκτης της θέσης επάνω στην εικόνα και N είναι ο συνολικός αριθμός των pixels. Ας υποθέσουμε ότι $D_R(y) = \{x : \|x - y\| \leq R\}$ είναι ο κύκλος του keypoint που εμφανίζεται στο heatmap με ακτίνα R και με κέντρο γύρω από το y και ότι το y_k είναι το k-ιστό keypoint. Για κάθε ένα keypoint k από τα 17 keypoints ο αλγόριθμος αυτός δημιουργεί ένα binary classification που προβλέπει ένα heatmap $p_k(x)$ όπου ισούται με 1 εάν το $x \in D_R(y, k)$ αλλιώς ισούται με 0. Με αυτό το τρόπο έχουμε 17 ανεξάρτητα binary classification για κάθε ξεχωριστό keypoint, και κάθε μία από αυτές ανέρχεται στην πρόβλεψη ενός κύκλου ακτίνας R με το R να ισούται με 32 pixels.

Εκτός από κάποια heatmaps για κάθε keypoint, επίσης το PoseNet προβλέπει τα διανύσματα μικρής εμβέλειας $S_k(x)$ για βελτίωση της ακρίβειας εντοπισμού του keypoint.

Μέσα από κάθε θέση x των κύκλων που δημιουργεί για κάθε ένα από τα keypoint, μέσα από το μικρής εμβέλειας δισδιάστατο διάνυσμα μετατόπισης $S_k(x) = y_k - x$ δείχνεται μέσα από την x θέση της εικόνας στο k -οστό keypoint του προσώπου. Ο συγκεκριμένος αλγόριθμος ανάκτησης των δισδιάστατων αρθρώσεων παράγει 17 διανυσματικά πεδία όσα και τα keypoints, που λύνουν ένα δισδιάστατο πρόβλημα παλινδρόμησης σε κάθε μέρος τις εικόνας και που βρίσκεται κάθε keypoint ανεξάρτητα. Μέσο της δημιουργίας του Hough voting συγκεντρώνονται τα heatmaps και τα διανύσματα μικρής εμβέλειας και δημιουργούν Hough score για κάθε keypoint χρησιμοποιώντας ανεξάρτητους Hough accumulators. Για κάθε keypoint ο αλγόριθμος υπολογίζει ένα score κατά πόσο ανιχνεύθηκε σωστά το keypoint. Για κάθε keypoint y_k , έστω ότι η τετραγωνική ρίζα του εμβαδού του bounding box που ανιχνεύθηκαν όλα τα keypoints του ανθρώπου που εμπεριέχετε, κάθε βαθμολογία υπολογίζεται από τον όρο $S_k = E\{OKS_k\} = p_k(y, k) \int_{x \in D_R(y, k)} \hat{h}_k(x) \exp(-\frac{(x-y_k)^2}{2\lambda^2 \kappa_k^2}) dx$, όπου $\hat{h}_k(x)$ είναι το Hough score κανονικοποιημένη μέσα στο $D_R(y, k)$. Το αναμενόμενο OKS keypoint score είναι το γινόμενο ότι η συνθήκη σε ένα keypoint είναι παρόν στην συνθήκη ακρίβειας εντοπισμού του OKS του συγκεκριμένου keypoint. Χρησιμοποιώντας το μέσο όρο από τις βαθμολογίες όλων των keypoints ως βαθμολογία σε instance-level, όπου φαίνεται από το $s^h = \frac{1}{K} \sum_k S_k$, και ακολουθείτε από ένα non-maximum suppression για να παραμείνουν μόνο τα keypoints με υψηλή βαθμολογία. Στο non-maximum suppression χρησιμοποιείτε σαν τελικό το instance-level score, το άθροισμά όλων των score από τα keypoints που δεν έχουν ακόμη υποστηριχθεί από περιστατικά με υψηλότερη βαθμολογία, και στο τέλος πριν την τελική βαθμολόγηση οι βαθμολογίες των keypoints κανονικοποιούνται.

Χρησιμοποιώντας το πιο πάνω αλγόριθμο αναλύω κάθε frame που κάνω capture σε πραγματικό χρόνο όταν πατάμε το κουμπί Capture έως να σταματήσει η καταγραφή των frames πατώντας το κουμπί STOP. Καθ' όλη την διάρκεια από την ώρα που επιλεκτικέ το κουμπί Capture ως την ώρα που σταματήσουμε την καταγραφή των frames, αποθηκεύω όσα frames καταγραφούν εκείνη την χρονική διάρκεια. Εκτός από αυτό ο χρήστης έχει την δυνατότητα να καταγράψει κάθε 20 frames την κίνηση να την αναλύει και στην συνέχεια να αναλύει τα επόμενα 20 που θα καταγράψει η κάμερα έως πατήσει το κουμπί stop για να σταματήσει η καταγραφή της κίνησης. Υλοποιήθηκε και αυτή η δυνατότητα στο χρήστη σε περίπτωση που θα θέλουμε να αναλυθεί για έλεγχο της κίνησης εάν κάνει σωστή κάποια άσκηση ο χρήστης σε σύντομο χρονικό διάστημα σε

συνεχή διάρκεια και να επιστρέφει μία απάντηση στο χρήστη εάν εκτελεί σωστά την άσκηση, ή τη πρέπει να διορθώσει στην συγκεκριμένη άσκηση.

Κάθε frame πριν το εισάγω μέσα στο μοντέλο που βρίσκεται ο αλγόριθμος του PoseNet για να μου επιστρέψει τα 17 2D joint rotation, το σμικρύνω από την αρχική του ανάλυση που είναι 720x720 σε 257x257 όπου είναι η μέγιστη ανάλυση που δέχεται σαν είσοδο τον μοντέλο του PoseNet. Ο λόγος που επιλεκτικέ αυτή η είσοδος είναι διότι όσο μεγαλύτερο αρχείο εκπαιδεύετε να υποδέχεται το μοντέλο τόσο περισσότερο χρόνο χρειάζεται να επιστρέψει αποτέλεσμα και θέλουμε η εκτέλεση του αλγορίθμου να γίνεται σε επιθυμητά πλαίσια. Όταν πάρω το αποτέλεσμα δηλαδή τα 17 2D joint rotation, αποθηκεύω τα συγκεκριμένα joint rotation εφόσον το συνολικό score των keypoints που μου επιστρέφει το PoseNet είναι μεγαλύτερο από 0.6 δηλαδή απόκτησα ένα ικανοποιητικό αποτέλεσμα με λιγοστή απώλεια λάθους(Το score μπορεί να είναι μεταξύ 0-1). Για κάθε joint rotation εφόσον είναι θετικό το score, αποθηκεύεται το x του και το y του, μέσα από αυτά υπολογίζω σε πιο τεταρτημόριο βρίσκεται και υπολογίζω την γωνία κλήση τους μέσα από την εφαπτομένη τους στην πλην ένα διαιρώντας το x από το y(x/y) και μετατρέποντας την γωνία που μου επιστρέφει από Rads σε degrees.

Μέσα από τα Figure 4.2.1.1 και Figure 4.2.1.2 μπορούμε να δούμε την εφαρμογή καθώς και την δυνατότητα επιλογής του χρήστη αξιολόγησης για όλη την διάρκεια του βίντεο η ανά 20 frames. Από το Figure 4.2.1.3 μπορούμε να δούμε την δυνατότητα του χρήστη να σταματήσει η καταγραφή των ανά 20 frames εάν επιλεκτικέ , ή να σταματήσει την καταγραφή του βίντεο και να ξεκινήσει ο αλγόριθμος εάν ο χρήστης επέλεξε το CAPTURE.

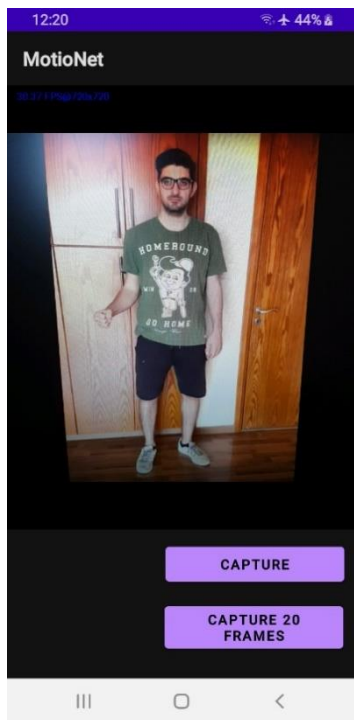


Figure 4.2.1.1

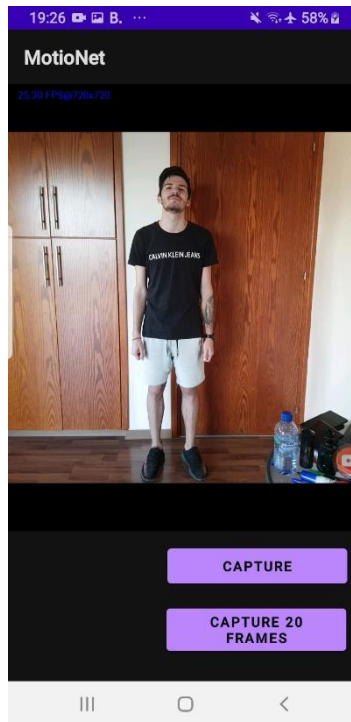


Figure 4.2.1.2

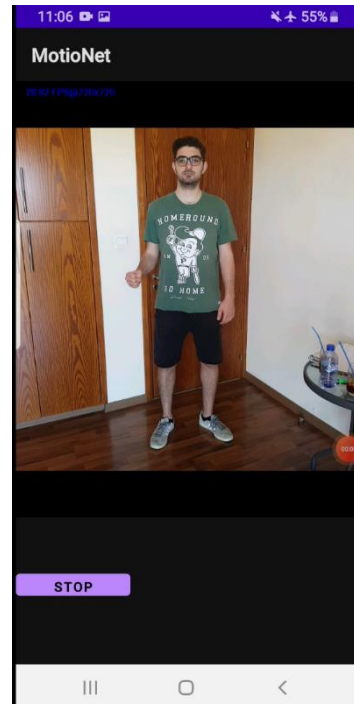


Figure 4.2.1.3

4.2.2 Υποδίκτυα E_s και E_q

Ο αλγόριθμος του MotioNet όπως αναφέραμε και προηγμένος εμπεριέχει δύο υποδίκτυα τα E_s και E_q όπως μπορούμε να δούμε από το Figure 3.3.1 όπου το καθένα έχει διαφορετική συνεισφορά στον αλγόριθμο πριν από το Forward Kinematics. Το E_s βάση του μήκους του κόκκαλου υπολογίζει το T-rose και το E_q υπολογίζει το global root position και τα foot contact labels.

Αρχικά για το E_q πρόσθεσα και τον υπολογισμό της άρθρωσης του θώρακα, μέσω από τα confident values του δεξιού και αριστερού ώμου, όπου σε αυτή την περίπτωση βρίσκω το μέσο σημείο τους επειδή ο θώρακας βρίσκεται στην μέση μεταξύ αυτών το δύο. Αυτός ο έξτρα υπολογισμός που πρόσθεσα χρειάζεται για αργότερα στο forward kinematic και τον πρόσθεσα διότι σε αντίθεση με το OpenPose που χρησιμοποιήθηκε από το πρωτότυπο του αλγορίθμου του MotioNet, το PoseNet δεν επιστρέφει που βρίσκεται ο θώρακας. Στη συνέχεια υπολογίζω το global root position δηλαδή που βρίσκεται η λεκάνη όπου είναι το κεντρικό σημείο του ανθρώπινου σώματος και είναι ο πατέρας όλων των άλλων σημείων εάν μετακινηθεί το συγκεκριμένο σημείο τότε σημαίνει

μετακινηθήκαν όλα τα σημεία. Για τον υπολογισμό του root position χρησιμοποιώ θέση του δεξιού και αριστερού ισχίου καθώς η λεκάνη είναι το μέσο σημείο μεταξύ των δύο αυτών σημείων. Για το foot contact label στο συγκεκριμένο υποδίκτιο έπρεπε να υπολογίζω εάν το πόδι ήταν στον αέρα το συγκεκριμένο frame η εάν πατούσε στο έδαφος. Για να το υλοποιήσω αυτό έβλεπα βάση το προηγούμενο frame, εάν η απόσταση των y του κατώτερου μέρος του ποδιού από αυτό του προηγούμενου frame ήταν μεγαλύτερη από το μισό μήκος του ποδιού από το γόνατο έως το αστράγαλο τότε σημαίνει πως το πόδι ήταν στον αέρα, όμως εάν η τιμή που έπαιρνα ήταν αρνητική τότε σήμαινε πως στο προηγούμενο frame δεν πατούσε στο έδαφος και τώρα βρίσκεται στον αέρα. Εάν η τιμή στο y ήταν η ίδια με πριν η μικρότερη από το μισό μήκος του ποδιού από το γόνατο έως το αστράγαλο, τότε το foot contact label σημαίνει δεν άλλαξε με το προηγούμενο frame και το foot contact label παρέμεινε το ίδιο. Το ίδιο κάνω και για τα δύο πόδια ξεχωριστά δεξιά και αριστερά.

Για τον υπολογισμό των foot contact label αρχικά διότι υπάρχει και ένας πιο εύκολος τρόπος για τον υπολογισμό να ελέγχεις εάν το y του αστραγάλου είναι 0. Εμάς ο αλγόριθμος που μας επιστρέφει τις τοποθεσίες των αρθρώσεων του σώματος βάση που βρίσκονται στο pixel του frame που στέλνουμε στο μοντέλο του αλγορίθμου για ανάλυση και το πόδι εξαρτάτε που βρίσκεται στο pixel και επιστρέφει διαφορετικές τιμές εάν μετακινηθεί το τηλέφωνο και επίσης η τιμές που παίρνω στους αστραγάλους συνήθως είναι μεταξύ 240 με 250 διότι το pixel όπως είναι γνωστό στον άξονα x δηλαδή οι γραμμές του ξεκινούν από 1 και τελειώνουν σε 257. Για αυτό το λόγο δυσκολεύτηκα να υπολογίσω εάν το πόδι πατά στο έδαφος και σκεφτικά αυτή την άλλη τεχνική που χρησιμοποιείται στα animation στα γραφικά να ελέγχεται βάση με το προηγούμενο frame εάν η διαφορά του Y είναι μεγαλύτερη από το μισό μήκος του ποδιού από το γόνατο έως στον αστράγαλο.

Στο υποδίκτιο E_5 υπολογίζω την πόζα T-rose όπου είναι όπως στο Figure 3.3.2 που στην συγκεκριμένη πόζα οι γωνιές όλων των joint rotation είναι 0 καθώς και z τους.

Αρχικά βρίσκω το μήκος των κόκκαλων βάση της ευκλείδεια απόστασης των σημείων που είναι στις δύο άκρες του κόκκαλου $d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$. Τα κόκκαλα που υπολογίζω είναι η σπονδυλική στήλη μέσω των σημείων της λεκάνης και του θώρακα, το μήκος του κέντρου της λεκάνης στα άκρα της λεκάνης, του μήκος του λαιμού από τα σημεία του θώρακα και της μύτης, το μήκος του μηρού από τα σημεία της άκριας της λεκάνης έως το γόνατο, το μήκος της κνήμης από τα σημεία του γονάτου και του

αστράγαλου, το μήκος του θώρακα με του ώμου βάση των σημείων του, το μέγεθος του βραχίονα από τα σημεία του ώμου και του αγκώνα και το μήκος της ωλένης από τα σημεία του αγκώνα και του καρπού. Για το μήκος των κοκάλων των ποδιού και των χεριών υπολογίζω διαφορετικά κόκαλα για την δεξιά και αριστερή μεριά βάση των σημείων σε κάθε πλευρά διότι υπάρχουν άτομα που έχουν το ένα κόκκαλο από την μία πλευρά πιο μεγάλο από την άλλη.

Στην συνέχεια υπολογίζω το T-pose με την τοποθέτηση του root joint rotation όπως το υπολόγισα και στα confident value και στα σημεία της δεξιάς και αριστερής μεριάς της λεκάνης το μήκος τους από την λεκάνη στο x άξονα και ο y άξονας μένει ο ίδιος με του joint position, τα σημεία του γονάτου στον άξονα το x είναι ο ίδιος με την δεξιά και αριστερή μεριά της λεκάνης διότι βρίσκονται στην ίδιο ευθεία και στον άξονα των y με το που βρίσκεται κάθε μεριά της λεκάνης στον άξονα τον y συν το μήκος του μηρού, το ίδιο και με τον αστράγαλο στον άξονα x και στον άξονα το y το ίδιο με αυτό της κάθε μεριάς της λεκάνης συν το μήκος της κνήμης. Στο σημείο του θώρακα είχα το ίδιο στον άξονα τον x με αυτό του πατέρα του το root joint rotation και στον άξονα το y αυτό του πατέρα του πλην το μήκος της σπονδυλικής στήλης, στο σημείων του ώμου είχα το ίδιο με αυτό του θώρακα στον άξονα y και στον άξονα x αυτό του θώρακα και στην δεξιά πλευρά συν την απόσταση του θώρακα από τον δεξιό ώμο και στην αριστερή μεριά μείον την απόσταση του θώρακα με τον αριστερό ώμο. Για τα σημεία του αγκώνα στον άξονα τον y έχω το ίδιο με αυτό του θώρακα και στον άξονα τον x για την δεξιά μεριά είναι το σημείο x του δεξιού ώμου συν το μήκος του δεξιού κόκκαλου του βραχίονα και στην αριστερή μεριά το σημείο x του αριστερού ώμου μείον του μήκους του αριστερού βραχίονα. Για τον καρπό στον άξονα y έχω το ίδιο με αυτό του θώρακα και του άξονα x στην δεξιά μεριά είναι το μήκος της ωλένης συν τον άξονα τον x του αγκώνα και στην αριστερή μεριά το σημείο του αριστερού αγκώνα στον άξονα x μείον το μήκος της αριστερής ωλένης. Για το κεφάλι στο T-pose έχει τον ίδιο άξονα x με τον πατέρα του τον θώρακα και ο άξονας y ισούται με που βρίσκεται ο θώρακας στον άξονα y μείον του μήκους του λαιμού.

4.2.3 Forward Kinematic

Το Forward Kinematic είναι η διαδικασία όπου δίνοντας τις συντεταγμένες των αρθρώσεων στην τρισδιάστατο χώρο, να υπολογίζει και να σου επιστρέφει στον χώρο αυτό τις περιστροφές των αρθρώσεων της ανθρώπινης κίνησης. Η περιστροφές της

κίνησης βασίζονται σε ένα δέντρο από τις αρθρώσεις του ανθρώπινου σώματος, με πατέρα όλων των αρθρώσεων να είναι η λεκάνη του ανθρώπινου σώματος και να καταλήγουμε στα φύλλα του δένδρου να είναι τα άκρα του ανθρώπινου σώματος δηλαδή οι καρποί των χεριών, οι αστράγαλοι των ποδιών και το κεφάλι. Εάν ένα από τους πατέρες μετακινηθεί τότε κατά συνέπεια μετακινούνται και όλα τα παιδιά του .

Για τον υπολογισμό του άξονα z ακολούθησα μια διαφορετική τεχνική σε σχέση με την πρώτη υλοποίηση του αλγορίθμου, αντί να κτίζω ένα βαθύ νευρονικό δίκτυο για τον υπολογισμό του άξονα z, χρησιμοποίησα διάφορες τεχνικές τριγωνομετρίας. Αποφάσισα να χρησιμοποιήσω αυτή την τεχνική διότι το νευρονικό δίκτυο θα καθυστερούσε στην εξαγωγή των αποτελεσμάτων και εκτός από αυτό δεν υπάρχει κάποια βιβλιοθήκη που να υποστηρίζει στο κτίσιμο νευρονικό δικτύων.

Εγώ χρησιμοποίησα μια τεχνική εμπνευσμένη από την ρομποτική όπου μέσα από την γωνία που σχηματίζεται στο γονέα και τα σημεία x και y της άρθρωσης που θέλουμε να βρούμε τον άξονα z. Εγώ στην δική μου υλοποίηση εκτός από εκείνα τα δύο χρησιμοποίησα και τα σημεία της άρθρωσης στο T-rose, όπου όπως είδαμε και στον αλγόριθμο τα z στη T-rose πόζα είναι . Μέσα από εκείνα τα τρία σημεία το πατέρα της άρθρωσης που θέλω να υπολογίσω το z της, την άρθρωση και το σημείο της άρθρωσης στην T-rose πόζα δημιουργείτε ένα τρίγωνο, από την τριγωνομετρία ξέρουμε πως $\sin \theta = \frac{\text{Απέναντη}}{\text{Προσκήμενη}}$, όπου θ είναι η γωνία που σχηματίζεται στον πατέρα, η προσκείμενη είναι το μήκος του κοκάλου που ενώνει το πατέρα και την άρθρωση και προσκείμενη το άγνωστος μήκος η διαφορά δηλαδή της άρθρωσης από την T-rose πόζα. Όμως η πιο πάνω συνθήκη ισχύει μόνο στα ορθογώνια τρίγωνα, εάν ένα τρίγωνο που δημιουργώ δεν είναι ορθογώνιο τρίγωνο τότε τοποθετώ μία διχοτόμο και μοιράζω το τρίγωνο και το μισό τρίγωνο είναι ο ορθογώνιο τρίγωνο και βρίσκω το μισή απόσταση του z. Για κάθε άρθρωση αποθηκεύω την τιμή αυτή που υπολογίζω επί δύο διότι όπως ανέφερα λίγο πιο πριν βρίσκω για το μισό τρίγωνο την τιμή στον άξονα τον z.

Στην συνέχεια μετά τον υπολογισμό του Z πρέπει να υπολογίσω και τις περιστροφές των αρθρώσεων. Για κάθε χρονική στιγμή t υπολογίζω τους τοπικούς μετασχηματισμούς(local transformation), για τους τοπικούς μετασχηματισμούς για τον υπολογισμό τους αρχικά υπολογίζω τις τοπικές περιστροφές των αρθρώσεων. Για τον υπολογισμό των τοπικών περιστροφών των αρθρώσεων χρησιμοποιώ τον μετασχηματισμό Euler που σου επιστρέφει την περιστροφή των σημείων. Η

συγκεκριμένη τεχνική που χρησιμοποιήθηκε και στην αρχική υλοποίηση του αλγορίθμου, αρχικά υλοποιήθηκε από τον Villegas. Στην συγκεκριμένη τεχνική για κάθε άρθρωση κάθε άρθρωση σαν μεταβλητές χρησιμοποιώ τις συντεταγμένες (x ,y ,z) και την γωνία w που σχηματίζεται μέσο του x και y και χρησιμοποιώ των πιο κάτω μετασχηματισμό :

$$R_t^{Euler} = \begin{pmatrix} 1 - (y^2 + z^2) & xy - wz & xz + wy \\ xy + wz & 1 - (x^2 + z^2) & yz - wx \\ xz - wy & yz + wx & 1 - (x^2 + y^2) \end{pmatrix}$$

Δεδομένου του πιο πάνω πίνακα περιστροφής $R_t^{Euler} \in SO(3)$, για κάθε άρθρωση ξεχωριστά ενημερώνονται η θέσεις των αρθρώσεων μέσο του Forward Kinematics με ένα αναδρομικό τρόπο. Το FK με αυτό το τρόπο λειτουργεί σαν εργαλείο για την χαρτογράφηση των περιστροφών των αρθρώσεων στον τοπικό μετασχηματισμό, με αυτό το τρόπο βοηθάμε στον αλγόριθμο να μαθαίνουνται η περιστροφές των αρθρώσεων ανεξάρτητα από τον σκελετό.

Στην συνέχεια υπολογίζονται ο παγκόσμιος μετασχηματισμός που είναι ένας πίνακας που για κάθε άρθρωση στην πρώτη γραμμή για εμπεριέχει την πρώτη γραμμή του τοπικού μετασχηματισμού και στις υπόλοιπες 2 οι τελευταίες γραμμές του παγκόσμιου μετασχηματισμού που είναι ένας πίνακας 4x4 πίνακας για κάθε άρθρωση που στην διαγώνιο έχει 1 δηλαδή είναι ο ταυτοτικός πίνακας I. Στην συνέχεια για κάθε άρθρωση την πολλαπλασιάζουμε με πολλαπλασιασμό πινάκων με των πατέρα της στο παγκόσμιο σύστημα με αυτό στο τοπικό σύστημα.

Μέσα από τα foot contact labels που υπολογισθήκαν πριν ελέγγω εάν το πόδι μετακινήθηκε εάν δεν μετακινήθηκε τότε η περιστροφή των αξόνων των ποδιών είναι 0 καθώς δεν έχει μετακινηθεί καθόλου το πόδι. Εάν όμως δεν υπάρχει επαφή του ποδιού με το πάτωμα τότε υπολογίζονται η περιστροφές των αρθρώσεων του ποδιού καθώς όπως γνωρίζουμε το πόδι μετακινείται.

Με την πιο πάνω τεχνική εντοπίζω τις περιστροφές των αρθρώσεων για κάθε άρθρωση ξεχωριστά και εάν κάποιος από τον πατέρα του μετακινηθεί τότε μετακινούνται όλα τα πεδία του. Για το λόγο ότι θέλουμε την κίνηση να εξάγεται σε πραγματικό χρόνο διότι θέλουμε να τεθεί σε εφαρμογές σε πραγματικό χρόνο χρησιμοποιώ ένα layer σε αντίθεση με την αρχική υλοποίηση που κτίζει ένα δίκτυο από διάφορα Forward Kinematics layers και στο τέλος αποφασίζει μέσω ενός βαθύ νευρονικό δίκτυο ποιος είναι ο άξονας z καθώς και οι περιστροφές των αρθρώσεων.

Στην συνέχεια μόλις υπολογίσω ποιες είναι οι τιμές στον άξονα z, υπολογίζω τις περιστροφές των αρθρώσεων του σώματος για όλα τα frames που κατέγραψε ο χρήστης την κίνηση του σώματος του. Τότε φτιάχνω ένα αρχείο κίνησης το BVH που κτίζει ιεραρχικά το σκελετό με το παγκόσμιο root position να είναι το κεντρικό σημείο του σώματος δηλαδή η λεκάνη και τα πεδία του να είναι τα δεξιό και αριστερό άκρο της λεκάνης και αυτά να καταλήγουν ιεραρχικά στα άκρα του σώματος δηλαδή το άκρο των ποδιών, των χεριών και το κεφάλι. Στο αρχείο κίνησης προσθέτω μόνο στην κεντρική θέση του σώματος έχουμε την θέση στους άξονες x, y και z σε αυτή την σειρά και μετά τις περιστροφές των αξόνων x, y και z με την ίδια σειρά, για τις υπόλοιπες αρθρώσεις τοποθετώ τις περιστροφές των αξόνων x, y και z σε αυτή ακριβώς την σειρά.

Στο Forward Kinematic βρήκα μεγάλη δυσκολία στο πως θα ανακτήσω τον άξονα z για το λόγο ότι όπως ανέφερα και προηγούμενος δεν υπάρχει η δυνατότητα σε android εφαρμογή να δημιουργείτε ένα νευρονικό δίκτυο παρά μόνο να χρησιμοποιούνται μοντέλα TensorFlow Lite όπως αυτό που χρησιμοποίησα για την ανάκτηση των keypoints. Επίσης που έτρεξα τον αλγόριθμο στον υπολογιστή μου για να κτιστή ένα νευρονικό δίκτυο πας σε ένα βίντεο των 30 δευτερολέπτων χρειάζονται περίπου 4:30 ώρες με μια κάρτα γραφικών μέτρια δυναμικότητας οπότε έπρεπε να σκεπτόμουν άλλες τεχνικές που να είναι το ίδιο αποδοτικές και να θέλουν πιο λίγο χρόνο. Σε αυτό το θέμα μπερδεύτηκα πολύ τη να έκανα για να υπολογίσω τον άξονα z με την χρήση απλών τριγωνομετρικών πράξεων και της άλγεβρας. Στην τελική βρήκα μια τεχνική που χρησιμοποιείτε στο forward kinematic στην ρομποτική και μαζί πρόσθεσα κάτι που σκεφτικά εγώ βάση της T-pose πόζας δηλαδή να υπολογίζω τον άξονα z με την δημιουργία ενός τριγώνου από τον πατέρα της άρθρωσης, της άρθρωσης και της άρθρωσης στην T-pose πόζα.

Κεφάλαιο 5

ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

5.1 Αξιολόγηση Confident values.....	36
5.2 Αξιολόγηση υποδικιών E_S και E_Q	41
5.2.1 Αξιολόγηση στο υποδίκτυο E_Q	41
5.2.2 Αξιολόγηση στο υποδίκτυο E_S	43
5.3 Αξιολόγηση του Forward Kinematics.....	46

Κατά τον προγραμματισμό των φάσεων στην πτυχιακή εργασία κρίθηκε αναγκαίο, πως μετά από την υλοποίηση της πτυχιακής εργασίας θα ακολουθούσε η αξιολόγηση της μέσα από διάφορους χρήστες διαφορετικής σωματικής ανάπλασης και σε διαφορετικές απόστασης ο χρήστης από την κάμερα. Βάση από αυτού θα δούμε πόσο αξιόπιστη είναι η εφαρμογή που αναπτύξαμε και εκτός από αυτού εάν υπάρχουν τρόποι βελτίωσης της. Επειδή δεν είχα στην διάθεση μου περισσότερα από 1 κινητά τηλέφωνα τύπου smartphone η αξιολόγηση του συστήματος έγινε δυνατό να αξιολογηθεί μέσω μόνο ενός κινητού τηλεφώνου.

Τα αποτελέσματα που θα σχολιαστούν θα είναι αυτά από την ανάκτηση των confident values, των υπολογισμό του root position και των foot contact labels και τα τελικά αποτελέσματα από το Forward Kinematics εάν υπολογίζεται σωστά ο άξονας z και οι περιστροφές των αξόνων.

5.1 Αξιολόγηση Confident values

Η ανάκτηση των confident values αξιολογείτε με μεγάλη επιτυχία καθώς της περισσότερες φορές υπολογίζονται σωστά οι τιμές των δισδιάστατων θέσεων του κάθε frame. Καθώς δοκιμάστηκαν από διαφορετικές οπτικές γωνίες, από διαφορετικές αποστάσεις, διαφορετικούς φωτισμούς και διαφορετικούς ανθρώπους δηλαδή διαφορετικό σωματότυπο και διαφορετικό ύψος. Ο λόγος με διαφορετικό σωματότυπο είναι διότι οι περισσότεροι άνθρωποι έχουν διαφορετικό τύπο σώματος κάποιοι έχουν μεγαλύτερα χέρια από το κανονικό, κάποιοι μεγαλύτερο κορμό και μικρότερα χέρια οι πόδια ή έχουν το ένα πόδι μεγαλύτερο από το άλλο πόδι.

Δοκιμάζοντας από διαφορετικές αποστάσεις και μέσα από τα αποτελέσματα που παραχθήκαν μπορούμε να δούμε πως είναι πολύ ικανοποιητικά καθώς σε όλες τις περιπτώσεις, που μπορούμε να δούμε από τα Figure 5.1.1 και Figure 5.1.2, είναι πάρα πολύ ικανοποιητικά. Από αυτά τα δύο αποτελέσματα που έχουν διαφορά απόστασης περίπου 1.5 με 2 μέτρα από εκεί που βρίσκεται ο χρήστης μπορούμε να δούμε πως η ανάκτηση των δισδιάστατων αρθρώσεων του σώματος είναι σχετικά πολύ καλή καθώς και στις δύο περιπτώσεις η ανάκτηση των δισδιάστατων αρθρώσεων φαίνεται να είναι πολύ ακριβής. Αυτό όμως που είναι άξιο παρατήρησης όμως πως όταν είμαστε σε μακρινή απόσταση μπορεί να χάνεται η ακρίβεια στην πρόβλεψη των άκρων του σώματος όπως είναι οι καρποί και οι αστράγαλοι καθώς όπως φαίνεται καθαρά και από τα αποτελέσματα στα Figure 5.1.1 και Figure 5.1.2 χάνει ελαφρά λίγο στην πρόβλεψη που βρίσκονται οι καρποί στο χέρι του χρήστη.

Από χρήστες διαφορετικής σωματικής ανάπτυξης του σώματος όπως μπορούμε να δούμε από τα αποτελέσματα στα Figure 5.1.2 και Figure 5.1.3 που ο χρήστης στην πρώτη εικόνα έχει ύψος περίπου 1.85 μέτρα και ο χρήστης στην δεύτερη εικόνα να είναι έχει ύψος περίπου 1.75 μέτρα και είναι καθαρό πως έχουν κάποια διαφορά στα μέρη του σώματος τους. Όπως μπορούμε να δούμε και στους δύο χρήστες φαίνεται καθαρά πως οι εκτίμηση των δισδιάστατων αρθρώσεων του σώματος τους να υπολογίζεται σωστά εκτός από τον πρώτο χρήστη που φαίνεται να χάνεται η εκτίμηση στο αριστερό του αυτί και στον δεύτερο χρήστη παρατηρώ πως ο αλγόριθμος χάνει στην εκτίμηση των καρπούς του σώματος καθώς τους υπολογίζει λίγο πιο πάνω και στα δύο χέρια.

Μέσα από την αξιολόγηση της ανάκτησης των δισδιάστατων αρθρώσεων αξιολόγησα και το σύστημα και μέσα από διαφορετικό φωτισμό δηλαδή πιο χαμηλό φωτισμό. Όπως μπορούμε να παρατηρήσουμε από τα Figure 5.1.4 και Figure 5.1.5, βλέπουμε πως η εκτίμηση των αρθρώσεων του σώματος εκτιμούνται με μεγάλη ακρίβεια εκτός από μερικές φορές που μπορεί να χάσει από την εκτίμηση των άκρων του ποδιού ή του χεριού. Αυτό μπορεί να βεβαιωθεί μέσα από το Figure 5.1.4 καθώς μπορούμε να παρατηρήσουμε μια μικρή απώλεια στην εκτίμηση του αριστερού ποδιού και στην εκτίμηση των δύο καρπών των χεριών.

Για διαφορετικές οπτικές γωνίες δηλαδή να αναλύσω και να αξιολογήσω την ανάκτηση των δισδιάστατων αρθρώσεων του σώματος τοποθέτησα τους χρήστες στα δεξιά της κάμερας και να χάνεται το κορμί στο βάθος της κάμερας. Μέσα από τα αποτελέσματα στα Figure 5.1.7 και Figure 5.1.8, μπορούμε να παρατηρήσουμε πως το ποσοστό

απώλειας στην πρόβλεψη των αρθρώσεων. Αυτό που προκαλεί μεγάλη εντύπωση είναι πως στο Figure 5.1.7, προβλέπεται πολύ σωστά ο δεξιός αγκώνα καθώς είναι πολύ δύσκολα να γίνει η πρόβλεψη τους καθώς δεν φαίνεται καθαρά. Αυτό όμως που παρατηρώ να προβλέπει λάθος είναι η τοποθεσία μερικών άκρων και ιδικά των ποδιών καθώς δεν προβλέπεται σωστά η τοποθεσία τους. Μέσα από διαφορετικές οπτικές γωνίες δοκίμασα να παρακολουθήσω εάν προβλέπει σωστά τις διαστάσεις εάν ο χρήστης είναι τέλειος γυρισμένος πλάτη στην κάμερα. Μέσα από το Figure 5.1.9 μπορούμε να επιβεβαιώσουμε πως υπολογίζεται πολύ σωστά όλες οι αρθρώσεις του χρήστη με πολύ μεγάλη ακρίβεια καθώς μπορούμε να παρατηρήσουμε πως είναι πολύ μικρό το ποσοστό λάθους.

Επίσης προσπάθησα να εκτιμήσω κατά πόσο θα εκτιμούσε τα χέρια κάποιου εάν δεν φαίνονται καθαρά δηλαδή τα έχει κριμένα μέσα στις τσέπες του. Όπως μπορούμε να διαπιστώσουμε από το Figure 5.1.6, η εκτίμηση και σε αυτή την περίπτωση είναι πολύ ορθή καθώς τα άκρα των χεριών δηλαδή οι καρποί υπολογίζονται με μεγάλη ακρίβεια. Τα αποτελέσματα μπορούμε να τα επιβεβαιώσουμε και μέσα από το Table 5.1.1 και μέσα από τις δύο γραφικές παραστάσεις Graph 5.1.1 και Graph 5.1.2. Οι τιμές που περιέχουν ο πίνακας Table 5.1.1 και οι δύο γραφικές παραστάσεις είναι από το 0 έως το 1 και παρουσιάζει πόσο σωστά βρήκα τα αποτελέσματα όπου το 1 είναι το 100%, τα αποτελέσματα αυτά είναι από 8 διαφορετικούς χρήστες διαφορετικής σωματικής ανάπλασης και ύψους. Μέσα από την γραφική παράσταση Graph 5.1.1 μπορούμε να επιβεβαιώσουμε πως σε χαμηλότερο φωτισμό η απόδοση πέφτει κατά πολύ και σε κάποιους χρήστες το ποσοστό πόσο σωστά υπολογίστηκαν οι διαστάσεις αρθρώσεις να είναι μεταξύ του 80% και 84%. Μέσα από την γραφική παράσταση Graph 5.1.2 μπορούμε να δούμε πως και στους 8 χρήστες το ποσοστό είναι μεγαλύτερο από 90%, καθώς μπορούμε να επιβεβαιώσουμε πως από κοντινή απόσταση υπολογίζονται με πολύ υψηλό ποσοστό σιγουριάς οι αρθρώσεις. Το ίδιο συμβαίνει και για διαφορετικό οπτικό πεδίο και από μακρινή απόσταση καθώς τα αποτελέσματα σε αυτές τις περιπτώσεις είναι μεγαλύτερα από 88%. Το μόνο που σε δύο χρήστες που μπορούμε να το επιβεβαιώσουμε ξανά είναι από χαμηλότερο φωτισμό καθώς ο μέσος όρος των αποτελεσμάτων είναι μεταξύ του 80% και 84% σε αυτούς του δύο χρήστες.



Figure 5.1.1

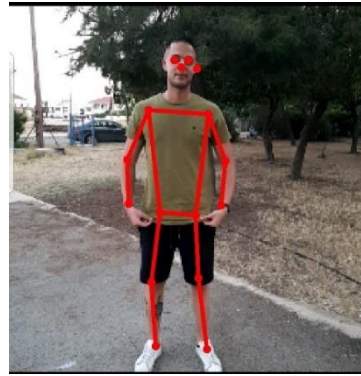


Figure 5.1.2



Figure 5.1.3



Figure 5.1.4



Figure 5.1.5



Figure 5.1.6



Figure 5.1.7



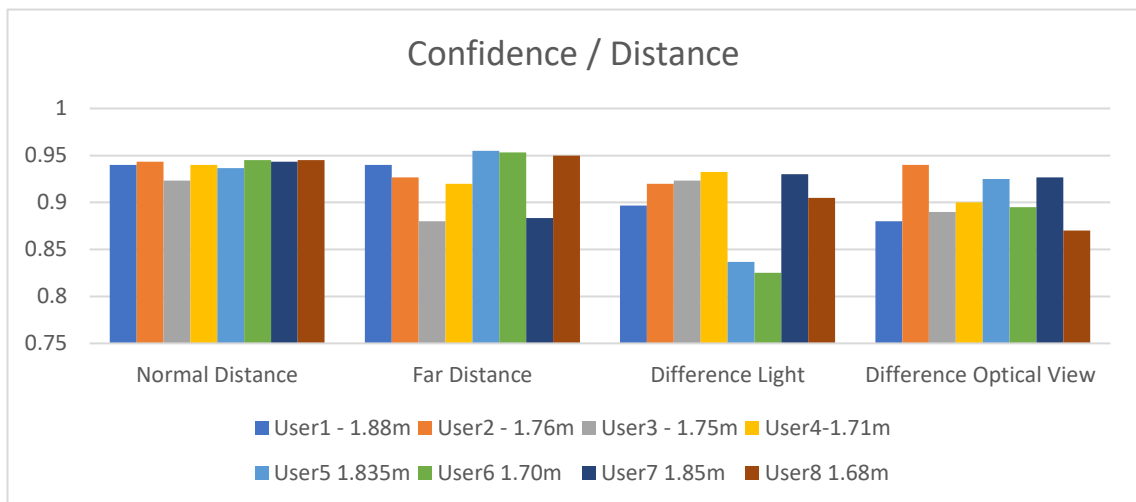
Figure 5.1.8



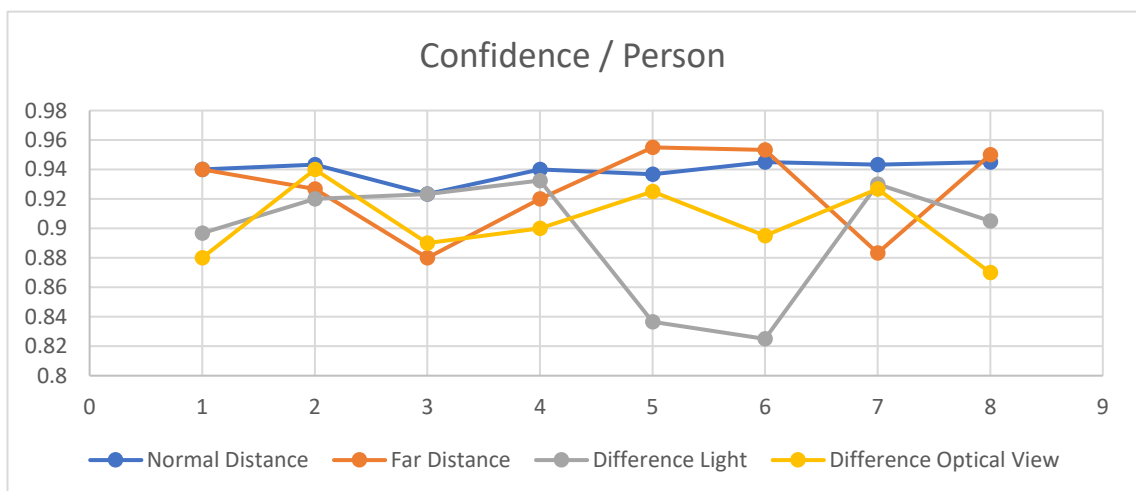
Figure 5.1.9

	Normal Distance	Far Distance	Difference Light	Difference Optical View
User1 - 1.88m	0.940	0.940	0.897	0.880
User2 - 1.76m	0.943	0.927	0.920	0.940
User3 - 1.75m	0.923	0.880	0.923	0.890
User4-1.71m	0.940	0.920	0.933	0.900
User5 1.835m	0.937	0.955	0.837	0.925
User6 1.70m	0.945	0.953	0.825	0.895
User7 1.85m	0.943	0.883	0.930	0.927
User8 1.68m	0.945	0.950	0.905	0.870

Table 5.1.1



Graph 5.1.1



Graph 5.1.2

5.2 Αξιολόγηση υποδικιών E_S και E_Q

Σε παρών υπό ενότητα θα αξιολογηθούν τα δύο υποδίκτυα του αλγορίθμου MotioNet E_S και το E_Q . Μέσα την συγκεκριμένη αξιολόγηση θα αξιολογηθεί κατά πόσο οι τεχνικές που χρησιμοποιήσα σε αυτά τα δύο υποδίκτυα είναι σωστές και παράγουν σωστό αποτέλεσμα. Επίσης θα αξιολογηθούν οι συγκεκριμένες τεχνικές του αλγορίθμου κατά πόσο αξιόπιστες είναι μέσα από τον έλεγχο τους από διαφορετικούς χρήστες.

5.2.1 Αξιολόγηση στο υποδίκτυο E_Q

Στο πρώτο υποδίκτυο αυτό αρχικά υπολογίζοντας την δισδιάστατη θέση του στήθους και της κεντρικής θέσης του σώματος που είναι το κεντρικό σημείο της λεκάνης παρατηρώ πως εφόσον για την άρθρωση της λεκάνης έχουν υπολογιστή σωστά με ψηλή βαθμολογία που επιστρέφεται από τον αλγόριθμο του δεξιά σημείου της λεκάνης και του αριστερά σημείου της λεκάνης τότε το κεντρικό σημείο της λεκάνης υπολογίζεται σωστά. Το ίδιο ακριβώς συμβαίνει και με τον υπολογισμό του σημείου που βρίσκεται το στήθος καθώς εάν υπολογιστεί σωστά που βρίσκεται ο δεξιός και αριστερός όμως τότε και ο υπολογισμός του στήθους θα είναι όσο πιο κοντά γίνεται σωστά υπολογισμένο.

Αυτό μπορούμε να παρατηρήσουμε από τα Figure 5.2.1.1 και Figure 5.2.1.2, μπορούμε να καταλάβουμε πως ο υπολογισμός των δύο αυτών αρθρώσεων υπολογίζονται πολύ ακριβές και σχεδόν με μεγάλη ακρίβεια. Αυτό όμως συμβαίνει για το λόγω ότι προβλέπονται με μεγάλη ακρίβεια τα άκρα του κεντρικού σημείου της λεκάνης. Αυτό συμβαίνει το ίδιο και για το στήθος καθώς οι αρθρώσεις του δεξιού και αριστερού ώμου δεν προβλέπονται σωστά.

Μέσα όμως από την αξιολόγηση μου από διαφορετικές αποστάσεις είναι άξιο παρατήρησης πως όσο πιο μακριά βρίσκεται ο χρήστης από την κάμερα δεν βλέπουμε κάποια διαφορά στον υπολογισμό της λεκάνης και του στήθους. Σύμφωνα και με το Figure 5.2.1.3 μπορεί να επιβεβαιωθεί, καθώς ο χρήστης βρίσκεται σε πιο μακρινή απόσταση και με λιγότερο φωτισμό σε σχέση με τα προηγούμενα δύο σχήματα και η εκτίμηση του στήθους και της λεκάνης είναι και πάλι πολύ σωστή.

Κατά την υλοποίηση στο συγκεκριμένου υποδίκτυο επίσης υπολογίζονται και τα foot contact labels των ποδιών δηλαδή εάν το πόδι βρίσκεται στο πάτωμα ή όχι. Αυτό που παρατήρησα μετά από μερικές αξιολογήσεις και δοκιμές της εφαρμογής που υλοποίησα είναι πως μερικές φορές υπολογίζεται σωστά τα foot contact labels και μερικές φορές λάθος. Αυτό συμβαίνει επειδή κάποιες φορές υπολογίζει λάθος το που βρίσκεται ο αστράγαλος και υπάρχει ένα μικρό ποσοστό σφάλματος στην τοποθεσία που βρίσκεται ο αστράγαλος. Επίσης μερικές φορές μπορεί ο αλγόριθμος να υπολογίσει λάθος την άρθρωση του αστραγάλου και να μας επιστρέψει αντίστροφα που βρίσκεται ο δεξιός και που βρίσκεται ο αριστερός αστράγαλος με αυτό να έχει σαν αποτέλεσμα να υπολογιστούν λάθος τα food contact labels και να υπάρχει και λάθος κατά την συνέχεια αφού συνδέεται ο υπολογισμός σε κάθε frame με τα αποτελέσματα που παράχθηκαν στο προηγούμενο frame.



Figure 5.2.1.1



Figure 5.2.1.2

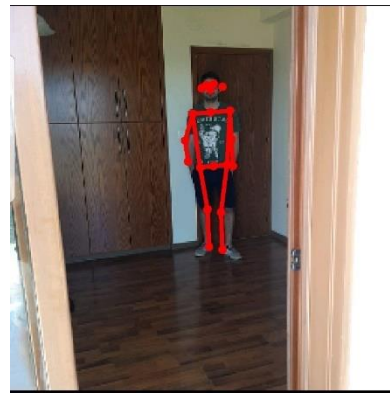


Figure 5.2.1.3

5.2.2 Αξιολόγηση στο υποδίκτυο Es

Μέσα στην αξιολόγηση του συγκεκριμένου υποδίκτυο για να παρακολουθήσω κατά πόσο κτίζεται σωστά η T-rose πόζα αυτό που έκανα είναι να μετρήσω το πραγματικό μέγεθος του κόκκαλου του κάθε χρήστη και στην συνέχεια να βρω την διαφορά του κάθε κόκκαλου με αυτό του αλγορίθμου. Μέσα από αυτή την αξιολόγηση μπορούμε να επιβεβαιώσουμε κατά πόσο υπάρχει ποσοστό λάθους στην δημιουργία της T-rose πόζας και σε πιο κόκκαλο δηλαδή μεταξύ πιο δύο αρθρώσεων είναι το μεγαλύτερο ποσοστό σφάλματος. Η συγκεκριμένη αξιολόγηση έγινε με τους ίδιους 8 χρήστες όπως και προηγούμενος που έχουν διαφορετικό σωματότυπο και ύψος.

Ο πίνακας Table 5.2.2.1 όπου παρουσιάζεται η απόλυτη διαφορά του κάθε κόκκαλου με αυτό που υπολογίζει ο αλγόριθμος για κάθε χρήστη ξεχωριστά και μας παρουσιάζεται ο μέσος όρος διαφοράς όλων των κόκκαλων του χρήστη και ο μέσος όρος της απόλυτης διαφοράς του κάθε κόκκαλου από όλους τους χρήστες σε εκατοστόμετρα. Μέσα από την γραφική παράσταση Graph 5.2.2.1 μας παρουσιάζει, για κάθε κόκκαλο την διαφορά που έχει το πραγματικό μήκος του κόκκαλου με αυτή που υπολογίζει ο αλγόριθμος στο συγκεκριμένο υποδίκτυο και τον μέσο όρο διαφοράς για κάθε κόκκαλο. Από την συγκεκριμένη γραφική παράσταση είναι άξιο παρατήρησης πως για τα κόκκαλα των χεριών είναι η μικρότερη διαφορά και η μεγαλύτερη διαφορά είναι για την σπονδυλική στήλη και για τα κάτω μέρος των ποδιών. Για τα κάτω μέρος των ποδιών αυτό συμβαίνει για το ότι όπως παρατηρήσαμε από το υποκεφαλαίου 5.1 Αξιολόγηση Confident values, συνήθως τα κάτω άκρα των ποδιών είχαν μεγάλο ποσοστό απώλειας και μερικές φορές η πρόβλεψη των συγκεκριμένων αρθρώσεων είχε ένα μικρό ποσοστό λάθους.

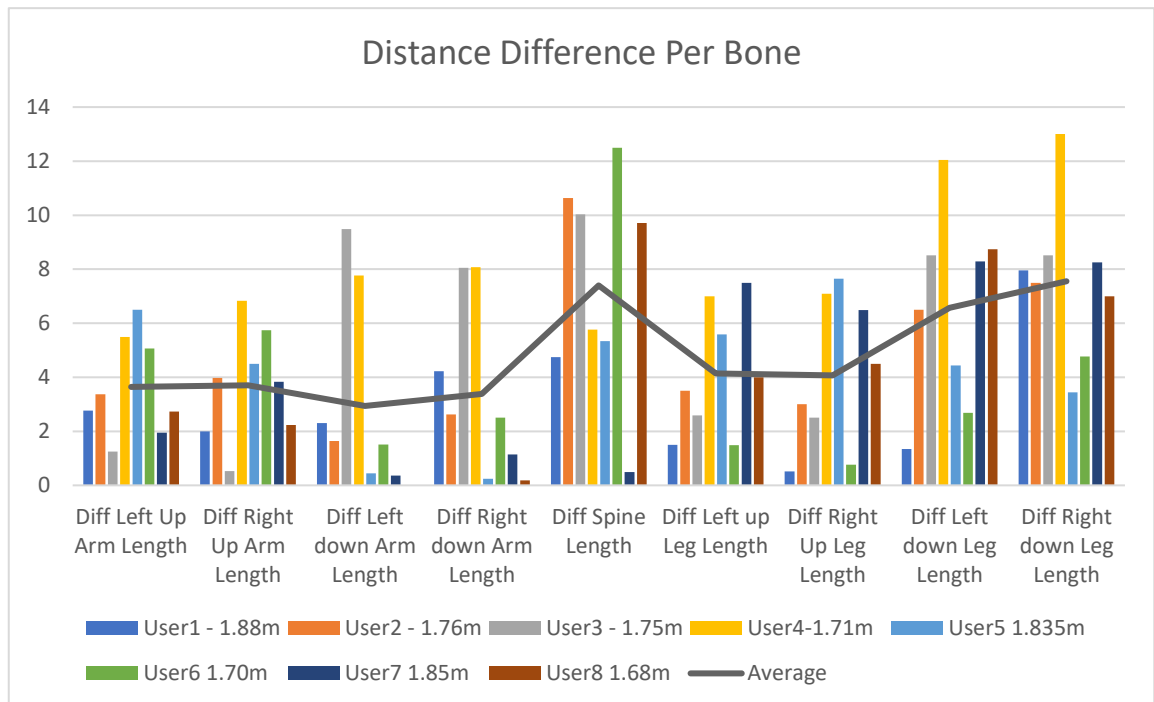
Από την γραφική παράσταση Graph 5.2.2.1 μας παρουσιάζει για κάθε χρήστη για κάθε του κόκκαλο την απόλυτη διαφορά που έχει το πραγματικό μήκος του κόκκαλου με το μήκος που υπολογίζει ο αλγόριθμος και τον μέσο όρο διαφοράς μήκους για όλων των κόκκαλων του κάθε χρήστη. Μέσα από την συγκεκριμένη γραφική παράσταση είναι άξιο παρατήρησης πως για ψηλότερους χρήστες, το ποσοστό διαφοράς του μήκους του κόκκαλου είναι μικρότερο εκτός από την σπονδυλική στήλη. Αυτό μπορεί να βεβαιωθεί και μέσω από την χρήστη user1, user5 και user7 που είναι 1.88 και 1.835 και 1.85 μέτρα αντίστοιχα. Για χρήστες που είναι ύψος μεταξύ 1.70 και 1.65 μετρά μπορούμε να παρατηρήσουμε πως έχουν μικρότερη διαφορά στα μήκη των κόκκαλων του χεριού μεταξύ 0 και 3 cm αλλά σε αυτά των ποδιών παρατηρώ να έχουν μεγαλύτερη διαφορά μεταξύ 4 και λίγο πιο πάνω από το 8 και αυτό φαίνεται καθαρά από τον χρήστη user6 και

user8 που είναι 1.70 και 1.68 αντίστοιχα. Για χρήστες μέτριου αναστήματος δηλαδή ύψος 1.71 με 1.79 παρατηρείται μια πιο ισορροπημένη διαφορά κόκκαλου μεταξύ όλων των κόκκαλων εκτός σε μία περίπτωση στον χρήστη user4. Κάτι ακόμα που παρατηρείται από την συγκεκριμένη γραφική παράσταση είναι πως για τον χρήστη με user4 υπάρχει η μεγαλύτερός μέσος όρος, από την διαφορά μήκους των κόκκαλων όπου είναι 8 σε αντίθεση με τους άλλους χρήστες που είναι περίπου 4 ο μέσος όρος της διαφοράς όλων των οστών.

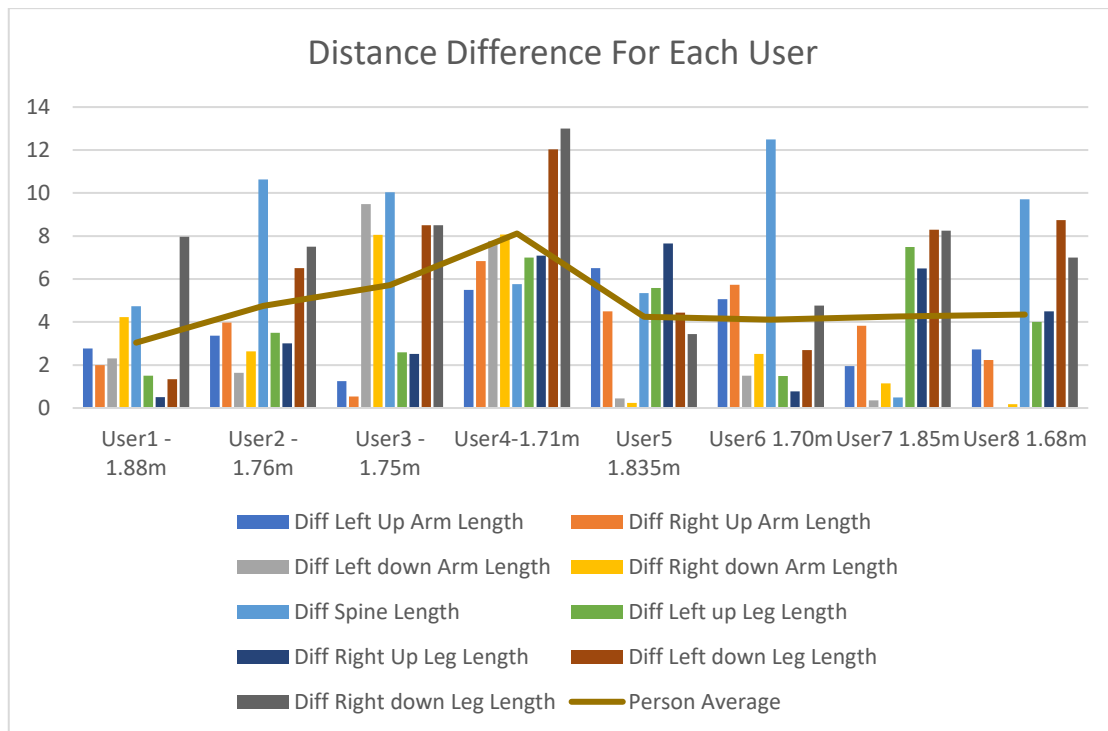
Η τελική παρατήρηση είναι πως υπάρχει μια μικρή απώλεια λάθους και να υπάρχει ένα μικρό ποσοστό κατά την δημιουργία της πόζας T-pose, αλλά το ποσοστό λάθους στην συγκεκριμένη περίπτωση θα είναι πολύ μικρό. Στην μόνη περίπτωση που θα παρακολουθείται μεγάλο ποσοστό λάθους είναι στην δημιουργία της σπονδυλικής στήλης και στην δημιουργία της άρθρωσης του στήθους στην T-pose πόζα, καθώς υπάρχει μεγάλο ποσοστό λάθους στην συγκεκριμένη περίπτωση. Εκεί που παρατηρείται να έχουμε το μικρότερο ποσοστό λάθους από όλα μας τα αποτελέσματα είναι στην περίπτωση των αρθρώσεων των ώμων και των χεριών καθώς στις περισσότερες φορές η διαφορά είναι πολύ μικροί από το κανονικό μήκος.

	Diff Left Up Arm Length	Diff Right Up Arm Length	Diff Left down Arm Length	Diff Right down Arm Length	Diff Spine Length	Diff Left up Leg Length	Diff Right Up Leg Length	Diff Left down Leg Length	Diff Right down Leg Length	Person Average
User1 - 1.88m	2.770	2.000	2.310	4.230	4.740	1.500	0.510	1.340	7.960	3.040
User2 - 1.76m	3.370	3.980	1.640	2.630	10.630	3.500	3.000	6.500	7.500	4.750
User3 - 1.75m	1.250	0.530	9.480	8.050	10.030	2.590	2.510	8.510	8.510	5.718
User4-1.71m	5.490	6.830	7.770	8.070	5.770	7.000	7.090	12.040	13.000	8.118
User5 1.835m	6.500	4.500	0.440	0.240	5.340	5.590	7.650	4.440	3.440	4.238
User6 1.70m	5.070	5.740	1.510	2.510	12.500	1.490	0.770	2.690	4.770	4.117
User7 1.85m	1.950	3.830	0.360	1.140	0.490	7.490	6.490	8.290	8.250	4.254
User8 1.68m	2.730	2.230	0.000	0.180	9.710	4.000	4.500	8.740	7.000	4.343
Average	3.641	3.705	2.939	3.381	7.401	4.145	4.065	6.569	7.554	4.822

Table 5.2.2.1



Graph 5.2.2.1



Graph 5.2.2.2

5.3 Αξιολόγηση του Forward Kinematics

Τα τελικά αποτελέσματα του αλγορίθμου όπως ανάφερα και προηγούμενος τοποθετούνται σε ένα ιεραρχικό αρχείο κίνησης. Στο συγκεκριμένο αρχείο περιέχονται μόνο οι συντεταγμένες του κεντρικού σημείου δηλαδή της λεκάνης μαζί με τις περιστροφές των αξόνων της, και όλες οι περιστροφές των αρθρώσεων σε μια ιεραρχική σειρά.

Το αποτέλεσμα που πάρθηκε όπως μπορούμε να δούμε από το Figure 5.3.1 είναι πως αρχικά μας παρουσιάζει των σκελετό του ανθρώπου στην στάση T-pose και στην συνέχεια μας παρουσιάζει την συνολική κίνηση του ανθρώπου βάση από τις περιστροφές των αξόνων της κάθε άρθρωσης. Στο Figure 5.3.2 μπορούμε να δούμε κατά την μετακίνηση των αρθρώσεων σε κάποιο από τα frames καθώς μπορούμε να παρατηρήσουμε πως μετακινητέ ελαφρώς το κεφάλι, τα γόνατα και τα χέρια.

Μέσα από τα αποτελέσματα που παράχθηκαν υπάρχουν και θετικά αλλά και αρνητικά δείγματα. Καθώς σε κάποιες περιπτώσεις παρατηρείτε μερική απώλεια λάθους και να μην εντοπίζονται σωστά κάποιες από τις περιστροφές των αξόνων. Αυτό παρατηρείτε περισσότερο όταν τα χέρια βρίσκονται μπροστά από το σώμα του χρήστη δηλαδή την σπονδυλική του στήλη. Επίσης μερικές φορές παρατηρείτε να περιστρέφονται τα πόδια ενώ αυτά δεν μετακινούνται ο λόγος που συμβαίνει αυτό είναι διότι όπως είδαμε

προηγούμενος ότι υπολογίζοντας λάθος τα foot contact labels, λόγω του ότι ο αλγόριθμος των δισδιάστατων αρθρώσεων μπερδεύει μερικές φορές που βρίσκονται τα κάτω άκρα των ποδιών ειδικά σε περιπτώσεις που είναι πολύ κοντά μεταξύ τους.

Όμως μέσα από τα αποτελέσματα πήρα και πολύ θετικά δείγματα όπως να υπολογίζεται σωστά πότε ο χρήστης μετακινεί το χέρι του επάνω η πότε μετακινεί το κορμό του. Σε περίπτωση που ο χρήστης μετακινεί κυκλικά τα χέρια του τότε υπολογίζεται σωστά η περιστροφή της κίνησης όμως σε μερικές περιπτώσεις μπορεί να χάσει λίγο το προσανατολισμό του χεριού κατά την ανακατασκευή της κίνησης και να μετακινηθεί λίγο παραπάνω ενώ δεν έπρεπε. Όμως σε αυτή την περίπτωση αυτό το ποσοστό λάθους μπορεί να μην το λάβουμε υπόψη καθώς είναι πολύ μικρό.

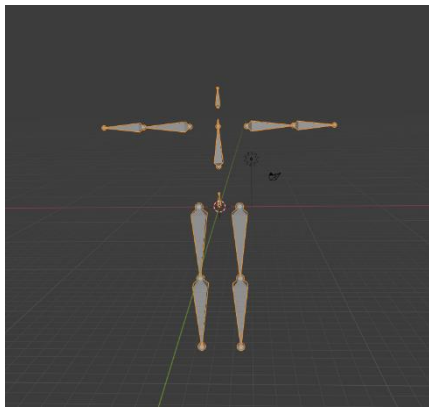


Figure 5.3.1

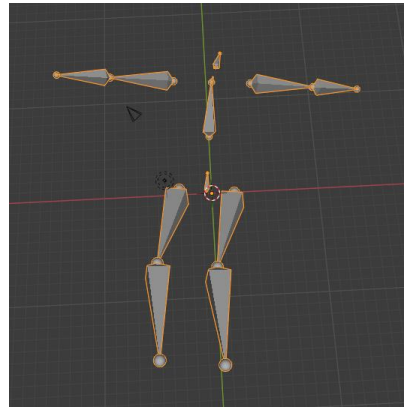


Figure 5.3.2

Κεφάλαιο 6

ΣΥΖΗΤΗΣΗ

6.1 Συζήτηση αποτελεσμάτων	48
----------------------------------	----

6.1 Συζήτηση αποτελεσμάτων

Στη συγκεκριμένη υπό ενότητα θα συζητηθούν τα δικά μου αποτελέσματα με αυτά της αρχικής εκδοχής του αλγορίθμου. Αυτό που θα αναληθθή είναι το τελικό αποτέλεσμα της κάθε υλοποίησης δηλαδή τα αρχεία κίνησης. Τα αποτελέσματα που θα αξιολογηθούν είναι 4 βίντεο από την βάση δεδομένων Human3.6M.

Μέσα από τα βίντεο της βάσης δεδομένων αναλύθηκαν 4 βίντεο τα Greeting, Discussion, Posing και το Smoking. Σε κάθε ένα από τα βίντεο ο χρήστης πραγματοποιώ διαφορετική κίνηση και εμείς αυτό που θα πραγματοποιήσουμε είναι κατά πόσο η δική μου υλοποίηση είναι κοντά στο πρωτότυπο του αλγορίθμου.

Μέσα από το Πρώτο Βίντεο το Greeting παρατηρείται πως η αρχική υλοποίηση του αλγορίθμου υπολογίζεται με μεγάλη ακρίβεια και υπολογίζεται σωστά πότε το πόδι δεν έχει επαφή με το έδαφος και πότε ο άνθρωπος που παρουσιάζεται στο βίντεο χαιρετάει και πότε υποκλίνεται. Μέσα από την δικιά μου υλοποίηση στο συγκεκριμένο βίντεο υπολογίζεται σωστά πότε ο χρήστης υποκλίνεται σωστά και πότε χαιρετάει το πλήθος. Μέσα από την δική μου υλοποίηση το μόνο ποσοστό λάθους είναι κατά την διάρκεια με το πρόβλημα στον υπολογισμό των foot contact labels, καθώς δεν υπολογίζεται σωστά πότε μετακινείτε το πόδι λόγο του προβλήματος που παρουσιάζει το PoseNet μερικές φορές στην ανάκτηση των άκριών των ποδιών, και μερικές φορές κάνει λάθος κίνηση δηλαδή την προηγούμενη κίνηση να υποκλίνεται. Στο Figure 6.1.1 μπορούμε να παρακολουθήσουμε μια στιγμή της κίνησης από το αρχείο κίνησης από την δική μου υλοποίηση που χαιρετάει σωστά το κόσμο και στο Figure 6.1.2 μπορούμε να δούμε την στιγμή που υποκλίνεται όπου είναι πολύ κοντά με την κίνηση από το βίντεο στην βάση δεδομένων.

Μέσα από το δεύτερο βίντεο αυτό του Discussion, από την αρχική υλοποίηση του αλγορίθμου παρατηρείται σε πολύ καλή οι κίνηση του χεριού μπροστά από το στόμα καθώς υπολογίζονται πολλή ορθά οι περιστροφές των αρθρώσεων. Επίσης το ίδιο συμβαίνει και στις περιστροφές των αρθρώσεων όπου κουνούνται λίγο τα πόδια εκεί που πρέπει. Στην δικιά μου υλοποίηση αυτό που παρατηρείται είναι ότι υπολογίζεται λάθος η περιστροφή των αρθρώσεων, καθώς τα χέρια μετακινούνται λάθος και υπάρχει ένα μικρό λάθος στο αριστερό πόδι καθώς περιστρέφεται σε μεγάλο βαθμό.

Μέσα από το τρίτο βίντεο του Posing, μέσα από το αποτέλεσμα από την αρχική υλοποίηση φαίνεται να είναι σωστά όταν ο χρήστης βρίσκεται στην αρχική στάση του σώματος σε αυτή της T-pose πόζας και στην συνέχεια που κατεβάζει το ένα χέρι κάτω και με το άλλο παίρνει μια πόζα όπου το ένα χέρι είναι το μισό στο αέρα σαν θα τον βγάλουν φωτογραφία. Από την δικιά μου υλοποίηση αυτό που παρατηρείται είναι ότι παρουσιάζεται ορθά η αρχική στάση που βρίσκεται στην T-pose πόζα και στην συνέχεια παρουσιάζεται με μικρό ποσοστό λάθους η πόζα που παίρνει ο χρήστης, το μόνο λάθος σε αυτή την περίπτωση είναι πως δεν κλίνει σωστά το χέρι κατά την αναπαράσταση από το bnh αρχείο. Μέσα από το Figure 6.1.3 και Figure 6.1.4 μπορούμε να δούμε που προσπαθεί να πάρει την στάση από το βίντεο όπως το Figure 6.1.5 και Figure 6.1.6 αντίστοιχα. Το μόνο λάθος είναι και πάλι μερικές φορές οι περιστροφές των αρθρώσεων του ποδιού.

Το τελευταίο βίντεο που αξιολογήθηκε από το συγκεκριμένο dataset είναι αυτό του smoking όπου ο χρήστης προσπαθεί να ανάψει τσιγάρο και να το καπνίσει. Το αποτέλεσμα από την αρχική υλοποίηση του αλγορίθμου είναι σχεδόν το ίδιο με αυτό του βίντεο καθώς κάνει τα ίδια βήματα με αυτά του βίντεο βάζει το χέρι του στην τσέπη του και ανάβει το τσιγάρο με τον ίδιο τρόπο όπως και στο βίντεο. Όμως από την δικιά μου υλοποίηση τα αποτελέσματα που παίρνω στο αρχείο κίνησης, δεν είναι και τα καλύτερα καθώς δεν υπολογίζονται σωστά οι περιστροφές τις κίνησης και υπάρχει μεγάλο ποσοστό σφάλματος με να μην αναπαρίσταται σωστά η κίνηση.

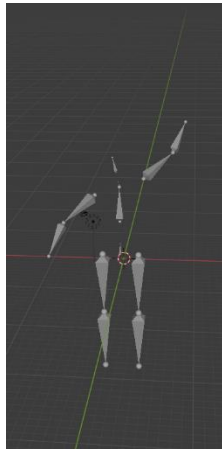


Figure 6.1.1

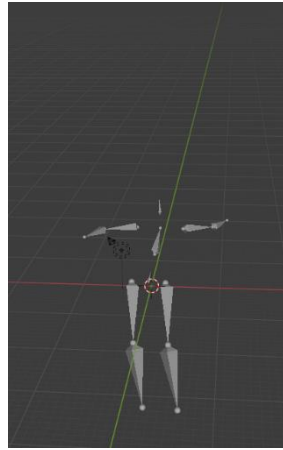


Figure 6.1.2

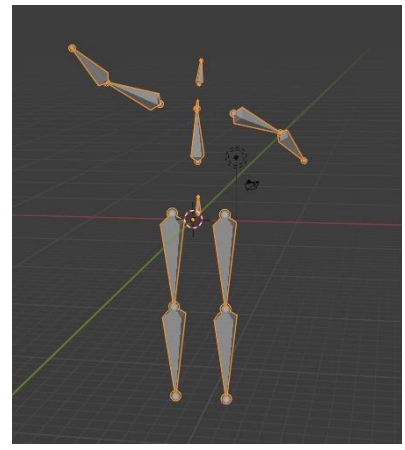


Figure 6.1.3

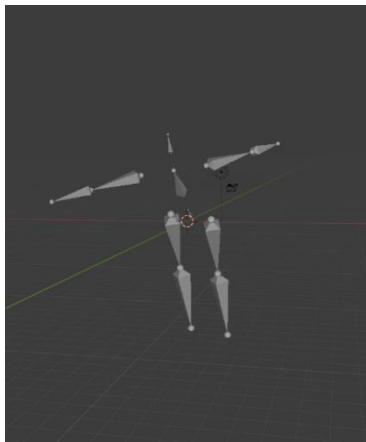


Figure 6.1.4



Figure 6.1.5

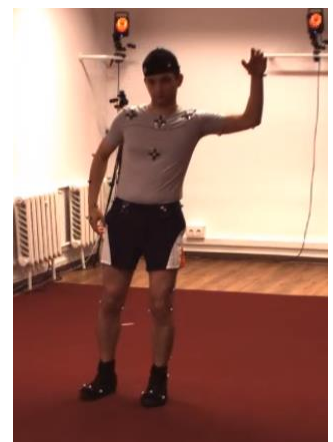


Figure 6.1.6

Κεφάλαιο 7

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

7.1 Συμπεράσματα	51
7.2 Μελλοντική εργασία	52

7.1 Συμπεράσματα

Τα αρχικά συμπεράσματα από την πτυχιακή εργασία αρχικά είναι πως δεν είναι έτυμη καμία συσκευή microcontroller για να υποστήριξη κάποιο είδος ανάκτησης της κίνησης από βίντεο, με την χρήση ενός μοντέλου που θα χρησιμοποιεί ένα νευρωνικό δίκτυο για την ανάκτηση τους. Για υλοποίηση μηχανικής μάθησης σε τέτοιου είδους συσκευές είναι ικανές για απλούς αλγορίθμους υλοποίησης νευρωνικό δικτύων για computer vision όπως αναγνώριση αντικειμένων, ή για αναγνώριση εάν αυτό που υπάρχει στην κάμερα είναι άνθρωπος ή όχι.

Το συμπέρασμα μου από την υλοποίηση της εφαρμογής σε android συσκευή είναι πολύ θετικά. Μέσα από τη ανάκτηση των confident values και όπως είδαμε από την ανάλυση των αποτελεσμάτων με την χρήση του αλγόριθμου του PoseNet με μικρό ποσοστό σφάλματος. Οπότε προτείνεται η χρήση του συγκεκριμένου αλγορίθμου για την ανάκτηση των δισδιάστατων αρθρώσεων. Όμως η καταγραφή των βίντεο ανά 20 frames δεν είναι και πολύ καλή καθώς κατά τον υπολογισμό 20 frames χρειάζονται περίπου 20 με 30 δευτερόλεπτα και με αυτό τον τρόπο χάνει κάποια frames κατά την καταγραφή του βίντεο και θα υπάρχει μεγάλο ποσοστό λάθους. Το συμπέρασμα που βγάζω μέσα από τα αποτελέσματα του υποδικτύου EQ είναι πολύ θετικά καθώς υπολογίζονται με μεγάλη ακρίβεια το δεξιό μέρος της λεκάνης και το ίδιο συμβαίνει και με το αριστερό μέρος της λεκάνης τότε το κεντρικό σημείο, δηλαδή το root position υπολογίζεται με πολύ μικρό ποσοστό λάθους. Όμως το συμπέρασμα για των υπολογισμό των foot contact labels δεν είναι και πολύ θετικό διότι μερικές φορές υπολογίζεται λάθος, λόγω του ότι ο αλγόριθμος του PoseNet για την ανάκτηση των δισδιάστατων αρθρώσεων μπερδεύει την θέση των

αστραγάλων και τα τοποθετεί αντίθετα. Από το υποδίκτυο E_s βγήκα στο συμπέρασμα ότι λόγω το ότι η διαφορά υπολογισμού του μήκους των κοκάλων από τον αλγόριθμο σε σχέση με το πραγματικό μήκος είναι πολύ μικροί πόζα T-pose κτίζεται πολύ σωστά με μικρό ποσοστό λάθους και αυτό με βοηθά στην μείωση ποσοστού λάθους στην συνέχεια. Το τελικό συμπέρασμα από την χρήση της τεχνικής του Forward Kinematic, που είναι εμπνευσμένη από τον τρόπο που λειτουργεί στην ρομποτική δεν είναι και πολύ θετικό καθώς μερικές φορές υπολογίζεται λάθος ο άξονας ο z με αποτέλεσμα να υπολογίζονται λάθος οι περιστροφές των αρθρώσεων. Οπότε το συμπέρασμα μου είναι να χρησιμοποιείτε σε περιπτώσεις όπου τα πόδια είναι αμελητέα δηλαδή μόνο το πάνω μέρος του σώματος μας ενδιαφέρει.

7.2 Μελλοντική εργασία

Η εργασία που κατασκεύασα επάνω στο microcontroller, διότι υλοποιήθηκε ο τρόπος που παράγονται frames και να καταγράφεται μία εικόνα από την κάμερα να μπορεί να χρησιμοποιηθεί για άλλες μελλοντικές εργασίες για επεξεργασία εικόνας. Όπου να μπορεί να αναλύετε μια εικόνα κατά πόσο περιέχει μια ασθένεια ένα όργανο και κατά πόσο όχι με την χρήση ενός μικρού μοντέλου που να περιέχει μέσα ένα νευρονικό δίκτυο που θα αποφασίζει. Μέσα από την υλοποίηση στην συγκεκριμένη πλατφόρμα μπορεί να υλοποιηθούν πολλά project που θα αναλύουν μια εικόνα η κάποιο βίντεο και το μοντέλο τους θα είναι πολύ μικρό και θα μπορεί να το επεξεργαστή ο microcontroller.

Η εφαρμογή που ανέπτυξα σε android συσκευή σε μελλοντική εργασία αυτό που μπορεί να βελτιωθεί είναι η ανάκτηση του άξονα z, δηλαδή να μειωθεί το ποσοστό λάθους σε αυτή την περίπτωση για καλύτερα αποτελέσματα. Σε αυτή την περίπτωση εάν χρησιμοποιηθεί κάποιο smartphone που έχει ενσωματωμένο ένα depth sensor και να καταφέρει να ανάκτηση τον άξονα z. Με παρόμοιο τρόπο το Azure Kinect της Microsoft, υπολογίζει τον άξονα τον z βάση του depth sensor που έχει ενσωματωμένο που υπολογίζει το βάθος που βρίσκεται ο χρήστης και με αυτό το τρόπο παράγει τον άξονα τον z. Το συγκεκριμένο αισθητήρα εμπεριέχουν τα κινητά τηλέφωνα τύπου smartphone των τελευταίων 2 με 3 χρόνων και το χρησιμοποιούν για υπολογισμό του βάθους.

Η βελτίωση ανάκτησης της κίνησης του άξονα z, μπορεί να χρησιμοποιηθεί με την χρήση accelerometers καθώς μπορούν να υπολογίζουν την απόσταση όταν μετακινείται ο χρήστης. Το αρνητικό αυτής όμως της βελτίωσης είναι πως ο χρήστης σε κάθε άρθρωση

του πρέπει να είναι ενσωματωμένος με τους συγκεκριμένους αισθητήρες για να μπορούν να καταγράψουν τον άξονα τον z.

Μια ακόμη μελλοντική εργασία που μπορεί να υλοποιηθεί βάση με τα αποτελέσματα των δισδιάστατων αρθρώσεων είναι η αξιολόγηση κάποιων ασκήσεων μέσω από της δισδιάστατες τοποθεσίες των αρθρώσεων. Όπως είδαμε και από το Κεφαλαίο 5, ο εντοπισμός των δισδιάστατων σημείων των αρθρώσεων εντοπίζονται με μικρό ποσοστό λάθους. Μέσα από ένα εκπαιδευμένο νευρονικό δίκτυο, που θα είναι εκπαιδευμένο σε συγκεκριμένες ασκήσεις σωματικής εξάσκησης και αποκατάστασης και θα επιστρέφει κατά πόσο εκτελείτε η άσκηση σωστά και πως να διορθώσει τον τρόπο εκτέλεσης της άσκησης. Μια παρόμοια υλοποίηση δημιουργήθηκε από τους Steven Chen and Richard R. Yang, όπου έχει σαν είσοδο τα διανύσματα της κάθε άρθρωσης που χρειάζεται για την κάθε άσκηση, που τα συγκεκριμένα διανύσματα μπορούν να εντοπισθούν μέσα από τις δισδιάστατες αρθρώσεις. Μέσα από τα διανύσματα αυτά που θα είναι η είσοδος στο μοντέλο που θα βρίσκει κατά πόσο εκτελούνται σωστά με την βοήθεια μια ευχετικής συνάρτησης.

Βιβλιογραφία

- [1] Mingyi Shi, Andreas Aristidou, Kfir Aberman, Taku Komura, Dani Lishinski, Daniel Cohen-OR and Baoquan Chen, MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency, Shandong University and AICFVE, Beijing Film Academy, 2020.
- [2] Pete Warden and Daniel Situnayake, TinyML “Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers”, First Edition, 2019.
- [3] Kirk Benell, Rob Reynolds, Scott Hanson and Pete Warden, Low-Power Machine Learning, 2019.
- [4] https://github.com/sparkfun/Tensorflow_AIOT2019(accessed December 2020).
- [5] <https://learn.sparkfun.com/>(accessed January 2021)
- [6] <https://opensource.com/resources/what-arduino>
- [7] <https://visualise.com/virtual-reality/virtual-reality-healthcare>(Accessed January 2021)
- [8] ARMv7-M Architecture Reference Manual(pp A1-31- A1-37), 2006-2010
- [9] ARM Cortex-M7 Processor Datasheet (pp 1-5)
- [10] “<https://www.circuitbasics.com/basics-uart-communication/#:~:text=UART%20stands%20for%20Universal%20Asynchronous,transmit%20and%20receive%20serial%20data.>”(Accessed January 2021)
- [11] “<https://embeddedinventor.com/quad-spi-everything-you-need-to-know/>”(Accessed January 2021)

- [12] “<https://towardsdatascience.com/an-introduction-to-tinyml-4617f314aa79>”(Accessed January 2021)
- [13] George Papandreou, Tyler Zhu, Liang-Chien Chen, Spyros Gidaris, Jonathan Tompson and Kevin Murphy, PersonLab: Person Pose Estimation and Instance Segmentation with Bottom-Up, Part-Based, Geometric Embedding Model, March 2018.
- [14] Ruben Villegas, Tyler Zhu, Jimei Yang, Duygu Ceylan and Honglak Lee, Neural Kinematic Networks for Unsupervised Motion Retargeting, University of Michigan, Adobe Research, Google Brain.
- [15] Jennifer R. Kwapisz, Gary M. Weiss and Samuel A. Moore, Activity Recognition using Cell Phone Accelerometers, Fordham University
- [16] Steven Chen and Richard R. Yang, Pose Trainer: Correcting exercise Posture using Pose Estimation, Department of Computer Science Stanford University.
- [17] “<https://developer.android.com/guide>”
- [18] Catalin Ionescu, Dragos Papava, Vlad Olaru, Christian Sminchisescu, Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments
- [19] Ahmad Jalal, Shaharyar Kamal, Daijin Kim, Human Depth Sensors-Based Activity Recognition Using Spatiotemporal Features and Hidden Markov Model for Smart Environments

Παράρτημα Α

Σε αυτό το παράρτημα θα αναλύσουμε ορισμούς και ακρώνυμα όπου σχετίζονται με την ατομική διπλωματική μου εργασία.

Ορισμοί

- ReLU : Η ReLU είναι μια μαθηματική συνάρτηση δραστηριοποίησης(activation function) όπου χρησιμοποιείται για την διαμόρφωση του τελικού αποτελέσματος του δικτιού. Η συγκεκριμένη αυτή συνάρτηση επιστρέφει 0 εάν το x είναι αρνητικό και εάν x είναι θετικό τότε επιστρέφει την τιμή.
- Dataset : Είναι μία βάση δεδομένων από κάποια δεδομένα. Τα dataset στην μηχανική μάθηση είναι εκπαιδευμένα σαν ένα μοντέλο για κάποιο συγκεκριμένο σκοπό. Για παράδειγμα να είναι εκπαιδευμένα καλά το dataset για pose estimation, human recognition κτλ.
- Deep neural network : Το deep neural network σημαίνει βάθος νευρωνικό δίκτυο όπου χρησιμοποιείται στην μηχανική μάθηση. Αποτελείται από το input layer όπου είναι τα δεδομένα εισόδου από 3 και περισσότερα hidden layer και στο τέλος είναι το output layers όπου είναι τα αποτελέσματα εξόδου. Κάθε hidden layer του δικτύου συμπεριλαμβάνεται από N κόμβους και κάποια ενδιάμεσα layers ενώνονται μεταξύ τους όπως τους νευρώνες του ανθρώπινου εγκεφάλου. Όλα τα layers συνεργάζονται μεταξύ τους για να ανακαλύψουν την καλύτερη δυνατή λύση σε σχέση με τη λύση θέλει ο αλγόριθμος που βρίσκετε το συγκεκριμένου βαθύ νευρωνικό δίκτυο. Μόλις βρεθεί η λύση τότε τοποθετείτε στο output layer.
- Global root position : Είναι οι κεντρικές συντεταγμένες (x, y, z) του ανθρώπινου σκελετού στην κάμερα. Όπου εκεί συγκεκριμένα βρίσκεται η λεκάνη του ανθρώπινου σώματος.
- Foot contact labels : Είναι οι συντεταγμένες (x, y, z) για την βάση του ποδιού και της φτέρνας για κάθε πόδι ξεχωριστά.
- Data agnostic inverse kinematic solver : Είναι μια μαθηματική διαδικασία όπου υπολογίζει τα δεδομένα δηλαδή τις συντεταγμένες των αρθρώσεων που απαιτούνται για την τοποθέτηση στο τέλος μια κινηματικής αλυσίδας. Επίσης χρησιμοποιείται για να ανακτήσουμε κινήσεις από άλλα δεδομένα όπως για

παράδειγμα σε ποια πορεία κατευθύνεται κάποιος άνθρωπος όπου κινείται στην κάμερα.

- Forward Kinematics : Μέσα από την χρήση κινηματικών εξισώσεων, για τον υπολογισμό της τελικής θέσεις της άρθρωσης μέσα από της προκαθορισμένες τιμές των αρθρώσεων.
- Bauds : Πόσες φορές ένα σήμα μπορεί να αλλάξει σε διαφορετικό σήμα μέσα σε ένα δευτερόλεπτο.

Ακρώνυμα

- RAM : Random Accessed Memory
- ARM :
- SPI : Serial Peripheral Interface
- VR : Virtual Reality
- MEMS : Micro-ElectroMechanical System
- FPS : Frames Per Second
- FPU : Floating-Point Unit
- KB : Kilobyte
- MB : Megabyte
- ML : Machine Learning
- FK : Forward Kinematics
- IK : Inverse Kinematics