

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΚΑΤΕΥΘΗΝΣΗ**

**ΥΛΙΚΟΥ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας τοποθετημένης σε αυτόματο περιστροφικό αναδιπλούμενο βραχίονα ελεγχόμενο από μικροϋπολογιστή**

**Χρήστος Γ. Τερζόγλου**

**Εισηγητές: Χρήστος Καμπούρης**

**Ιωάννης Βογιατζής**

**ΑΘΗΝΑ ΙΟΥΛΙΟΣ 2022**

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας τοποθετημένης μέσω κάμερας τοποθετημένης σε αυτόματο περιστροφικό αναδιπλούμενο βραχίονα ελεγχόμενο από μικροϋπολογιστή

Χρήστος Γ. Τερζόγλου

A.M. 71344977

Εισηγητής διπλωματικής εργασίας: Χρήστος Καμπούρης

Ιωάννης Βογιατζής

Η διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική επιτροπή

Α/α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	Ιωάννης Βογιατζής	Καθηγητής	
2	Σπυρίδων Ματιάτος	Λέκτορας Εφαρμογών	
3	Χρήστος Καμπούρης	Ακαδημαϊκός Υπότροφος	

Ημερομηνία εξέτασης:

ΙΟΥΛΙΟΣ 2022

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Χρήστος Τερζόγλου του Γεωργίου με αριθμό μητρώου 44977 φοιτητής του Πανεπιστημίου Δυτικής Αττικής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου»

Ο Δηλών

Χρήστος Τερζόγλου



Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή, κύριο Χρήστο Καμπούρη για την υπομονή και την καθοδήγηση του σε όλα τα στάδια ανάπτυξης της εργασίας.

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την υποστήριξη και την βοήθεια που μου προσέφεραν.

Τέλος θα ήθελα να ευχαριστήσω την συνάδελφο Ναταλία Ζαχαρία για την προσφορά του ρομποτικού βραχίονα που έκανε δυνατή την επέκταση της παρούσας εργασίας.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## **ΤΙΤΛΟΣ**

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας τοποθετημένης σε αυτόματο περιστροφικό αναδιπλούμενο μηχανικό βραχίονα ελεγχόμενο από μικροϋπολογιστή

## **ΠΕΡΙΛΗΨΗ**

Η πτυχιακή περιλαμβάνει την ανάπτυξη λογισμικού για υπάρχον σύστημα, το οποίο λογισμικό θα είναι υπεύθυνο για τον έλεγχο κίνησης του βραχίονα.

## **ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ**

Όραση υπολογιστών, ανίχνευση και εντοπισμός προσώπων, python, openCV, Μέθοδος τοπικών δυαδικών προτύπων, μέθοδος καταρράκτη Viola & Jones

## **TITLE**

Software development for face recognition and tracking via a camera attached on an automatic rotary-foldable mechatronics arm controlled via a microcomputer

## **ABSTRACT**

This dissertation includes the software development for an existing arm. The software is responsible for controlling the arm.

## **KEYWORDS**

Computer vision, Face detection and recognition, python OpenCV, Local Binary pattern histogram, Cascade Classifier, Viola and Jones Method

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ</b> .....	3
<b>ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ</b> .....	5
<b>ΕΥΧΑΡΙΣΤΙΕΣ</b> .....	7
<b>ΠΕΡΙΛΗΨΗ</b> .....	9
<b>ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ</b> .....	9
<b>ABSTRACT</b> .....	9
<b>KEYWORDS</b> .....	9
<b>ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ</b> .....	15
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ</b> .....	19
<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ</b> .....	21
<b>ΕΙΣΑΓΩΓΗ</b> .....	23
<b>ΚΕΦΑΛΑΙΟ 1: ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ</b> .....	25
<b>1.1: Εισαγωγή</b> .....	25
<b>1.2: Ανάλυση απαιτήσεων</b> .....	25
<b>1.3: Ανάλυση λειτουργιών</b> .....	27
<b>ΚΕΦΑΛΑΙΟ 2: Εντοπισμός και αναγνώριση προσώπων, εισαγωγή και βασικές έννοιες</b> .....	29
<b>2.1: Ο εντοπισμός του προσώπου</b> .....	29
2.1.1: Εισαγωγή .....	29
2.1.2: Βασικές αρχές εντοπισμού προσώπου .....	29
2.1.3: Ανάλυση και σύγκριση τεχνικών εντοπισμού προσώπου.....	29
2.1.3.1: Τεχνική Viola & Jones:.....	30
2.1.3.2: Τεχνική ιστογραμμάτων κλίσεων : .....	30
2.1.4: Επιλογή κατάλληλης μεθόδου: .....	32
<b>2.2: Η αναγνώριση του προσώπου</b> .....	32
2.2.1: Εισαγωγή .....	32
2.2.2: Βασικές αρχές αναγνώρισης προσώπων .....	33
2.2.3: Ανάλυση και σύγκριση τεχνικών αναγνώρισης προσώπων .....	33
2.2.3.1: Η τεχνική ανάλυσης κύριων συνιστωσών .....	33
2.2.3.2: Η τεχνική των τοπικών δυαδικών προτύπων .....	35
2.2.4: Επιλογή κατάλληλης τεχνικής .....	35
<b>Κεφάλαιο 3: Εντοπισμός προσώπων: Η μέθοδος Viola &amp; Jones και η μέθοδος τοπικών δυαδικών προτύπων</b> .....	37

<b>3.1: Η μέθοδος Viola &amp; Jones</b> .....	37
3.1.1: Εισαγωγή: .....	37
3.1.2: Η δημιουργία και ανάλυση της ολοκληρωτικής εικόνας:.....	37
3.1.3: Τα χαρακτηριστικά της μεθόδου Viola & Jones.....	40
3.1.4: Εκπαίδευση του ταξινομητή ή χρήση έτοιμων αρχείων .....	41
3.1.5: Χρήση έτοιμων αρχείων .....	43
3.1.6: Επιλογή κατάλληλης μεθόδου .....	43
3.1.7: Ο καταρράκτης ταξινομητών.....	44
<b>3.2: Η διαδικασία των τοπικών δυαδικών προτύπων για αναγνώριση προσώπων</b> .....	44
<b>Κεφάλαιο 4: Εντοπισμός κινήσεων</b> .....	49
<b>4.1: Ανάλυση τεχνικών εντοπισμού κίνησης</b> .....	49
4.1.1: Η μέθοδος αφαίρεσης φόντου.....	49
4.1.2: Η μέθοδος της οπτικής ροής .....	49
<b>4.2: Επιλογή κατάλληλης μεθόδου</b> .....	50
<b>4.3: Ανάλυση της μεθόδου αφαίρεσης φόντου</b> .....	50
<b>Κεφάλαιο 5: Ο υλικός εξοπλισμός της εργασίας</b> .....	55
<b>5.1: Εισαγωγή</b> .....	55
<b>5.2: Διαθέσιμοι μικροϋπολογιστές: NVIDIA Jetson Nano</b> .....	55
5.2.1: Εισαγωγή .....	55
5.2.2: Τεχνικά χαρακτηριστικά.....	56
<b>5.3: Διαθέσιμοι μικροϋπολογιστές Raspberry Pi</b> .....	57
5.3.1: Εισαγωγή .....	57
5.3.2: Τεχνικά χαρακτηριστικά.....	58
<b>5.4: Επιλογή κατάλληλου μικροϋπολογιστή</b> .....	58
<b>5.5: Παραμετροποίηση του Raspberry Pi</b> .....	59
5.5.1: Εγκατάσταση λειτουργικού συστήματος.....	59
5.5.2: Προσβασιμότητα από το τοπικό δίκτυο.....	62
5.5.3: Εγκατάσταση πακέτων.....	64
<b>5.6: Υλοποίηση σειριακής επικοινωνίας μεταξύ Raspberry Pi και Arduino</b> .....	65
5.6.1: Εισαγωγή .....	65
5.6.2: Σύνδεση Raspberry Pi και Arduino .....	65
<b>5.7: Λοιπός υλικός εξοπλισμός</b> .....	67
5.7.1: Η κάμερα.....	67
5.7.2: Οι αισθητήρες κίνησης .....	68

5.7.3: Οι σερβοκινητήρες.....	68
5.7.4: Ο κωδικοποιητής.....	69
<b>Κεφάλαιο 6: Ανάλυση της εφαρμογής .....</b>	<b>71</b>
<b>6.1: Ανάλυση πακέτων και προγραμμάτων που χρησιμοποιήθηκαν .....</b>	<b>71</b>
6.1.1: Εισαγωγή: .....	71
6.1.2: Το πρόγραμμα MATLAB .....	71
6.1.3: η βιβλιοθήκη simpleCV:.....	71
6.1.4: η βιβλιοθήκη openCV:.....	71
6.1.5: Επιλογή γλώσσας προγραμματισμού.....	72
6.1.6: Διαχείριση βάσεων δεδομένων .....	72
<b>6.2: Ανάλυση της εφαρμογής .....</b>	<b>73</b>
<b>6.3: Το σχήμα της βάσης δεδομένων .....</b>	<b>73</b>
<b>6.4: Εκτέλεση της εφαρμογής .....</b>	<b>74</b>
6.4.1: Διαχείριση δεδομένων χρηστών .....	77
6.4.1.1: Εισαγωγή χρήστη.....	79
6.4.1.2: Ενημέρωση στοιχείων χρήστη .....	82
6.4.1.3: Διαγραφή στοιχείων χρήστη .....	84
6.4.2: Εκπαίδευση του συστήματος μέσω των τοπικών δυαδικών προτύπων .....	88
6.4.3: Επιτήρηση χώρου και αναγνώριση περιοχών ενδιαφέροντος.....	90
6.4.3.1: Σενάριο με εντοπισμό προσώπου .....	93
6.4.3.2: Σενάριο με εντοπισμό άνω μέρους σώματος .....	99
6.4.3.3: Σενάριο με εντοπισμό πλήρους σώματος .....	103
6.4.3.4: Σενάριο με εντοπισμό κίνησης .....	107
6.4.3.5: Δεδομένα σειριακής επικοινωνίας .....	113
<b>6.5: Ρυθμίσεις .....</b>	<b>113</b>
6.5.1: Εισαγωγή ή διαγραφή παραμέτρων .....	116
6.5.2: Εμφάνιση πλέγματος.....	119
6.5.3: Εμφάνιση πλέγματος με εντοπισμό προσώπου.....	120
6.5.4: Εμφάνιση πλέγματος με εντοπισμό άνω μέρους σώματος .....	125
6.5.5: Εμφάνιση πλέγματος με εντοπισμό πλήρους σώματος.....	129
6.5.6: Εμφάνιση πλέγματος με εντοπισμό κίνησης.....	133
6.5.7: Εμφάνιση πλέγματος με εντοπισμό αντικειμένων .....	137
<b>6.6: Ενίσχυση βίντεο για εντοπισμό αντικειμένων .....</b>	<b>140</b>

<b>Κεφάλαιο 7: Προβλήματα που παρουσιάστηκαν</b> .....	147
<b>7.1: Προβλήματα με το Raspberry Pi</b> .....	147
<b>7.2: Προβλήματα με την σειριακή επικοινωνία</b> .....	147
<b>7.3: Πρόβλημα με την καταγραφή των βίντεο</b> .....	147
<b>7.4: Πρόβλημα με την καταγραφή σε βίντεο</b> .....	148
<b>7.5: Πρόβλημα κατά την καταγραφή την εμφάνιση των αντικειμένων</b> .....	149
<b>Κεφάλαιο 8: Συμπεράσματα και μελλοντικές επεκτάσεις</b> .....	151
<b>8.1: Συμπεράσματα</b> .....	151
<b>8.2: Μελλοντικές επεκτάσεις</b> .....	151
<b>Κεφάλαιο 9: Ανάλυση SWOT</b> .....	153
<b>Παραρτήματα</b> .....	155
<b>Παράρτημα 1</b> .....	155
<b>Παράρτημα 2</b> .....	157
<b>Βιβλιογραφία</b> .....	163

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1- 1: Τα βήματα ανάπτυξης της εργασίας .....	25
Εικόνα 2- 1: Μέθοδος Viola & Jones .....	30
Εικόνα 2- 2: Αναπαράσταση κλίσεων και πλατών (το μήκος του βέλους αναπαριστά το πλάτος).....	31
Εικόνα 2- 3: Δημιουργία Ιστογράμματος Κλίσεων .....	31
Εικόνα 2- 4: Αναπαράσταση προσώπου μέσω HOG.....	32
Εικόνα 2- 5: Τα στάδια αναγνώρισης προσώπων .....	33
Εικόνα 2- 6: Δημιουργία διανυσμάτων από τις εικόνες.....	33
Εικόνα 2- 7: Μέση εικόνα και EigenFace εικόνες.....	34
Εικόνα 2- 8: Κατασκευή ιστογραμμάτων για την μέθοδο LBPH.....	35
Εικόνα 3- 1: Σχηματική αναπαράσταση της ολοκληρωτικής εικόνας.....	38
Εικόνα 3- 2: Αναπαράσταση προσώπων σε αρχική και ολοκληρωτική εικόνα .....	38
Εικόνα 3- 3: Υπολογισμός αθροίσματος.....	39
Εικόνα 3- 4: Το πρώτο στάδιο του υπολογισμού .....	39
Εικόνα 3- 5: Το δεύτερο στάδιο υπολογισμού (Αφαίρεση περιττών φωτεινότητων).....	39
Εικόνα 3- 6: Το τρίτο στάδιο υπολογισμού (αφαίρεση του τελευταίου μέρους).....	40
Εικόνα 3- 7: Άθροιση της περιοχής που έχει αφαιρεθεί δύο φορές.....	40
Εικόνα 3- 8: Τα χρησιμότερα χαρακτηριστικά.....	41
Εικόνα 3- 9: Σχηματική αναπαράσταση του καταρράκτη ταξινομητών.....	44
Εικόνα 3- 10: Σχηματική αναπαράσταση των παραμέτρων ακτίνας και γειτόνων.....	45
Εικόνα 3- 11: Ο υπολογισμός των τοπικών δυαδικών προτύπων.....	45
Εικόνα 3- 12: Κατασκευή ιστογραμμάτων της εικόνας.....	46
Εικόνα 3- 13: Σύγκριση ιστογραμμάτων .....	46
Εικόνα 4- 1: Η τεχνική αφαίρεσης φόντου .....	49
Εικόνα 4- 2: Η τεχνική της οπτικής ροής .....	50
Εικόνα 4- 3: Μετατροπή της εικόνας σε μονοχρωματική .....	51
Εικόνα 4- 4: Εφαρμογή Gaussian Blur στην εικόνα.....	51
Εικόνα 4- 5: Σύγκριση μεταξύ δύο διαδοχικών καρέ .....	52
Εικόνα 4- 6: Διαστολή φωτεινών εικονοστοιχείων .....	53
Εικόνα 4- 7: Το τελικό καρέ που εμφανίζεται στην κάμερα .....	53
Εικόνα 5- 1: Το Jetson Nano της NVIDIA .....	55
Εικόνα 5- 2: Το λειτουργικό σύστημα L4T .....	56
Εικόνα 5- 3: Σειριακή επικοινωνία μεταξύ Jestson και του μικροϋπολογιστή.....	57
Εικόνα 5- 4: Το Raspberry Pi .....	57
Εικόνα 5- 5: Το πρόγραμμα SD Card Formatter .....	60
Εικόνα 5- 6: Λήψη του προγράμματος Raspberry Pi Imager .....	60
Εικόνα 5- 7: Το πρόγραμμα Raspberry Pi Imager .....	61

Εικόνα 5- 8: Επιλογή ρυθμίσεων για το Raspberry Pi.....	61
Εικόνα 5- 9: Παραμετροποίηση του Raspberry Pi.....	62
Εικόνα 5- 10: Εύρεση της IP του Raspberry Pi.....	63
Εικόνα 5- 11: Σύνδεση μέσω Remote Desktop Connection.....	63
Εικόνα 5- 12: Επιφάνεια εργασίας μέσω του Remote Desktop.....	64
Εικόνα 5- 13: Σύνδεση μέσω της θύρας USB.....	66
Εικόνα 5- 14: Σύνδεση μέσω των ακροδεκτών RX/TX.....	66
Εικόνα 5- 15: Η δομή του πακέτου της ασύγχρονης επικοινωνίας.....	67
Εικόνα 5- 16: Η κάμερα για το Raspberry Pi.....	68
Εικόνα 5- 17: Ο αισθητήρας κίνησης HC-SR501.....	68
Εικόνα 5- 18: Ενδεικτικοί σερβοκινητήρες.....	69
Εικόνα 5- 19: KY-040 Rotary Encoder.....	69
Εικόνα 6- 1: Η γλώσσα προγραμματισμού python.....	72
Εικόνα 6- 2: Το λογότυπο της MySQL.....	72
Εικόνα 6- 3: Το αρχικό μενού της εφαρμογής.....	76
Εικόνα 6- 4: Το παράθυρο διαχείρισης χρηστών.....	79
Εικόνα 6- 5: Μηνύματα λάθους κατά την εισαγωγή χρήστη.....	79
Εικόνα 6- 6: Το διάγραμμα ροής για την εισαγωγή χρήστη.....	81
Εικόνα 6- 7: Στιγμιότυπο από την εισαγωγή χρήστη.....	82
Εικόνα 6- 8: Μηνύματα λάθους κατά την ενημέρωση χρήστη.....	82
Εικόνα 6- 9: Διάγραμμα ροής για την ενημέρωση στοιχείων χρήστη.....	84
Εικόνα 6- 10: Τα μηνύματα λάθους της διαδικασίας διαγραφής.....	85
Εικόνα 6- 11: Το διάγραμμα ροής για την διαγραφή δεδομένων.....	87
Εικόνα 6- 12: Εμφάνιση φωτογραφιών κατά την διαδικασία εκπαίδευσης.....	88
Εικόνα 6- 13: Το διάγραμμα ροής της διαδικασίας εκπαίδευσης.....	90
Εικόνα 6- 14: Μηνύματα λάθους κατά την ενεργοποίηση της κάμερας.....	90
Εικόνα 6- 15: Επιτυχής αναγνώριση προσώπου.....	94
Εικόνα 6- 16: Μη επιτυχής αναγνώριση προσώπων.....	95
Εικόνα 6- 17: Το διάγραμμα ροής για τον εντοπισμό προσώπου.....	98
Εικόνα 6- 18: Εντοπισμός άνω μέρους σώματος.....	99
Εικόνα 6- 19: Διάγραμμα ροής για τον εντοπισμό άνω μέρους σώματος.....	102
Εικόνα 6- 20: Διάγραμμα ροής για τον εντοπισμό πλήρους σώματος.....	106
Εικόνα 6- 21: Εντοπισμός κίνησης.....	107
Εικόνα 6- 22: Διάγραμμα ροής για τον εντοπισμό κίνησης.....	110
Εικόνα 6- 23: Το παράθυρο των ρυθμίσεων.....	114
Εικόνα 6- 24: Εισαγωγή δεδομένων.....	118
Εικόνα 6- 25: Εμφάνιση απλού πλέγματος.....	119
Εικόνα 6- 26: Οπτική αναπαράσταση πλέγματος με πρόσωπο.....	121
Εικόνα 6- 27: Εντοπισμός προσώπων για διαφορετικές τιμές παραμέτρων.....	121
Εικόνα 6- 28: Διάγραμμα ροής παραμετροποίησης με πρόσωπα.....	124
Εικόνα 6- 29: Αναπαράσταση εντοπισμού άνω μέρους σώματος.....	125
Εικόνα 6- 30: Αναπαράσταση άνω μέρους σώματος για διαφορετικές τιμές παραμέτρων.....	125
Εικόνα 6- 31: Παραμετροποίηση άνω μέρους σώματος.....	128



Εικόνα 6- 32: Εντοπισμός πλήρους σώματος .....	129
Εικόνα 6- 33: Παραδείγματα με διαφορετικές τιμές παραμέτρων.....	129
Εικόνα 6- 34: Το διάγραμμα ροής για τον εντοπισμό πλήρους σώματος.....	132
Εικόνα 6- 35: Εντοπισμός κίνησης .....	133
Εικόνα 6- 36: Παραδείγματα με διαφορετικές τιμές παραμέτρων.....	133
Εικόνα 6- 37: Διάγραμμα ροής για την παραμετροποίηση με εντοπισμό κίνησης.....	136
Εικόνα 6- 38: Παράδειγμα με εντοπισμό αντικειμένων .....	137
Εικόνα 6- 39: Παραδείγματα με διαφορετικές τιμές παραμέτρων.....	137
Εικόνα 6- 40: Παραδείγματα κλάσεων του συνόλου δεδομένων .....	140
Εικόνα 6- 41: Στιγμιότυπα από τα 2 βίντεο .....	141
Εικόνα 7- 1: Το πρόβλημα εμφάνισης των βίντεο μέσα από την απομακρυσμένη σύνδεση .....	148
Εικόνα 7- 2: Προβολή του βίντεο στον τοπικό υπολογιστή .....	149

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1- 1: Πίνακας απαιτήσεων.....	25
Πίνακας 1- 2: Πίνακας ανάλυσης απαιτήσεων.....	26
Πίνακας 1- 3: Πίνακας πρότασης υλοποίησης.....	27
Πίνακας 1- 4: Πίνακας ανάλυσης λειτουργιών.....	27
Πίνακας 2- 1: Περιοχές ενδιαφέροντος και επιθυμητές λειτουργίες.....	29
Πίνακας 3- 1: Οι νέες έννοιες των Viola & Jones.....	37
Πίνακας 3- 2: Τα χαρακτηριστικά των Viola & Jones.....	41
Πίνακας 5- 1: Τα βασικά στοιχεία του Jetson Nano.....	55
Πίνακας 5- 2: Τα τεχνικά χαρακτηριστικά του Jetson Nano.....	56
Πίνακας 5- 3: Τα τεχνικά χαρακτηριστικά του Raspberry Pi.....	58
Πίνακας 5- 4: Σύγκριση των μικροϋπολογιστών.....	59
Πίνακας 5- 5: Εγκατάσταση των πακέτων.....	64
Πίνακας 5- 6: Τεχνικά χαρακτηριστικά της κάμερας.....	68
Πίνακας 6- 1: Περιγραφή του πίνακα User.....	73
Πίνακας 6- 2: περιγραφή του πίνακα.....	74
Πίνακας 6- 3: Σύνοψη σειριακής επικοινωνίας.....	113
Πίνακας Π-2- 1: Εντολή cv2.CascadeClassifier.....	157
Πίνακας Π-2- 2: Εντολή cv2.VideoCapture.....	157
Πίνακας Π-2- 3: Η εντολή cv2.cvtColor.....	157
Πίνακας Π-2- 4: Εντολή detectMultiScale.....	158
Πίνακας Π-2- 5: Εντολή cv2.rectangle.....	158
Πίνακας Π-2- 6: Εντολή cv2.imwrite.....	158
Πίνακας Π-2- 7: Η εντολή cv2.absdiff.....	158
Πίνακας Π-2- 8: Η εντολή GaussianBlur.....	159
Πίνακας Π-2- 9: cv2.face.LBPHFaceRecognizer_create.....	159
Πίνακας Π-2- 10: Η εντολή faceRecognizer.train.....	159
Πίνακας Π-2- 11: Η εντολή faceRecognizer.save.....	159
Πίνακας Π-2- 12: Η εντολή faceRecognizer.predict.....	160
Πίνακας Π-2- 13: Η εντολή cv2.dilate.....	160
Πίνακας Π-2- 14: Η εντολή drawContours.....	160
Πίνακας Π-2- 15: Η εντολή cv2.findContours.....	161
Πίνακας Π-2- 16: Η εντολή cv2.threshold.....	162

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

### ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

LBPH	Local Binary Pattern Histogram	Τοπικό Δυαδικό πρότυπο
RAM	Random Access Memory	Μνήμη τυχαίας προσπέλασης
GSM	Global System for Mobile Communication	Παγκόσμιο σύστημα κινητής επικοινωνίας
SMS	Short Message Service	Υπηρεσία μικρών μηνυμάτων
e-mail	Electronic mail	Ηλεκτρονικό ταχυδρομείο
UPS	Uninterruptible Power Supply	Αδιάκοπη παροχή ισχύος

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## ΕΙΣΑΓΩΓΗ

Το ερευνητικό περιεχόμενο της παρούσας εργασίας εντάσσεται στον κλάδο της υπολογιστικής όρασης. Το πεδίο αυτό εξελίσσεται ταχύτατα από το 1980 έως και σήμερα με ποικίλες εφαρμογές στην καθημερινότητα μας.

Η υπολογιστική όραση είναι ένας κλάδος της τεχνικής νοημοσύνης που προσπαθεί να αναπαράγει την αίσθηση της όρασης μέσω αλγορίθμων. Ο βασικός αισθητήρας είναι μια κάμερα με την οποία συλλέγονται τα δεδομένα. Μέτα την συλλογή, το σύστημα κατεργάζεται και αξιολογεί τα οπτικά δεδομένα μέσω των κατάλληλων τεχνικών και στο τέλος εκτελεί τις κατάλληλες πράξεις.

Μέχρι σήμερα έχουν πραγματοποιηθεί πάρα πολλές έρευνες και έχουν αναπτυχθεί αρκετές εφαρμογές όπως η ανίχνευση/αναγνώριση προσώπων, η χαρτογράφηση χώρων και ο εντοπισμός πεζών ή εμποδίων στα αυτόνομα οχήματα.

Στην παρούσα εργασία παρουσιάζονται τρεις εφαρμογές της υπολογιστικής όρασης: ο εντοπισμός και αναγνώριση προσώπων και ο εντοπισμός κινήσεων. Η συλλογή των δεδομένων γίνεται μέσω μιας κάμερας και η επεξεργασία τους γίνεται μέσω ενός μικροϋπολογιστή.

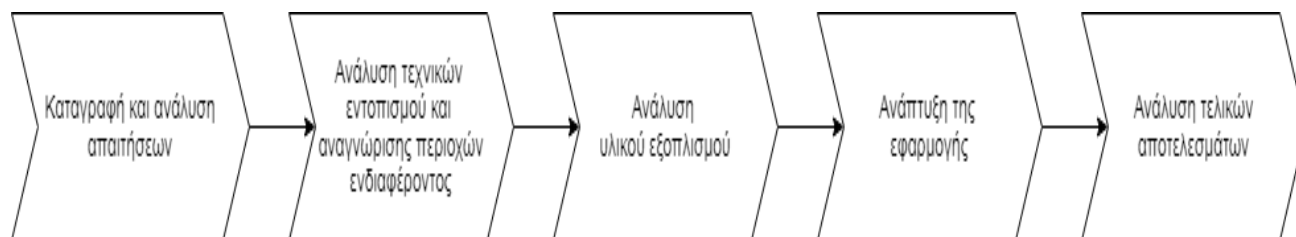
Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



## ΚΕΦΑΛΑΙΟ 1: ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

### 1.1: Εισαγωγή

Ο αρχικός στόχος της εργασίας ήταν η ανάπτυξη ενός συστήματος εντοπισμού και αναγνώρισης προσώπων, σώματα από το κεφάλι μέχρι τον θώρακα πλήρη σώματα και κινήσεων μέσω μιας κάμερας συνδεδεμένης με έναν μικροϋπολογιστή. Στην πορεία η εργασία επεκτάθηκε με την τοποθέτηση της κάμερας σε έναν υπάρχον περιστροφικό βραχίονα ο οποίος ελέγχεται από διαφορετικό μικροελεγκτή και την προσθήκη σειριακής επικοινωνίας με σκοπό την κίνηση του βραχίονα ώστε να συνεχιστεί η καταγραφή της περιοχής ενδιαφέροντος. Ο βραχίονας είναι κατασκευασμένος σε τρισδιάστατο εκτυπωτή, η κίνηση γίνεται μέσω σερβοκινητήρων. Ο μικροελεγκτής καθώς και ο μικροϋπολογιστής είναι open Source καθώς και οι βιβλιοθήκες και τα απαραίτητα προγράμματα. Πριν ξεκινήσει η υλοποίηση της κατασκευής πρέπει να καταγραφούν και να αναλυθούν οι απαιτήσεις.



Εικόνα 1- 1: Τα βήματα ανάπτυξης της εργασίας

### 1.2: Ανάλυση απαιτήσεων

Για την επίτευξη των στόχων της εργασίας και την ορθή λειτουργία ολόκληρου του συστήματος είναι απαραίτητο να καταγραφούν και να αναλυθούν οι ανάγκες και οι απαιτήσεις. Στον παρακάτω πίνακα φαίνονται οι απαιτήσεις που πρέπει να ικανοποιήσει το σύστημα.

Απαιτήσεις	Διαχείριση χρηστών
	Εντοπισμός χαρακτηριστικών και αναγνώριση προσώπων
	Καταγραφή συμβάντων
	Παραμετροποίηση παραθύρου
	Διαχείριση λειτουργιών εφαρμογής

Πίνακας 1- 1: Πίνακας απαιτήσεων

Μετά την καταγραφή των απαιτήσεων πρέπει να γίνει ανάλυση. Στον πίνακα 1.2 αναλύονται οι απαιτήσεις που αναφέρθηκαν. Το βήμα της ανάλυσης είναι πολύ σημαντικό καθώς με βάση την ανάλυση θα γίνει η προσέγγιση και τελική επιλογής της τεχνικής υλοποίησης

Διαχείριση χρηστών	Οι λειτουργίες που πρέπει γίνονται στους χρήστες είναι οι εξής <ol style="list-style-type: none"> <li>1) Δημιουργία χρήστη</li> <li>2) Ενημέρωση στοιχείων χρήστη</li> <li>3) Διαγραφή χρήστη</li> </ol> <p>Το σύστημα πρέπει να δέχεται τα στοιχεία του έμπιστου ατόμου και να καταγράφει φωτογραφίες του, μέσω των οποίων θα εκπαιδευτεί το σύστημα</p>
Εντοπισμός χαρακτηριστικών και αναγνώριση προσώπων	Μέσω απλών και αποτελεσματικών τεχνικών το σύστημα θα εντοπίζει τις απαραίτητες περιοχές ενδιαφέροντος <ol style="list-style-type: none"> <li>1) Πρόσωπα(και ιδιαίτερα τα άγνωστα)</li> <li>2) Άνω μέρη σώματος</li> <li>3) Πλήρες σώμα</li> <li>4) Κινήσεις</li> </ol>
Καταγραφή συμβάντων	Μέσω της καταγραφής συμβάντων οι χρήστες μπορούν να δουν σε βίντεο αν έχει γίνει κάτι στον χώρο παρακολούθησης. Τα βίντεο αφορούν εντοπισμούς των περιοχών ενδιαφέροντος
Παραμετροποίηση παραθύρου	Μέσω της παραμετροποίησης είναι δυνατή η αλλαγή του κέντρου του παραθύρου, παρέχοντας μεγαλύτερη ευελιξία ως προς τους εντοπισμούς. Το παράθυρο της κάμερας χωρίζεται σε τμήματα μεγέθους 3x3.
Διαχείριση λειτουργιών εφαρμογής	Για την διαχείριση των λειτουργιών που αναφέρθηκαν χρειάζεται μια εφαρμογή με κατανοητές οδηγίες για την χρήση της εφαρμογής

**Πίνακας 1- 2: Πίνακας ανάλυσης απαιτήσεων**

Με βάση την ανάλυση που έχει γίνει στις απαιτήσεις επιλέγεται η τεχνική υλοποίησης

Διαχείριση χρηστών	Κατασκευή ενός απλού παραθύρου με δύο πεδία προς συμπλήρωση. Πέρα από τα πεδία υπάρχουν και τρία πλήκτρα για την απαραίτητη λειτουργία. Τα στοιχεία του χρήστη αποθηκεύονται σε βάση δεδομένων και οι φωτογραφίες σε φάκελο.
Εντοπισμός χαρακτηριστικών - Αναγνώριση προσώπων	Για τον εντοπισμό των ανθρώπινων χαρακτηριστικών επιλέχθηκε η μέθοδος Viola & Jones με τους παρακάτω ταξινομητές <ol style="list-style-type: none"> <li>1) Haarcascade_frontalface_default</li> <li>2) Haarcascade_upperbody</li> <li>3) Haarcascade_fullbody</li> </ol> <p>Για την αναγνώριση προσώπων εφαρμόστηκε η τεχνική LBPH και για τον εντοπισμό κινήσεων η τεχνική αφαίρεσης φόντου</p>

Καταγραφή συμβάντων	Σε περίπτωση εντοπισμού συμβάντος, μέσω της εφαρμογής γίνεται καταγραφή σε βίντεο από την έναρξη του συμβάντος μέχρι και ένα σύντομο χρονικό διάστημα μετά την λήξη του
Παραμετροποίηση παραθύρου	Καταγραφή των ελάχιστων και μέγιστων τιμών στους άξονες X και Y μετά από έλεγχο εγκυρότητας δεδομένων. Κατά την λειτουργία της κάμερας μόλις εντοπιστεί περιοχή ενδιαφέροντος ο βραχίονας κινείται ώστε το κέντρο του αντικειμένου να βρεθεί στο κεντρικό υποπλαίσιο
Διαχείριση λειτουργιών εφαρμογής	Ανάπτυξη εφαρμογής για τον χρήστη με γραφικό περιβάλλον και κατανοητές οδηγίες σχετικά με την συμπλήρωση των πεδίων

**Πίνακας 1- 3: Πίνακας πρότασης υλοποίησης**

### 1.3: Ανάλυση λειτουργιών

Για την επίτευξη το στόχου και του επιθυμητού αποτελέσματος το σύστημα εκτελεί μια σειρά από λειτουργίες. Τα επιμέρους μέρη του συστήματος είναι τα εξής

- Ο μικροϋπολογιστής
- Ο μικροελεγκτής (δεν εμπίπτει στα πλαίσια της συγκεκριμένης εργασίας)
- Ο ρομποτικός βραχίονας (δεν εμπίπτει στα πλαίσια της συγκεκριμένης εργασίας)

Στον παρακάτω πίνακα αναλύονται οι λειτουργίες κάθε μέρους του συστήματος

<b>Μικροϋπολογιστής</b>	<b>Μικροελεγκτής</b>	<b>Βραχίονας</b>
Αποθήκευση στοιχείων	Παρακολούθηση μέσω αισθητήρων κίνησης	Επιτήρηση χώρου μέσω κάμερας
Διαχείριση χρηστών	Έλεγχος των σερβοκινητήρων	Κίνηση του βραχίονα
Εκπαίδευση με έμπιστους χρήστες	Ενεργοποίηση των LED ανάλογα με συνθήκες φωτισμού	Καταγραφή της θέσης του βραχίονα
Ενεργοποίηση της κάμερας και επικοινωνία με τον μικροελεγκτή για την κίνηση του βραχίονα		
Καταγραφή και αποθήκευση βίντεο		

**Πίνακας 1- 4: Πίνακας ανάλυσης λειτουργιών**

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## ΚΕΦΑΛΑΙΟ 2: Εντοπισμός και αναγνώριση προσώπων, εισαγωγή και βασικές έννοιες

### 2.1: Ο εντοπισμός του προσώπου

#### 2.1.1: Εισαγωγή

Όπως αναφέρθηκε στο πρώτο κεφάλαιο η εφαρμογή που αναπτύχθηκε πρέπει να εντοπίζει τις παρακάτω περιοχές ενδιαφέροντος

Είδος περιοχής ενδιαφέροντος	Λειτουργία
Πρόσωπα	Εντοπισμός και αναγνώριση
Ανω μέρος σώματος	Εντοπισμός
Πλήρες σώμα	Εντοπισμός
Κινήσεις	Εντοπισμός

Πίνακας 2- 1: Περιοχές ενδιαφέροντος και επιθυμητές λειτουργίες

Από τις τέσσερις αυτές περιοχές ενδιαφέροντος η πιο κρίσιμη είναι τα πρόσωπα καθώς η απόδοση του συστήματος εξαρτάται απόλυτα από την ορθή αναγνώριση (και κατά συνέπεια εντοπισμό) του προσώπου.

Ο εντοπισμός προσώπου παρουσιάζει αρκετές προκλήσεις με τις πιο βασικές από αυτές να είναι οι συνθήκες που επικρατούν στο πεδίο λήψης καθώς και η ποιότητα/ανάλυση της κάμερας. Για να μπορέσει η κάμερα να αναγνωρίσει τα πρόσωπα θα πρέπει αρχικά να μπορεί να τα εντοπίσει σε ένα μεγάλο εύρος καταστάσεων στον χώρο όπως είναι ο φωτισμός και το μέγεθος.

Καθώς υπάρχουν διαφορετικές τεχνικές που εντοπίζουν πρόσωπα, πρέπει να γίνει μια ανάλυση των τεχνικών αυτών, να αξιολογηθούν τα πλεονεκτήματα και μειονεκτήματα της κάθε μιας και με βάση αυτά να γίνει επιλογή της κατάλληλης μεθόδου.

#### 2.1.2: Βασικές αρχές εντοπισμού προσώπου

Για τον εντοπισμό προσώπου σε μια εικόνα στην πλειονότητα των μεθόδων χρειάζονται μια μορφή εκπαίδευσης. Η πιο συνηθισμένη μορφή είναι η παροχή εικόνων εκ των οποίων ορισμένες περιέχουν πρόσωπα. Επίσης οι εικόνες διαφέρουν ως προς την παρουσίαση των προσώπων μεταβάλλοντας την κλίση και τον φωτισμό. Με βάση αυτές τις εικόνες ο αλγόριθμος αξιολογεί την είσοδο και αποφασίζει αν υπάρχει πρόσωπο ή όχι. Ο έλεγχος γίνεται μέσω ενός ανιχνευτή ο οποίος περνάει την εικόνα που δόθηκε και συμπεραίνει αν υπάρχει πρόσωπο.

#### 2.1.3: Ανάλυση και σύγκριση τεχνικών εντοπισμού προσώπου

Οι τεχνικές που αξιολογήθηκαν για τον εντοπισμό προσώπου είναι οι εξής:

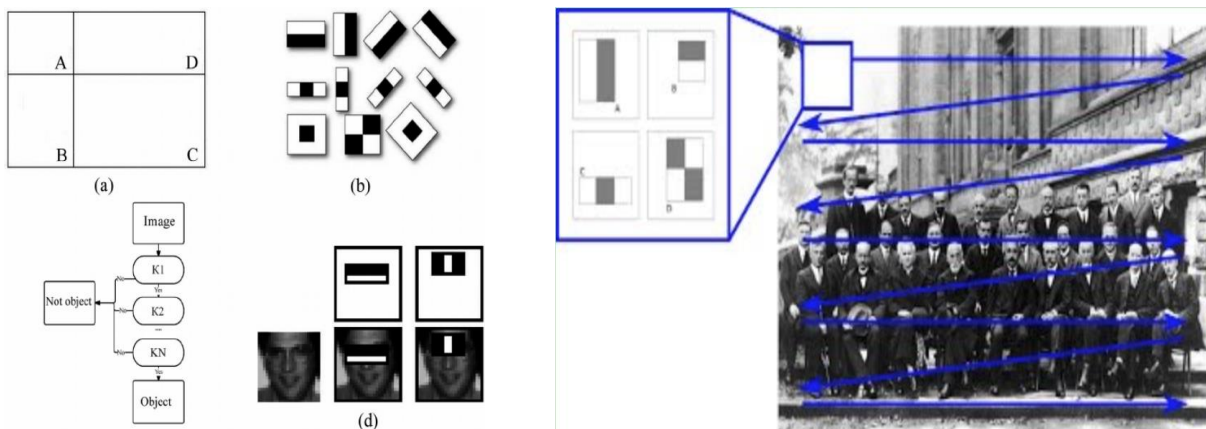
- Viola & Jones
- Histogram of Oriented Gradients (HOG)

### 2.1.3.1: Τεχνική Viola & Jones:

Η τεχνική Viola & Jones είναι τεχνική που βασίζεται στην εμφάνιση. Όσο αφορά τους εντοπισμούς χρειάζεται εκπαίδευση είτε χρήση έτοιμων αρχείων. Κατά την εκτέλεση εφαρμόζει ένα ορθογώνιο περίγραμμα το οποίο ολισθαίνει στην εικόνα για να βρει τα πρόσωπα. Ο εντοπισμός γίνεται μέσω Haar ταξινομητών και η εκπαίδευση μέσω adaBoost.[1,2,3].

Στα πλεονεκτήματα της τεχνικής αυτής είναι η ταχύτητα στους εντοπισμούς ακόμα και σε βίντεο πραγματικού χρόνου. Επίσης η υλοποίηση του αλγόριθμου είναι απλή με βιβλιοθήκες υπολογιστικής όρασης.

Τα μειονεκτήματα είναι ότι χρειάζονται αρκετά δείγματα και πολύ χρόνο για να εκπαιδευτεί. Επιπλέον είναι αποτελεσματική όταν το πρόσωπο είναι ανφάς



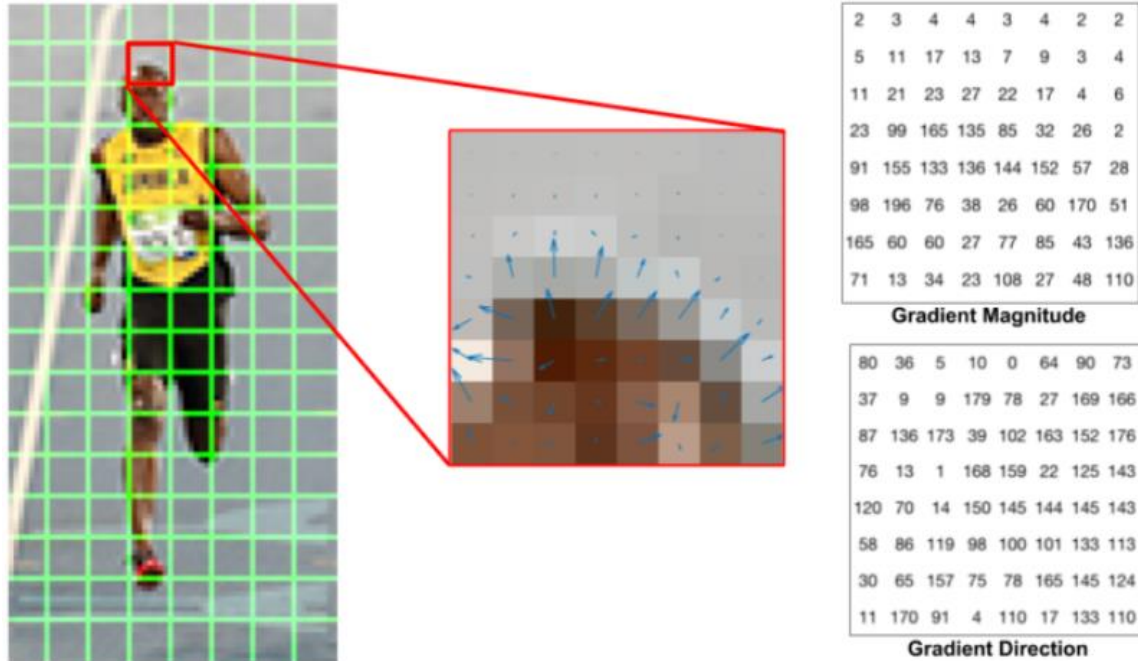
Εικόνα 2- 1: Μέθοδος Viola & Jones

### 2.1.3.2: Τεχνική ιστογραμμάτων κλίσεων :

Στην τεχνική των ιστογραμμάτων κλίσεων (Histogram of oriented Gradients) περιγράφονται τα χαρακτηριστικά του προσώπου (feature description). Με βάση το πλάτος και τις κλίσεις των χαρακτηριστικών δημιουργούνται ιστογράμματα.

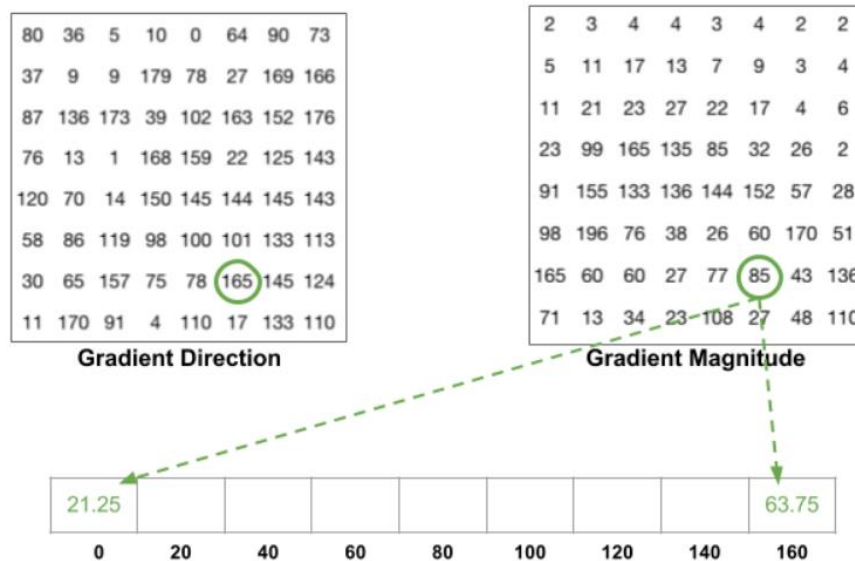
Στα εικονοστοιχεία της εικόνας υπολογίζεται η διαφορά στις φωτεινότητες στον οριζόντιο και κατακόρυφο άξονα. Για κάθε στοιχείο υπολογίζεται στον άξονα X η απόλυτη τιμή της διαφοράς ανάμεσα στο αριστερό και δεξιό στοιχείο και στον άξονα Y το από πάνω με το από κάτω. Εφαρμόζοντας τον τύπο της ευκλείδειας απόστασης υπολογίζεται το πλάτος για το συγκεκριμένο κελί. Για τον υπολογισμό της γωνίας υπολογίζεται το τόξο εφαπτομένης της διαίρεσης των διαφορών Y και X. Με αυτόν τον τρόπο σχηματίζονται δύο πίνακες με πληροφορίες για τα πλάτη και τις γωνίες.

Το ιστόγραμμα κλίσεων χωρίζεται σε 9 διακριτές περιοχές που αναπαριστούν τις κλίσεις. Κάθε περιοχή αναπαριστά γωνίες από 0-160 μοίρες. Στο παράδειγμα της παρακάτω εικόνας διακρίνονται οι κλίσεις και τα πλάτη των εικονοστοιχείων.[5]



Εικόνα 2- 2: Αναπαράσταση κλίσεων και πλατών (το μήκος του βέλους αναπαριστά το πλάτος)

Για την συμπλήρωση του ιστογράμματος εφαρμόζεται ο εξής κανόνας. Η κατάλληλη από τις 9 περιοχές του ιστογράμματος καθορίζεται με βάση την κατεύθυνση και η τιμή με βάση το πλάτος. Αν ένα στοιχείο δεν πέφτει ακριβώς σε περιοχή τότε υπολογίζεται το ποσοστό που αναλογεί σε κάθε περιοχή. Έτσι δημιουργείται το ιστόγραμμα με τις κατανομές των κλίσεων.[5,6]



Εικόνα 2- 3: Δημιουργία Ιστογράμματος Κλίσεων





**Εικόνα 2- 4: Αναπαράσταση προσώπου μέσω HOG**

Πλεονεκτήματα είναι η σχετικά χαμηλή υπολογιστική ισχύς ειδικά σε εικόνες μικρής κλίμακας. Σημαντικά μειονεκτήματα είναι η ταχύτητα σε εικόνες μεγάλης κλίμακας καθώς βασίζεται σε κυλιόμενα παράθυρα για την εξαγωγή πληροφοριών από τα εικονοστοιχεία

#### **2.1.4: Επιλογή κατάλληλης μεθόδου:**

Η επιλογή της κατάλληλης τεχνικής για τον εντοπισμό προσώπου είναι ένα πολυδιάστατο πρόβλημα. Οι κύριοι παράγοντες επιλογής είναι η ακρίβεια και το είδος του εντοπισμού καθώς και η υπολογιστική ισχύς που είναι διαθέσιμη

Με βάση τα παραπάνω επιλέχθηκε η μέθοδος των Viola & Jones για τρεις λόγους. Αρχικά είναι αρκετά απλή και αξιόπιστη μέθοδος. Δεύτερον υπάρχουν έτοιμα αρχεία εκπαίδευσης για τους εντοπισμούς των περιοχών ενδιαφέροντος. Τέλος δεν περιορίζεται μόνο σε πρόσωπα αλλά μπορεί να αξιοποιηθεί και για άλλα είδη εντοπισμών και στην συγκεκριμένη εργασία έχει αξιοποιηθεί επιπλέον για τον εντοπισμό άνω μέρους και πλήρους σώματος.

## **2.2: Η αναγνώριση του προσώπου**

### **2.2.1: Εισαγωγή**

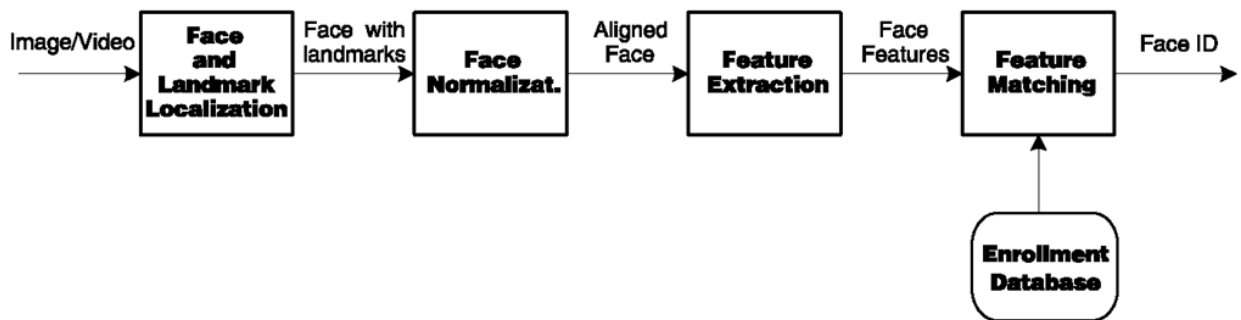
Επίσης κρίσιμο κομμάτι τη εφαρμογής είναι η επιτυχής αναγνώριση προσώπων. Στην εφαρμογή είναι αποθηκευμένες πληροφορίες για τα έμπιστα άτομα. Όταν εμφανίζεται ένα πρόσωπο στο προς αναγνώριση, τότε συγκρίνονται οι πληροφορίες και το αποτέλεσμα καθορίζει αν η αναγνώριση ήταν επιτυχής.

Όπως και με τον εντοπισμό προσώπων έτσι και με την αναγνώριση προσώπων υπάρχουν διαφορές τεχνικές που πρέπει να αξιολογηθούν και να επιλεγεί η καλύτερη με βάση της συνθήκες



### 2.2.2: Βασικές αρχές αναγνώρισης προσώπων

Η αναγνώριση προσώπων αναπτύσσεται σε τέσσερα στάδια. Το πρώτο στάδιο δεν αφορά την αναγνώριση του προσώπου αλλά τον εντοπισμό του. Μόλις εντοπιστεί το πρόσωπο αρχίζει η διαδικασία αναγνώρισης με την εύρεση των στοιχείων. Το επόμενο βήμα αφορά την κανονικοποίηση του προσώπου και των στοιχείων που έχουν βρεθεί (κυρίως ως προς τις συνθήκες φωτισμού και μεγέθους). Στην συνέχεια ακολουθεί η εξαγωγή των χαρακτηριστικών με βάση τα οποία θα γίνει η αναγνώριση. Στο τέλος είναι η αντιστοίχιση/σύγκριση των προσώπων. Εδώ, όπως γίνεται κατανοητό και από το όνομα συγκρίνονται τα στοιχεία της εισόδου με αυτά που είναι αποθηκευμένα.[6,7]



Εικόνα 2- 5: Τα στάδια αναγνώρισης προσώπων

### 2.2.3: Ανάλυση και σύγκριση τεχνικών αναγνώρισης προσώπων

Για τους σκοπούς της αναγνώρισης προσώπων αξιολογήθηκαν οι παρακάτω τεχνικές

- Ανάλυση κύριων συνιστωσών.
- Μέθοδος των τοπικών δυαδικών προτύπων.

#### 2.2.3.1: Η τεχνική ανάλυσης κύριων συνιστωσών

Η τεχνική αυτή βασίζεται σε στατιστική για να αναγνωρίσει τα πρόσωπα. Το πρώτο βήμα είναι η εισαγωγή των έμπιστων προσώπων. Για να είναι πιο αποδοτικό το σύστημα χρειάζονται περισσότερες από μία εικόνες για κάθε πρόσωπο που θα αναγνωριστεί.

Για κάθε πρόσωπο πρέπει να δημιουργηθεί ένα διάνυσμα. Αυτό σχηματίζεται με την συνένωση των γραμμών της εικόνας σε μια σειρά από εικονοστοιχεία (σε grayscale κλίμακα).



Εικόνα 2- 6: Δημιουργία διανυσμάτων από τις εικόνες

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Αυτό γίνεται για όλες τις εικόνες που υπάρχουν στο σύνολο δεδομένων. Με αυτόν τον τρόπο δημιουργείται ένας πίνακας (matrix)  $M$  μεγέθους  $N \times K^2$  (όπου  $N$  ο αριθμός των εικόνων στο σύνολο δεδομένων).

Στην συνέχεια εφαρμόζεται η ανάλυση των κύριων συνιστωσών με βάση τον μέσο όρο κάθε κολώνας του πίνακα και την διαφορά ανάμεσα σε κάθε στοιχείο της κολώνας και τον μέσο όρο, υπολογίζεται ένας πίνακας με διακυμάνσεις. Με τον υπολογισμό των ιδιοτιμών και των ιδιοδιανυσμάτων του πίνακα διακυμάνσεων βρίσκονται τα δεδομένα με τις μεγαλύτερες διακυμάνσεις. Τα ιδιοδιανύσματα με τις μεγαλύτερες ιδιοτιμές λέγονται EigenFaces.

Μετά την εφαρμογή του EigenFace στον πίνακα  $M$  δημιουργείται ένας πίνακας (έστω  $Z$ ) που περιέχει  $N \times K^2$  εγγραφές. Αυτές οι εγγραφές μπορούν να ξαναμετατραπούν σε εικόνες  $K \times K$ .

Με βάση τα παραπάνω στοιχεία μπορεί να εξαχθεί μια «μέση» εικόνα για όλα τα πρόσωπα που υπάρχουν στο σύνολο δεδομένων και εικόνες που δείχνουν την απόκλιση κάθε πρόσωπου από την μέση εικόνα. Όσο πιο φωτεινές είναι οι περιοχές στα δεξιά πρόσωπα τόσο μεγαλύτερες οι διακυμάνσεις



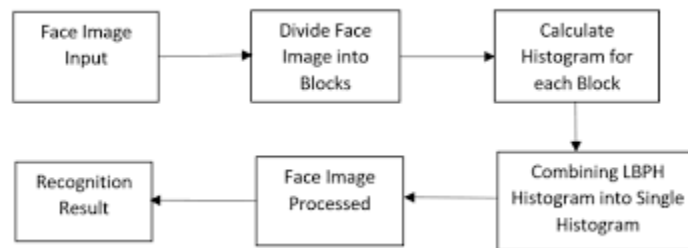
**Εικόνα 2- 7: Μέση εικόνα και EigenFace εικόνες**

Όταν δοθεί ένα πρόσωπο προς αναγνώριση υπολογίζεται το βαθμωτό γινόμενο ανάμεσα στο πρόσωπο προς αναγνώριση και στα  $N$  Eigenfaces. Έτσι το πρόσωπο μπορεί να μετατραπεί σε συνδυασμό κύριων συνιστωσών. Για την αναγνώριση προσώπων μπορεί να εφαρμοστεί η ευκλείδεια απόσταση ανάμεσα στα EigenFaces. Όσο πιο μικρή είναι η τιμή τόσο περισσότερο ταιριάζουν τα πρόσωπα.

Το πλεονεκτήματα αυτής της τεχνικής είναι η απλότητα. Τα μειονεκτήματα είναι ότι επηρεάζεται πολύ εύκολα από μεταβολές ειδικά από τον φωτισμό

### 2.2.3.2: Η τεχνική των τοπικών δυαδικών προτύπων

Μια διαφορετική προσέγγιση ακολουθείται με την μέθοδο των τοπικών δυαδικών προτύπων (Local binary Pattern histogram). Στην συγκεκριμένη τεχνική σχεδιάζονται ιστογράμματα για τα έμπιστα πρόσωπα με την βοήθεια μιας ενδιάμεσης εικόνας. Για κάθε πρόσωπο προς αναγνώριση σχεδιάζονται και τα δικά του ιστογράμματα και συγκρίνονται και αν η διαφορά είναι μικρότερη από ένα κατώφλι τότε το πρόσωπο έχει αναγνωριστεί επιτυχώς



Εικόνα 2- 8: Κατασκευή ιστογραμμάτων για την μέθοδο LBPH

Τα πλεονεκτήματα της μεθόδου LBPH είναι ότι δεν επηρεάζονται από αλλαγές και διακυμάνσεις στον φωτισμό και παρέχουν αρκετές λεπτομέρειες. Τα μειονεκτήματα είναι τα μεγάλα ιστογράμματα που παράγονται και η απώλεια κάποιων πληροφοριών.

### 2.2.4: Επιλογή κατάλληλης τεχνικής

Όπως και με την διαδικασία επιλογής εντοπισμού προσώπου, έτσι και για την αναγνώριση προσώπων πρέπει να ληφθούν αρκετοί παράγοντες υπόψη. Ένας βασικός παράγοντας είναι η επιθυμητή ακρίβεια, καθώς και οι συνθήκες φωτισμού που υπάρχουν στο πεδίο λήψης της κάμερας.

Με βάση τα παραπάνω επιλέχθηκε η μέθοδος των τοπικών δυαδικών προτύπων (LBPH) καθώς είναι αρκετά αξιόπιστη σε σχέση με άλλες μεθόδους καθώς κάθε εικόνα που συλλέγεται για την εκπαίδευση του συστήματος επεξεργάζεται ανεξάρτητα από τις υπόλοιπες και όχι ως ένα ενιαίο σύνολο. Επίσης δεν επηρεάζεται τόσο πολύ από διακυμάνσεις στον φωτισμό και τέλος έχει αρκετά απλή υλοποίηση.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## Κεφάλαιο 3: Εντοπισμός προσώπων: Η μέθοδος Viola & Jones και η μέθοδος τοπικών δυαδικών προτύπων

### 3.1: Η μέθοδος Viola & Jones

#### 3.1.1: Εισαγωγή:

Η μέθοδος των Viola & Jones αναπτύχθηκε από τους Paul Viola και Michael Jones και παρουσιάστηκε στο άρθρο «Robust real-time face detection». Ήθελαν να δημιουργήσουν μια καινούργια διαδικασία η οποία θα είναι γρήγορη και αποδοτική

Παρόλο που σήμερα είναι λίγο ξεπερασμένη η συγκεκριμένη μέθοδος είναι αρκετά δυνατή και είναι πολύ καλή στην ανίχνευση προσώπων σε βίντεο πραγματικού χρόνου.

Η συγκεκριμένη μέθοδος βασίζεται σε τρεις έννοιες. Οι έννοιες αυτές κάνουν την τεχνική αυτή πιο αποδοτική. Η συγκεκριμένη μέθοδος εφαρμόζεται σε μονοχρωματικές εικόνες.

Έννοια	Περιγραφή
Ολοκληρωτική/ολοκληρωμένη εικόνα	Διαφορετικός τρόπος περιγραφής των εικόνων/προσώπων. Μέσω αυτής της εικόνας αυξάνεται η ταχύτητα των υπολογισμών
Ταξινομητής	Εκπαιδεύεται μέσω adaBoost, και επιλέγει τα βασικότερα χαρακτηριστικά.
Καταρράκτης ταξινομητών	Συνδέονται οι ταξινομητές μεταξύ τους. Με αυτόν τον τρόπο αυξάνεται η απόδοση καθώς ο ταξινομητής απορρίπτει πιο γρήγορα τις περιοχές που δεν υπάρχουν πρόσωπα

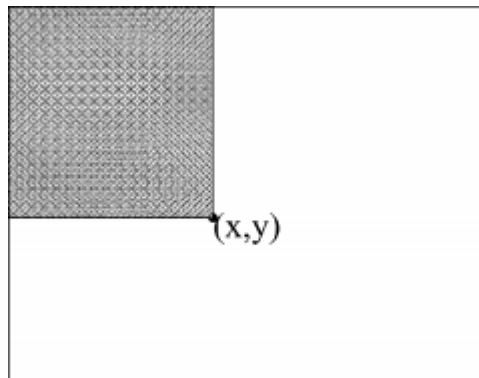
Πίνακας 3- 1: Οι νέες έννοιες των Viola & Jones

Η μέθοδος των Viola and Jones εκτελεί τα παρακάτω τέσσερα βήματα. Αρχικά υπολογίζεται η ολοκληρωτική εικόνα (Integral image), στην συνέχεια υπολογίζονται τα χαρακτηριστικά με βάση την εικόνα που έχει δημιουργηθεί. Με αυτά τα χαρακτηριστικά εκπαιδεύονται οι ταξινομητές μέσω AdaBoost και ταξινομούνται με έναν καταρράκτη ταξινομητών (Cascade Classifier)[3]

#### 3.1.2: Η δημιουργία και ανάλυση της ολοκληρωτικής εικόνας:

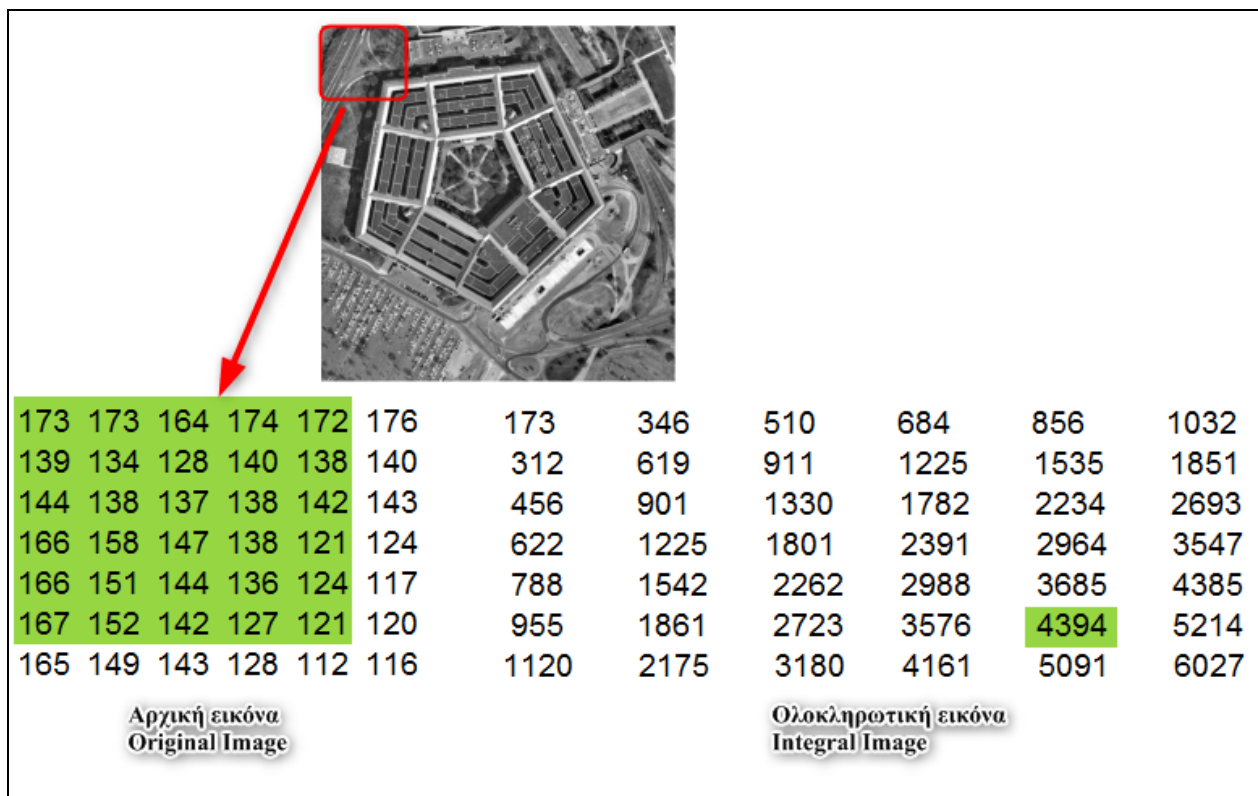
Η ολοκληρωτική εικόνα επιτρέπει τον γρήγορο υπολογισμό των στοιχείων του προσώπου. Στην εικόνα αυτή το εικονοστοιχείο σε κάθε σημείο ορίζεται ως το σύνολο των φωτεινοτήτων των εικονοστοιχείων από το άνω αριστερό άκρο έως το συγκεκριμένο σημείο.

Στην παρακάτω εικόνα διακρίνεται μια σχηματική αναπαράσταση της ολοκληρωτικής εικόνας. Η ένταση της φωτεινότητας στο σημείο  $(x,y)$  ισούται με το άθροισμα των φωτεινοτήτων που βρίσκονται στο σκούρο πλαίσιο.[3]



Εικόνα 3- 1: Σχηματική αναπαράσταση της ολοκληρωτικής εικόνας

Για παράδειγμα στην παρακάτω εικόνα διακρίνεται μια αναπαράσταση της ολοκληρωτικής εικόνας. Στον αριστερό πίνακα παρουσιάζεται η αρχική εικόνα και στα δεξιά η αντίστοιχη ολοκληρωτική εικόνα. Στο πράσινο κελί της ολοκληρωτικής εικόνας υπάρχει το άθροισμα των πράσινων κελιών της αρχικής εικόνας.



Εικόνα 3- 2: Αναπαράσταση προσώπων σε αρχική και ολοκληρωτική εικόνα

Το μεγάλο πλεονέκτημα της ολοκληρωτικής εικόνας και ο λόγος που επιλέχτηκε από τους Viola & Jones είναι η ταχύτητα στους υπολογισμούς. Για παράδειγμα το άθροισμα των φωτεινότητων ενός παραλληλογράμμου σε μια κανονική εικόνα γίνεται αθροίζοντας τα στοιχεία ένα-ένα. Σε μια ολοκληρωτική εικόνα υπολογίζεται με τέσσερις πράξεις όπως φαίνεται στο παρακάτω παράδειγμα.

Έστω ότι χρειάζεται να υπολογιστεί το άθροισμα των φωτεινότητων που βρίσκονται στο μέσα στο κόκκινο παραλληλόγραμμο της ολοκληρωτικής εικόνας.

173	346	510	684	856	1032
312	619	911	1225	1535	1851
456	901	1330	1782	2234	2693
622	1225	1801	2391	2964	3547
788	1542	2262	2988	3685	4385
955	1861	2723	3576	4394	5214
1120	2175	3180	4161	5091	6027

Εικόνα 3- 3: Υπολογισμός αθροίσματος

Αρχικά υπολογίζεται το άθροισμα στο κάτω δεξιά σημείο του παραλληλογράμμου (σημείο P). Αυτό το σημείο περιέχει όλες τις φωτεινότητες μέχρι εκεί (πράσινο πλαίσιο στην παρακάτω εικόνα).

173	346	510	684	856	1032	173	346	510	684	856	1032
312	619	911	1225	1535	1851	312	619	911	1225	1535	1851
456	901	1330	1782	2234	2693	456	901	1330	1782	2234	2693
622	1225	1801	2391	2964	3547	622	1225	1801	2391	2964	3547
788	1542	2262	2988	3685	4385	788	1542	2262	2988	3685	4385
955	1861	2723	3576	4394	5214	955	1861	2723	3576	4394	5214
1120	2175	3180	4161	5091	6027	1120	2175	3180	4161	5091	6027

Εικόνα 3- 4: Το πρώτο στάδιο του υπολογισμού

Στην συνέχεια αφαιρείται η φωτεινότητα του άνω δεξιά σημείου του παραλληλογράμμου (σημείο Q). Έτσι αφαιρούνται οι φωτεινότητες από το πάνω μέρος της εικόνας.

173	346	510	684	856	1032	173	346	510	684	856	1032
312	619	911	1225	1535	1851	312	619	911	1225	1535	1851
456	901	1330	1782	2234	2693	456	901	1330	1782	2234	2693
622	1225	1801	2391	2964	3547	622	1225	1801	2391	2964	3547
788	1542	2262	2988	3685	4385	788	1542	2262	2988	3685	4385
955	1861	2723	3576	4394	5214	955	1861	2723	3576	4394	5214
1120	2175	3180	4161	5091	6027	1120	2175	3180	4161	5091	6027

Εικόνα 3- 5: Το δεύτερο στάδιο υπολογισμού (Αφαίρεση περιττών φωτεινότητων)

Με την αφαίρεση της φωτεινότητας στο κάτω αριστερό σημείο (σημείο S) απαλείφεται το τελευταίο πράσινο πλαίσιο.

173	346	510	684	856	1032	173	346	510	684 <sup>Q</sup>	856	1032
312	619	911	1225	1535	1851	312	619	911	1225	1535	1851
456	901	1330	1782	2234	2693	456	901	1330	1782	2234	2693
622	1225	1801	2391	2964	3547	622	1225	1801	2391	2964	3547
788	1542	2262	2988	3685	4385	788 <sup>S</sup>	1542	2262	2988	3685	4385
955	1861	2723	3576	4394	5214	955	1861	2723	3576	4394	5214
1120	2175	3180	4161	5091	6027	1120	2175	3180	4161	5091	6027

Εικόνα 3- 6: Το τρίτο στάδιο υπολογισμού (αφαίρεση του τελευταίου μέρους)

Το τελευταίο βήμα είναι η άθροιση της φωτεινότητας στο άνω αριστερό σημείο του παραλληλογράμμου υπολογισμού (σημείο R) καθώς η περιοχή με κόκκινο στην εικόνα 3.6 έχει αφαιρεθεί δύο φορές, μια από το σημείο Q και μια από το σημείο S

173	346	510	684	856	1032	173	346	510 <sup>R</sup>	684 <sup>Q</sup>	856	1032
312	619	911	1225	1535	1851	312	619	911	1225	1535	1851
456	901	1330	1782	2234	2693	456	901	1330	1782	2234	2693
622	1225	1801	2391	2964	3547	622	1225	1801	2391	2964	3547
788	1542	2262	2988	3685	4385	788 <sup>S</sup>	1542	2262	2988	3685	4385
955	1861	2723	3576	4394	5214	955	1861	2723	3576	4394	5214
1120	2175	3180	4161	5091	6027	1120	2175	3180	4161	5091	6027

Εικόνα 3- 7: Άθροιση της περιοχής που έχει αφαιρεθεί δύο φορές

Ο τελικός τύπος για το άθροισμα των φωτεινότητων ενός παραλληλογράμμου σε μια ολοκληρωτική εικόνα είναι ο εξής

$$Sum = I_P - I_Q - I_S + I_R$$

Αυτό που αξίζει να τονιστεί είναι ότι αυτός ο τύπος μπορεί να εφαρμοστεί σε μια ολοκληρωτική εικόνα ανεξαρτήτως του μεγέθους του παραλληλογράμμου και είναι το πιο σημαντικό στοιχείο της ολοκληρωτικής εικόνας


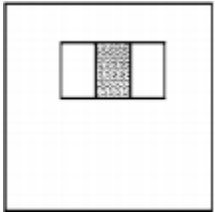
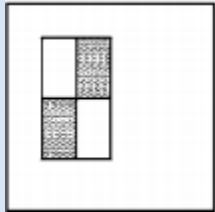
### 3.1.3: Τα χαρακτηριστικά της μεθόδου Viola & Jones

Μέσω των χαρακτηριστικών εντοπίζονται τα πρόσωπα. Τα χαρακτηριστικά τύπου Haar είναι περιοχές ορθογώνιου σχήματος και σε αυτές τις περιοχές ανάλογα με τον αριθμό τους αθροίζονται οι εντάσεις των στοιχείων και υπολογίζεται η διαφορά ανάμεσα στα αθροίσματα. Με βάση αυτές τις διαφορές, κατηγοριοποιούνται τα τμήματα μιας εικόνας.

Καθώς τα συγκεκριμένα χαρακτηριστικά είναι ορθογώνια μπορούν να υπολογιστούν πολύ γρήγορα ανεξαρτήτως του μεγέθους τους με την χρήση της ολοκληρωτικής εικόνας.



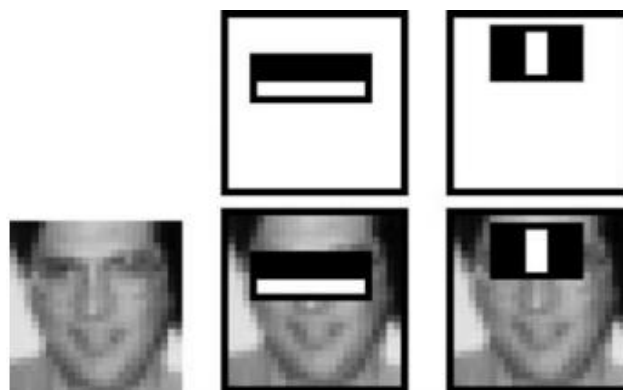
Τα χαρακτηριστικά Haar που χρησιμοποιούνται από τους Viola & Jones είναι τρία και διακρίνονται στον παρακάτω πίνακα[3]

<p>Δύο ορθογώνια</p>	<p>Το αποτέλεσμα των χαρακτηριστικών είναι η διαφορά (αφαίρεση) στα άθροισμα της άσπρης και της μαύρης περιοχής</p>	
<p>Τρία ορθογώνια</p>	<p>Η τιμή τους ορίζεται ως το άθροισμα των άσπρων περιοχών μείον το άθροισμα της μαύρης περιοχής</p>	
<p>Τέσσερα ορθογώνια</p>	<p>Σε αυτά τα ορθογώνια η τιμή των χαρακτηριστικών ορίζεται ως το άθροισμα των άσπρων μείον το άθροισμα των μαύρων περιοχών</p>	

Πίνακας 3- 2: Τα χαρακτηριστικά των Viola & Jones

### 3.1.4: Εκπαίδευση του ταξινομητή ή χρήση έτοιμων αρχείων

Οι Viola & Jones επέλεξαν τον αλγόριθμο τον AdaBoost για την επιλογή των ορθότερων στοιχείων που θα χρησιμοποιηθούν στην εκπαίδευση. Για παράδειγμα στα περισσότερα πρόσωπα υπάρχει μια διαφορά στην φωτεινότητα ανάμεσα στα μάτια και στα μάγουλα. Ένα άλλο κύριο χαρακτηριστικό είναι η διαφορά ανάμεσα στην μύτη και στα μάτια.[3]



Εικόνα 3- 8: Τα χρησιμότερα χαρακτηριστικά

Το πρώτο στάδιο του AdaBoost είναι η επιλογή των χαρακτηριστικών. Με αυτά τα χαρακτηριστικά δημιουργούνται οι λεγόμενοι αδύναμοι ταξινομητές. Κάθε αδύναμος ταξινομητής εστιάζει σε αυτά τα χαρακτηριστικά. Μόλις κατασκευαστούν οι αδύναμοι ταξινομητές «συνενώνονται» όλοι μαζί για να παράξουν έναν ισχυρό ταξινομητή ο οποίος υπολογίζεται προσθέτοντας τα βάρη των αδύναμων ταξινομητών.[3]

Η μαθηματική αναπαράσταση του αδύναμου ταξινομητή είναι η παρακάτω:

$$h(x, f, p, \theta) = \begin{cases} 1, & pf(x) < p\theta \\ 0, & otherwise \end{cases}$$

Οι παράμετροι είναι οι εξής:

- $h(x,f,p,\theta)$ : Ο αδύναμος ταξινομητής (weak classifier)
- $f$ : Το χαρακτηριστικό (feature)
- $\theta$ : Το κατώφλι (threshold)
- $p$ : πολικότητα (δείχνει την κατεύθυνση της ανισότητας) (parity)

Στην συνέχεια ξεκινάει η διαδικασία κατασκευής του ισχυρού ταξινομητή από τον αλγόριθμο AdaBoost.

Αρχικά εισάγονται οι εικόνες για την εκπαίδευση. Οι εικόνες εισάγονται με την μορφή  $(x_i, y_i)$ ,  $i = 1 \dots N$ , όπου  $y_i$  αναπαριστούν τις εικόνες που δεν περιέχουν πρόσωπα (αρνητικές εικόνες) και εικόνες που έχουν πρόσωπα (θετικές εικόνες).[3]

Μετά την εισαγωγή των εικόνων αρχικοποιούνται τα βάρη με βάση τον παρακάτω τύπο:

$$\omega_{1,i} = \frac{1}{2m}, \frac{1}{2l}$$

Όπου:

- $m$ : αρνητικό
- $l$ : Θετικό

Για  $t=1, \dots, T$  γίνονται τα παρακάτω βήματα. Το πρώτο βήμα αφορά την κανονικοποίηση των βαρών και γίνεται με τον παρακάτω τύπο:

$$W_{t,i} \leftarrow \frac{W_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

Μετά την κανονικοποίηση επιλέγεται ο ορθότερος ταξινομητής με βάση το σφάλμα τους

$$\epsilon_t = \min_{f,p,\theta} \sum_i W_i |h(x_i, f, p\theta) - y_i|$$

Ο ταξινομητής  $h(x) = h(x, f, p, \theta)$  ορίζεται με βάση τις τιμές που ελαχιστοποιούν το σφάλμα. Μετά τον ορισμό ενημερώνονται τα βάρη με βάση τους παρακάτω τύπους

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

Όπου:

$$\beta_t = \frac{e_t}{1 - e_t}$$

Το  $e$  είναι 0 αν το δείγμα έχει ταξινομηθεί σωστά

Στο τέλος οι αδύναμοι ταξινομητές συνδυάζονται για να δημιουργήσουν τον τελικό ισχυρό ταξινομητή[3]

$$h(x) = \begin{cases} 1, \sum_{t=1}^T \alpha_t h_t(x) \geq 0,5 \sum_{t=1}^T \alpha_t \\ 0, otherwise \end{cases}$$
$$a_t = \log\left(\frac{1}{\beta_t}\right)$$

### 3.1.5: Χρήση έτοιμων αρχείων

Παρόλο που είναι δυνατή η εκπαίδευση του συστήματος η διαδικασία είναι πολύ χρονοβόρα. Επίσης χρειάζεται μεγάλος αριθμός δειγμάτων για τον ορθή εκπαίδευση και κατά συνέπεια την λειτουργία του.

Για αυτόν τον λόγο υπάρχει πληθώρα έτοιμων αρχείων που μπορούν να χρησιμοποιηθούν για τον εντοπισμό περιοχών ενδιαφέροντος. Τα αρχεία αυτά είναι τύπου Extensible Markup Language (XML) και είναι εύκολη η λήψη τους από το διαδίκτυο. Έτσι είναι αρκετά εύκολη η χρήση τους χωρίς να σπαταλείται χρόνος και πόροι για την εκπαίδευση του συστήματος.

### 3.1.6: Επιλογή κατάλληλης μεθόδου

Η επιλογή της κατάλληλης τεχνικής εξαρτάται από αρκετές παραμέτρους. Ο πρώτος και πιο βασικός είναι το είδος του εντοπισμού. Αν δεν υπάρχει έτοιμο αρχείο τότε πρέπει να γίνει εκπαίδευση του συστήματος. Ένας επίσης σημαντικός παράγοντας είναι ο χρόνος για την εκπαίδευση του συστήματος.

Με βάση τα παραπάνω επιλέχθηκε η χρήση ετοιμών αρχείων για τον εντοπισμό των περιοχών ενδιαφέροντος και συγκριμένα οι παρακάτω

- haarcascade\_frontalface\_default.xml
- haarcascade\_fullbody.xml
- haarcascade\_upperbody.xml

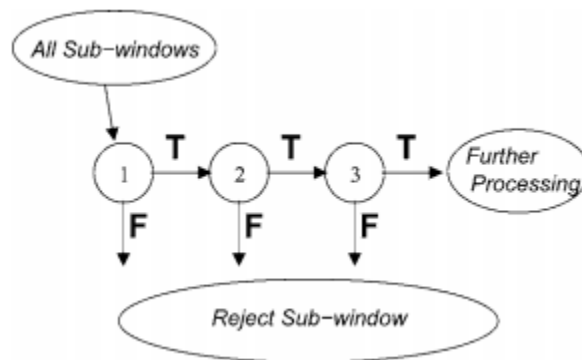
### 3.1.7: Ο καταρράκτης ταξινομητών

Σε μια εικόνα στα περισσότερα παράθυρα που θα σαρώσει ο ανιχνευτής το πιο πιθανό είναι να μην υπάρχουν πρόσωπα, κάνοντας την διαδικασία εντοπισμού αρκετά αργή. Οι Viola & Jones ήθελαν να αναπτύξουν αποτελεσματική και (το πιο σημαντικό) γρήγορη μέθοδο.

Για αυτόν τον λόγο εφάρμοσαν κάτι διαφορετικό. Η νέα έννοια πρότειναν είναι ο καταρράκτης ταξινομητών (cascade classifier). Η ιδέα της προσέγγισης αυτής είναι ότι οι απλοί ταξινομητές μπορούν να «κόψουν» τη πλειονότητα των αρνητικών παραθύρων.

Για αυτόν τον λόγο έφτιαξαν τον καταρράκτη να αποτελείται από ισχυρούς ταξινομητές συνδεδεμένους στην σειρά ξεκινώντας από τους απλούς και πηγαίνοντας στους σύνθετους. Κάθε ταξινομητής αποφασίζει αν στο σημείο που είναι ο ανιχνευτής υπάρχουν πρόσωπα. Αν δεν υπάρχει πρόσωπο τότε το παράθυρο απορρίπτεται και σταματάει η επεξεργασία από τους ταξινομητές που βρίσκονται στην σειρά. Σε περίπτωση που βρεθεί πιθανό πρόσωπο τότε συνεχίζεται η επεξεργασία στο επόμενο στάδιο.[3]

Αν ένας ταξινομητής βγάλει θετικό αποτέλεσμα καλεί τον επόμενο, και αν ο επόμενος βγάλει θετικό αποτέλεσμα καλεί τον επόμενο και ου το κάθε εξής. Αν ένας ταξινομητής βγάλει αρνητικό αποτέλεσμα τότε απορρίπτεται όλο το παράθυρο χωρίς επιπλέον επεξεργασία.[3]



Εικόνα 3- 9: Σχηματική αναπαράσταση του καταρράκτη ταξινομητών

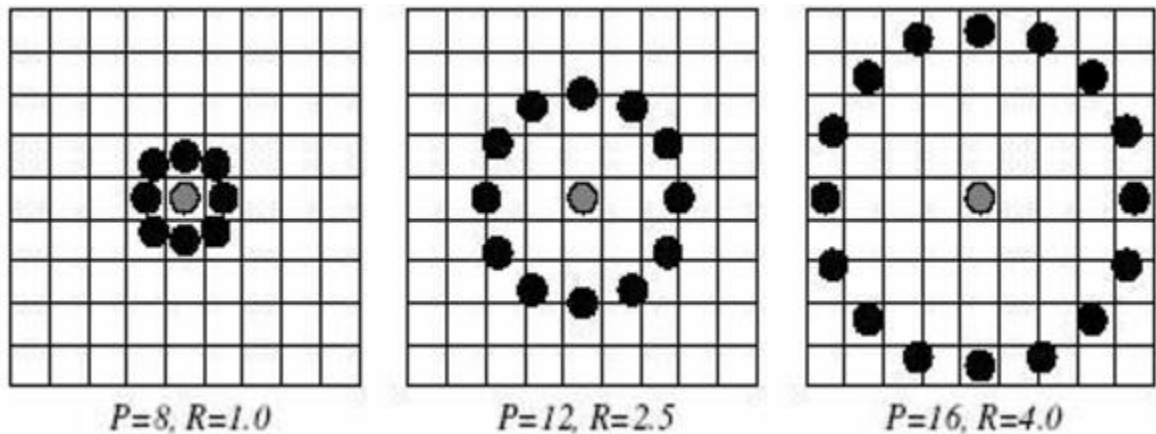
### 3.2: Η διαδικασία των τοπικών δυαδικών προτύπων για αναγνώριση προσώπων

. Μέσω της συγκεκριμένης διαδικασίας μπορούν να αναγνωριστούν πρόσωπα με αρκετά μεγάλη επιτυχία ακόμα και όταν ο φωτισμός δεν είναι αρκετά καλός.

Οι βασικές παράμετροι της μεθόδου LBPH είναι οι εξής:

1. Ακτίνα (Radius): Η ακτίνα καθορίζει την απόσταση γύρω από το κεντρικό εικονοστοιχείο που θα ληφθούν τα δείγματα. Η συνηθισμένη τιμή είναι 1
2. Γείτονες(Neighbors): Οι γείτονες αποτελούν τον αριθμό των δειγμάτων. Συνήθως χρησιμοποιούνται 8 γείτονες. Αξίζει να σημειωθεί ότι όσο περισσότεροι γείτονες χρησιμοποιηθούν τόσο καλύτερη ανάλυση θα έχει το σύστημα αλλά θα αυξηθεί και το υπολογιστικό κόστος.
3. Πλέγμα X (Grid X): Αφορά τον αριθμό των εικονοστοιχείων στον οριζόντιο άξονα με βάση τα οποία θα κατασκευαστούν τα ιστογράμματα. Η ανάλυση και το υπολογιστικό κόστος είναι ανάλογα του αριθμού των στοιχείων στο πλέγμα

4. Πλέγμα Y (Grid Y): Αφορά τον αριθμό των εικονοστοιχείων στον κάθετο άξονα με βάση τα οποία θα κατασκευαστούν τα ιστογράμματα. Η ανάλυση και το υπολογιστικό κόστος είναι ανάλογα του αριθμού των στοιχείων στο πλέγμα[14,15]

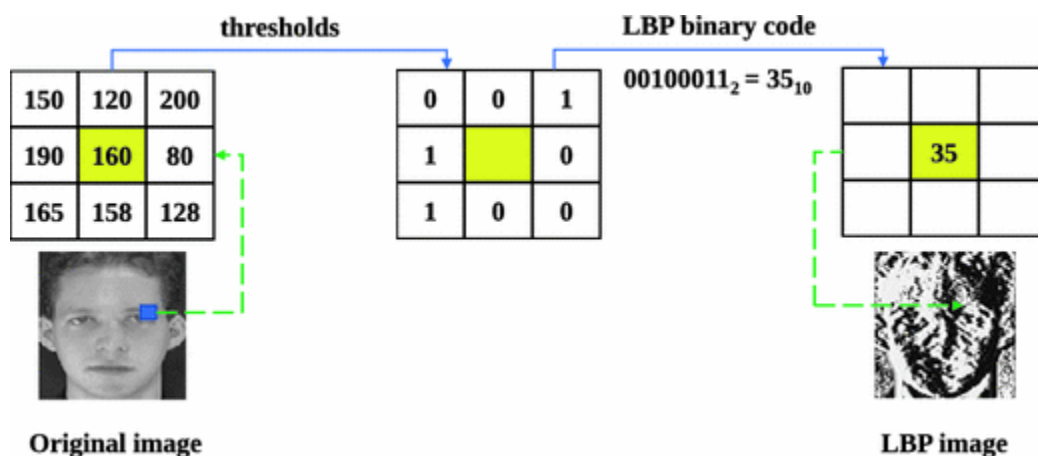


Εικόνα 3- 10: Σχηματική αναπαράσταση των παραμέτρων ακτίνας και γειτόνων

Το πρώτο υπολογιστικό βήμα είναι η δημιουργία μιας ενδιάμεσης εικόνας. Η ενδιάμεση εικόνα περιγράφει την αρχική εικόνα καλύτερα και τονίζει και περιγράφει καλύτερα τα χαρακτηριστικά του προσώπου. Ο αλγόριθμος χρησιμοποιεί ένα κυλιόμενο παράθυρο βασισμένο στις παραμέτρους της ακτίνας και των γειτόνων.

Η εικόνα, η οποία πρέπει να είναι σε κλίμακα του γκρι (grayscale image) χωρίζεται σε τμήματα. Για οκτώ γείτονες με ακτίνα ένα χωρίζεται σε τμήματα 3x3. Στα εννέα αυτά στοιχεία χρησιμοποιείται η ένταση του κεντρικού εικονοστοιχείου και με βάση αυτό ως κατώφλι συγκρίνεται με τα υπόλοιπα οκτώ. Στα εικονοστοιχεία που έντασή τους είναι μικρότερη παίρνουν την τιμή μηδέν, ενώ τα υπόλοιπα παίρνουν την τιμή ένα. Με βάση αυτές τις τιμές σχηματίζεται ένας δυαδικός 8-bit αριθμός. Αυτός ο αριθμός γίνεται δεκαδικός και αντικαθιστά το αρχικό κεντρικό εικονοστοιχείο. Αυτό γίνεται σε όλη την εικόνα, και με αυτόν τον τρόπο σχηματίζονται τα τοπικά δυαδικά πρότυπα.[12,13,14]

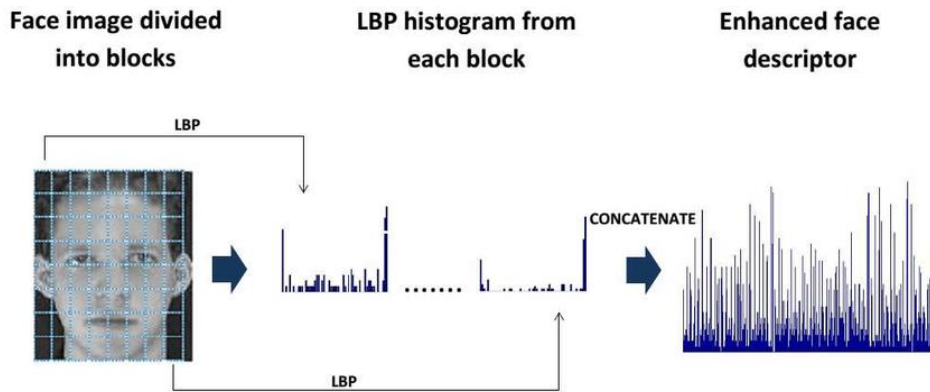
Με αυτές τις νέες δεκαδικές τιμές κατασκευάζεται ένα ιστόγραμμα για κάθε εικόνα το οποίο χρησιμοποιείται για την διαδικασία της αναγνώρισης του προσώπου



Εικόνα 3- 11: Ο υπολογισμός των τοπικών δυαδικών προτύπων

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

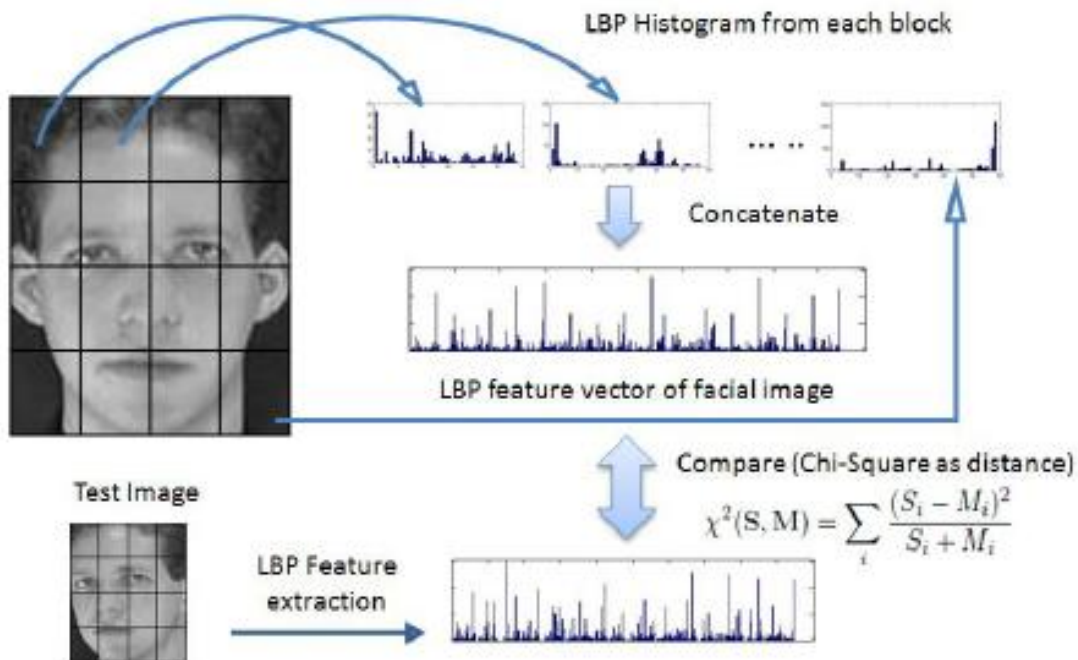
Με βάση την καινούργια εικόνα κατασκευάζονται ιστογράμματα για τις περιοχές του προσώπου. Στην συνέχεια τα ιστογράμματα συνενώνονται και κατασκευάζουν το τελικό ιστογράμματα όλης της εικόνας.



Εικόνα 3- 12: Κατασκευή ιστογραμμάτων της εικόνας

Το επόμενο βήμα είναι η διαδικασία εκπαίδευσης. Εισάγονται οι φωτογραφίες με τα πρόσωπα προς εκπαίδευση και ένα αναγνωριστικό για κάθε διαφορετικό πρόσωπο. Μέσω του αλγορίθμου κατασκευάζονται τα ιστογράμματα και αποθηκεύονται με το αναγνωριστικό τους.[13,14]

Κατά την λειτουργία του συστήματος αναγνώρισης προσώπων γίνεται η εξής διαδικασία. Μόλις εντοπιστεί ένα πρόσωπο, κατασκευάζεται το ιστογράμματα με την παραπάνω μέθοδο και συγκρίνεται με τα αποθηκευμένα για την ύπαρξη ταύτισης.



Εικόνα 3- 13: Σύγκριση ιστογραμμάτων

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Μετά την εκτέλεση δίνεται το αναγνωριστικό του προσώπου με το ιστόγραμμα που ταιριάζει περισσότερο. Πέρα από το αναγνωριστικό επιστρέφεται και η διαφορά ανάμεσα στα δύο ιστογράμματα,. Αυτή η απόσταση χρησιμοποιείται για τον έλεγχο της ορθής αναγνώρισης. Αξίζει να σημειωθεί ότι το επίπεδο εμπιστοσύνης δείχνει την απόσταση μεταξύ των δύο ιστογραμμάτων οπότε πρέπει να είναι μικρότερο από το κατώφλι που έχει οριστεί για να είναι επιτυχής η αναγνώριση.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



## Κεφάλαιο 4: Εντοπισμός κινήσεων

### 4.1: Ανάλυση τεχνικών εντοπισμού κίνησης

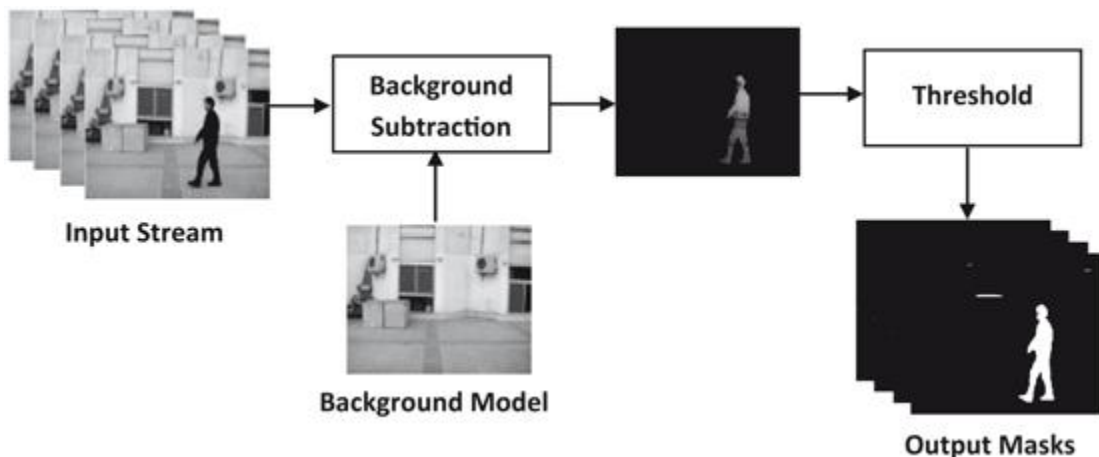
Ένα επίσης πολύ σημαντικό μέρος της παρακολούθησης χώρου είναι ο εντοπισμός κινήσεων στο πεδίο λήψης της κάμερας. Στην όραση υπολογιστών υπάρχουν δύο βασικές κατηγορίες εντοπισμού κινήσεων.

- Η μέθοδος της αφαίρεσης φόντου (Background subtraction)
- Η μέθοδος της οπτικής ροής (Optical flow)

#### 4.1.1: Η μέθοδος αφαίρεσης φόντου

Η μέθοδος αφαίρεσης φόντου βασίζεται στην σύγκριση ανάμεσα σε στο τρέχων καρέ και στο προηγούμενο αφαιρώντας τα pixel-pixel. Όσα στοιχεία είναι ίδια θα έχουν την τιμή 0. Στα υπόλοιπα ανάλογα με την τιμή τους καθορίζεται η ταχύτητα της κίνησης. Όσο περισσότερο τείνουν προς το λευκό τόσο πιο έντονη ή γρήγορη είναι η κίνηση. Το κύριο μειονέκτημα είναι ότι δεν μπορεί να εφαρμοστεί αποτελεσματικά σε εξωτερικούς χώρους καθώς επηρεάζεται με κάθε αλλαγή του φόντου.

Η συγκεκριμένη τεχνική είναι αρκετά απλή και αποτελεσματική.

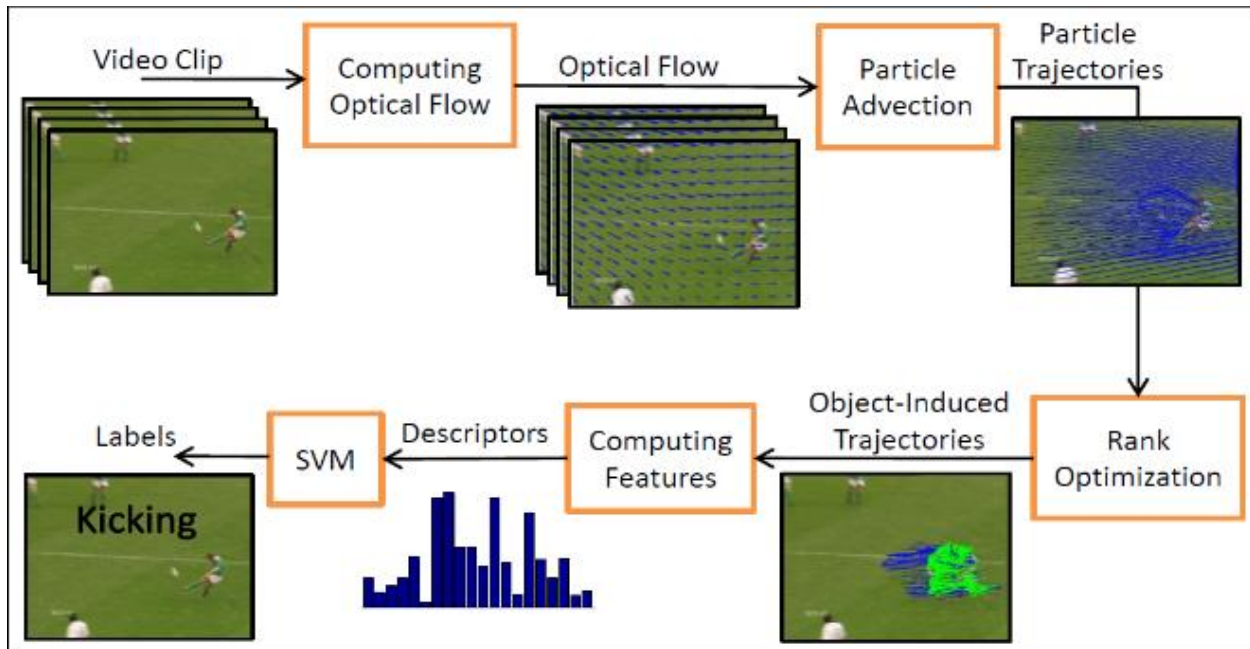


Εικόνα 4- 1: Η τεχνική αφαίρεσης φόντου

#### 4.1.2: Η μέθοδος της οπτικής ροής

Η οπτική ροή εντοπίζει τις κινήσεις των αντικειμένων σε διαδοχικά καρέ τα οποία προκαλούνται από την σχετική κίνηση ανάμεσα στο αντικείμενο και στο σημείο λήψης της εικόνας. Γίνεται και έλεγχος της κίνησης ενός στοιχείου σε σχέση με τα γειτονικά του [17,18]

Τα κύριο πλεονέκτημα είναι η υψηλή ακρίβεια στους εντοπισμούς. Το βασικό μειονέκτημα είναι η πολυπλοκότητα η οποία μεταφράζεται σε αυξημένο υπολογιστικό κόστος, ειδικά σε ανιχνεύσεις πραγματικού χρόνου.



Εικόνα 4- 2: Η τεχνική της οπτικής ροής

#### 4.2: Επιλογή κατάλληλης μεθόδου

Η επιλογή της κατάλληλης μεθόδου δεν εξαρτάται μόνο από τις συνθήκες που επικρατούν στο πεδίο λήψης της κάμερας αλλά και από τις απαιτήσεις σε ακρίβεια καθώς την ισχύ του υπολογιστικού συστήματος. Στα πλαίσια της συγκεκριμένης εργασίας επιλέχθηκε η μέθοδος της αφαίρεσης φόντου για δύο σημαντικούς λόγους. Ο πρώτος λόγος αφορά το γεγονός ότι είναι αρκετά απλή μέθοδος. Ο δεύτερος σημαντικός λόγος είναι ότι καθώς η παρακολούθηση του χώρου γίνεται σε χώρους σπιτιού δεν θα υπάρχουν σημαντικές αλλαγές στο φόντο οπότε είναι πολύ αποτελεσματική με σχετικά μικρό υπολογιστικό κόστος

#### 4.3: Ανάλυση της μεθόδου αφαίρεσης φόντου

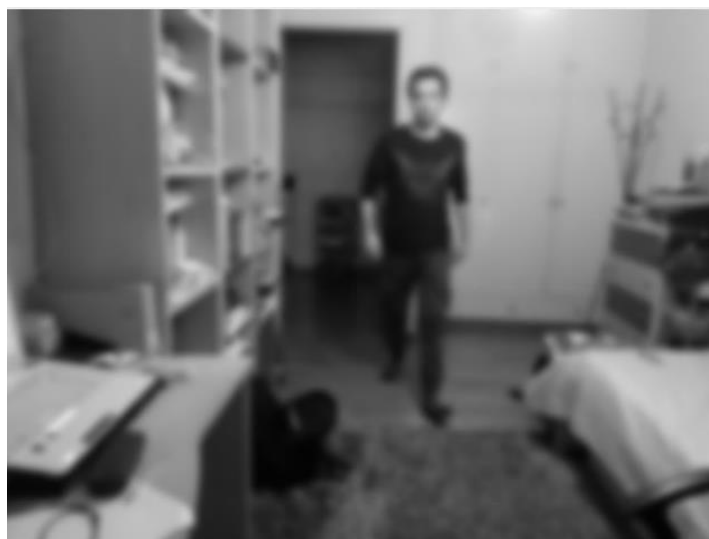
Η διαδικασία της αφαίρεσης φόντου βασίζεται στην σύγκριση δύο διαδοχικών καρέ. Μετά την σύγκριση απομονώνονται οι διαφορές με σκοπό τον εντοπισμό των κινήσεων.

Το πρώτο βήμα αφορά την μετατροπή της εικόνας σε κλίμακα του γκρι (grayscale image) καθώς τα χρώματα δεν παίζουν κανέναν ρόλο στον εντοπισμό κινήσεων.



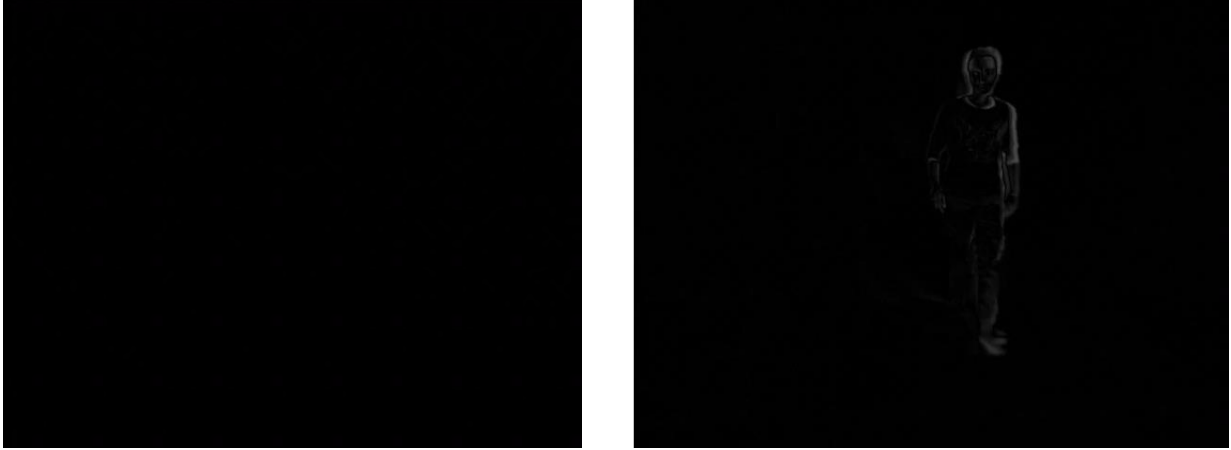
**Εικόνα 4- 3: Μετατροπή της εικόνας σε μονοχρωματική**

Αξίζει να τονιστεί ότι ακόμα και δύο διαδοχικά καρέ σε ένα βίντεο δεν είναι πανομοιότυπα ακόμα και αν δεν κινηθεί τίποτα. Αυτό οφείλεται κυρίως σε μικροσκοπικές διαφορές στους αισθητήρες της κάμερας. Ορισμένα εικονοστοιχεία θα έχουν διαφορές στην φωτεινότητα. Αυτό πρέπει να αντιμετωπιστεί καθώς μπορεί να προκαλέσει ψευδώς αναγνωρισμένες κινήσεις. Δηλαδή η εικόνα πρέπει να εξομαλυνθεί με την εφαρμογή ενός Γκαουσιανού φίλτρου που θα δώσει μια πιο ομοιόμορφη «υφή» στην εικόνα



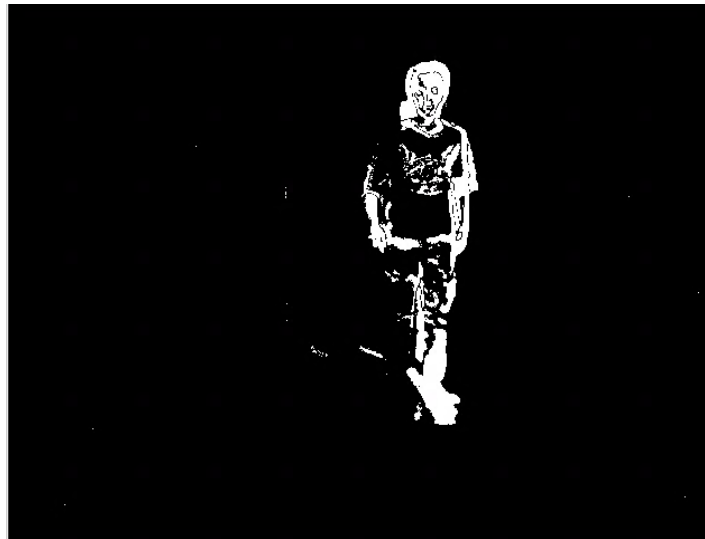
**Εικόνα 4- 4: Εφαρμογή Gaussian Blur στην εικόνα**

Με την ομοιόμορφη εικόνα υπολογίζεται η διαφορά ανάμεσα σε δύο διαδοχικά καρέ με σκοπό τον εντοπισμό κινήσεων



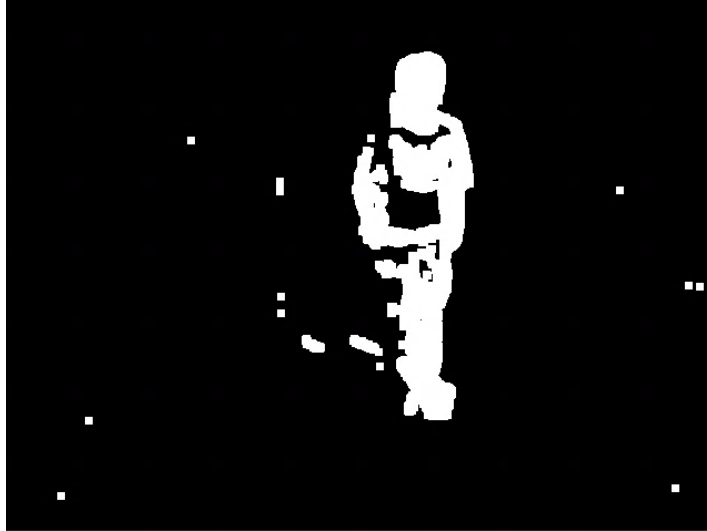
**Εικόνα 4- 5: Σύγκριση μεταξύ δύο διαδοχικών καρτέ**

Αν υπάρχει κίνηση υπολογίζεται η διαφορά στην φωτεινότητα και αν είναι μεγαλύτερη από ένα συγκεκριμένο κατώφλι τα εικονοστοιχεία αποκτούν την μεγίστη φωτεινότητα. Τα υπόλοιπα αποκτούν την ελάχιστη. Πρακτικά η εικόνα μετατρέπεται σε δυαδική (ασπρόμαυρη).



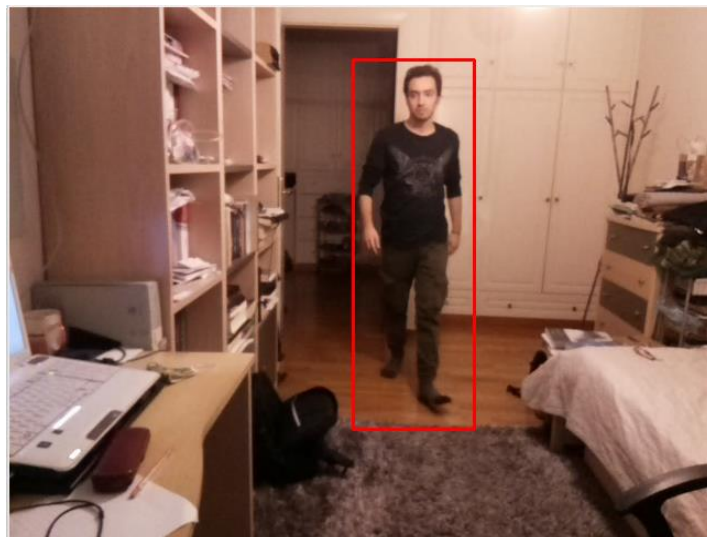
Μετά την αφαίρεση του φόντου το προσκήνιο διαστέλλεται. Ο σκοπός της διαστολής αυτής σκοπεύει στην πλήρη απομόνωση της κίνησης. Ένα δυνητικό πρόβλημα με την διαστολή είναι ότι διογκώνονται και στοιχεία που οφείλονται σε θόρυβο. Το πρόβλημα αυτό επιλύεται με τον έλεγχο του μεγέθους του περιγράμματος των λευκών εικονοστοιχείων.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



**Εικόνα 4- 6: Διαστολή φωτεινών εικονοστοιχείων**

Το τελευταίο βήμα είναι ο εντοπισμός των περιγραμμάτων για τα λευκά στοιχεία. Μόλις υπολογιστεί το περίγραμμα υπολογίζεται αν το μέγεθος του είναι μεγαλύτερο από μια προκαθορισμένη τιμή. Σε περίπτωση που είναι εμφανίζεται στην οθόνη ένα κόκκινο παραλληλόγραμμο στο περίγραμμα με το μεγαλύτερο μέγεθος.



**Εικόνα 4- 7: Το τελικό καρτέ που εμφανίζεται στην κάμερα**

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## Κεφάλαιο 5: Ο υλικός εξοπλισμός της εργασίας

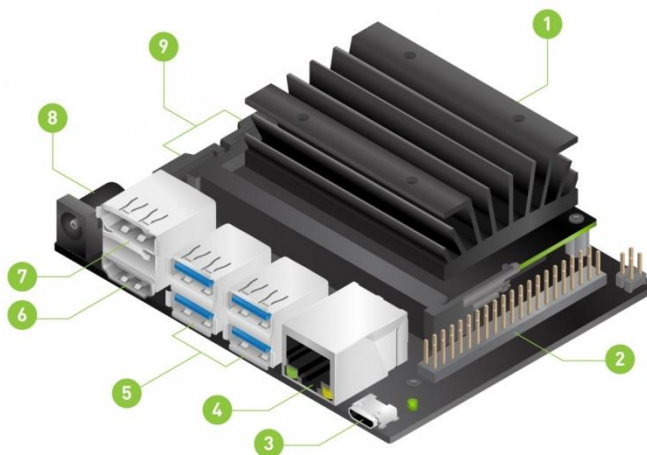
### 5.1: Εισαγωγή

Η σωστή επιλογή του υλικού εξοπλισμού είναι πολύ σημαντική για την ορθή λειτουργία του συστήματος. Η επιλογή αποτελεί ένα πρόβλημα πολλαπλών παραμέτρων με τις πιο βασικές να είναι το κόστος, η υπολογιστική ισχύς, το μέγεθος. Ένας επιπλέον παράγοντας επιλογής είναι η υποστήριξη βιβλιοθηκών υπολογιστικής όρασης.

### 5.2: Διαθέσιμοι μικροϋπολογιστές: NVIDIA Jetson Nano

#### 5.2.1: Εισαγωγή

Το Jetson Nano είναι ένας ισχυρός και ταυτόχρονα σχετικά μικρού μεγέθους μικροϋπολογιστής από την NVIDIA. Έχει ποικίλες εφαρμογές από έναν καθημερινό υπολογιστή μέχρι διακομιστή ιστοσελίδων)[19]



Εικόνα 5- 1: Το Jetson Nano της NVIDIA

Τα στοιχεία της εικόνας 5-1 διακρίνονται στον παρακάτω πίνακα[19]

Αρίθμηση	Ονομασία μέρους
1	Κάρτα SD για αποθήκευση
2	40 ακροδέκτες για σύνδεση περιφερειακών
3	Micro-USB για παροχή ισχύος ή για σύνδεση συσκευών
4	Θύρα Gigabit Ethernet για σύνδεση στο διαδίκτυο
5	4x θύρες USB 3.0
6	Θύρα HDMI
7	Θύρα Display Port
8	Παροχή ισχύων 5V DC Jack
9	MIPI CSI-2 camera Connector

Πίνακας 5- 1: Τα βασικά στοιχεία του Jetson Nano

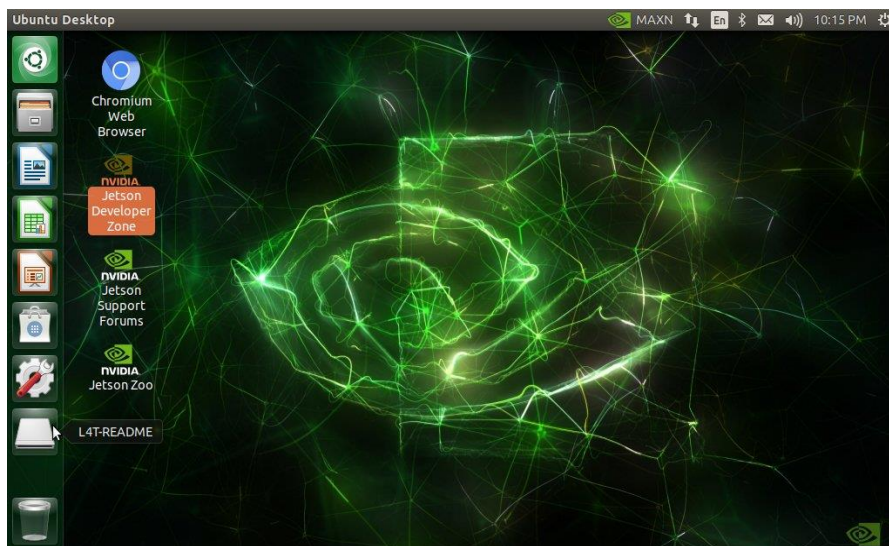
### 5.2.2: Τεχνικά χαρακτηριστικά

Στην συνέχεια διακρίνονται τα τεχνικά χαρακτηριστικά του Jetson Nano[20]

NVIDIA Jetson Nano	
Κεντρική μονάδα επεξεργασίας	Quad-core ARM Cortex-A57 MPCore Processor
Μνήμη τυχαίας προσπέλασης	4GB 64-bit LPDDR4, 1600MHz 25.6 GB/s
Αποθηκευτικός χώρος	16GB eMMC 5.1
Κάμερα	12 lanes (3x4 or 4x2) MIPI CSI-2 D-PHY 1.1 (1.5 GB/s per pair)
Θύρες για περιφερειακά και συσκευές	HDMI 2.0 and eDP 1.4
	4x USB 3.0, USB 2.0 Micro B
	40x Header Pins
Σύνδεση στο διαδίκτυο	Gigabit Ethernet M.2 Key E
Υποστηριζόμενα πρωτόκολλα	GPIO
	I2C
	I2S
	SPI
	UART
Λειτουργικό σύστημα	Linux For Tegra (L4T)

Πίνακας 5- 2: Τα τεχνικά χαρακτηριστικά του Jetson Nano

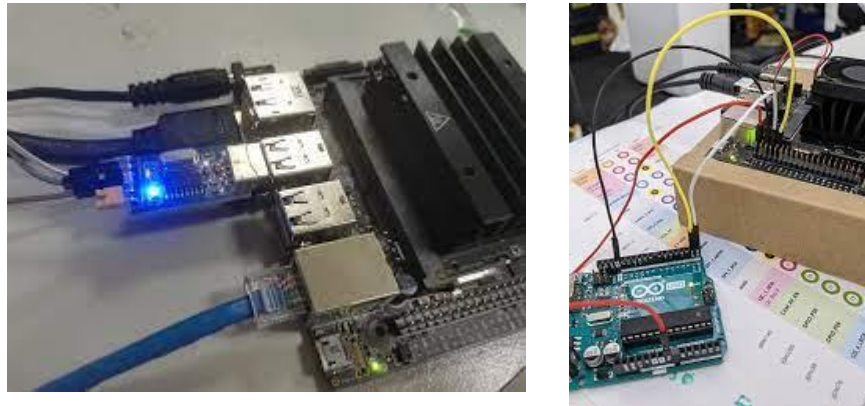
Η λήψη του λειτουργικού συστήματος μπορεί να γίνει από την NVIDIA. Καθώς βασίζεται σε LINUX είναι σχετικά απλή διαδικασία η εγκατάσταση πακέτων υπολογιστικής όρασης και διαχείρισης βάσεων δεδομένων.[21]



Εικόνα 5- 2: Το λειτουργικό σύστημα L4T

Πέρα από την υποστήριξη πακέτων υπολογιστικής όρασης και διαχείρισης βάσεων δεδομένων το jetson έχει την δυνατότητα να υποστηρίζει και σειριακή επικοινωνία με τον μικροελεγκτή για την κίνηση του βραχίονα. Η επικοινωνία γίνεται είτε μέσω USB είτε μέσω των ακροδεκτών



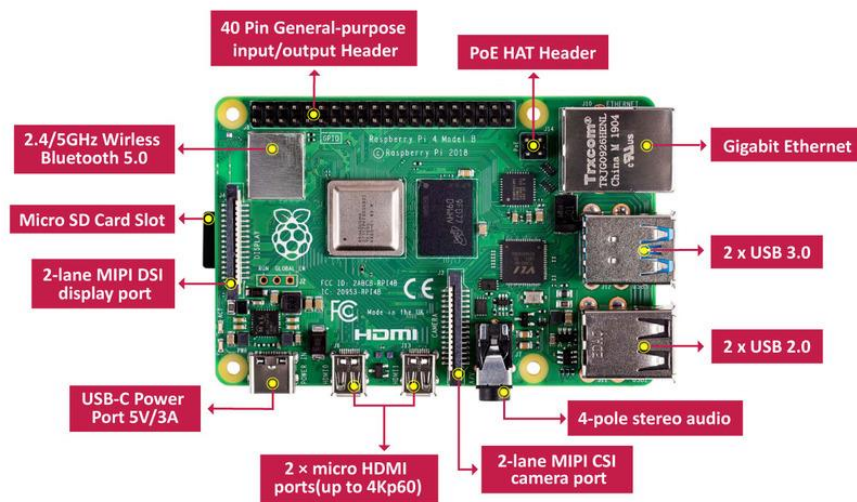


Εικόνα 5- 3: Σειριακή επικοινωνία μεταξύ Jetson και του μικροϋπολογιστή

### 5.3: Διαθέσιμοι μικροϋπολογιστές Raspberry Pi

#### 5.3.1: Εισαγωγή

Το Raspberry Pi είναι ένας υπολογιστής μικρού μεγέθους (περίπου σε μέγεθος τηλεκάρτας) που αναπτύχθηκε από τον οργανισμό Raspberry Pi Foundation στην Αγγλία. Ο αρχικός σκοπός του ήταν η εκμάθηση βασικών αρχών της επιστήμης των υπολογιστών σε γυμνάσια και λύκεια. Σήμερα είναι αρκετά δημοφιλές λόγω του χαμηλού και λόγω του χαμηλού του κόστους χρησιμοποιείται σε πληθώρα εφαρμογών όπως το internet of things, τα έξυπνα σπίτια και η ρομποτική. Το τελευταίο μοντέλο είναι το Raspberry Pi Model 4[22]



Εικόνα 5- 4: Το Raspberry Pi

### 5.3.2: Τεχνικά χαρακτηριστικά

Στην συνέχεια παρουσιάζονται τα τεχνικά χαρακτηριστικά του Raspberry Pi.[23]

Τεχνικά χαρακτηριστικά Raspberry Pi 4	
Επεξεργαστής	Quad core Cortex A-72 64-bit Broadcom BCM 2711, @1.5GHz
Μνήμη RAM	LDDR4-3200 SDRAM Μέγεθος 2,4,8 GB ανάλογα με το μοντέλο
Λειτουργικά συστήματα	Raspbian
	Ubuntu Mate
	Windows 10 IoT Core
	RISC OS
Ασύρματα πρωτόκολλα	2.4 και 5GHz IEEE 802.11 Wireless
	Bluetooth 5.0 BLE
Ενσύρματη σύνδεση στο διαδίκτυο	Gigabit Ethernet που υποστηρίζει και Power Over Ethernet για την τροφοδοσία του Raspberry Pi
Ακροδέκτες	40 ακροδέκτες για περιφερειακά που υποστηρίζουν διάφορα πρωτόκολλα.
Θύρες για περιφερειακά και εξωτερικές συσκευές	2x USB 3.0
	2x USB 2.0
	2x Micro HDMI
	1x Micro SD
	1x DSI Display Port
	1x CSI Camera Port
	1x Audio Port
Τάση και ρεύμα λειτουργίας	5V Συνεχόμενη Τάση DC μέσω USB Type C. Ρεύμα έντασης 3A
	5V Συνεχόμενη τάση DC μέσω GPIO. Ρεύμα έντασης 3A
Επίπεδα υπερχρονισμού	5 επίπεδα
Υποστηριζόμενα πρωτόκολλα	PWM (Pulse Width Modulation)
	SPI (Serial Peripheral Interface)
	I2C (Inter-Integrated Circuit)
	UART (Universal Asynchronous Receiver-Transmitter)

Πίνακας 5- 3: Τα τεχνικά χαρακτηριστικά του Raspberry Pi

### 5.4: Επιλογή κατάλληλου μικροϋπολογιστή

Η επιλογή του κατάλληλου μικροϋπολογιστή εξαρτάται από πολλά κριτήρια. Ένα βασικό κριτήριο είναι το κόστος. Η μέση τιμή για ένα Raspberry Pi στην Ελλάδα είναι στα 69 ευρώ ενώ για ένα jetson είναι στα 90 ευρώ.

Ένας επίσης σημαντικός παράγοντας είναι η υπολογιστική ισχύς. Εδώ οι δυο μικροϋπολογιστές είναι αρκετά όμοιοι αν και όταν δεν απαιτείται υψηλή GPU Power το Raspberry Pi μέσω με καλύτερη απόδοση μέσω υπερχρονισμού. Όσον αφορά την προσβασιμότητα το Raspberry Pi πέρα από Ethernet υποστηρίζει και ασύρματη σύνδεση στο διαδίκτυο που το κάνει πιο προσβάσιμο.

Ο τελευταίος παράγοντας σύγκρισης είναι το μέγεθος. Το Raspberry Pi είναι μικρότερο από το Jetson που σημαίνει ότι μπορεί να χωρέσει πιο εύκολα στην βάση του βραχίονα.

Με βάση τα παραπάνω επιλέχθηκε το Raspberry pi καθώς είναι πιο μικρό μέγεθος, παρέχει μεγαλύτερη ευελιξία ως προς την συνδεσιμότητα και έχει την καλύτερη απόδοση ως προς το κόστος. Στον παρακάτω πίνακα φαίνονται συνοπτικά η σύγκριση

Παράγοντας σύγκρισης/ μικροϋπολογιστής	Raspberry Pi	Jetson Nano
Κόστος	X	
Υπολογιστική ισχύς/Ταχύτητα επεξεργασίας		X
Μέγεθος	X	

Πίνακας 5- 4: Σύγκριση των μικροϋπολογιστών

### 5.5: Παραμετροποίηση του Raspberry Pi

Μετά την επιλογή του Raspberry Pi, είναι απαραίτητη η παραμετροποίηση ορισμένων σημείων του Raspberry Pi. Τα βήματα της παραμετροποίησης είναι

1. Εγκατάσταση λειτουργικού συστήματος
2. Προσβασιμότητα από τοπικό δίκτυο
3. Εγκατάσταση πακέτων

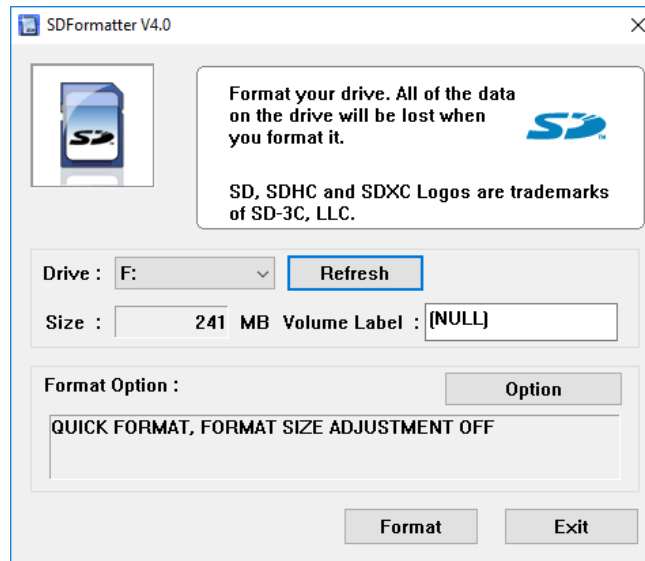
#### 5.5.1: Εγκατάσταση λειτουργικού συστήματος

Το Raspberry Pi υποστηρίζει αρκετά λειτουργικά συστήματα. Για τις ανάγκες της εργασίας επιλέχθηκε το λειτουργικό σύστημα Raspbian καθώς είναι ελαφρύ και είναι εύκολη η εγκατάσταση των απαραίτητων πακέτων.

Η διαδικασία εγκατάστασης του λειτουργικού συστήματος είναι μια αρκετά απλή διαδικασία η οποία μέσω του προγράμματος Raspberry Pi Imager έχει γίνει ακόμα πιο απλή. Απαραίτητος εξοπλισμός είναι το Raspberry Pi, μια κάρτα SD μια οθόνη, το τροφοδοτικό, ένα πληκτρολόγιο και ένα ποντίκι.

Πριν την έναρξη της διαδικασίας εγκατάστασης πρέπει να γίνει μορφοποίηση της κάρτας SD (Format). Υπάρχει πληθώρα προγραμμάτων που μπορούν να χρησιμοποιηθούν για αυτόν τον σκοπό. Το πρόγραμμα που επιλέχθηκε τελικά είναι το SD Card Formatter καθώς είναι αποτελεσματικό και δωρεάν. Μετά την εκκίνηση του προγράμματος στο πεδίο drive επιλέγεται η κάρτα SD και με το πλήκτρο format ξεκινάει η διαδικασία. Με την επιτυχή ολοκλήρωση εμφανίζεται αντίστοιχο μήνυμα.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



Εικόνα 5- 5: Το πρόγραμμα SD Card Formatter

Εφόσον ολοκληρωθεί το format της κάρτας SD πρέπει να γίνει λήψη του προγράμματος το οποίο θα γράψει το λειτουργικό σύστημα στην SD. Το πρόγραμμα για αυτή την διαδικασία είναι το Raspberry Pi Imager, το οποίο μπορεί να βρεθεί στην επίσημη ιστοσελίδα του Raspberry Pi.[24]

### Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

Εικόνα 5- 6: Λήψη του προγράμματος Raspberry Pi Imager

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

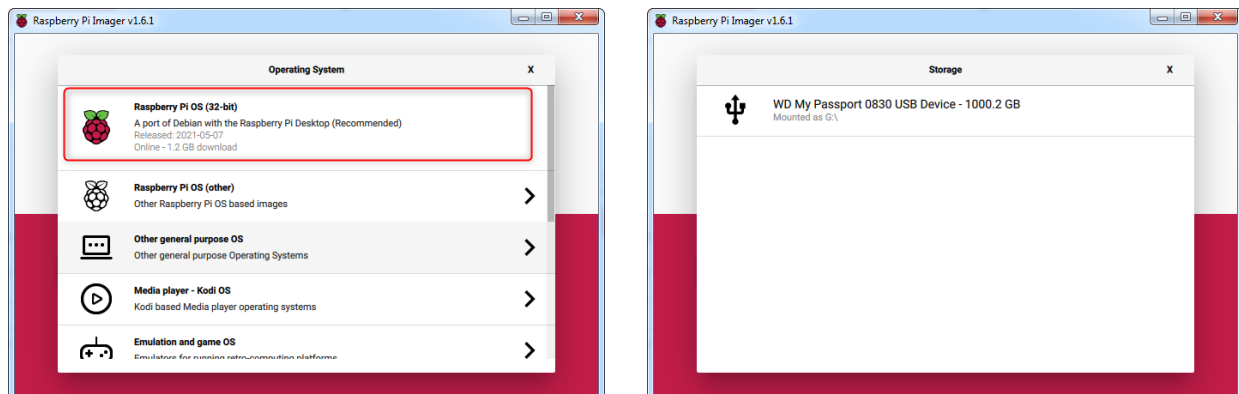
Το Raspberry Pi Imager έχει απλοποιήσει σημαντικά την διαδικασία εγκατάστασης του λειτουργικού συστήματος. Πρακτικά ο χρήστης μόλις ανοίξει το πρόγραμμα το μόνο που έχει να κάνει είναι να επιλέξει το επιθυμητό λειτουργικό σύστημα, και την κάρτα SD.[24]

Με την εκκίνηση του προγράμματος εμφανίζεται η παρακάτω εικόνα



**Εικόνα 5- 7: Το πρόγραμμα Raspberry Pi Imager**

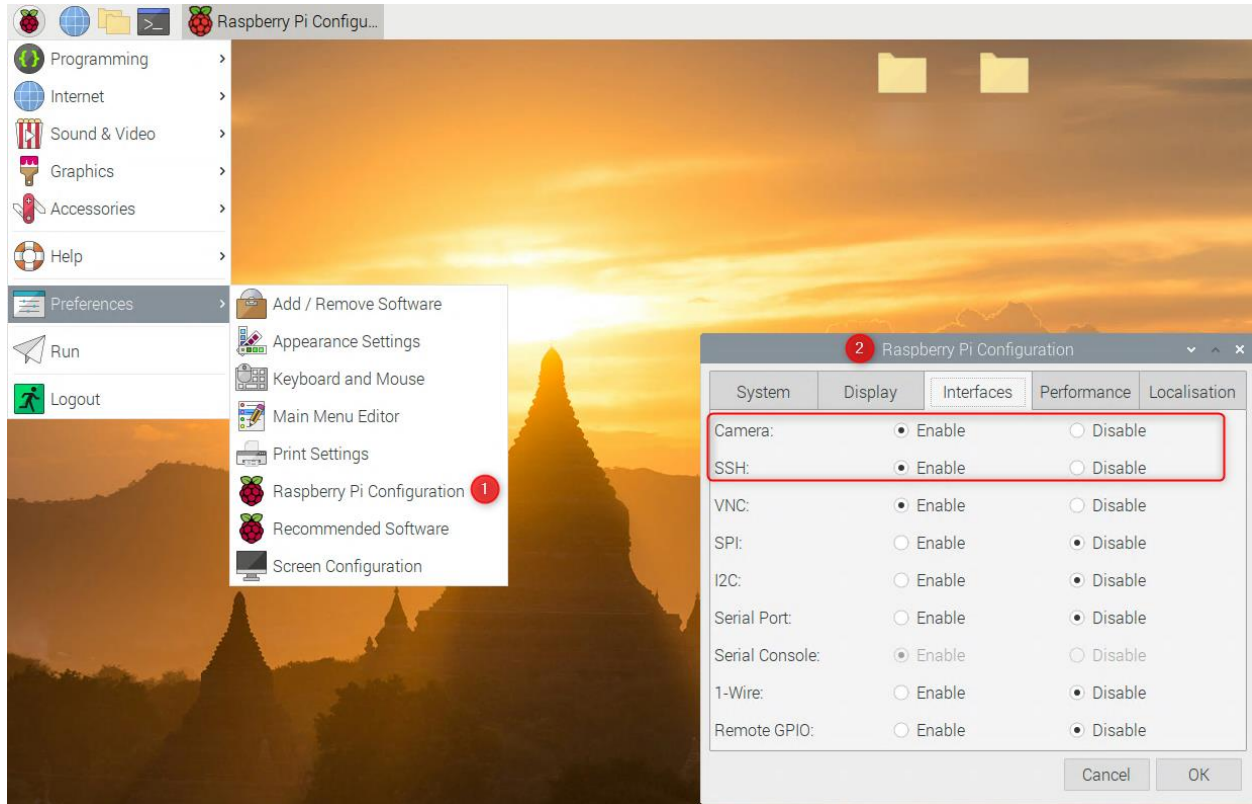
Μέσω του προγράμματος είναι δυνατή η επιλογή του επιθυμητού λειτουργικού συστήματος (CHOOSE OS) καθώς και η κάρτα SD (CHOOSE STORAGE). Τέλος με το πλήκτρο WRITE γίνεται η εγγραφή του λειτουργικού συστήματος στην κάρτα SD.[24]



**Εικόνα 5- 8: Επιλογή ρυθμίσεων για το Raspberry Pi**

Μόλις ολοκληρωθεί η εγγραφή, η κάρτα SD τοποθετείται στην υποδοχή του Raspberry Pi. επίσης συνδέονται στο Raspberry Pi το πληκτρολόγιο, η οθόνη, το ποντίκι και τελευταία η παροχή ισχύος. Μια σημαντική σημείωση είναι ότι το Raspberry Pi δεν έχει διακόπτη λειτουργίας, ενεργοποιείται μόλις συνδεθεί στο ρεύμα. Καθώς το λειτουργικό σύστημα έχει εγκατασταθεί μέσω του Raspberry Pi Imager χρειάζεται να γίνει η παραμετροποίηση της χώρας και των λοιπών παραμέτρων. Μια σημαντική παραμετροποίηση είναι στο Raspi-config η ενεργοποίηση της κάμερας και του SSH. Με την

ενεργοποίηση των συγκεκριμένων παραμετροποιήσεων είναι δυνατή η χρήση της κάμερας καθώς και η επικοινωνία του μικροϋπολογιστή μέσω secure shell με άλλες συσκευές όπως ένας υπολογιστής μέσω προγραμμάτων όπως το Putty και το Remote Desktop Protocol. Με αυτόν τον τρόπο δεν είναι απαραίτητη η χρήση οθόνης. Μετά την ενεργοποίηση των επιλογών είναι απαραίτητη η επανεκκίνηση του Raspberry Pi. Η παραμετροποίηση γίνεται από το μενού Preferences και Raspberry Pi configuration. Η καρτέλα που περιέχει τις παραμέτρους για την κάμερα και το SSH είναι η Interfaces όπως φαίνεται στην εικόνα



Εικόνα 5- 9: Παραμετροποίηση του Raspberry Pi

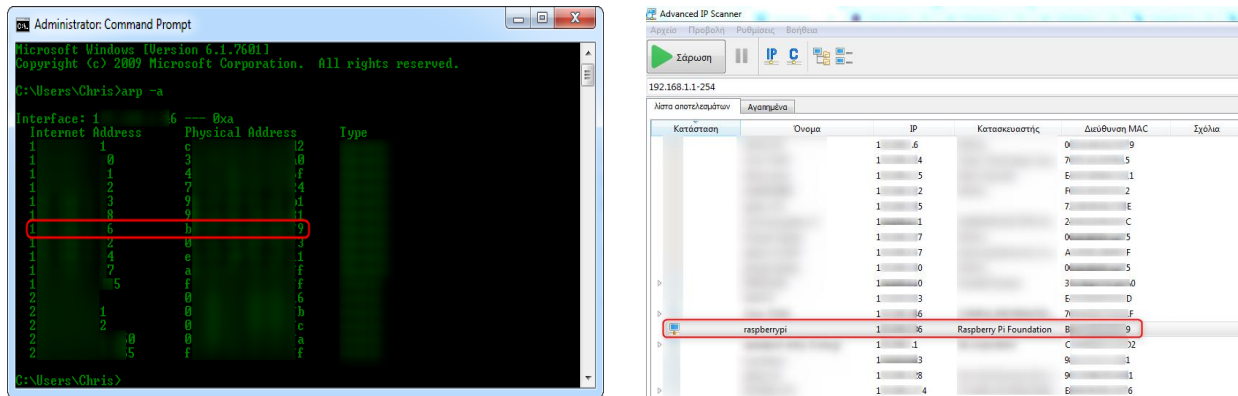
### 5.5.2: Προσβασιμότητα από το τοπικό δίκτυο

Η προσβασιμότητα από το τοπικό δίκτυο είναι μια προαιρετική παραμετροποίηση που αφορά την προσβασιμότητα του Raspberry Pi από το τοπικό δίκτυο. Με αυτόν τον τρόπο δεν χρειάζονται το ποντίκι, το πληκτρολόγιο και η οθόνη. Καθώς το SSH είναι ενεργοποιημένο μέσω απλών προγραμμάτων είναι δυνατή η πρόσβαση στο Raspberry Pi. Το πρώτο βήμα ολόκληρης της διαδικασίας είναι να βρεθεί η διεύθυνση IP του Raspberry. Αυτό μπορεί να επιτευχθεί με δύο τρόπους

1. Αν είναι γνωστή η διεύθυνση MAC του Raspberry Pi τότε η IP μπορεί να βρεθεί μέσω του Command Prompt των Windows υπό την προϋπόθεση ότι και οι δύο συσκευές είναι στο ίδιο υποδίκτυο. Η εντολή είναι arp -a και εμφανίζει τις διευθύνσεις MAC και IP όλων των συσκευών που βρίσκονται στο τοπικό δίκτυο



2. Σε περίπτωση που η MAC δεν είναι γνωστή τότε η IP μπορεί να βρεθεί μέσω προγραμμάτων. Υπάρχουν αρκετά προγράμματα τα οποία βρίσκουν τις IP και τις MAC των συσκευών του δικτύου. Ένα από αυτά είναι το advanced IP Scanner το οποίο είναι δωρεάν. Με το πλήκτρο Σάρωση εμφανίζονται οι συσκευές και οι διευθύνσεις.[25]

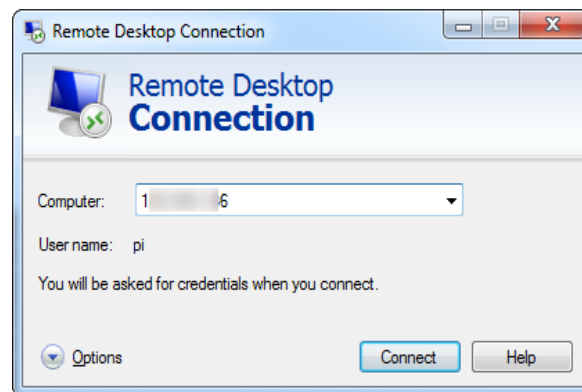


Εικόνα 5- 10: Εύρεση της IP του Raspberry Pi

Με την γνωστή πλέον IP είναι δυνατή η σύνδεση με το πρόγραμμα PuTTY. Το PuTTY είναι ένα δωρεάν πρόγραμμα που παρέχει σύνδεση SSH. Η σύνδεση SSH παρέχει ασφαλή σύνδεση με απομακρυσμένες συσκευές σε ένα μη ασφαλές δίκτυο. Συνήθως η σύνδεση γίνεται μέσω διεπαφής CLI και είναι δυνατή η απομακρυσμένη εκτέλεση εντολών. Στο πεδίο Host Name εισάγεται η IP του Raspberry Pi. Αν εμφανιστεί κάποιο μήνυμα με το πλήκτρο Yes συνεχίζεται η διαδικασία σύνδεσης

Στη συνέχεια εισάγονται το όνομα χρήστη και ο κωδικός πρόσβασης όπως έχουν οριστεί κατά την παραμετροποίηση του Raspberry Pi. η διεπαφή γίνεται μέσω γραμμής εντολών(command line interface). Με την χρήση των εντολών `sudo apt-get install xrdp` και `sudo apt-get install tightvncserver` πραγματοποιείται εγκατάσταση των πακέτων xrdp και tightvncserver που είναι απαραίτητα για την απομακρυσμένη σύνδεση.

Μόλις ολοκληρωθεί η εγκατάσταση είναι δυνατή η σύνδεση μέσω του Remote Desktop Connection των Windows. Στο πεδίο Computer εισάγεται η διεύθυνση IP του Raspberry Pi και με το πλήκτρο connect ξεκινάει η διαδικασία σύνδεσης.



Εικόνα 5- 11: Σύνδεση μέσω Remote Desktop Connection



**Εικόνα 5- 12: Επιφάνεια εργασίας μέσω του Remote Desktop**

### 5.5.3: Εγκατάσταση πακέτων

Για την ορθή λειτουργία της εφαρμογής είναι απαραίτητη η εγκατάσταση των πακέτων MySQL και openCV για βάσεις δεδομένων και για αναγνώριση προσώπων. Στον παρακάτω πίνακα περιγράφονται με την σειρά όλες οι εντολές που πρέπει να εισαχθούν για την ενημέρωση και εγκατάσταση των πακέτων.[26]

Περιγραφή	Εντολή
Ενημέρωση και αναβάθμιση υπαρχόντων πακέτων	<ul style="list-style-type: none"> <li>• sudo apt-get update</li> <li>• sudo apt-get upgrade</li> </ul>
Εγκατάσταση MySQL Server και δημιουργία χρήστη	<ul style="list-style-type: none"> <li>• sudo apt install mariadb-server</li> <li>• sudo mysql_secure_installation (στα μηνύματα επιβεβαίωσης Y)</li> <li>• sudo mysql -u root -p (συνδεση στον mysql server)</li> <li>• CREATE USER [IF NOT EXISTS] 'new_user_name'@'host_name' IDENTIFIED BY 'user_password'</li> </ul>
Εγκατάσταση πακέτων OpenCV	<ul style="list-style-type: none"> <li>• sudo apt install libatlas-base-dev -y</li> <li>• sudo apt install libjasper-dev -y</li> <li>• sudo apt install libqtgui4 -y</li> <li>• sudo apt install python3-pyqt5 -y</li> <li>• sudo apt install libqt4-test -y</li> <li>• sudo apt install libhdf5-dev libhdf5-serial-dev -y</li> <li>• sudo pip3 install opencv-contrib-python==3.4.4.19</li> </ul>
Εγκατάσταση πακέτου Py Serial	<ul style="list-style-type: none"> <li>• python3 -m pip install pyserial.</li> </ul>

**Πίνακας 5- 5: Εγκατάσταση των πακέτων**



## 5.6: Υλοποίηση σειριακής επικοινωνίας μεταξύ Raspberry Pi και Arduino

### 5.6.1: Εισαγωγή

Ένα επίσης κομβικό σημείο της εργασίας είναι η επικοινωνία ανάμεσα στο Raspberry Pi και στο Arduino. Η ανάγκη επικοινωνίας ανάμεσα τους έγκειται στις διαφορετικές λειτουργίες που εκτελούν. Η επεξεργασία των δεδομένων της κάμερας γίνεται μέσω του Raspberry Pi, όταν ένα πρόσωπο αρχίζει να κινείται προς τα άκρα του πεδίου λήψης της κάμερας είναι απαραίτητη η περιστροφή του βραχίονα. Ο έλεγχος των σερβοκινητήρων γίνεται από το Arduino, πράγμα που σημαίνει ότι το Raspberry Pi πρέπει να ενημερώσει το Arduino προς ποια κατεύθυνση πρέπει να στραφεί ο βραχίονας καθώς στο άκρο του βρίσκεται η κάμερα.

Η σειριακή επικοινωνία είναι ένας απλός τρόπος μεταφοράς δεδομένων. Σε αντίθεση με την παράλληλη επικοινωνία που αποστέλλονται πολλαπλά bit ταυτόχρονα, στην σειριακή επικοινωνία αποστέλλονται διαδοχικά, ένα bit την φορά. Η ταχύτητα που αποστέλλονται τα δεδομένα ονομάζεται baud rate (είναι ο ρυθμός που μεταφέρεται η πληροφορία στο κανάλι) και πρέπει να είναι κοινή και στις δύο συσκευές. Για την επίτευξη της επικοινωνίας χρησιμοποιείται το πρωτόκολλο UART καθώς υποστηρίζεται και από τις δύο συσκευές. Επίσης υπάρχουν αρκετές έτοιμες βιβλιοθήκες για την διαχείριση του συγκεκριμένου πρωτοκόλλου.

Το UART είναι ασύγχρονο πρωτόκολλο πολλαπλών κυρίων (Multi Master Asynchronous Protocol) που σημαίνει ότι οι συσκευές μπορούν να στέλνουν δεδομένα όποτε είναι απαραίτητο. Αυτό είναι πολύ σημαντικό καθώς οι συσκευές πρέπει να ανταλλάσσουν δεδομένα χωρίς να είναι γνωστό ποιος θα ξεκινήσει την επικοινωνία.

Το Arduino διαθέτει ένα UART (το μοντέλο Mega διαθέτει περισσότερα) και μπορεί να εφαρμοστεί μέσω της θύρας USB είτε μέσω των pins RX/TX τα οποία δεν μπορούν να χρησιμοποιηθούν ταυτόχρονα.

Το Raspberry Pi, μπορεί να υποστηρίξει σειριακές επικοινωνίες στις θύρες USB, όπου κάθε μία έχει διαφορετικό όνομα. Επιπλέον στα GPIO Pins TX0, RX0 μπορεί να εφαρμοστεί ένα ακόμα UART

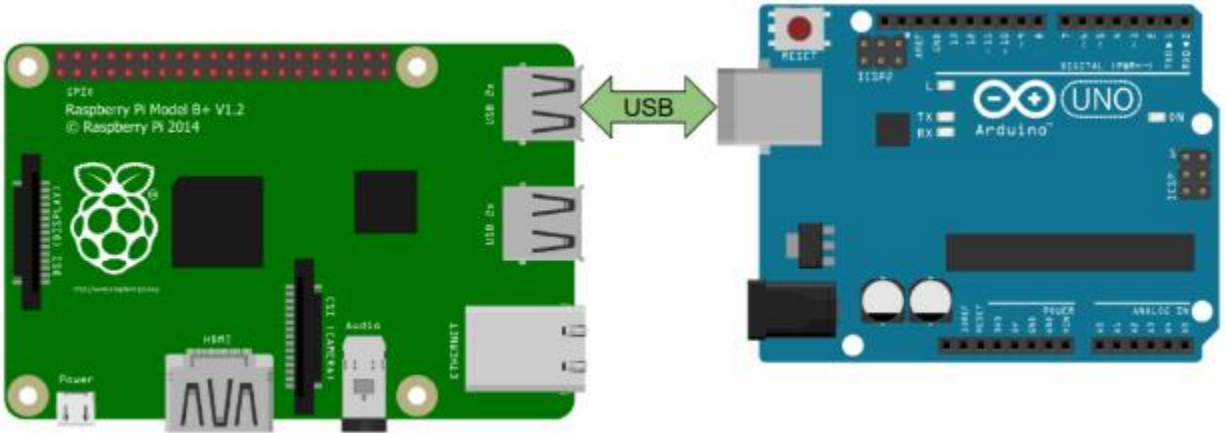
Υπάρχουν δύο τρόποι που μπορεί να εφαρμοστεί η σειριακή επικοινωνία. Ο ένας τρόπος είναι μέσω της θύρας USB. Ο δεύτερος τρόπος γίνεται μέσω των ακροδεκτών RX/TX [67]

### 5.6.2: Σύνδεση Raspberry Pi και Arduino

Η σύνδεση ανάμεσα στο Raspberry Pi και στο Arduino μπορεί να γίνει με δύο δυνατούς τρόπους:

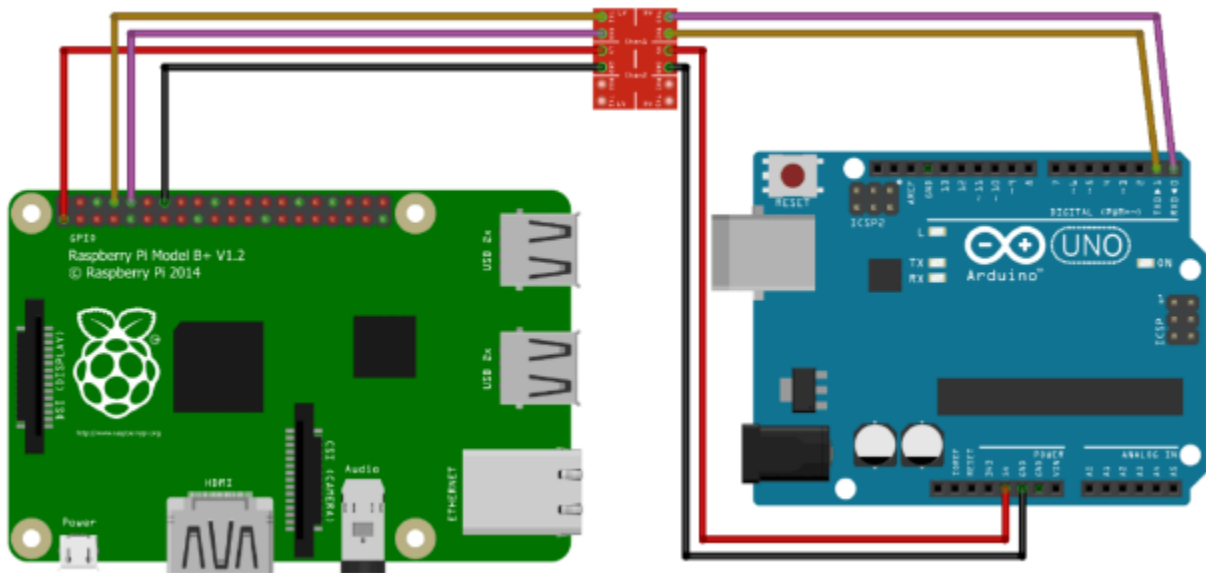
Η πρώτη και πιο απλή μέθοδος είναι μέσω καλωδίου στις θύρες USB. Το Raspberry Pi έχει τέσσερις θύρες USB και μπορεί να χρησιμοποιηθεί οποιαδήποτε θύρα. Το Arduino έχει μια θύρα (την ίδια που πραγματοποιείται το φόρτωμα των προγραμμάτων). Η σύνδεση φαίνεται στο σχεδιάγραμμα της εικόνας 5-13. [67]

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



Εικόνα 5- 13: Σύνδεση μέσω της θύρας USB

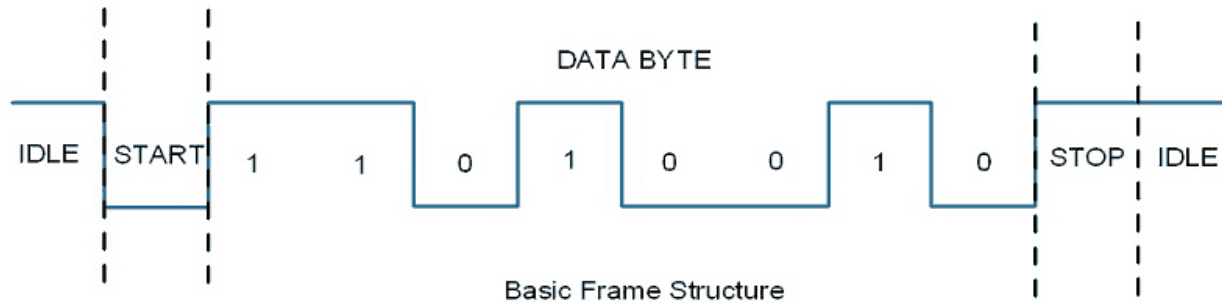
Ο δεύτερος τρόπος σύνδεσης γίνεται μέσω των GPIO pins του Raspberry Pi και των Pins του Arduino. Η σύνδεση γίνεται στα pins RX,TX. Καθώς η πλακέτα του Arduino χρησιμοποιεί 5V χρειάζεται κύκλωμα μετατροπής της τάσης σε 3.3V για την προστασία του Raspberry pi όπως φαίνεται στην εικόνα 5-14[67]



Εικόνα 5- 14: Σύνδεση μέσω των ακροδεκτών RX/TX

Στα πλαίσια ανάπτυξης επιλέχθηκε η μέθοδος της θύρας USB καθώς είναι πιο απλή και δεν απαιτεί επιπλέον κυκλώματα.

Η επικοινωνία ανάμεσα στις δύο συσκευές γίνεται μέσω του πρωτοκόλλου UART(Universal Asynchronous Receiver/Transmitter). Η δομή των δεδομένων είναι κοινή ανάμεσα στις δύο συσκευές. Στην ασύγχρονη επικοινωνία το πακέτο έχει την παρακάτω βασική δομή.[28]



**Εικόνα 5- 15: Η δομή του πακέτου της ασύγχρονης επικοινωνίας**

- bit εκκίνησης (START bit): Το bit αυτό είναι πάντα 0 και υποδηλώνει ότι η σειριακή επικοινωνία έχει αρχίσει
- Bit δεδομένων (Data bit packet): Συνήθως είναι πακέτα από 8 bit (η τιμή κυμαίνεται από 5 έως 9), τα οποία αρχίζουν αμέσως μετά το bit εκκίνησης
- Bit τερματισμού (END bit): Είναι συνήθως ένα με δύο bit και αποστέλλεται μετά τα δεδομένα για να υποδηλώσει το τέλος του πακέτου. Πάντα είναι το/τα bit τερματισμού είναι 1.

Συνοπτικά το πλήρες πακέτο αποτελείται από ένα bit εκκίνησης τα δεδομένα (συνήθως 8 bit) και το bit τερματισμού (ένα ή δύο bit). Προαιρετικά μπορεί να υπάρχουν και bit ισοτιμίας για τον εντοπισμό λαθών.[28]

### 5.7: Λοιπός υλικός εξοπλισμός

Στην ενότητα αυτή γίνεται μια βασική αναφορά στον υλικό εξοπλισμό του βραχίονα ο οποίος δεν θα αναλυθεί σε βάθος καθώς αποτελεί μέρος της συγκεκριμένης εργασίας. Τα εξαρτήματα είναι τα εξής:

- Η κάμερα
- Οι αισθητήρες κίνησης (προαιρετικά)
- Οι σερβοκινητήρες
- Ο κωδικοποιητής

Στα πλαίσια της συγκεκριμένης εργασίας εμπίπτει μόνο η κάμερα καθώς μέσω αυτής ο μικροϋπολογιστής εντοπίζει τις περιοχές ενδιαφέροντος. Οι αισθητήρες κίνησης, οι σερβοκινητήρες και ο κωδικοποιητής ελέγχονται από τον μικροελεγκτή Arduino.

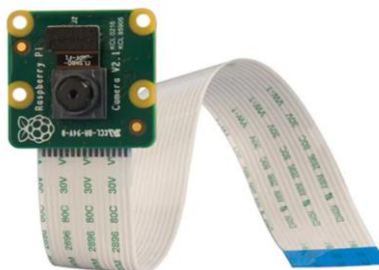
#### 5.7.1: Η κάμερα

Η κάμερα που έχει χρησιμοποιηθεί είναι η Raspberry Pi Camera Module V2.1. στον παρακάτω πίνακα διακρίνονται τα τεχνικά χαρακτηριστικά της κάμερας. Οι λόγοι επιλογής της συγκεκριμένης κάμερας είναι το χαμηλό κόστος καθώς και η ευκολία τοποθέτησης της στον βραχίονα λόγω διαστάσεων και βάρους. Στον παρακάτω πίνακα διακρίνονται τα τεχνικά χαρακτηριστικά της κάμερας.[29]

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Ανάλυση εικόνων	8 Megapixel, 3280x2464 pixel
Ανάλυση βίντεο	640x480p90, 720p60, 1080p30
Διαστάσεις	25mm x 23mm x9mm
Βάρος	~3g
Διεπαφή	CSI Interface

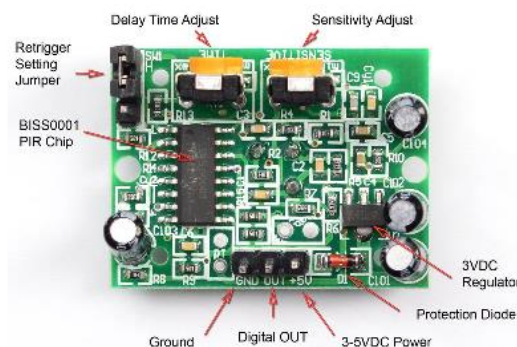
Πίνακας 5- 6: Τεχνικά χαρακτηριστικά της κάμερας



Εικόνα 5- 16: Η κάμερα για το Raspberry Pi

### 5.7.2: Οι αισθητήρες κίνησης

Οι αισθητήρες κίνησης είναι οι HC-SR501 PIR Motion Sensors. Είναι συνδεδεμένοι με το Arduino και μέσω αυτών γίνεται παρακολούθηση του χώρου όταν δεν υπάρχει αντικείμενο ενδιαφέροντος στο πεδίο λήψης της κάμερας. Υπάρχουν τέσσερις αισθητήρες κίνησης σε γωνία ενενήντα μοιρών. Σε περίπτωση που δεν υπάρχει αντικείμενο ενδιαφέροντος τότε γίνεται επιτήρηση. Μόλις εντοπιστεί κίνηση, ο βραχίονας στρέφεται προς τον αισθητήρα που εντόπισε την κίνηση με σκοπό την λήψη δεδομένων από την κάμερα[30]



Εικόνα 5- 17: Ο αισθητήρας κίνησης HC-SR501

### 5.7.3: Οι σερβοκινητήρες

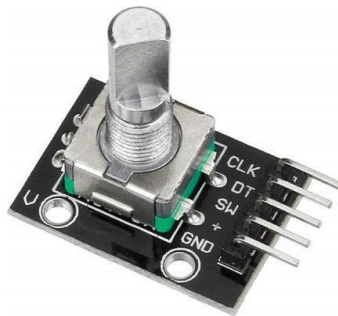
Για την κίνηση του βραχίονα χρησιμοποιούνται σερβοκινητήρες. Για την περιστροφή του βραχίονα μπορεί να χρησιμοποιηθεί ο DS04-NFC 360 Degrees. Για τις υπόλοιπες κινήσεις του βραχίονα πέρα από την περιστροφή μια καλή επιλογή είναι οι F SG90[31,32]



**Εικόνα 5- 18: Ενδεικτικοί σερβοκινητήρες**

#### **5.7.4: Ο κωδικοποιητής**

Η κύρια δουλειά του κωδικοποιητή είναι να καταγράφει την θέση του βραχίονα. Με αυτόν τον τρόπο όταν ένας αισθητήρας εντοπίσει κίνηση ο βραχίονας ξέρει προς ποια κατεύθυνση να στραφεί. Μια καλή επιλογή είναι ο KY-040 Rotary Encoder που παρέχει 30 θέσεις. Οι αλλαγές καταγράφονται από το Arduino και αποστέλλονται στο Raspberry Pi και έτσι διατηρείται η θέση του βραχίονα.[33]



**Εικόνα 5- 19: KY-040 Rotary Encoder**

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## Κεφάλαιο 6: Ανάλυση της εφαρμογής

### 6.1: Ανάλυση πακέτων και προγραμμάτων που χρησιμοποιήθηκαν

#### 6.1.1: Εισαγωγή:

Η ορθή επιλογή της βιβλιοθήκης είναι κρίσιμη για την σωστή λειτουργία της εφαρμογής. Υπάρχουν αρκετά προγράμματα και βιβλιοθήκες που μπορούν να πετύχουν τον σκοπό αυτό όπως το openCV, το simpleCV και το MATLAB. Κάθε πακέτο έχει πλεονεκτήματα και μειονεκτήματα με αποτέλεσμα να χρειάζεται ανάλυση του κάθε πακέτου[34]

#### 6.1.2: Το πρόγραμμα MATLAB

Αρχικά το MATLAB είναι μια προγραμματιστική πλατφόρμα για πληθώρα εφαρμογών, από τεχνητή νοημοσύνη μέχρι και επεξεργασία σημάτων. Υπάρχει και βιβλιοθήκη για συστήματα υπολογιστικής όρασης. Όσον αφορά τα πλεονεκτήματα καθώς είναι γλώσσα προγραμματισμού η συγγραφή κώδικα είναι εύκολη. Επίσης μέσα στην πλατφόρμα υπάρχει ένα σύνολο εργαλείων αποσφαλμάτωσης. Ένα επίσης μεγάλο πλεονέκτημα είναι ότι όλες οι συναρτήσεις είναι πολύ καλά τεκμηριωμένες. Στα μειονεκτήματα το MATLAB μπορεί να χρησιμοποιηθεί μόνο επί πληρωμή. Επίσης είναι σχετικά αργό για εφαρμογές πραγματικού χρόνου και δεν είναι εύκολο να ενσωματωθεί με εξωτερικά προγράμματα.[34]

#### 6.1.3: η βιβλιοθήκη simpleCV:

Το simpleCV είναι ένα σύνολο βιβλιοθηκών που επιτρέπει την ανάπτυξη εφαρμογών υπολογιστικής όρασης. Υψηλού επιπέδου χωρίς να χρειάζεται γνώση σε βάθος πολύπλοκων εννοιών όπως μορφές αρχείων ή χρωματικούς χώρους. Στα πλεονεκτήματα της βιβλιοθήκης είναι ότι είναι δωρεάν και οι αλγόριθμοι είναι βελτιστοποιημένοι σε πολύ μεγάλο βαθμό. Τέλος παρέχεται καλή τεκμηρίωση των αλγορίθμων. Στα μειονεκτήματα δεν υποστηρίζει άλλη γλώσσα πέρα από την python με αποτέλεσμα οι εφαρμογές της να είναι λίγο πιο περιορισμένες.[34]

#### 6.1.4: η βιβλιοθήκη openCV:

Το OpenCV (Open Source Computer Vision) είναι μια βιβλιοθήκη με ευρεία χρήση σε εφαρμογές όρασης υπολογιστών(computer vision) καθώς και σε επεξεργασία εικόνων και βίντεο με ευρεία χρήση από ερευνητικές ομάδες και εταιρίες. Η openCV είναι δωρεάν και είναι open source που σημαίνει ότι όποιος θέλει μπορεί να τροποποιήσει στοιχεία της βιβλιοθήκης για να εξυπηρετήσει συγκεκριμένους σκοπούς. Ένα άλλο μεγάλο πλεονέκτημα είναι ότι είναι ανεξάρτητη λειτουργικού συστήματος και υπάρχει μεγάλη υποστήριξη από κοινότητες. Ένα πρόβλημα είναι ότι δεν παρέχεται αποσφαλμάτωση και χρειάζεται χρόνος προκειμένου να μάθει κάποιος εφαρμογές με openCV.[34]

### 6.1.5: Επιλογή γλώσσας προγραμματισμού

Το Raspberry Pi υποστηρίζει αρκετές γλώσσες προγραμματισμού. Για την ανάπτυξη του συστήματος χρειάζεται μια γλώσσα προγραμματισμού η οποία να είναι εύκολη στην σύνταξη και να μπορεί να ενσωματώσει την βιβλιοθήκη της υπολογιστικής όρασης και σειριακής επικοινωνίας.

Αυτά τα κριτήρια τα πληροί η γλώσσα προγραμματισμού Python. Η Python είναι γλώσσα υψηλού επιπέδου και είναι σχεδιασμένη να είναι εύκολη στην ανάγνωση, σύνταξη και συγγραφή.

Οι κύριοι λόγοι επιλογής της Python είναι η ευελιξία που παρέχει. Ενδεικτικά υποστηρίζει συναρτησιακό προγραμματισμό και αντικειμενοστραφή προγραμματισμό και βιβλιοθήκη σειριακής επικοινωνίας, στοιχεία τα οποία είναι σημαντικά για την ανάπτυξη της εφαρμογής.



Εικόνα 6- 1: Η γλώσσα προγραμματισμού python

### 6.1.6: Διαχείριση βάσεων δεδομένων

Μια βάση δεδομένων είναι μια οργανωμένη δομή δεδομένων για αποθήκευση πληροφοριών. Η πιο συνηθισμένη δομή αποθήκευσης είναι οι πίνακες και οι πίνακες αποτελούνται από πεδία. Μια γνωστή “γλώσσα” για την διαχείριση βάσεων δεδομένων ονομάζεται SQL(Structured Query Language).

Το MySQL είναι ένα περιβάλλον που δίνει την δυνατότητα διαχείρισης σχεσιακών βάσεων δεδομένων. Είναι ανοιχτού κώδικα (GNU Public License) ανεξαρτήτως λειτουργικού συστήματος. Επιτρέπει στους χρήστες να εργάζονται στην βάση δεδομένων, να ελέγχουν την ακεραιότητα των δεδομένων και να παίρνουν αντίγραφα της βάσης.



Εικόνα 6- 2: Το λογότυπο της MySQL



## 6.2: Ανάλυση της εφαρμογής

Ο κώδικας της εφαρμογής για τον εντοπισμό περιοχών ενδιαφέροντος και την επικοινωνία με τον μικροελεγκτή έχει γραφτεί σε γλώσσα Python. Η ανίχνευση των περιοχών ενδιαφέροντος γίνεται μέσω της βιβλιοθήκης OpenCV. Η αποθήκευση των ονομάτων και των παραμέτρων αποθηκεύονται σε βάση δεδομένων των οποίων η διαχείριση γίνεται με MySQL. Όλα τα δεδομένα βρίσκονται σε έναν φάκελο με όνομα Application.

Μέσα στον φάκελο βρίσκονται όλα τα απαραίτητα αρχεία για την επιτυχή αναγνώριση των περιοχών ενδιαφέροντος και την εκπαίδευση του συστήματος. Ο φάκελος περιέχει τα εξής:

- Τον φάκελο Enhanced όπου παράγονται τα βίντεο μετά την διαδικασία εντοπισμού αντικειμένων. Υπάρχουν τρεις φάκελοι όπου αποθηκεύονται τα βίντεο ανάλογα με το είδος τους.
- Τον φάκελο FaceRecognizer όπου αποθηκεύεται το αρχείο που παράγεται από την διαδικασία εκπαίδευσης.
- Τον φάκελο PhotosRecognizer όπου αποθηκεύονται οι φωτογραφίες με τις οποίες θα γίνει η εκπαίδευση του συστήματος
- Τον φάκελο Unknown όπου αποθηκεύονται τα βίντεο ανάλογα με το είδος τους σε τέσσερις φακέλους
  - Στον φάκελο Faces αποθηκεύονται τα βίντεο που περιέχουν εντοπισμούς άγνωστων προσώπων
  - Στον φάκελο UpperBody αποθηκεύονται τα βίντεο που περιέχουν εντοπισμούς άνω μέρους σώματος
  - Στον φάκελο FullBody αποθηκεύονται τα βίντεο που περιέχουν εντοπισμούς πλήρους σώματος
  - Στον φάκελο Movement αποθηκεύονται τα βίντεο που περιέχουν εντοπισμούς κινήσεων
- Το αρχείο EncoderPos.txt περιέχει την τελευταία αποθηκευμένη θέση του βραχίονα
- Τους haar cascade classifiers για τον εντοπισμό των περιοχών ενδιαφέροντος και συγκεκριμένα
  - Haarcascade\_frontalface\_default.xml
  - Haarcascade\_fullbody.xml
  - haarcascade\_upperbody.xml

## 6.3: Το σχήμα της βάσης δεδομένων

Για την αποθήκευση των αναγνωριστικών, ονομάτων των χρηστών καθώς και των λοιπών παραμέτρων χρησιμοποιείται βάση δεδομένων MySQL. Η βάση περιέχει δύο πίνακες που ονομάζονται user και offset.

Ο πίνακας face περιέχει δύο πεδία, το αναγνωριστικό(id) και το όνομα κάθε χρήστη που το σύστημα αναγνωρίζει

User					
Field	Type	NULL	KEY	Default	Extra
User_id	Int(11)	NO	PRI		
User_name	Varchar(50)	YES		NULL	

Πίνακας 6- 1: Περιγραφή του πίνακα User

Ο πίνακας offset περιέχει τις συντεταγμένες με βάση τις οποίες χωρίζεται η εικόνα σε πλαίσια με σκοπό την επιλογή της κατάλληλης κίνησης του βραχίονα. Η περιγραφή του πίνακα αναλύεται στον παρακάτω πίνακα

Offset					
Field	Type	NULL	Key	Default	Extra
Offset_minx	Int(11)	YES		NULL	
Offset_maxx	Int(11)	YES		NULL	
Offset_miny	Int(11)	YES		NULL	
Offset_maxy	Int(11)	YES		NULL	

Πίνακας 6- 2: περιγραφή του πίνακα

#### 6.4: Εκτέλεση της εφαρμογής

Για την εκτέλεση του κώδικα της εφαρμογής έχει χρησιμοποιηθεί το περιβάλλον Python IDLE το οποίο παρέχει κειμενογράφο και shell για την εκτέλεση των προγραμμάτων. Κατά την εκκίνηση του προγράμματος αρχικά φορτώνονται οι βιβλιοθήκες και οι καθολικές μεταβλητές

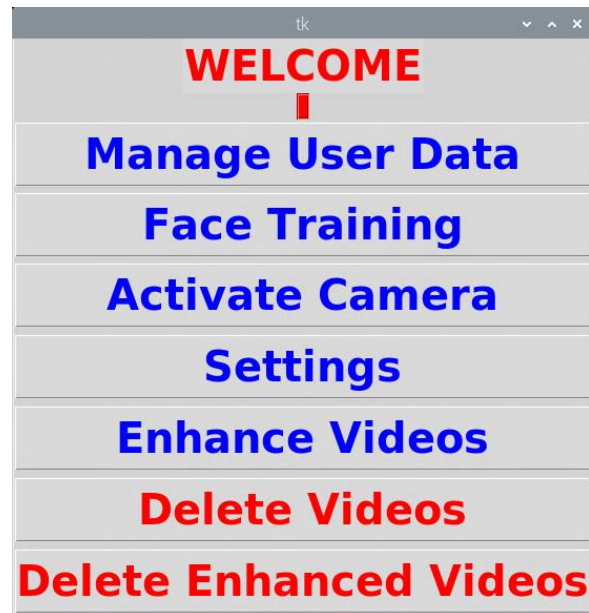
```
1 # algorithm for face recognition and movement detection using opencv and python
2 # ALL RIGHTS RESERVED
3
4 #<import all packages>
5 import cv2
6 import numpy as np
7 from tkinter import *
8 from tkinter import messagebox
9 from tkinter import filedialog
10 import pymysql
11 import os
12 import shutil
13 from PIL import Image,ImageTk
14 from datetime import datetime
15 from datetime import date
16 from time import sleep
17 import time
18 import datetime
19 import serial
20 import keyboard
21 import math
22 #</import all packages>
23
24 #DB Data: |face_test
25 #table user |user_id int (PK)|user_name varchar(50)
26 #table offset |offset_minx int|offset_maxx int|offset_miny int|offset_maxy int
27
28 #<Global variables for sql,cascade classifiers and camera>
29 conn = pymysql.connect(host = 'localhost',user = 'root',password = '1234',db = 'face_test')
30 sqlCursor = conn.cursor()
```

```
31
32 faceDetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
33 ubodyDetect = cv2.CascadeClassifier('haarcascade_upperbody.xml')
34 fbodyDetect = cv2.CascadeClassifier('haarcascade_fullbody.xml')
35
36 if __name__ == '__main__':
37     ser = serial.Serial('/dev/ttyACM0',9600,timeout = 1)
38
39 current_position = 0
40 #</Global variables for sql,cascade classifiers and camera>
```

Αφού ολοκληρωθεί η παραπάνω διαδικασία το επόμενο βήμα είναι η ενημέρωση του μικροελεγκτή για με την τελευταία αποθηκευμένη θέση του βραχίονα. Ανοίγεται το αρχείο EncoderPos.txt και ανακτάται η τελευταία τιμή. Αυτή η τιμή αποστέλλεται στον μικροελεγκτή ο οποίος απαντά ότι την έλαβε μέσω του χαρακτήρα 'K'. Οι χαρακτήρες με κίτρινο χρώμα στην παρακάτω εικόνα είναι σχόλια και είχαν χρησιμοποιηθεί κατά την διάρκεια των ελέγχων και της αποσφαλμάτωσης.

```
1581 rcvd = False
1582 position_file = open('EncoderPos.txt')
1583
1584 content = position_file.read().splitlines()
1585
1586 file_line = ""
1587
1588
1589 for line in content:
1590     file_line = str(line)+'/'
1591
1592 #print(file_line)
1593
1594 while rcvd is False:
1595     print("Sending->" +str(file_line))
1596     #ser.write(file_line.encode())
1597     #rec_line = #ser.readline().decode('utf-8').rstrip()
1598     #print(rec_line)
1599     if(rec_line == 'K'):
1600         ##ser.write(b'OOK')
1601         rcvd = True
1602     print("All Good")
1603     position_file.close()
1604
1605 ""
1606 while True:
1607     position_file = open('EncoderPos.txt','a')
1608     rec_line = #ser.readline().decode('utf-8').rstrip()
1609     print(rec_line)
1610     position_file.write(str(rec_line))
1611     position_file.write("\n")
1612     position_file.close()
1613 ""
```

Μόλις ολοκληρωθεί η ενημέρωση εμφανίζεται το αρχικό μενού της εφαρμογής με τις λειτουργίες της εφαρμογής. Οι λειτουργίες αφορούν την διαχείριση των δεδομένων χρηστών, την εκπαίδευση του συστήματος, την παρακολούθηση του χώρου, τις ρυθμίσεις και την διαγραφή των αποθηκευμένων βίντεο.



Εικόνα 6- 3: Το αρχικό μενού της εφαρμογής

Ακολουθεί ο κώδικας για την εμφάνιση του μενού

```
1614
1615 root = Tk()
1616 root.geometry("550x550")
1617 root.configure(bg = "Light Gray")
1618 Label1 = Label(root,text = "WELCOME",font = ("",30,"bold"),fg = "RED")
1619 Label1.pack()
1620 f1 = LabelFrame(root,bg = "red")
1621 f1.pack()
1622 L1 = Label(f1,bg = "red")
1623 L1.pack()
1624 Button_MRD = Button(root,text = "Manage User Data",font = ("",30,"bold"),fg =
"Blue",command = ManageRecognizerData)
1625 Button_MRD.pack(fill=X,expand=True)
1626 Button_TRN = Button(root,text = "Face Training",font = ("",30,"bold"),fg =
"Blue",command = LBPHFaceTrain)
1627 Button_TRN.pack(fill=X,expand=True)
1628 Button_CMR = Button(root,text = "Activate Camera",font = ("",30,"bold"),fg =
"Blue",command = ActivateCamera)
1629 Button_CMR.pack(fill=X,expand=True)
```

```
1630 Button_STG = Button(root,text = "Settings",font = ("",30,"bold"),fg = "Blue",command
= Settings)
1631 Button_STG.pack(fill=X,expand=True)
1632 Button_ENH = Button(root,text = "Enhance Videos",font = ("",30,"bold"),fg =
"Blue",command = EnhanceVideos)
1633 Button_ENH.pack(fill=X,expand=True)
1634 Button_DLT = Button(root,text = "Delete Videos",font = ("",30,"bold"),fg =
"Red",command = hardResetVideos)
1635 Button_DLT.pack(fill=X,expand=True)
1636 Button_EL = Button(root,text = "Delete Enhanced Videos",font = ("",30,"bold"),fg =
"Red",command = hardResetEnVideos)
1637 Button_EL.pack(fill=X,expand=True)
1638
1639 #EnhanceVideos
1640
1641#####</MAIN
GUI>#####
```

#### 6.4.1: Διαχείριση δεδομένων χρηστών

Με το πλήκτρο Manage User Data καλείται η συνάρτηση ManageRecognizerData στην γραμμή 1624 και εμφανίζεται το παράθυρο για την διαχείριση των έμπιστων χρηστών της εφαρμογής.

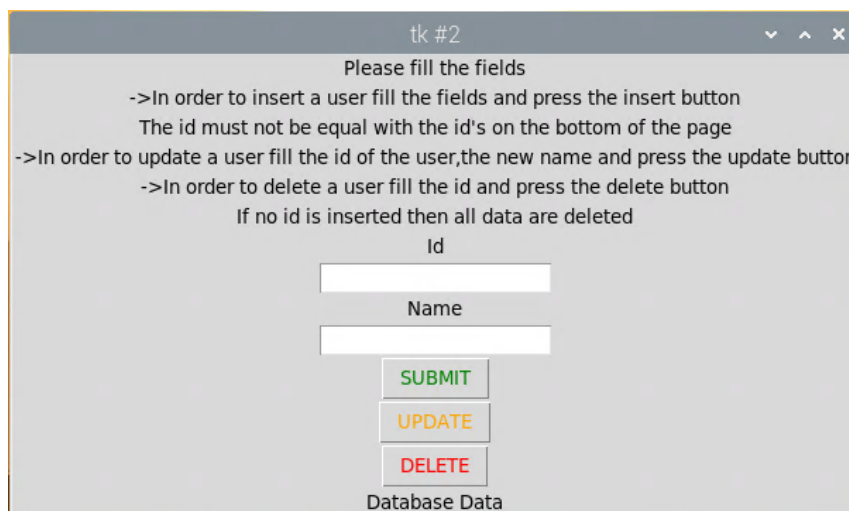
Σε αυτό το παράθυρο είναι δυνατή η εισαγωγή, τροποποίηση και διαγραφή των στοιχείων των ατόμων. Η εισαγωγή και η διαγραφή δεδομένων αφορούν τα στοιχεία του πίνακα user καθώς και τις φωτογραφίες στο φάκελο PhotosRecognizer. Η ενημέρωση περιορίζεται μόνο στα στοιχεία της βάσης δεδομένων. Στο πάνω μέρος του παραθύρου υπάρχουν οδηγίες για την ορθή χρήση του παραθύρου. Στην μέση βρίσκονται τα πεδία προς συμπλήρωση και τα πλήκτρα των λειτουργιών. Στο κάτω μέρος του παραθύρου εμφανίζονται τα ήδη υπάρχοντα στοιχεία στην βάση δεδομένων.

Στην παρακάτω εικόνα διακρίνεται ο κώδικας για την εμφάνιση του παραθύρου. Τα κομμάτια του κώδικα που βρίσκονται ανάμεσα στις τρεις τελείες θα αναλυθούν στις επόμενες υποπαραγράφους της ενότητας.

```
42 #####<MANAGE RECOGNIZER DATA>#####
43 def ManageRecognizerData():
44     camera = cv2.VideoCapture(0)
45     ...
188
189 #<User Data Panel>
190 form = Tk()
191 lowPartForm = Frame(form)
192 lowPartForm.pack(side = BOTTOM)
193 basicInfoLabel = Label(form,text = "Please fill the fields")
194 basicInfoLabel.pack()
195 InfoInsertLabel1 = Label(form,text = "->In order to insert a user fill the fields and
press the insert button")
196 InfoInsertLabel1 .pack()
```

```
197 InfoInsertLabel2 = Label(form,text = "The id must not be equal with the id's on the
bottom of the page")
198 InfoInsertLabel2.pack()
199 InfoUpdateLabel1 = Label(form,text = "->In order to update a user fill the id of the
user,the new name and press the update button")
200 InfoUpdateLabel1.pack()
201 InfoDeleteLabel1 = Label(form,text = "->In order to delete a user fill the id and press
the delete button")
202 InfoDeleteLabel1.pack()
203 InfoDeleteLabel2 = Label(form,text = "If no id is inserted then all data are deleted")
204 InfoDeleteLabel2.pack()
205 labelId = Label(form,text = "Id",width = 20,font = ("bold",10))
206 labelId.pack()
207 entryId = Entry(form)
208 entryId.pack()
209 labelName = Label(form,text = "Name",width = 20,font = ("bold",10))
210 labelName.pack()
211 entryName = Entry(form)
212 entryName.pack()
213 buttonSubmit = Button(form,text = "SUBMIT",fg = "GREEN",command =
formSubmit)
214 buttonSubmit.pack()
215 buttonUpdate = Button(form,text = "UPDATE",fg = "ORANGE",command =
formUpdate)
216 buttonUpdate.pack()
217 buttonDelete = Button(form,text = "DELETE",fg = "RED",command = formDelete)
218 buttonDelete.pack()
219 #<Show Database Data>
220 ifnoDBLabel = Label(lowPartForm,text = "Database Data")
221 ifnoDBLabel.pack()
222 storedDataDB = "SELECT * FROM user"
223 sqlCursor.execute(storedDataDB)
224 storedRecords = sqlCursor.execute(storedDataDB)
225 storedData = sqlCursor.fetchall()
226 for storedData2Display in storedData:
227     labelData = Label(lowPartForm,text = str(storedData2Display[0])+
'+str(storedData2Display[1]))
228     labelData.pack()
229 #</Show Database Data>
230 #</user Data panel>
231 #####<MANAGE RECOGNIZER DATA>#####
```

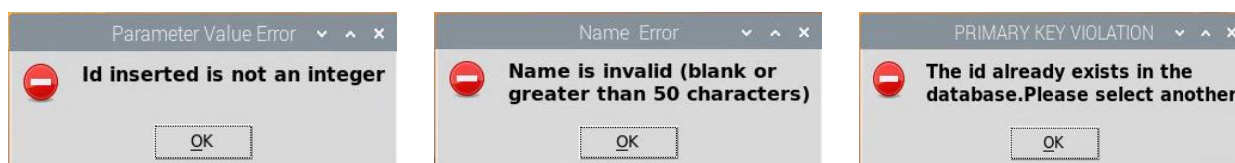
Το παράθυρο για την διαχείρισης χρηστών περιέχει στο πάνω μέρος οδηγίες για την ορθή συμπλήρωση των πεδίων, στην μέση έχει τα δύο πεδία προς συμπλήρωση και στο κάτω μέρος έχει τα πλήκτρα για τις λειτουργίες εισαγωγής, διαγραφής ή ενημέρωσης χρήστη. Κάτω από τα πλήκτρα εμφανίζονται τα ήδη αποθηκευμένα στοιχεία χρηστών.



Εικόνα 6- 4: Το παράθυρο διαχείρισης χρηστών

#### 6.4.1.1: Εισαγωγή χρήστη

Για την εισαγωγή ενός νέου προσώπου εισάγονται τα πεδία του αναγνωριστικού και του ονόματος. Με το πλήκτρο SUBMIT ξεκινάει η διαδικασία εισαγωγής των στοιχείων. Αρχικά γίνονται δύο έλεγχοι στο αναγνωριστικό. Ο πρώτος αφορά την εγκυρότητα ως προς το είδος της μεταβλητής και συγκεκριμένα ότι είναι ακέραιος αριθμός. Ο δεύτερος έλεγχος αφορά τον έλεγχο παραβίασης του κύριου κλειδιού στην βάση δεδομένων. Ο επόμενος έλεγχος είναι στο μήκος του ονόματος και συγκεκριμένα ότι είναι μικρότερο από πενήντα χαρακτήρες. Σε περίπτωση που κάποιος έλεγχος αποτύχει εμφανίζεται το αντίστοιχο μήνυμα λάθους



Εικόνα 6- 5: Μηνύματα λάθους κατά την εισαγωγή χρήστη

Αν όλοι οι έλεγχοι είναι επιτυχείς τα στοιχεία εισάγονται στην βάση δεδομένων. Στην συνέχεια ενεργοποιείται η κάμερα και εισάγονται πενήντα φωτογραφίες με το πρόσωπο που έχει ένα κόκκινο παραλληλόγραμμο (αν δεν βρεθεί πρόσωπο το πρόγραμμα αναμένει να βρεθεί). Οι φωτογραφίες μετατρέπονται σε μονοχρωματική κλίμακα και αποθηκεύονται στο φάκελο PhotosRecognizer.



```

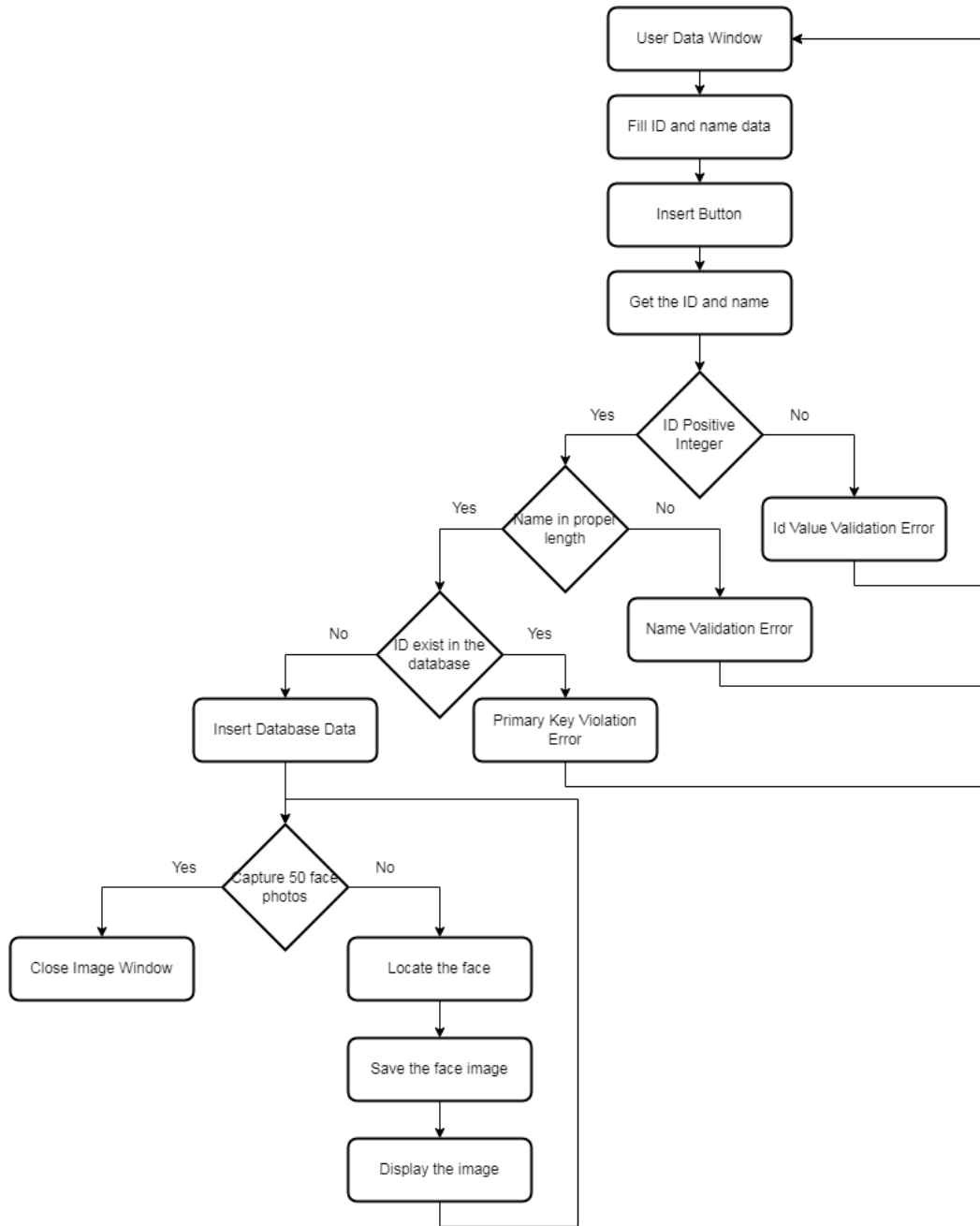
46 def formSubmit():
47     idnum = entryId.get()
48     name = entryName.get()
49     #<check if the id is integer and greater than zero>
50     try:
51         intidnum = int(idnum)
52         if(intidnum <= 0):
53             messagebox.showerror("Parameter Value Error","The id is less than zero")
54             return
55     except ValueError:
56         messagebox.showerror("Parameter Value Error","The id is not an integer")
57         return
58     #<check if the id is integer and greater than zero>
59     #<Check if the name is valid not blank and less than 50 characters>
60     if(len(name) > 50 or len(name) == 0):
61         messagebox.showerror("Name Error","Name is invalid (blank or greater than 50
characters)")
62         return
63     #</Check if the name is valid not blank and less than 50 characters>
64     #<check if the id exists>
65     checkQuery = "SELECT COUNT(*) FROM user WHERE user_id = %s"
66     sqlCursor.execute(checkQuery,intidnum)
67     id2check = sqlCursor.fetchone()
68     if(id2check[0] > 0):
69         messagebox.showerror("PRIMARY KEY VIOLATION","The id already exists in
the database")
70         return
71     #</check if the id exists>
72     #<Insert the id and the name in the database>
73     insertQuery = "INSERT INTO user (user_id,user_name) VALUES(%s,%s)"
74     queryParameters = (intidnum,name)
75     sqlCursor.execute(insertQuery,queryParameters)
76     #</Insert the id and the name in the database>
77     #<Capture the 30 face images and insert them in the folder Photosrecognizer>
78     photoIndex = 0
79     while(photoIndex < 50):
80         return_value,image = camera.read()
81         grayImage = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
82         face =
faceDetect.detectMultiScale(grayImage,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),fl
ags=cv2.CASCADE_SCALE_IMAGE)
83         for(point1,point2,width,height) in face:
84             photoIndex += 1
85         cv2.putText(image,str(photoIndex),(point1,point2),cv2.FONT_HERSHEY_SIMPLEX,1.2,(
55,255,255),2)
86         cv2.rectangle(image,(point1,point2),(point1+width,point2+height),(0,0,255),2)
87         cv2.imwrite("PhotosRecognizer/"+str(idnum)+"."+str(photoIndex)+".jpg",grayImage[point2:
point2+height,point1:point1+width])
88         cv2.waitKey(100)

```



```

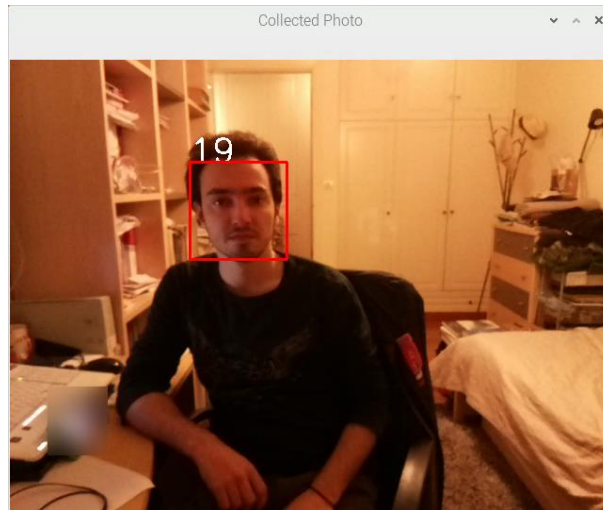
89     cv2.imshow("Collected Photo",image)
90     cv2.waitKey(1)
91     cv2.destroyAllWindows()
92     #<Capture the 30 face images and insert them in the folder Photosrecognizer>
93     #<commit the insert into the database>
94     conn.commit()
95     #</commit the insert into the database>
96     #</fromSubmit>
    
```



Εικόνα 6- 6: Το διάγραμμα ροής για την εισαγωγή χρήστη

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Κατά την συλλογή των φωτογραφιών του έμπιστου χρήστη εμφανίζεται σε ένα κόκκινο παραλληλόγραμμο το πρόσωπο που έχει εντοπιστεί καθώς και ο αριθμός της τρέχουσας φωτογραφίας.



Εικόνα 6- 7: Στιγμιότυπο από την εισαγωγή χρήστη

#### 6.4.1.2: Ενημέρωση στοιχείων χρήστη

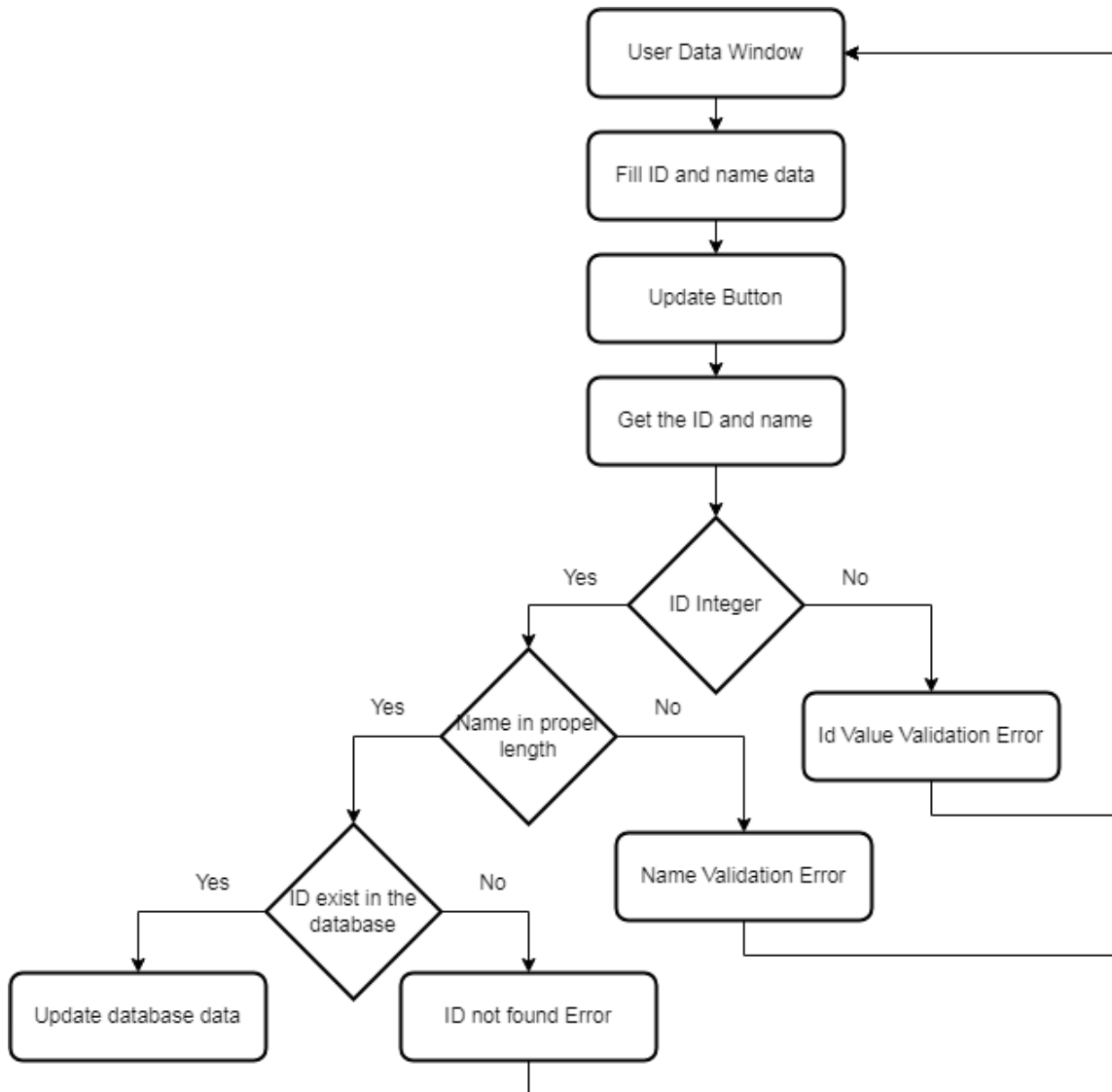
Για την ενημέρωση του ονόματος ενός χρήστη εισάγεται το αναγνωριστικό του υπάρχοντος χρήστη και το νέο όνομα. Με το πλήκτρο update αρχικά γίνεται έλεγχος αν το αναγνωριστικό είναι έγκυρος αριθμός. Στην συνέχεια ελέγχεται αν το αναγνωριστικό υπάρχει στην βάση δεδομένων. Ο τελευταίος έλεγχος αφορά το όνομα και συγκεκριμένα να μην είναι κενό η μεγαλύτερο από πενήντα χαρακτήρες. Εφόσον όλοι οι έλεγχοι είναι επιτυχείς πραγματοποιείται η αλλαγή του ονόματος χρήστη. Σε περίπτωση που κάποιος έλεγχος αποτύχει εμφανίζεται το αντίστοιχο μήνυμα λάθους.



Εικόνα 6- 8: Μηνύματα λάθους κατά την ενημέρωση χρήστη

Στην συνέχεια ακολουθούν ο κώδικας καθώς και το διάγραμμα ροής για την ενημέρωση των στοιχείων

```
98 #</form Update>
99 def formUpdate():
100     updateId = entryId.get()
101     updateName = entryName.get()
102     #<check if the id is an integer>
103     try:
104         intidnum = int(updateId)
105     except ValueError:
106         messagebox.showerror("Parameter type Error","The id is not an integer")
107     return
108 #</check if the id is an integer>
109 #<check if the name is valid not blank and less than 50 characters>
110 if(len(updateName) > 50 or len(updateName) == 0):
111     messagebox.showerror("Name Invalid","The name is invalid (blank or greater
than 50 characters)")
112     return
113 #<check if the name is valid not blank and less than 50 characters>
114 #<Check if the id exists>
115 checkQuery = "SELECT COUNT(*) FROM user WHERE user_id = (%s)"
116 sqlCursor.execute(checkQuery,intidnum)
117 check = sqlCursor.fetchone()
118 if(check[0] == 0):
119     messagebox.showwarning("Parameter Warning","The id does not exist")
120     return
121 #</Check if the id exists>
122 #<perform the update>
123 updateQuery = "UPDATE user SET user_name = %s WHERE user_id = %s"
124 parameters =(str(updateName),intidnum)
125 sqlCursor.execute(updateQuery,parameters)
126 conn.commit()
127 #</perform the update>
128 #</form Update>
129
```



Εικόνα 6- 9: Διάγραμμα ροής για την ενημέρωση στοιχείων χρήστη

### 6.4.1.3: Διαγραφή στοιχείων χρήστη

Για την διαγραφή ενός χρήστη το μόνο που απαιτείται είναι το αναγνωριστικό του χρήστη. Μετά την εισαγωγή του αναγνωριστικού και την επιβεβαίωση με το πλήκτρο Delete ξεκινάει η διαδικασία διαγραφής του χρήστη. Αρχικά ελέγχεται η εγκυρότητα του αναγνωριστικού, αν είναι ακέραιος αριθμός και αν υπάρχει στην βάση δεδομένων. Μετά την επιτυχή ολοκλήρωση των ελέγχων διαγράφονται τα στοιχεία από την βάση δεδομένων και στην συνέχεια διαγράφονται οι φωτογραφίες του χρήστη. Στην συνέχεια πραγματοποιείται η εκπαίδευση με τις υπόλοιπες φωτογραφίες.

Αν δεν εισαχθεί αναγνωριστικό τότε διαγράφονται όλα τα δεδομένα από την βάση δεδομένων, όλες οι φωτογραφίες καθώς και το αρχείο με τα δεδομένα από την εκπαίδευση των χρηστών.

Αν κάποιος έλεγχος δεν είναι επιτυχής εμφανίζεται το αντίστοιχο μήνυμα λάθους



Εικόνα 6- 10: Τα μηνύματα λάθους της διαδικασίας διαγραφής

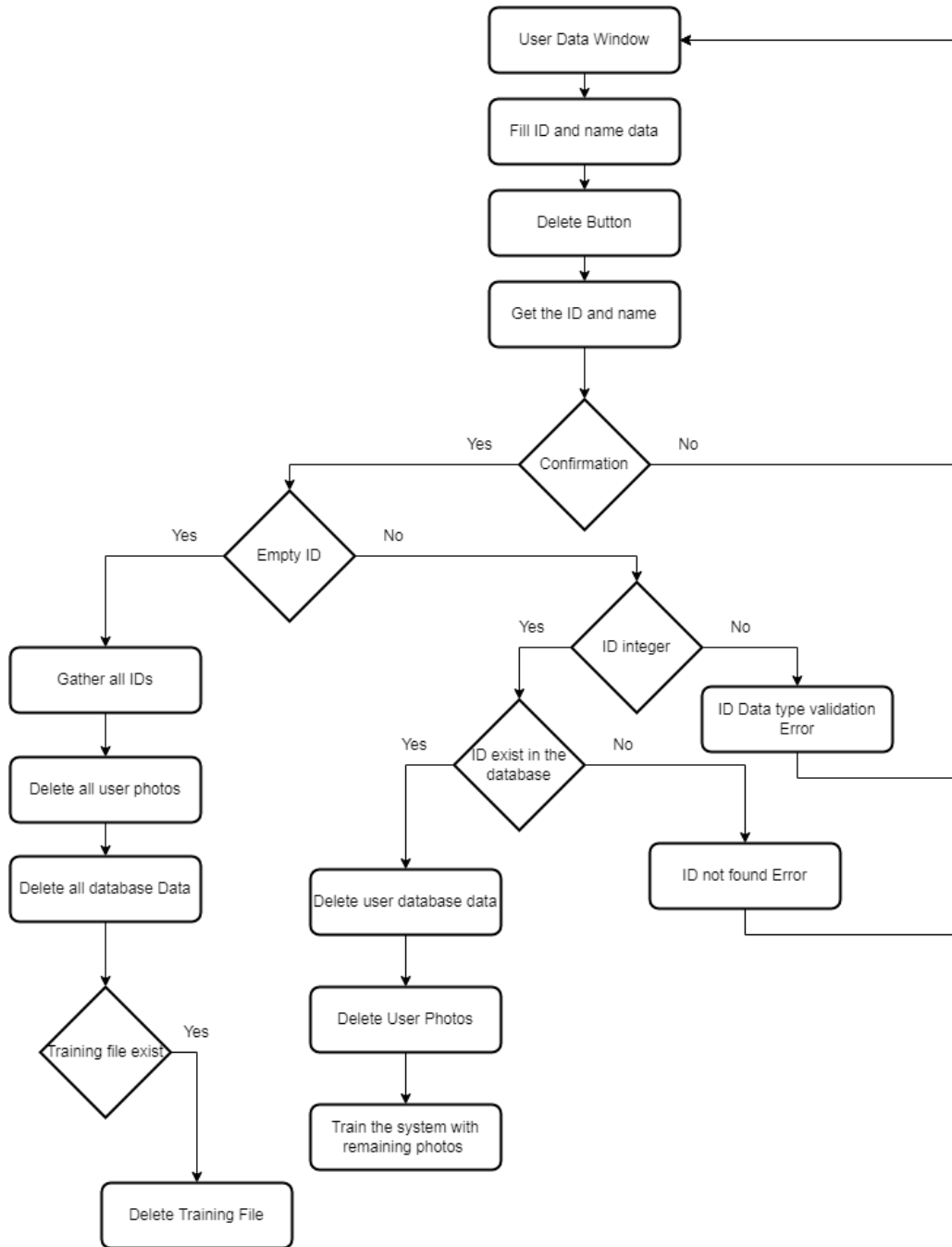
Ακολουθεί ο κώδικας για την διαγραφή των δεδομένων

```
130 #<formDelete>
131 def formDelete():
132     idnum = entryId.get()
133     confirmBox = messagebox.askquestion("Confirmation","Are you sure ?")
134     if confirmBox == 'yes':
135         #<If no id is given then delte everything>
136         if(str(idnum)==""):
137             retrieveQuery = "SELECT user_id FROM user"
138             sqlCursor.execute(retrieveQuery)
139             photoIds = sqlCursor.fetchall()
140             for photoId in photoIds:
141                 photoIndex = 1
142                 while(photoIndex <= 50):
143                     os.remove("PhotosRecognizer/"+str(photoId[0])+"."+str(photoIndex)+".jpg")
144                     photoIndex += 1
145                     deleteQuery = "DELETE FROM user"
146                     sqlCursor.execute(deleteQuery)
147                     conn.commit()
148                     if(os.path.exists("FaceRecognizer/FaceTraining.yml")):
149                         os.remove("FaceRecognizer/FaceTraining.yml")
150         #</If no id is given then delte everything>
151         #<If id is inserted then delete specific Info>
152     else:
153         #<check if the id is an integer>
154         try:
155             intidnum = int(idnum)
156         except ValueError:
157             messagebox.showerror("Parameter Type Error","Id is not an integer")
158         return
159     #</check if the id is an integer>
160     #<Check if the id exsists>
```

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

```
161     checkQuery = "SELECT COUNT(*) FROM user WHERE user_id = (%s)"
162     sqlCursor.execute(checkQuery,intidnum)
163     checkValue = sqlCursor.fetchone()
164     if(checkValue[0] == 0):
165         messagebox.showwarning("Parameter Warning", "The id does not exist")
166         return
167     #</Check if the id exists>
168     #<Delete Database Data>
169     deleteQuery = "DELETE FROM user WHERE user_id = (%s)"
170     sqlCursor.execute(deleteQuery,intidnum)
171     #</Delete Database Data>
172     #<Delete the users photos>
173     photoIndex = 1
174     while(photoIndex <= 50):
175         os.remove("PhotosRecognizer/"+str(intidnum)+"."+str(photoIndex)+".jpg")
176         photoIndex += 1
177     #<Delete the users photos>
178     #<train again*>
179     LBPHFaceTrain()
180     #</train again*>
181     #<Commit the delete>
182     conn.commit()
183     #</commit the delete>
184     #<If id is inserted then delete specific Info>
185     else:
186         return
187     #</fromDelete>
```

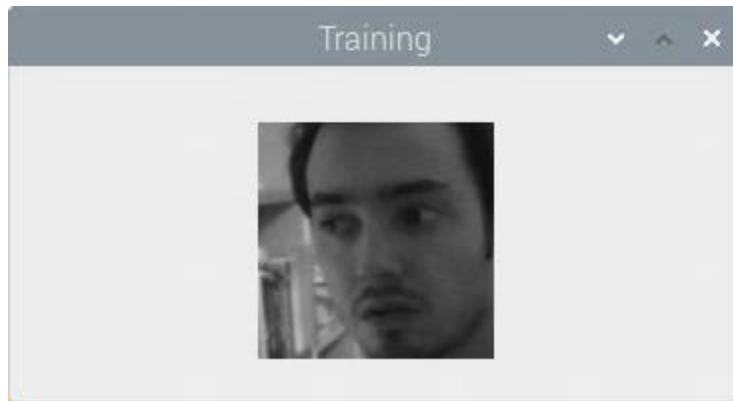
Το διάγραμμα ροής βρίσκεται στην παρακάτω εικόνα.



Εικόνα 6- 11: Το διάγραμμα ροής για την διαγραφή δεδομένων

#### 6.4.2: Εκπαίδευση του συστήματος μέσω των τοπικών δυαδικών προτύπων

Με το πλήκτρο training πραγματοποιείται η εκπαίδευση για την αναγνώριση προσώπων. Αρχικά δημιουργείται ο αναγνωριστής τοπικών δυαδικών προτύπων. Στην συνέχεια μετατρέπονται οι εικόνες σε numpy Array καθώς είναι ο τύπος που αναγνωρίζει το openCV. Από τα ονόματα των φωτογραφιών εξάγονται τα αναγνωριστικά των χρηστών. Τα αναγνωριστικά και οι φωτογραφίες εισάγονται σε δύο πίνακες. Μέσω αυτών των δύο πινάκων γίνεται η εκπαίδευση του αναγνωριστή. Κατά την διάρκεια της εκπαίδευσης εμφανίζονται οι φωτογραφίες.



Εικόνα 6- 12: Εμφάνιση φωτογραφιών κατά την διαδικασία εκπαίδευσης

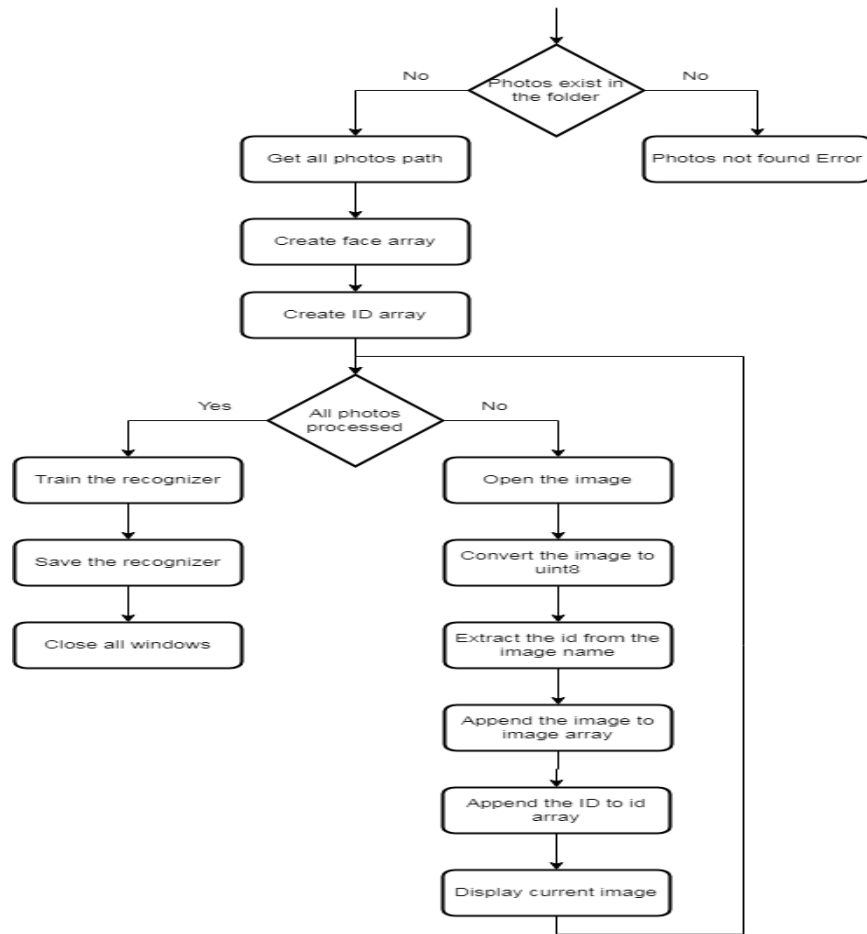
Το αρχείο που παράγεται είναι τύπου .yaml και περιέχουν τις πληροφορίες για τα ιστογράμματα κάθε φωτογραφίας έμπιστου χρήστη. Το αρχείο αυτό αποθηκεύεται στον φάκελο FaceRecognizer. Κατά την λειτουργία της κάμερας γίνεται ανάγνωση του αρχείου. Όταν εντοπιστεί πρόσωπο θα επιστραφεί το αναγνωριστικό του πρόσωπου καθώς και η απόσταση ανάμεσα στα ιστογράμματα τους. Με βάση αυτή την απόσταση κρίνεται αν το πρόσωπο είναι έμπιστο

```
233 #####<LBPH Face Training>#####
234 def LBPHFaceTrain():
235     faceRecognizer = cv2.face.LBPHFaceRecognizer_create()
236     pathImage = 'PhotosRecognizer'
237
238     def getPaths(pathWithImage):
239         arrayPath = [os.path.join(pathWithImage,pthImg) for pthImg in
os.listdir(pathWithImage)]
240         return arrayPath
241
242     def data2LBPHRecognizer(Path):
243         arrayPathPhoto = getPaths(pathImage)
244         pathImages = [arrayPathPhoto]
245         faces = []
246         idTags = []
247         for image in arrayPathPhoto:
248             PhotoFace = Image.open(image).convert('L')
249             numpyFace = np.array(PhotoFace,'uint8')
```



```
250     IDS1 = os.path.split(image)[1]
251     IDS2 = IDS1.split('.')[0]
252     IDNo = int(IDS2)
253     faces.append(numpyFace)
254     idTags.append(IDNo)
255     cv2.imshow("Training",numpyFace)
256     cv2.waitKey(150)
257     return faces,idTags
258
259     if not os.listdir(pathImage):
260         messagebox.showerror("Photos Error","No phtos found")
261         return
262     else:
263         Face2Recognizer,ID2Recognizer = data2LBPHRecognizer(pathImage)
264         faceRecognizer.train(Face2Recognizer,np.array(ID2Recognizer))
265         faceRecognizer.save('FaceRecognizer/FaceTraining.yml')
266         cv2.destroyAllWindows()
267
268     #####</LBPH Face Training>#####
```

Ακολουθεί το διάγραμμα ροής της διαδικασίας εκπαίδευσης του συστήματος



Εικόνα 6- 13: Το διάγραμμα ροής της διαδικασίας εκπαίδευσης

### 6.4.3: Επιτήρηση χώρου και αναγνώριση περιοχών ενδιαφέροντος

Με το πλήκτρο κάμερα ενεργοποιείται η κάμερα και ξεκινάει η επιτήρηση του χώρου. Το πρώτο βήμα είναι ο έλεγχος για το αν υπάρχει το αρχείο με τα έμπιστα πρόσωπα. Στην συνέχεια γίνεται έλεγχος για τον αν υπάρχουν στην βάση δεδομένων οι παράμετροι για την διαίρεση της εικόνας. Σε περίπτωση που κάποιος έλεγχος αποτύχει εμφανίζεται το αντίστοιχο μήνυμα σφάλματος



Εικόνα 6- 14: Μηνύματα λάθους κατά την ενεργοποίηση της κάμερας

Εφόσον οι έλεγχοι ολοκληρωθούν χωρίς προβλήματα ακολουθεί η αρχικοποίηση των παραμέτρων για τους χρόνους καταγραφής του κάθε συμβάντος και για τα βίντεο

```
269
270 #####<CAMERA>#####
271 def ActivateCamera():
272
273     camera = cv2.VideoCapture(0)
274     faceRecognizer = cv2.face.LBPHFaceRecognizer_create()
275     #<Check if the LBPH Face Recognizer File Exists>
276     if (os.path.exists('FaceRecognizer/FaceTraining.yml')):
277         faceRecognizer.read('FaceRecognizer/FaceTraining.yml')
278     else:
279         messagebox.showerror("LBPH file Error", "No LBPH File found")
280     return
281 #</Check if the LBPH Face Recognizer File Exists>
282 #<Check if offset data exist in the database>
283 checkQuery = "SELECT COUNT(*) FROM offset"
284 sqlCursor.execute(checkQuery)
285 number = sqlCursor.fetchone()
286 if (number[0] == 0):
287     messagebox.showerror("ERROR", "No offset Data Found")
288     return
289 #</Check if offset data exist in the database>
290 fourcc_code = cv2.VideoWriter_fourcc(*'DIVX')
291 #<Unknown face video settings>
292 face_consecutive_frames = 0
293 face_idle_timer = 80
294 face_writer = None
295 (face_height,face_width) = (None,None)
296 face_npZeros = None
297 face_made_recording = False
```

```
298 def face_record_video():
299     face_output = np.zeros((face_height2,face_width2,3),dtype = "uint8")
300     face_output[0:face_height2,0:face_width2] = frame
301     face_writer.write(face_output)
302 #</Unknown face video settings
303 #<upperBody video Parameters>
304 upbody_consecutive_frames = 0
305 upbody_idle_timer = 80
306 upbody_writer = None
307 (upbody_height,upbody_width) = (None,None)
308 upbody_npZeros = None
309 upbody_output = None
310 upbody_made_recording = False
311 def upbody_record_video():
312     upbody_output = np.zeros((upbody_height2,upbody_width2,3),dtype = "uint8")
313     upbody_output[0:upbody_height2,0:upbody_width2] = frame
314     upbody_writer.write(upbody_output)
315 #</upperBody video Parameters>
316 #<fullbody Video Parameters>
317 flbody_consecutive_frames = 0
318 flbody_idle_timer = 80
319 flbody_writer = None
320 (flbody_height2,flbody_width2) = (None,None)
321 flbody_output = None
322 flbody_made_recording = False
323 def flbody_record_video():
324     flbody_output = np.zeros((flbody_height2,flbody_width2,3),dtype = "uint8")
325     flbody_output[0:flbody_height2,0:flbody_width2] = frame
326     flbody_writer.write(flbody_output)
327 #</fullbody Video Parameters>
328 #<Movement Video Paramteres>
329 movement_average = None
330 movement_consecutive_frames = 0
331 movement_idle_time = 80
332 movement_writer = None
333 (movement_height,movement_width) = (None,None)
334 movement_npZeros = None
335 movement_output = None
336 movement_made_recording = False
337 def movement_record_video():
338     movement_output =
np.zeros((movement_height2,movement_width2,3),dtype="uint8")
339     movement_output[0:movement_height2,0:movement_width2] = frame
340     movement_writer.write(movement_output)
341 #</Movement Video Paramteres>
342 #<Get Offset Data>
343 offsetData = "SELECT * FROM offset"
344 sqlCursor.execute(offsetData)
345 offsetDBData = sqlCursor.fetchall()
```

```
346 for offset in offsetDBData:
347     minx = offset[0]
348     maxx = offset[1]
349     miny = offset[2]
350     maxy = offset[3]
351 upbody_remaining_time = False
352 flbody_remaining_time = False
353 movement_remaining_time = False
354 #</Get offset Data>
355 cof_x = int((minx+maxx)/2)
356 cof_y = int((miny+maxy)/2)
357 array = [(0,0,0)]
358 array2 = [(0,0,0)]
359 array3 = [(0,0,0)]
360 array.clear()
361 array2.clear()
362 array3.clear()
363 Csent = True
364 escPressed = False
```

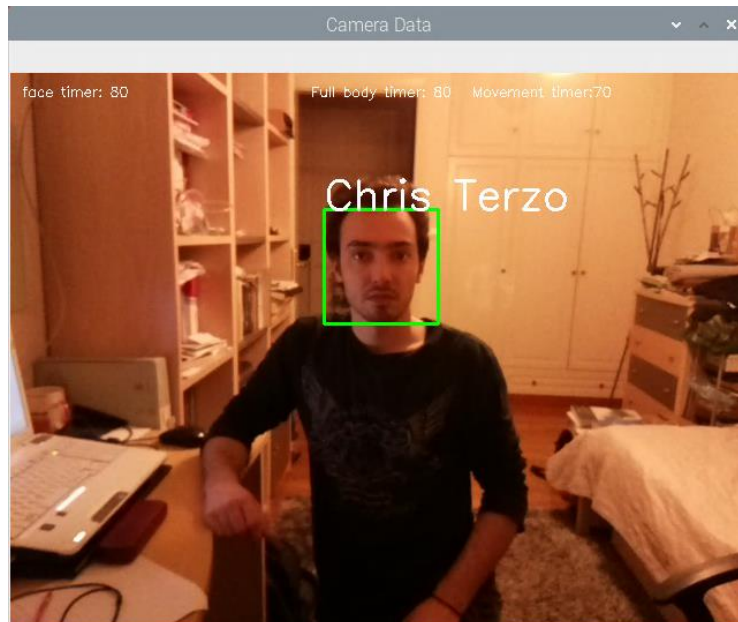
Η παρακολούθηση του χώρου γίνεται σε έναν ατέρμονα βρόχο ο οποίος διακόπτεται με το πλήκτρο Escape. Με την εκκίνηση κάθε βρόχου επαναφέρονται οι μεταβλητές εντοπισμού περιοχών ενδιαφέροντος σε ψευδή τιμή. Στην συνέχεια διαβάζεται ένα καρέ από την κάμερα και μετατρέπεται σε κλίμακα του γκρι διότι είναι ο τύπος εικόνων που θα γίνουν οι εντοπισμοί των περιοχών ενδιαφέροντος. Οι περιοχές ενδιαφέροντος εντοπίζονται με την χρήση haar classifiers που έχουν φορτωθεί από την αρχή εκκίνησης του προγράμματος.

```
365 while True:
366     #5#ser.flush()
367     face_det_no = 0
368     ubod_det_no = 0
369     fbod_det_no = 0
370     move_det_no = 0
371     known_face_detected = False
372     unknown_face_detected = False
373     upbody_detected = False
374     flbody_detected = False
375     motion_detected = False
376
377     (grabbed,frame) = camera.read()
378
379     #cv2.rectangle(frame,(0,0),(0+int(camera.get(cv2.CAP_PROP_FRAME_WIDTH)),25),(0,0,0),-1)
380     cv2.putText(frame,"face timer:
381     "+str(face_idle_timer),(10,20),cv2.FONT_HERSHEY_SIMPLEX,0.4,(255,255,255),1)
381     gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
```

#### 6.4.3.1: Σενάριο με εντοπισμό προσώπου

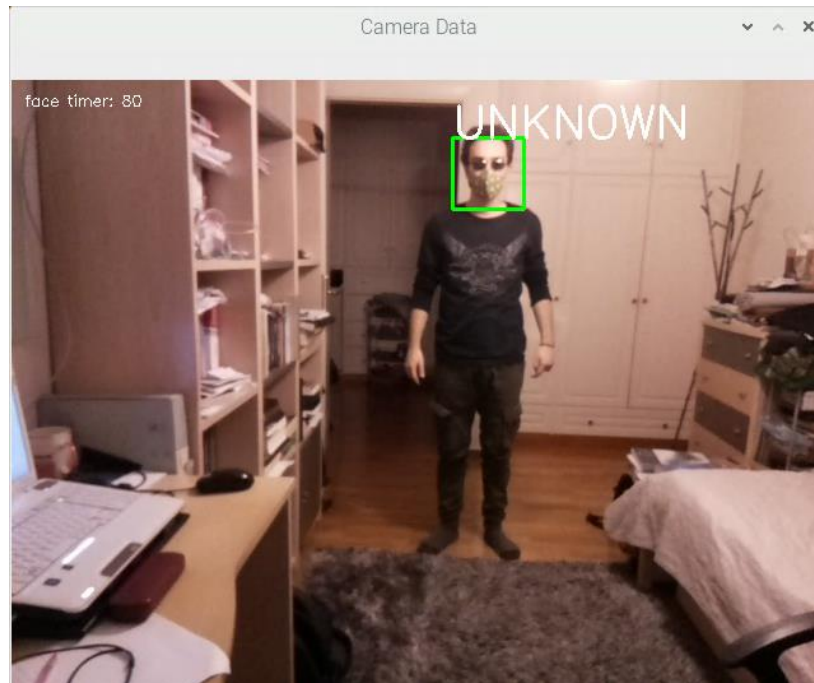
Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Αν κατά την επιτήρηση του χώρου βρεθεί ένα πρόσωπο τότε εμφανίζεται ένα πράσινο παραλληλόγραμμο και ξεκινάει η διαδικασία αναγνώρισης του προσώπου. Σε περίπτωση του πρόσωπο είναι έμπιστο τότε πάνω από το πρόσωπο εμφανίζεται το όνομα του.



**Εικόνα 6- 15: Επιτυχής αναγνώριση προσώπου**

Σε περίπτωση που το πρόσωπο δεν αναγνωριστεί επιτυχώς τότε πάνω από το παραλληλόγραμμο εμφανίζεται η λέξη UNKNOWN και ξεκινάει μια αλληλουχία διεργασιών. Η πρώτη διεργασία αφορά την επικοινωνία ανάμεσα στον μικροϋπολογιστή και στον μικροελεγκτή για την κίνηση του βραχίονα ως εξής. Αρχικά υπολογίζεται το κέντρο του παραλληλογράμμου του προσώπου. Στην συνέχεια ελέγχεται η θέση του κέντρου σε σχέση με το κεντρικό υποπλαίσιο όπως αυτό έχει διαχωριστεί (αναλύεται στην επόμενη ενότητα). Ο βραχίονας κινείται με τέτοιο τρόπο ώστε το κέντρο του προσώπου να βρεθεί στο κεντρικό πλαίσιο. Αν υπάρχουν πολλαπλά πρόσωπα τότε κινείται με βάση αυτό που βρίσκεται πιο κοντά στο κέντρο του μεσαίου πλαισίου. Η δεύτερη διεργασία αφορά την καταγραφή σε βίντεο του άγνωστου προσώπου. Η καταγραφή γίνεται σε προσωρινό βίντεο μέχρι το άγνωστο πρόσωπο να φύγει από το πεδίο λήψης της κάμερας. Αν το άγνωστο πρόσωπο απομακρυνθεί ο μετρητής μειώνεται μέχρι το μηδέν. Αν φτάσει στο μηδέν το βίντεο αποθηκεύεται με όνομα FACE\_TIMESTAMP.avi, ενώ αν εντοπιστεί άγνωστο πρόσωπο πριν το τέλος ο μετρητής επαναφέρεται στην αρχική του τιμή και η καταγραφή συνεχίζεται στο προσωρινό βίντεο. Αν πατηθεί το πλήκτρο Escape τότε σταματάει η παρακολούθηση του προσώπου και η επαναφορά του μετρητή, οπότε το βίντεο αποθηκεύεται μόλις μηδενίσει ο χρόνος.



Εικόνα 6- 16: Μη επιτυχής αναγνώριση προσώπων

Ο κώδικας για τον εντοπισμό, αναγνώριση και παρακολούθηση προσώπου είναι ο παρακάτω:

```
382     #<Face Detection>
383     faces = faceDetect.detectMultiScale(gray,1.3,3)
384     for(face_point1,face_point2,face_width,face_height) in faces:
385         face_det_no = face_det_no + 1
386
387         cv2.rectangle(frame,(face_point1,face_point2),(face_point1+face_width,face_point2+face_height),(0,255,0),2)
388         idTag,sure =
389         faceRecognizer.predict(gray[face_point2:face_point2+face_height,face_point1:face_point1+face_width])
390         if(sure < 100):
391             getUserQuery = "SELECT user_name FROM user WHERE user_id = %s"
392             intidTag = int(idTag)
393             sqlCursor.execute(getUserQuery,intidTag)
394             name = sqlCursor.fetchone()
395             cv2.putText(frame,str(name[0]),(face_point1,face_point2),cv2.FONT_HERSHEY_SIMPLE
396             X,1.2,(255,255,255),2)
397             known_face_detected = True
398         else:
399             print(unknown_face_detected)
400             cv2.putText(frame,"UNKNOWN",(face_point1,face_point2),
401             cv2.FONT_HERSHEY_SIMPLEX,1.2,(255,255,255),2)
```

```

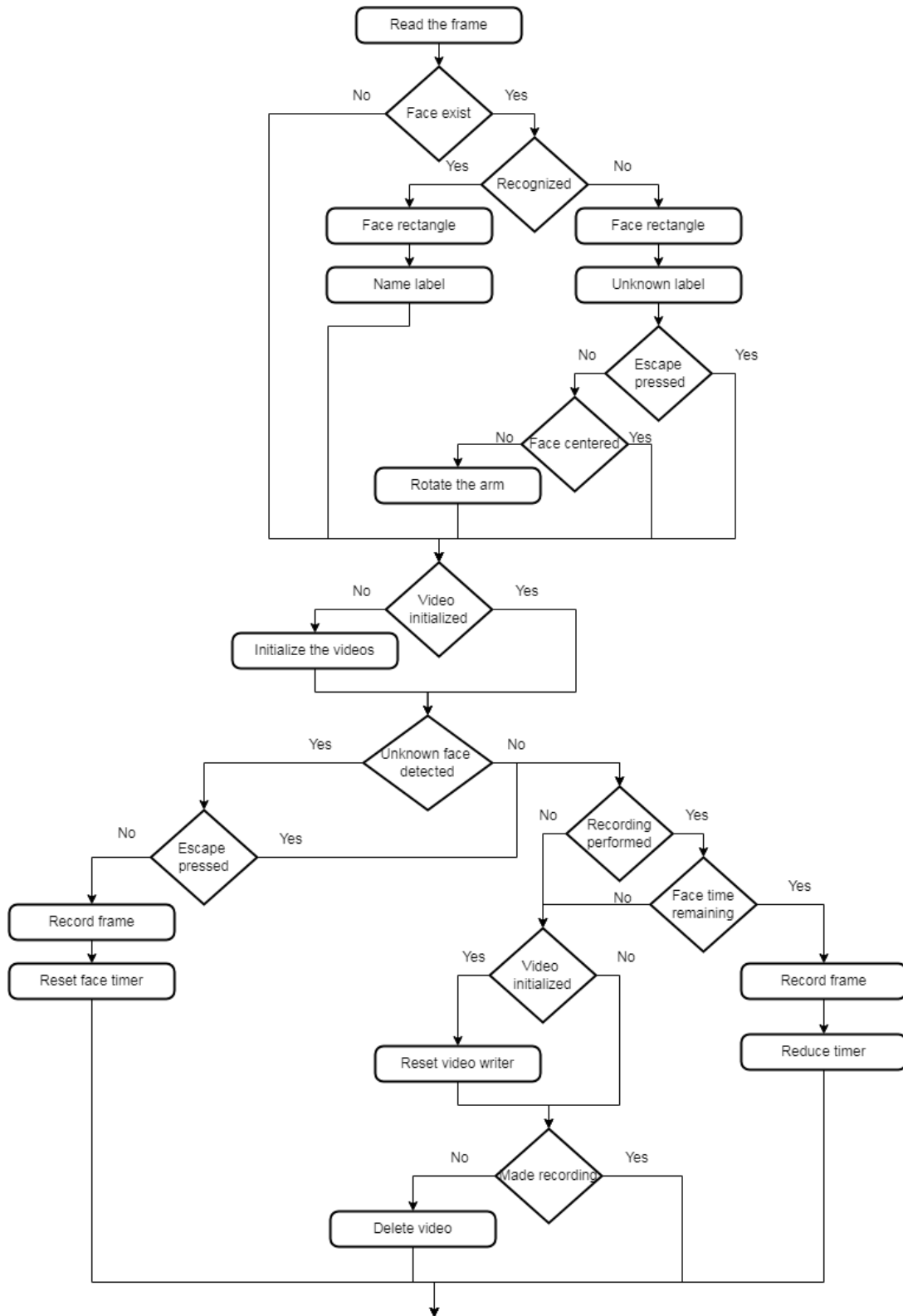
397         cv2.putText(frame,"UNKNOWN",(face_point1,face_point2),
cv2.FONT_HERSHEY_SIMPLEX,1.2,(255,255,255),2)
398         known_face_detected = False
399         unknown_face_detected = True
400         Csent = False
401         p1_farside = face_point1 + face_width
402         p2_farside = face_point2 + face_height
403         center_x = int((face_point1+p1_farside)/2)
404         center_y = int((face_point2+p2_farside)/2)
405         x_dst = math.pow((center_x-cof_x),2)
406         y_dst = math.pow((center_y-cof_y),2)
407         totalDST = math.sqrt(x_dst+y_dst)
408         el2array = [center_x,center_y,round(totalDST,2)]
409         array.append(el2array)
410         sorted(array,key = lambda x:x[2],reverse = True)
411         track_kenX = array[0][0]
412         track_kenY = array[0][1]
413         if(escPressed is False):
414             if((track_kenX < minx) and (track_kenY < miny)):
415                 #print("Face UL")
416                 ser.write(b"LN")
417             if((track_kenX >= minx and track_kenX < maxx) and (track_kenY <
miny)):
418                 ser.write(b"U")
419                 #print("Face UN")
420             if((track_kenX >= maxx) and (track_kenY < miny)):
421                 #print("Face UR")
422                 ser.write(b"RN")
423             if((track_kenX < minx) and (track_kenY >= miny and track_kenY <
maxy)):
424                 ser.write(b"LN")
425                 #print("Face L")
426             if((track_kenX > minx and track_kenX < maxx) and (track_kenY >= miny
and track_kenY < maxy)):
427                 ser.write(b"CN")
428                 #print("Face C")
429             if((track_kenX >= maxx) and (track_kenY >= miny and track_kenY <
maxy)):
430                 ser.write(b"RN")
431                 #print("Face R")
432             if((track_kenX < minx) and (track_kenY >= maxy)):
433                 #print("Face DL")
434                 ser.write(b"LN")
435             if((track_kenX >= minx and track_kenX < maxx) and (track_kenY >=
maxy)):
436                 ser.write(b"D")
437                 #print("Face DN")
438             if((track_kenX >= maxx) and (track_kenY >= maxy)):
439                 #print("Face DR")
440                 ser.write(b"RN")

```



```
440         ser.write(b"RN")
441     if(len(array) == 10 or unknown_face_detected is False):
442         array.clear()
443
444
445     if face_writer is None:
446         face_filename =
"Face_"+datetime.datetime.now().strftime("%Y_%m_%d_%H_%M_%S")
447         face_file_path = ("Unknown/Faces/"+face_filename+".avi")
448         face_file_path = face_file_path.format(face_filename = face_filename)
449         (face_height2,face_width2) = frame.shape[:2]
450         face_writer =
cv2.VideoWriter(face_file_path,fourcc_code,10,(face_width2,face_height2),True)
451         face_npZeros = np.zeros((face_height2,face_width2),dtype = "uint8")
452     if (unknown_face_detected and escPressed is False):
453         face_consecutive_frames += 1
454     if(face_consecutive_frames >= 10):
455         face_record_video()
456         face_made_recording = True
457         face_idle_timer = 80
458     else:
459     if face_made_recording is True and face_idle_timer > 0:
460         face_idle_timer -= 1
461         face_record_video()
462     else:
463         face_consecutive_frames = 0
464     if face_writer is not None:
465         face_writer.release()
466         face_writer = None
467     if face_made_recording is False:
468         os.remove(face_file_path)
469         face_made_recording = False
470         face_idle_timer = 80
471     #</Face Detection>
```

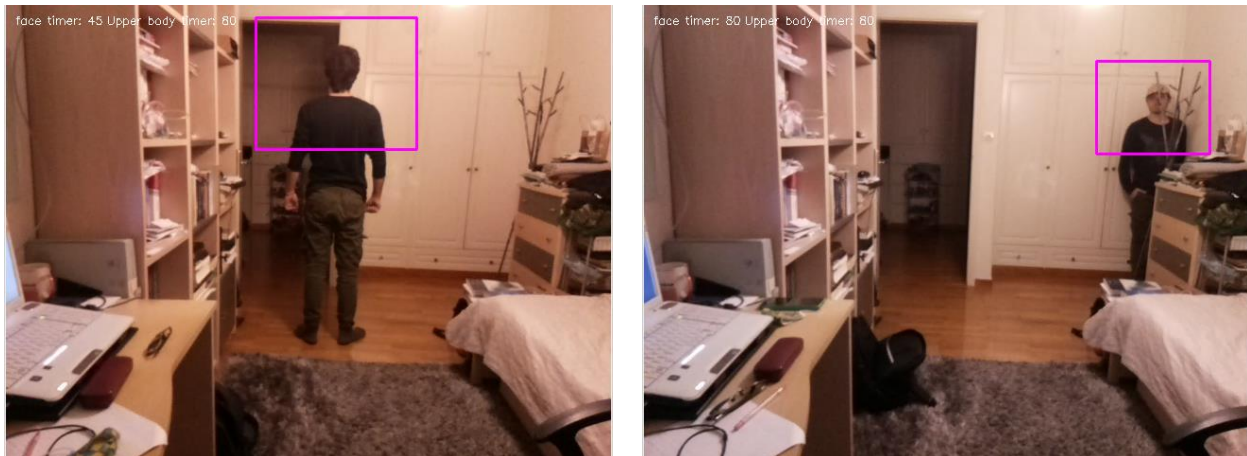
Ακολουθεί το διάγραμμα ροής παρουσιάζεται στην παρακάτω εικόνα.



Εικόνα 6- 17: Το διάγραμμα ροής για τον εντοπισμό προσώπου

### 6.4.3.2: Σενάριο με εντοπισμό άνω μέρους σώματος

Με τον εντοπισμό άνω μέρους σώματος εμφανίζεται ένα μωβ παραλληλόγραμμο γύρω από την περιοχή ενδιαφέροντος και ξεκινάνε οι διεργασίες μετακίνησης του βραχίονα και καταγραφής σε βίντεο. Όπως και με τον εντοπισμό άγνωστου προσώπου η κίνηση γίνεται με βάση το κέντρο του παραλληλογράμμου και σε πολλαπλές περιοχές ενδιαφέροντος σε σχέση με αυτή που βρίσκεται πιο κοντά στο κέντρο. Η καταγραφή γίνεται σε προσωρινό βίντεο όσο υπάρχει η περιοχή ενδιαφέροντος στο πεδίο λήψης της κάμερας. Μόλις φύγει ο μετρητής μειώνεται μέχρι το 0. Αν φτάσει το 0 το βίντεο αποθηκεύεται, ενώ αν βρεθεί άνω μέρος σώματος ο μετρητής επανέρχεται και η καταγραφή συνεχίζεται στο προσωρινό βίντεο.



Εικόνα 6- 18: Εντοπισμός άνω μέρους σώματος

Για να ξεκινήσει η διαδικασία για τον εντοπισμό άνω μέρους σώματος πρέπει να πληρείται τουλάχιστον ένα από τα δύο κριτήρια. Το πρώτο κριτήριο είναι να μην υπάρχει άγνωστο πρόσωπο στο πεδίο λήψης της κάμερας. Το δεύτερο κριτήριο είναι ο μετρητής του άνω μέρους σώματος να είναι μεγαλύτερος από το μηδέν. Επίσης για να περιστραφεί ο βραχίονας πρέπει να μην υπάρχει άγνωστο πρόσωπο. Αυτό συμβαίνει καθώς οι περιοχές ενδιαφέροντος είναι ιεραρχημένες και την υψηλότερη προτεραιότητα έχουν τα άγνωστα πρόσωπα έναντι του άνω μέρους σώματος.

Ακολουθεί ο κώδικας για τον εντοπισμό άνω μέρους σώματος

```
474     if((unknown_face_detected is False) or (upbody_remaining_time is True)):
475         cv2.putText(frame," Upper body timer:
"+str(upbody_idle_timer),(100,20),cv2.FONT_HERSHEY_SIMPLEX,0.4,(255,255,255),1)
476         upbody =
ubodyDetect.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=
cv2.CASCADE_SCALE_IMAGE)
```

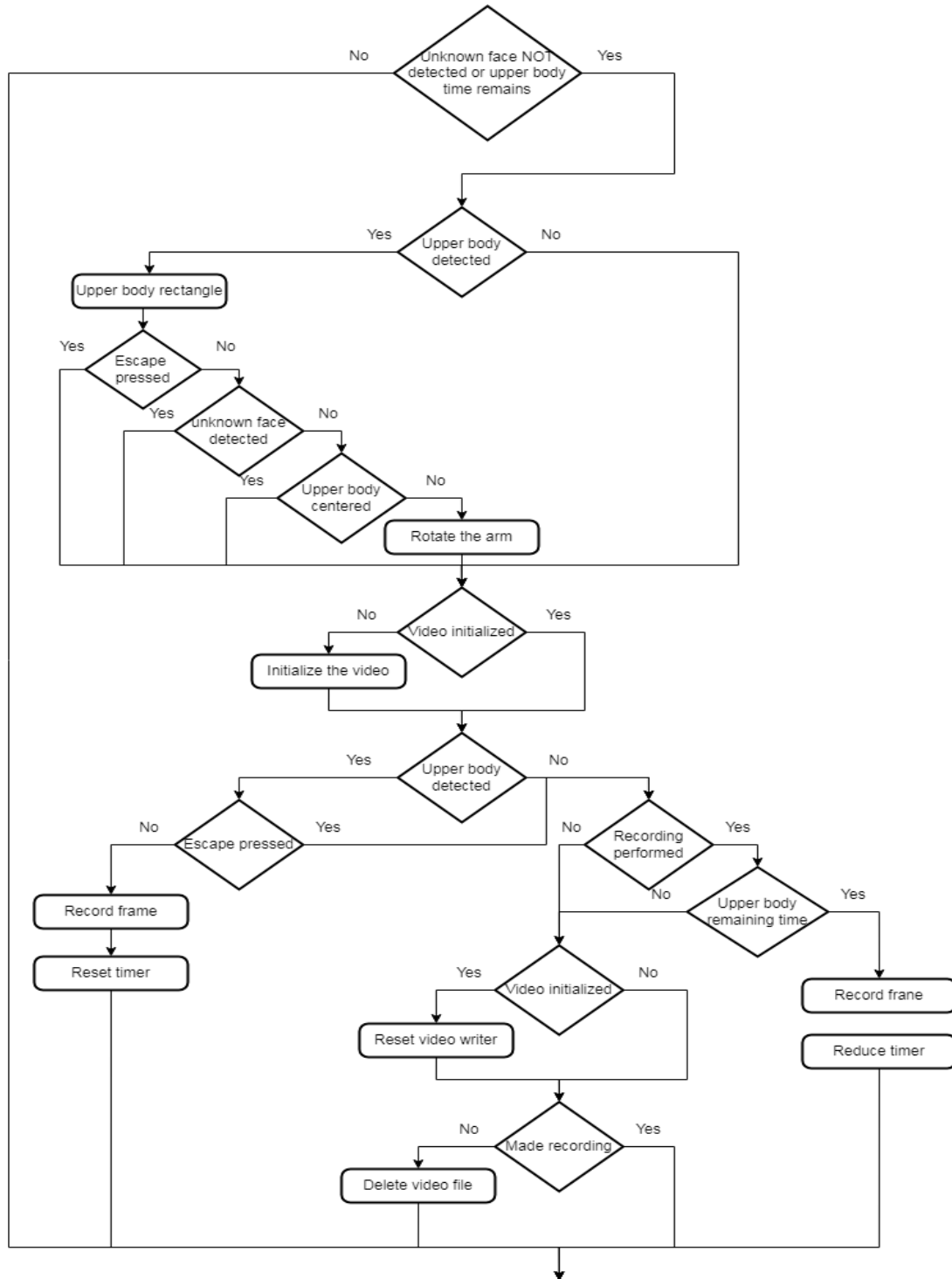
```

477     for(x,y,w,h) in upbody:
478         ubod_det_no = ubod_det_no + 1
479         cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,255),2)
480         upbody_detected = True
481         Csent = False
482         x_farside = x+w
483         y_farside = y+h
484         x_center = int((x+x_farside)/2)
485         y_center = int((y+y_farside)/2)
486         ub_dstx = math.pow((x_center-cof_x),2)
487         ub_dsty = math.pow((y_center-cof_y),2)
488         ub_totalDST = math.sqrt(ub_dstx+ub_dsty)
489         ub_el2array = [ub_dstx,ub_dsty,ub_totalDST]
490         array2.append(ub_el2array)
491         sorted(array2,key = lambda x:x[2],reverse = False)
492         track_kenX = array2[0][0]
493         track_kenY = array2[0][1]
494         if(escPressed is False):
495             if(unknown_face_detected is False):
496                 if(unknown_face_detected is False):
497                     if((track_kenX < minx) and (track_kenY < miny)):
498                         #print("Upper UL")
499                         ser.write(b"LN")
500                     if((track_kenX >= minx and track_kenX < maxx) and (track_kenY <
501 miny)):
502                         #print("Upper U")
503                         ser.write(b"UN")
504                     if((track_kenX >= maxx) and (track_kenY < miny)):
505                         #print("Upper UR")
506                         ser.write(b"RN")
507                     if((track_kenX < minx) and (track_kenY >= miny and track_kenY <
508 maxy)):
509                         #print("Upper L")
510                         ser.write(b"LN")
511                     if((track_kenX > minx and track_kenX < maxx) and (track_kenY >=
512 miny and track_kenY < maxy)):
513                         #print("Upper C")
514                         ser.write(b"CN")
515                     if((track_kenX >= maxx) and (track_kenY >= miny and track_kenY <
516 maxy)):
517                         ser.write(b"RN")
518                         #print("Upper R")
519                     if((track_kenX < minx) and (track_kenY >= maxy)):
520                         #print("Upper DL")
521                         ser.write(b"LN")
522                     if((track_kenX >= minx and track_kenX < maxx) and (track_kenY >=
523 maxy)):
524                         #print("Upper D")
525                         ser.write(b"DN")
526                     if((track_kenX >= maxx) and (track_kenY >= maxy)):

```

```
522         print("Upper DR")
523         #ser.write(b"RN")
524     if(len(array2) == 10 or upbody_detected is False):
525         array2.clear()
526
527     if upbody_writer is None:
528         upbody_filename =
529         "UpBody_"+datetime.datetime.now().strftime("%Y_%m_%d_%H_%M_%S")
530         upbody_filepath = ("Unknown/UpperBody/"+upbody_filename+".avi")
531         upbody_filepath = upbody_filepath.format(upbody_filename =
532         upbody_filename)
533         (upbody_height2,upbody_width2) = frame.shape[0:2]
534         upbody_writer =
535         cv2.VideoWriter(upbody_filepath,fourcc_code,10,(upbody_width2,upbody_height2),True)
536         upbody_npZeros = np.zeros((face_height2,face_width2),dtype = "uint8")
537         if((upbody_idle_timer == 80) or (upbody_idle_timer == 0)):
538             upbody_remaining_timer = False
539         else:
540             upbody_remaining_timer = True
541         if (upbody_detected and escPressed is False):
542             upbody_consecutive_frames += 1
543             if upbody_consecutive_frames >= 5:
544                 upbody_record_video()
545                 upbody_made_recording = True
546                 upbody_idle_timer = 80
547             else:
548                 if upbody_made_recording is True and upbody_idle_timer > 0:
549                     upbody_idle_timer -= 1
550                     upbody_record_video()
551                 else:
552                     upbody_consecutive_frames = 0
553                     if upbody_writer is not None:
554                         upbody_writer.release()
555                         upbody_writer = None
556                     if upbody_made_recording is False:
557                         os.remove(upbody_filepath)
558                         upbody_made_recording = False
559                         upbody_idle_timer = 80
560         #</Upbody Detection>
```

Ακολουθεί το διάγραμμα ροής



Εικόνα 6- 19:Διάγραμμα ροής για τον εντοπισμό άνω μέρους σώματος

### 6.4.3.3: Σενάριο με εντοπισμό πλήρους σώματος

Σε περίπτωση που η περιοχή ενδιαφέροντος είναι πλήρες σώμα τότε εμφανίζεται ένα μαύρο παραλληλόγραμμο γύρω από το σώμα. Για να ξεκινήσει η διαδικασία για τον εντοπισμό πλήρους σώματος πρέπει να πληρείται τουλάχιστον ένα από τα δύο κριτήρια. Το πρώτο κριτήριο είναι να μην υπάρχει άγνωστο πρόσωπο, και να μην υπάρχει άνω μέρος σώματος στο πεδίο λήψης της κάμερας. Το δεύτερο κριτήριο είναι ο μετρητής του πλήρους σώματος να είναι μεγαλύτερος από το μηδέν. Επίσης για να περιστραφεί ο βραχίονας πρέπει να μην υπάρχει καμία από τις περιοχές ενδιαφέροντος που αναφέρθηκαν. Αυτό συμβαίνει καθώς οι περιοχές ενδιαφέροντος είναι ιεραρχημένες και την υψηλότερη προτεραιότητα έχουν τα άγνωστα πρόσωπα και το άνω μέρος σώματος έναντι του πλήρους σώματος.

Στην εικόνα της επόμενης σελίδας ακολουθεί ο κώδικας για τον εντοπισμό πλήρους σώματος

```

558
559     #<FullBody Detection>
560     if((unknown_face_detected is False and upbody_detected is False) or
(flbody_remaining_time is True)):
561         cv2.putText(frame," Full body timer:
"+str(flbody_idle_timer),(235,20),cv2.FONT_HERSHEY_SIMPLEX,0.4,(255,255,255),1)
562         fullbody =
fbodyDetect.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=
cv2.CASCADE_SCALE_IMAGE)
563         for(x,y,w,h) in fullbody:
564             fbod_det_no = fbod_det_no + 1
565             cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,0),2)
566             flbody_detected = True
567             Csent = False
568             x_farside = x+w
569             y_farside = y+h
570             x_center = int((x+x_farside)/2)
571             y_center = int((y+y_farside)/2)
572             fbx_dst = math.pow((x_center-cof_x),2)
573             fby_dst = math.pow((y_center-cof_y),2)
574             fbttotalDST = math.sqrt(fbx_dst+fby_dst)
575             dbel2array = [fbx_dst,fby_dst,round(fbttotalDST,2)]
576             array3.append(dbel2array)
577             sorted(array3,key = lambda x:x[2],reverse = False)
578             track_kenX = array3[0][0]
579             track_kenY = array3[0][1]
580             if(escPressed is False):
581                 if(known_face_detected is False and upbody_detected is False):
582                     if(unknown_face_detected is False and upbody_detected is False):
583                         if((track_kenX < minx) and (track_kenY < miny)):
584                             #print("Full UL")
585                             ser.write(b"LN")
586                         if((track_kenX >= minx and track_kenX < maxx) and (track_kenY <
miny)):
587                             #print("FULL U")
588                             ser.write(b"UN")

```

```

589         if((track_kenX >= maxx) and (track_kenY < miny)):
590             #print("FULL UR")
591             ser.write(b"RN")
592         if((track_kenX < minx) and (track_kenY >= miny and track_kenY <
maxy)):
593             #print("FULL L")
594             ser.write(b"LN")
595         if((track_kenX > minx and track_kenX < maxx) and (track_kenY >=
miny and track_kenY < maxy)):
596             #print("FULL C")
597             ser.write(b"CN")
598         if((track_kenX >= maxx) and (track_kenY >= miny and track_kenY <
maxy)):
599             #print("FULL R")
600             ser.write(b"RN")
601         if((track_kenX < minx) and (track_kenY >= maxy)):
602             #print("FULL DL")
603             ser.write(b"LN")
604         if((track_kenX >= minx and track_kenX < maxx) and (track_kenY >=
maxy)):
605             #print("FULL D")
606             ser.write(b"DN")
607         if((track_kenX >= maxx) and (track_kenY >= maxy)):
608             #print("FULL DR")
609             ser.write(b"RN")
610         if(len(array3) == 10 or flbody_detected is False):
611             array3.clear()
612         if flbody_writer is None:
613             flbody_filename =
"FlBody_"+datetime.datetime.now().strftime("%Y_%m_%d_%H_%M_%S")
614             flbody_filepath = ("Unknown/FullBody/"+flbody_filename+".avi")
615             (flbody_height2,flbody_width2) = frame.shape[0:2]
616             flbody_writer =
cv2.VideoWriter(flbody_filepath,fourcc_code,10,(flbody_width2,flbody_height2),True)
617             flbody_npZeros = np.zeros((flbody_height2,flbody_width2),dtype = "uint8")
618         if((flbody_idle_timer == 80) or (flbody_idle_timer == 0)):
619             flbody_remaining_time = False
620         else:
621             flbody_remaining_time = True
622         if (flbody_detected and escPressed is False):
623             flbody_consecutive_frames += 1
624         if flbody_consecutive_frames >= 5:
625             flbody_record_video()
626             flbody_made_recording = True
627             flbody_idle_timer = 80

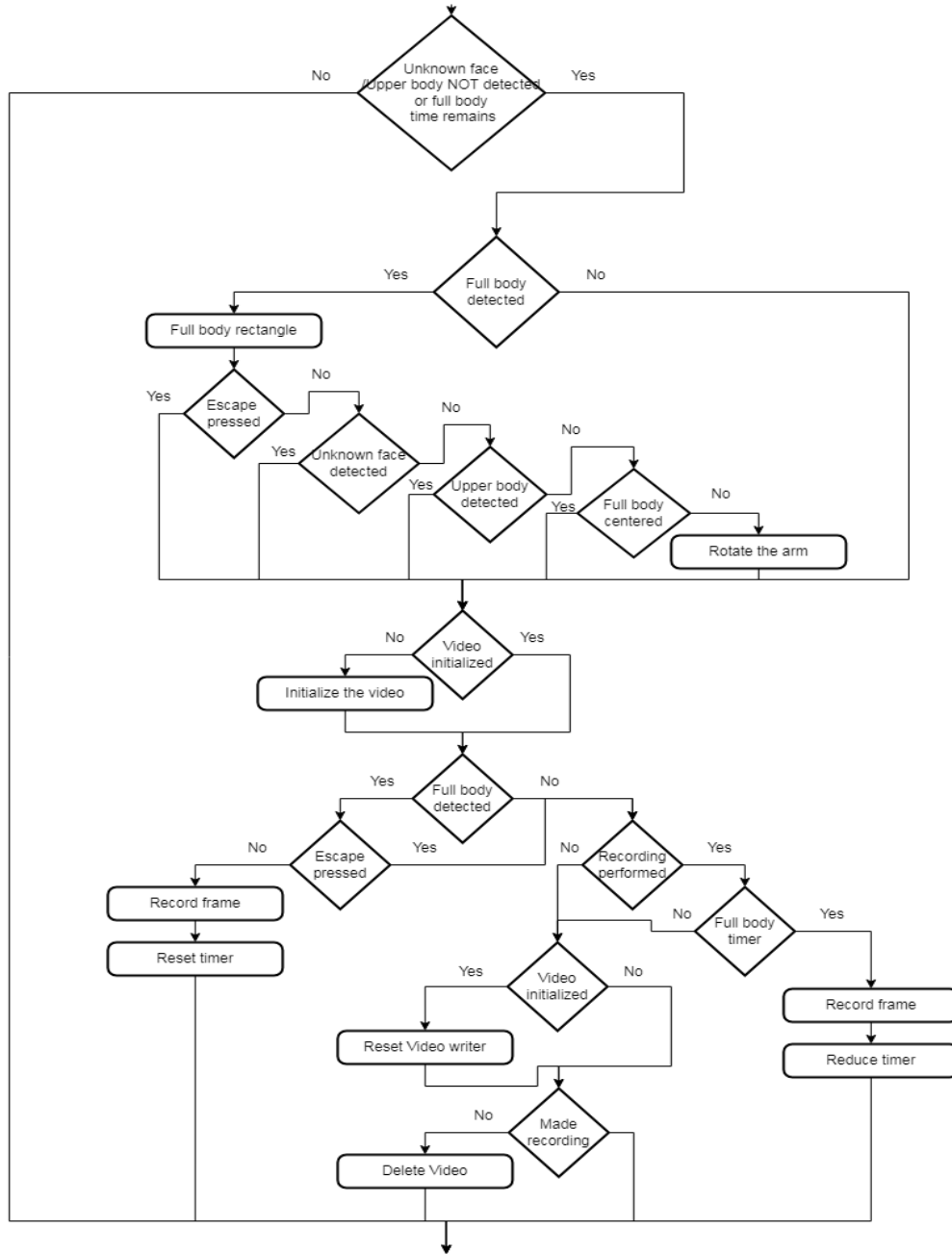
```



Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

```
628     else:
629         if((flbody_made_recording is True) and (flbody_idle_timer > 0)):
630             flbody_idle_timer -= 1
631             flbody_record_video()
632         else:
633             flbody_consecutive_frames = 0
634             if flbody_writer is not None:
635                 flbody_writer.release()
636                 flbody_writer = None
637             if flbody_made_recording is False:
638                 os.remove(flbody_filepath)
639                 flbody_made_recording = False
640                 flbody_idle_timer = 80
641     #</FullBody Detection>
```

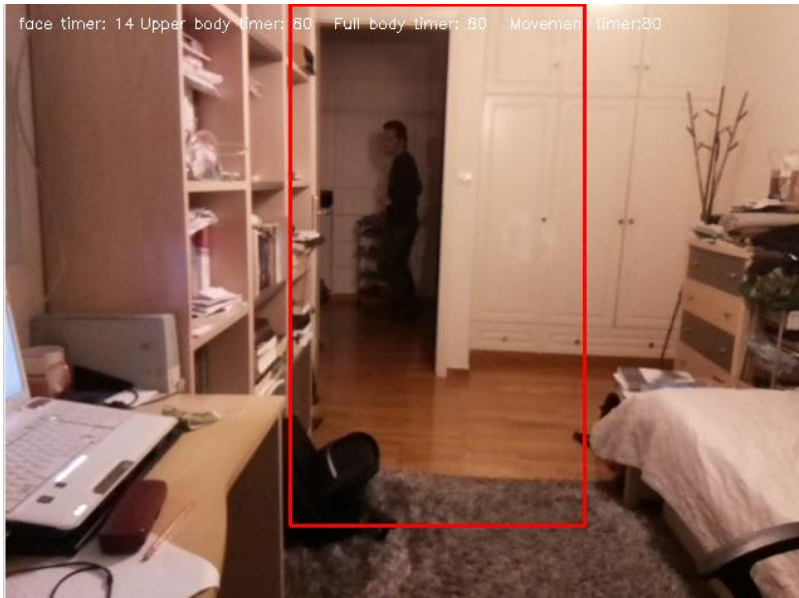
Στην συνέχεια ακολουθεί το διάγραμμα ροής για τον εντοπισμό κίνησης



Εικόνα 6- 20: Διάγραμμα ροής για τον εντοπισμό πλήρους σώματος

#### 6.4.3.4: Σενάριο με εντοπισμό κίνησης

Ο εντοπισμός κίνησης γίνεται μέσω αφαίρεσης φόντου όπως έχει περιγραφεί στο τέταρτο κεφάλαιο. Στην οθόνη εμφανίζεται μόνο η κίνηση με το μεγαλύτερο μέγεθος με ένα κόκκινο παραλληλόγραμμο. Το μέγεθος του παραλληλογράμμου πρέπει να είναι μικρότερο από το μέγεθος του frame. Αυτό γίνεται ώστε να μην πιάνεται ως κίνηση η περιστροφή του βραχίονα. Για να ξεκινήσει η διαδικασία εντοπισμού κίνησης πρέπει να μην υπάρχει άλλη περιοχή ενδιαφέροντος στο πεδίο λήψης της κάμερας ή να απομένει χρόνος για την καταγραφή σε βίντεο. Επιπλέον για να περιστραφεί ο βραχίονας πρέπει να μην υπάρχει άλλη περιοχή ενδιαφέροντος στο πεδίο λήψης της κάμερας λόγω της ιεράρχησης των περιοχών ενδιαφέροντος.



Εικόνα 6- 21: Εντοπισμός κίνησης

Ο κώδικας για τον εντοπισμό είναι ο εξής:

```
642
643
644     #<Movement Detection>
645     if( unknown_face_detected is False and upbody_detected is False and
646     flbody_detected is False) or (movement_remaining_time is True):
647         gray = cv2.GaussianBlur(gray,(21,21),0)
648         cv2.putText(frame,"Movement
649         timer:"+str(movement_idle_time),(400,20),cv2.FONT_HERSHEY_SIMPLEX,0.4,(255,255,
650         255),1)
651         if movement_average is None:
652             movement_average = gray.copy().astype("float")
653             continue
654         cv2.accumulateWeighted(gray,movement_average,0.5)
655         frame_dif = cv2.absdiff(gray,cv2.convertScaleAbs(movement_average))
656         thresholded = cv2.threshold(frame_dif,5,255,cv2.THRESH_BINARY)[1]
657         thresholded = cv2.dilate(thresholded, None, iterations = 3)
```

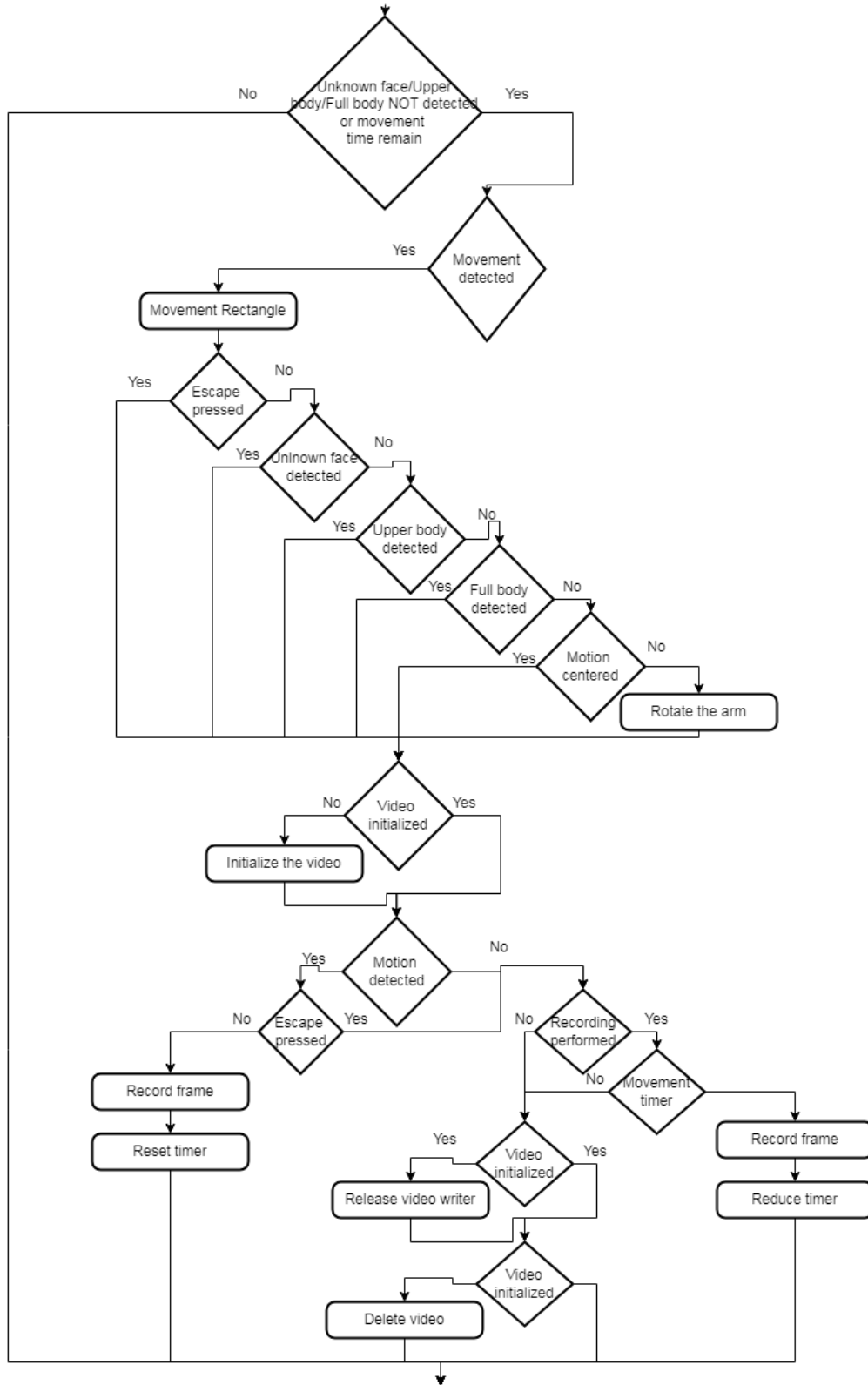
```

655     (contured,contured2) =
cv2.findContours(thresholded.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMP
LE)[-2:]
656     cnts = sorted(contured,key = cv2.contourArea,reverse = True)[:1]
657     for c in cnts:
658         if cv2.contourArea(c) < 10000:
659             continue
660         elif cv2.contourArea(c) >= 10000 and cv2.contourArea(c) <105530:
661             move_det_no = 1
662             motion_detected = True
663             Csent = False
664             (x,y,w,h) = cv2.boundingRect(c)
665             cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)
666             p1_m = x+w
667             p2_m = y+h
668             kenX = int((x+p1_m)/2)
669             kenY = int((y+p2_m)/2)
670             if(escPressed is False):
671                 if(known_face_detected is False and upbody_detected is False and
flbody_detected is False):
672                     if((kenX < minx) and (kenY < miny)):
673                         #print("MOVEMENT UL")
674                         ser.write(b"LN")
675                     if((kenX >= minx and kenX < maxx) and (kenY < miny)):
676                         #print("MOVEMENT U")
677                         ser.write(b"UN")
678                     if((kenX >= maxx) and (kenY < miny)):
679                         #print("MOVEMENT UR")
680                         ser.write(b"RN")
681                     if((kenX < minx) and (kenY >= miny and kenY < maxy)):
682                         #print("MOVEMENT LEFT")
683                         ser.write(b"LN")
684                     if((kenX >= minx and kenX < maxx) and (kenY >= miny and kenY <
maxy)):
685                         #print("MOVEMENT CENTER")
686                         ser.write(b"CN")
687                     if((kenX >= maxx) and (kenY >= miny and kenY < maxy)):
688                         #print("MOVEMENT RIGHT")
689                         ser.write(b"RN")
690                     if((kenX < minx) and (kenY >= maxy)):
691                         #print("MOVEMENT DOWN & LEFT")
692                         ser.write(b"LN")
693                     if((kenX >= minx and kenX < maxx) and (kenY >= maxy)):
694                         #print("MOVEMENT DOWN")
695                         ser.write(b"DN")
696                     if((kenX >= maxx) and (kenY >= maxy)):
697                         #print("MOVEMENT DOWN & RIGHT")
698                         ser.write(b"RN")

```

```
699     if movement_writer is None:
700         movement_filename =
701         "Movement_" + datetime.datetime.now().strftime("%Y_%m_%d_%H_%M_%S")
702         movement_filepath = ("Unknown/Movement/" + movement_filename + ".avi")
703         movement_filepath = movement_filepath.format(movement_filename =
704         movement_filename)
705         (movement_height2, movement_width2) = frame.shape[0:2]
706         movement_writer =
707         cv2.VideoWriter(movement_filepath, fourcc_code, 10, (movement_width2, movement_height2)
708         , True)
709         movement_npZeros = np.zeros((movement_height2, movement_width2), dtype
710         = "uint8")
711         if((movement_idle_time == 80) or (movement_idle_time == 0)):
712             movement_remaining_time = False
713         else:
714             movement_remaining_time = True
715         if (motion_detected and escPressed is False):
716             movement_consecutive_frames += 1
717             if movement_consecutive_frames >= 5:
718                 movement_record_video()
719                 movement_made_recording = True
720                 movement_idle_time = 80
721             else:
722                 if movement_made_recording is True and movement_idle_time > 0:
723                     movement_idle_time -= 1
724                     movement_record_video()
725                 else:
726                     movement_consecutive_frames = 0
727                     if movement_writer is not None:
728                         movement_writer.release()
729                         movement_writer = None
730                     if movement_made_recording is False:
731                         os.remove(movement_filepath)
732                         movement_made_recording = False
733                         movement_idle_time = 80
734         #</Movement Detection>
```

Η συγγραφή του κώδικα έγινε με βάση το παρακάτω διάγραμμα ροής



Εικόνα 6- 22: Διάγραμμα ροής για τον εντοπισμό κίνησης

Αφού τελειώσουν οι έλεγχοι το πρόγραμμα συνεχίζει με τις παρακάτω διεργασίες. Το πρώτο βήμα είναι ο έλεγχος για τις μεταβλητές εντοπισμού περιοχών ενδιαφέροντος και της μεταβλητής τερματισμού κίνησης. Αν είναι όλες οι μεταβλητές σε τιμή False τότε αποστέλλεται μια φόρα ο χαρακτήρας της περιοχής του κέντρου. Αυτό γίνεται ώστε αν να σταματήσει ο βραχίονας την περιστροφή. Το επόμενο μέρος γίνεται ένας ακόμα έλεγχος των μεταβλητών περιοχής ενδιαφέροντος. Αν είναι όλες σε False τότε αποστέλλεται ο χαρακτήρας που δηλώνει ότι δεν υπάρχει περιοχή ενδιαφέροντος στο πεδίο λήψης της κάμερας. το επόμενο βήμα είναι η εμφάνιση του frame.

```
730
731     if(unknown_face_detected is False and upbody_detected is False and
flbody_detected is False and motion_detected is False and Csent is False):
732         ser.write(b"CN")
733         #print("STOP C")
734         Csent = True
735     if(unknown_face_detected is False and upbody_detected is False and
flbody_detected is False and motion_detected is False):
736         ser.write(b"ON")
737         #print("SEBND O")
738
739     cv2.imshow("Camera Data",frame)
```

Μετά την εμφάνιση του frame, γίνεται έλεγχος για το αν ο μικροελεγκτής έχει στείλει δεδομένα. Τα δεδομένα αφορούν την θέση του βραχίονα η οποία παρακολουθείται από τον κωδικοποιητή. Οι τιμές καταγράφονται σε αρχείο. Έτσι την επόμενη φορά που θα εκτελεστεί το πρόγραμμα ο βραχίονας ξέρει ακριβώς που βρίσκεται

```
740
741     if(ser.in_waiting > 0):
742         pos_file = open('EncoderPos.txt','a')
743         num = ser.readline().decode('utf-8').rstrip()
744         #print("Received encoder Position -> "+str(num))
745         pos_file.write(str(num))
746         pos_file.write("\n")
747         pos_file.close()
748
749     #</RECEIVE POSITION FROM ARDUINO>
```

Με το πλήκτρο Escape ενημερώνεται το σύστημα ότι τελειώνει η παρακολούθηση του χώρου. Η τιμή για τον τερματισμό ενημερώνεται με τιμή true και μόλις τελειώσουν όλοι οι μετρητές καταγραφής το σύστημα προχωράει στην τελευταία διαδικασία

```
750
751     if(cv2.waitKey(1) == 27):
752         escPressed = True
753
754     if(face_idle_timer == 80 and upbody_idle_timer == 80 and flbody_idle_timer ==
80 and movement_idle_time == 80 and escPressed is True):
755         break
756
757
```

Η τελευταία διαδικασία αφορά την διαγραφή των βίντεο που δεν έχουν frames. Αυτό προέκυψε καθώς κατά την ανάπτυξη της εφαρμογής ένα λάθος στους κώδικες καταγραφής των βίντεο είχε ως αποτέλεσμα μετά το τέλος του βίντεο να καταγραφεί κενά βίντεο. Το πρόβλημα τελικά λύθηκε, αλλά αποφασίστηκε να μείνει αυτή η διαδικασία ως ένα επιπλέον μέτρο.

```
758 #<DELETE VIDEOS WITH NO FRAMES>
759
760 def checkLength(vid):
761     get_vid = cv2.VideoCapture(vid)
762     frames_count = 0
763     while True:
764         grabbed,frame = get_vid.read()
765         if not grabbed:
766             break
767         frames_count += 1
768     if frames_count == 0:
769         os.remove(vid)
770
771 for video in os.listdir("Unknown/Movement"):
772     if video.endswith(".avi"):
773         vid = os.path.join("Unknown/Movement",video)
774         checkLength(vid)
775
776 for video in os.listdir("Unknown/Faces"):
777     if video.endswith(".avi"):
778         vid = os.path.join("Unknown/Faces",video)
779         checkLength(vid)
780
781 for video in os.listdir("Unknown/UpperBody"):
782     if video.endswith(".avi"):
783         vid = os.path.join("Unknown/UpperBody",video)
784         checkLength(vid)
785
786 for video in os.listdir("Unknown/FullBody"):
787     if video.endswith(".avi"):
788         vid = os.path.join("Unknown/FullBody",video)
789         checkLength(vid)
790 #</DELETE VIDEOS WITH NO FRAMES>
```



```

791
792 camera.release()
793 cv2.destroyAllWindows()
794 #####</CAMERA>#####
795

```

#### 6.4.3.5: Δεδομένα σειριακής επικοινωνίας

Τα δεδομένα που στέλνει ο μικροϋπολογιστής για την περιοχή ενδιαφέροντος με σκοπό την περιστροφή του βραχίονα είναι χαρακτήρες και με βάση αυτούς γίνεται η επικοινωνία

Στον παρακάτω πίνακα συνοψίζονται ο χαρακτήρας που αποστέλλεται σειριακά, η σημασία του καθώς και η αναμενόμενη συμπεριφορά από τον ρομποτικό βραχίονα.

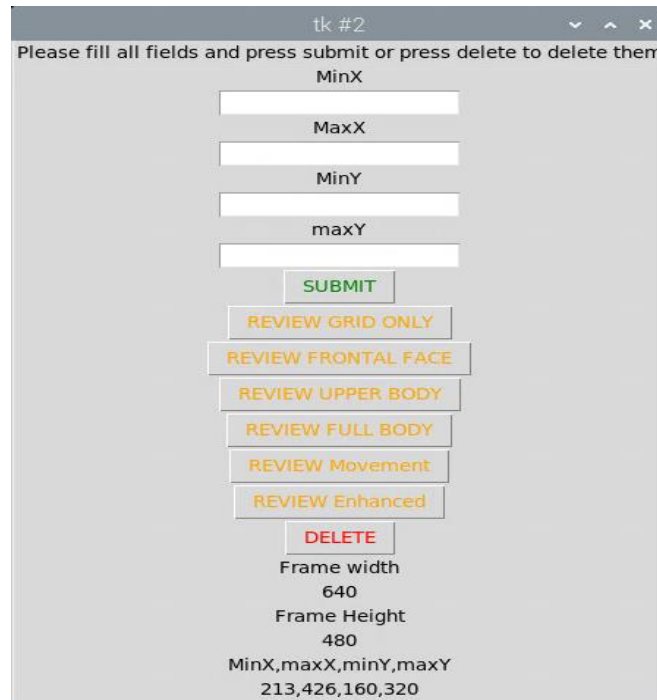
Χαρακτήρας	Σημασία	Αναμενόμενη συμπεριφορά βραχίονα
R	Δεξιά (Right)	Η περιοχή ενδιαφέροντος είναι δεξιά του κεντρικού πλαισίου. Ο βραχίονας στρέφεται δεξιά
L	Αριστερά (Left)	Η περιοχή ενδιαφέροντος βρίσκεται αριστερά του κεντρικού πλαισίου. Ο βραχίονας στρέφεται αριστερά
C	Κέντρο (Center)	Η περιοχή ενδιαφέροντος βρίσκεται στο κεντρικό πλαίσιο Ο βραχίονας δεν κινείται
U	Πάνω (Up)	Η περιοχή ενδιαφέροντος βρίσκεται πάνω από το κεντρικό πλαίσιο Η άρθρωση της κάμερας κινείται προς τα πάνω
D	Κάτω (Down)	Η περιοχή ενδιαφέροντος βρίσκεται κάτω από το κεντρικό πλαίσιο Η άρθρωση της κάμερας κινείται προς τα κάτω
O	Τίποτα	Δεν υπάρχει περιοχή ενδιαφέροντος στο πεδίο λήψης της κάμερας. Γίνεται παρακολούθηση από τους αισθητήρες κίνησης αν υπάρχουν.

**Πίνακας 6- 3: Σύνοψη σειριακής επικοινωνίας**

Ο μικροελεγκτής ανάλογα με την τιμή που θα λάβει κινεί κατάλληλα τον βραχίονα ή βασίζεται στους αισθητήρες κίνησης για να εντοπίσει κινήσεις και να στραφεί προς τον κατάλληλο αισθητήρα με σκοπό να εντοπίσει την περιοχή ενδιαφέροντος

#### 6.5: Ρυθμίσεις

Με το πλήκτρο Settings είναι δυνατή η παραμετροποίηση του διαχωρισμού της οθόνης σε πλαίσια. Με βάση αυτόν τον διαχωρισμό καθορίζεται η κίνηση του βραχίονα



**Εικόνα 6- 23: Το παράθυρο των ρυθμίσεων**

Με την χρήση των παραμέτρων  $minX, maxX, minY, maxY$  το καρέ χωρίζεται σε εννέα πλαίσια. Σε κάθε παραλληλόγραμμο εντοπισμού συγκρίνεται το κέντρο του σε σχέση με το κεντρικό υποπλαίσιο και ο βραχίονας κινείται κατάλληλα ώστε να συμπέσουν το κέντρο του παραλληλογράμμου με το κεντρικό υποπλαίσιο. Οι παράμετροι αποθηκεύονται στην βάση δεδομένων στον πίνακα offset. Ο κώδικας για το παράθυρο είναι ο εξής.

```
797 def Settings():
798     camera = cv2.VideoCapture(0)
799     frame_width = int(camera.get(cv2.CAP_PROP_FRAME_WIDTH))
800     frame_height = int(camera.get(cv2.CAP_PROP_FRAME_HEIGHT))
1318
1319     #<Settings Panel>
1320     form = Tk()
1321     LowerPartForm = Frame(form)
1322     infoLabel = Label(form,text = "Please fill all fields and press submit or press delete to
delete them")
1323     infoLabel.pack()
1324     labelminX = Label(form,text = "MinX",width = 20,font = ("bold",10))
1325     labelminX.pack()
1326     entryMinX = Entry(form)
1327     entryMinX.pack()
1328     labelmaxX = Label(form,text = "MaxX",width = 20,font = ("bold",10))
1329     labelmaxX.pack()
1330     entryMaxX = Entry(form)
1331     entryMaxX.pack()
1332     labelminY = Label(form,text = "MinY",width = 20,font = ("bold",10))
1333     labelminY.pack()
1334     entryminY = Entry(form)
1335     entryminY.pack()
1336     labelMaxY = Label(form,text = "maxY",width = 20,font = ("bold",10))
1337     labelMaxY.pack()
1338     entryMaxY = Entry(form)
1339     entryMaxY.pack()
1340     buttonSubmit = Button(form,text = "SUBMIT",fg = "GREEN",command = sub-
mitData)
1341     buttonSubmit.pack()
1342     buttonReview = Button(form,text = "REVIEW GRID ONLY",fg = "OR-
ANGE",command = gridOnly)
1343     buttonReview.pack()
1344     buttonReview = Button(form,text = "REVIEW FRONTAL FACE",fg = "OR-
ANGE",command = frontalFace)
1345     buttonReview.pack()
1346     buttonReviewup = Button(form,text = "REVIEW UPPER BODY",fg = "OR-
ANGE",command = upperBody)
1347     buttonReviewup.pack()
1348     buttonReviewfb = Button(form,text = "REVIEW FULL BODY",fg = "OR-
ANGE",command = fullBody)
1349     buttonReviewfb.pack()
1350     buttonReviewfb = Button(form,text = "REVIEW Movement",fg = "OR-
ANGE",command = Movement)
1351     buttonReviewfb.pack()
```

```
1352 buttonReviewen = Button(form,text = "REVIEW Enhanced",fg =
"ORANGE",command = Enhanced)
1353 buttonReviewen.pack()
1354 buttonDelete = Button(form,text = "DELETE",fg = "RED",command = deleteData)
1355 buttonDelete.pack()
1356
1357 #<Show already stored values>
1358 labelWidth = Label(form,text = "Frame width")
1359 labelWidth.pack()
1360 labelWidth = Label(form,text = str(frame_width))
1361 labelWidth.pack()
1362 labelHeight = Label(form,text = "Frame Height")
1363 labelHeight.pack()
1364 labelHeight = Label(form,text = str(frame_height))
1365 labelHeight.pack()
1366 offsetsLabel = Label(form,text = "MinX,maxX,minY,maxY")
1367 offsetsLabel.pack()
1368 #<DB Offsets Info>
1369 offsetDB = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy FROM
offset"
1370 sqlCursor.execute(offsetDB)
1371 data2display = sqlCursor.fetchone()
1372 labelData = Label(form,text =
str(data2display[0])+" "+str(data2display[1])+" "+str(data2display[2])+" "+str(data2display[
3]))
1373 labelData.pack()
1374 #</DB Offsets Info>
1375 #</Show already stored values>
1376 #</Settings Panel>
1377#####</SETTINGS>#####
```

### 6.5.1: Εισαγωγή ή διαγραφή παραμέτρων

Για την εισαγωγή παραμέτρων το πρώτο βήμα είναι η συμπλήρωση των πεδίων. Με το πλήκτρο SUBMIT αρχίζει η διαδικασία εισαγωγής των πεδίων. Αρχικά πραγματοποιούνται έλεγχοι ορθής συμπλήρωσης των πεδίων με την παρακάτω σειρά

1. Έλεγχος κενού πεδίου
2. Έλεγχος εισαγωγής ακεραίου αριθμού
3. Έλεγχος εισαγωγής θετικού αριθμού
4. Έλεγχος αν οι ελάχιστες τιμές είναι μικρότερες από τις μέγιστες τιμές
5. Έλεγχος αν οι μέγιστες τιμές είναι μικρότερες από το μέγεθος του καρέ σε ύψος και πλάτος

Σε περίπτωση που κάποιος έλεγχος αποτύχει εμφανίζεται μήνυμα λάθους που ενημερώνει τον χρήστη ποιο πεδίο είναι και ποιος έλεγχος απέτυχε. Μόλις ολοκληρωθούν επιτυχώς οι έλεγχοι διαγράφονται τα παλιά δεδομένα και εισάγονται τα καινούρια

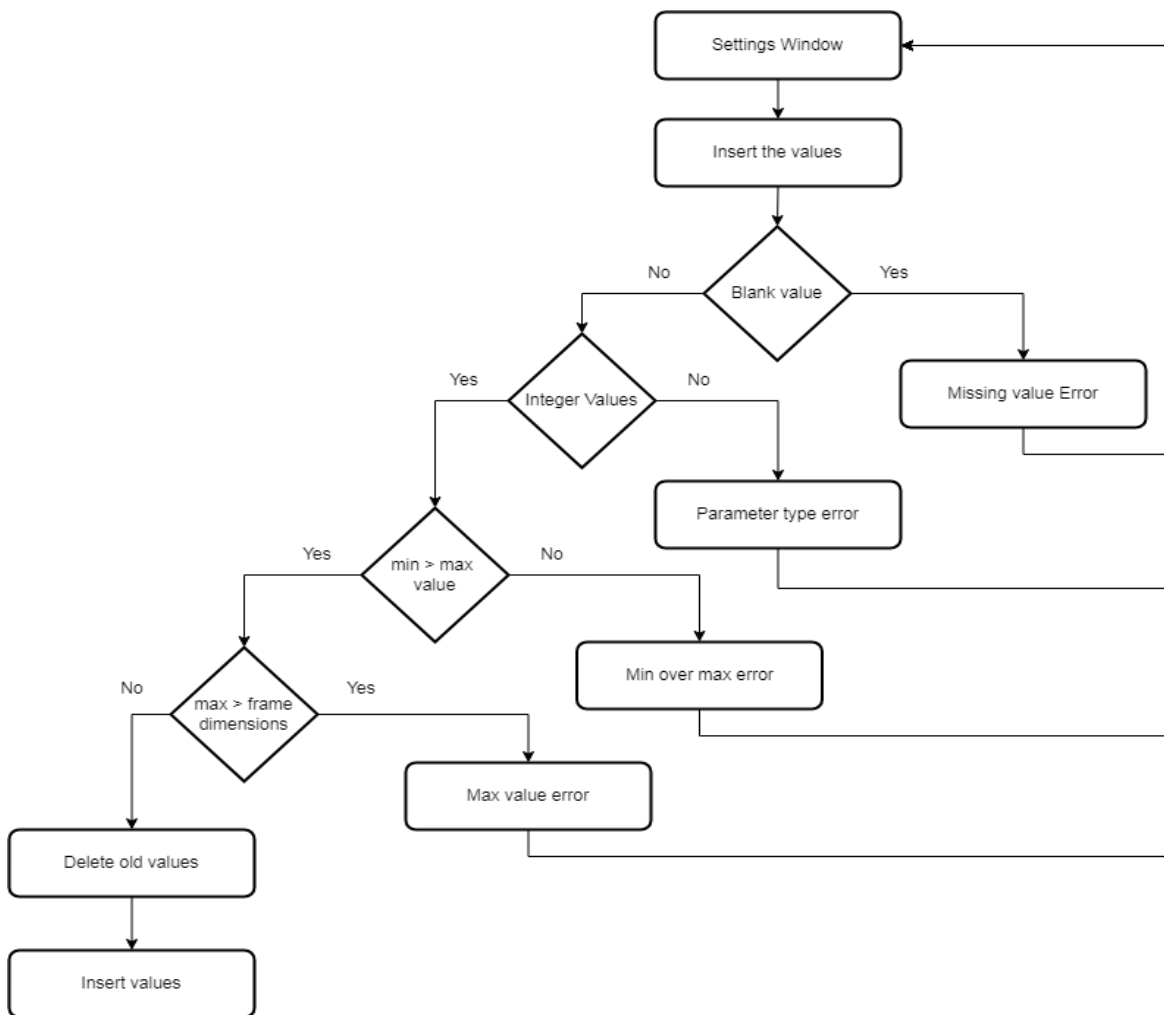
Η διαδικασία που εισάγει τα δεδομένα είναι η παρακάτω

```
802 def submitData():
803     #<Get the parameters>
804     minx = entryMinX.get()
805     maxx = entryMaxX.get()
806     miny = entryminY.get()
807     maxy = entryMaxY.get()
808     #</Get the parameters>
809     #<Parameters Check>
810     #<Check if the fields are blank>
811     if((len(minx) == 0) or (len(maxx) == 0) or (len(miny) == 0) or (len(maxy) == 0)):
812         messagebox.showerror("Parameter Blank Error", "A parameter is blank")
813         return
814     #<Check if the fields are blank>
815     #<Check if the values are integers>
816     try:
817         intminx = int(minx)
818     except ValueError:
819         messagebox.showerror("Parameter Type Error", "minX is not an integer")
820         return
821     try:
822         intmaxx = int(maxx)
823     except ValueError:
824         messagebox.showerror("Parameter Type Error", "maxX is not an integer")
825         return
826     try:
827         intminy = int(miny)
828     except ValueError:
829         messagebox.showerror("Parameter Type Error", "minY is not an integer")
830         return
831     try:
832         intmaxy = int(maxy)
833     except ValueError:
834         messagebox.showerror("Parameter Type Error", "maxY is not an integer")
835         return
836     #</Check if the values are integers>
837     #<Check if values are >= 0>
838     if((intminx < 0) or (intmaxx < 0) or (intminy < 0) or (intmaxy < 0)):
839         messagebox.showerror("Negative Value Error", "A parameter is a negative num-
ber")
840         return
841     #</Check if values are >= 0>
842     #<Check if min values are >= than max values>
843     if((intminx >= intmaxx) or (intminy >= intmaxy)):
844         messagebox.showerror("Min-Max Value Error", "a Min Value is grater than a
Max Value")
845         return
846     #</Check if min values are >= than max values>
847     #<Check if max values exceed the frame width or height>
848     if((intmaxx > frame_width) or (intmaxy > frame_height)):
849         messagebox.showerror("Max Value Error", "A max value exceeds the frame width
or height")
```

```

850     return
851     #</Check if max values exceed the frame width or height>
852     #</Parameters Check>
853     #<Delete old values from the database>
854     deleteQuery = "DELETE FROM offset"
855     sqlCursor.execute(deleteQuery)
856     conn.commit()
857     #</Delete old values from the database>
858     #<Insert the new values to DB>
859     insertQuery = "INSERT INTO
offset(offset_minx,offset_maxx,offset_miny,offset_maxy) VALUES(%s,%s,%s,%s)"
860
861     offsetValues = (intminx,intmaxx,intminy,intmaxy)
862     sqlCursor.execute(insertQuery,offsetValues)
863     conn.commit()
864     #</Insert the new values to DB>
865     #</Submit Data>
    
```

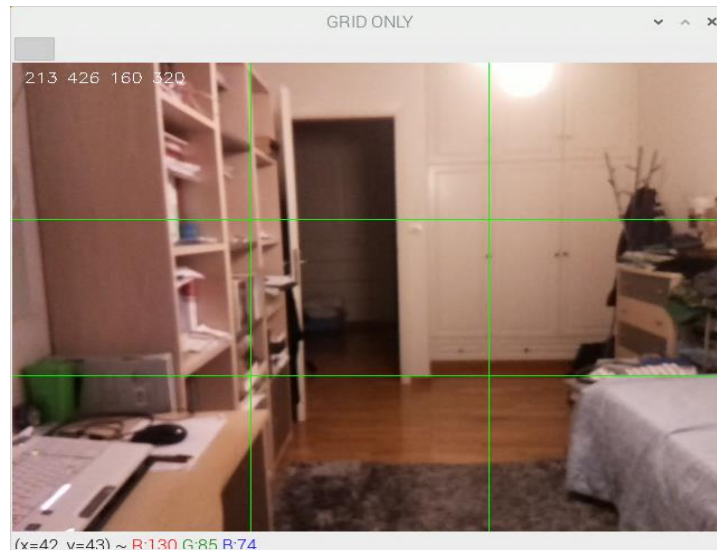
Το διάγραμμα ροής για την παραπάνω διαδικασία είναι το εξής



Εικόνα 6- 24: Εισαγωγή δεδομένων

### 6.5.2: Εμφάνιση πλέγματος

Με το πλήκτρο Review Grid Only εμφανίζεται το παράθυρο της κάμερας με πράσινες γραμμές τα πλαίσια που έχουν δημιουργηθεί. Με αυτό το παράθυρο ο χρήστης μπορεί να δει την παραμετροποίηση των πλαισίων χωρίς τα παραλληλόγραμμα από πρόσωπα, σώματα ή κινήσεις. Οι γραμμές μετατοπίζονται ανάλογα με τις τιμές των παραμέτρων



Εικόνα 6- 25: Εμφάνιση απλού πλέγματος

Η εμφάνιση του πλέγματος γίνεται από το παρακάτω κομμάτι κώδικα

```
866
867 #<Review Grid>
868 def gridOnly():
869     camera = cv2.VideoCapture(0)
870     #<Check if offset are loaded>
871     checkQuery = "SELECT COUNT(*) FROM offset"
872     sqlCursor.execute(checkQuery)
873     return_data = sqlCursor.fetchone()
874     if(return_data[0] == 0):
875         messagebox.showerror("OFFSET DATA ERROR", "No offset data found")
876         return
877     #</Check if offset are loaded>
878     #<Get offset Data>
879     retrieveQuery = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy FROM
offset"
880     sqlCursor.execute(retrieveQuery)
881     offsetDataDB = sqlCursor.fetchall()
```

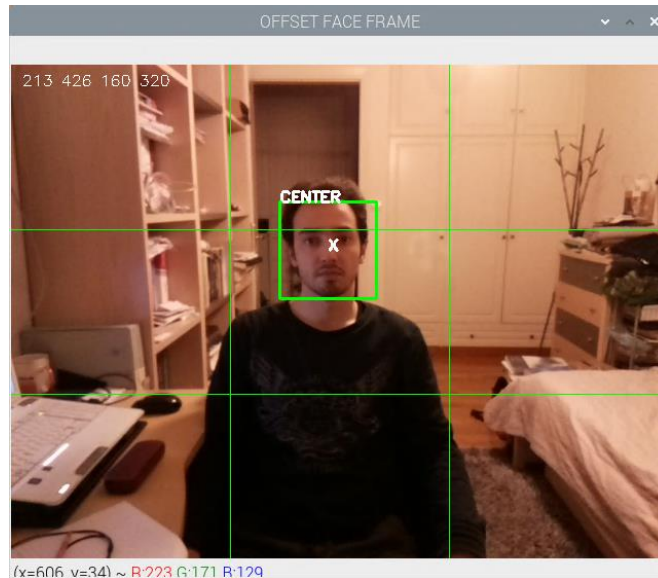
```
882     for offsetData in offsetDataDB:
883         intminx = offsetData[0]
884         intmaxx = offsetData[1]
885         intminy = offsetData[2]
886         intmaxy = offsetData[3]
887     cor_min_x_stt = (intminx,0)
888     cor_min_x_end = (intminx,frame_height)
889     cor_max_x_stt = (intmaxx,0)
890     cor_max_x_end = (intmaxx,frame_height)
891     cor_min_y_stt = (0,intminy)
892     cor_min_y_end = (frame_width,intminy)
893     cor_max_y_stt = (0,intmaxy)
894     cor_max_y_end = (frame_width,intmaxy)
895
896     while True:
897         gr,frame = camera.read()
898         cv2.putText(frame,(str(intminx)+" "+str(intmaxx)+" "+str(intminy)+"
"+str(intmaxy)),(10,20), cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)
899         cv2.line(frame,cor_min_x_stt,cor_min_x_end,(0,255,0),1)
900         cv2.line(frame,cor_max_x_stt,cor_max_x_end,(0,255,0),1)
901         cv2.line(frame,cor_min_y_stt,cor_min_y_end,(0,255,0),1)
902         cv2.line(frame,cor_max_y_stt,cor_max_y_end,(0,255,0),1)
903         cv2.imshow("GRID ONLY",frame)
904         if(cv2.waitKey(1) == 27):
905             break
906         camera.release()
907         cv2.destroyAllWindows()
908     #</Get offset Data>
909     #<Review Grid>
```

### 6.5.3: Εμφάνιση πλέγματος με εντοπισμό προσώπου

Με το πλήκτρο Review Frontal Face εμφανίζεται η εικόνα με το πλέγμα. Σε περίπτωση που εντοπιστούν πρόσωπα στο πεδίο λήψης της κάμερας εμφανίζεται ένα πράσινο παραλληλόγραμμο με ένα άσπρο X στο κέντρο του. Πάνω από το παραλληλόγραμμο εμφανίζεται το αντίστοιχο μήνυμα.

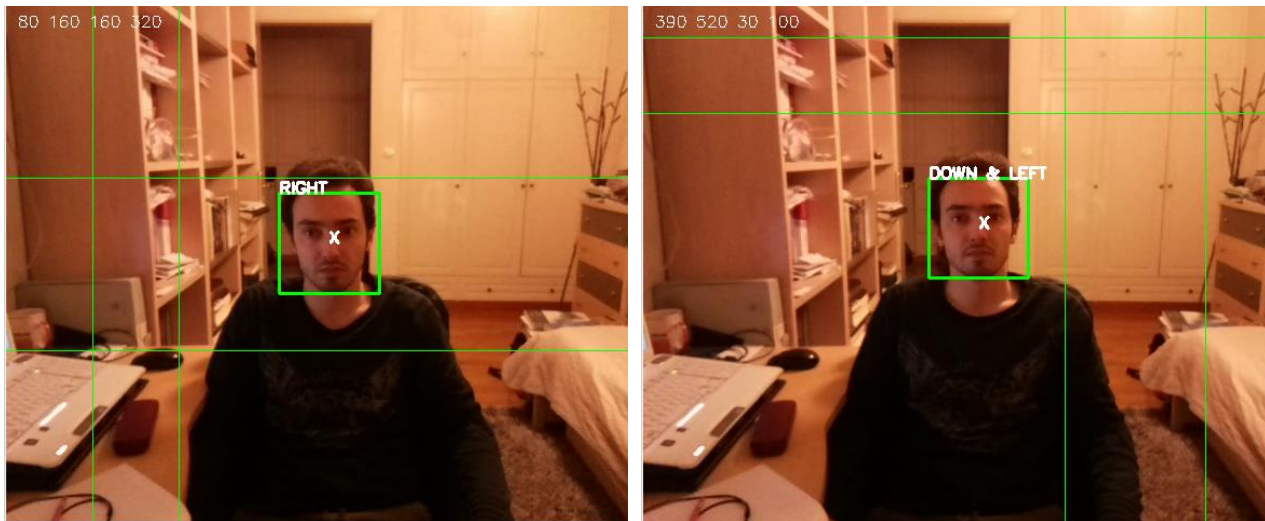


Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...



**Εικόνα 6- 26: Οπτική αναπαράσταση πλέγματος με πρόσωπο**

Αλλάζοντας τις τιμές των παραμέτρων μετατοπίζονται οι γραμμές. Επίσης αλλάζει το μήνυμα που εμφανίζεται ανάλογα με το κέντρο της περιοχής ενδιαφέροντος



**Εικόνα 6- 27: Εντοπισμός προσώπων για διαφορετικές τιμές παραμέτρων**

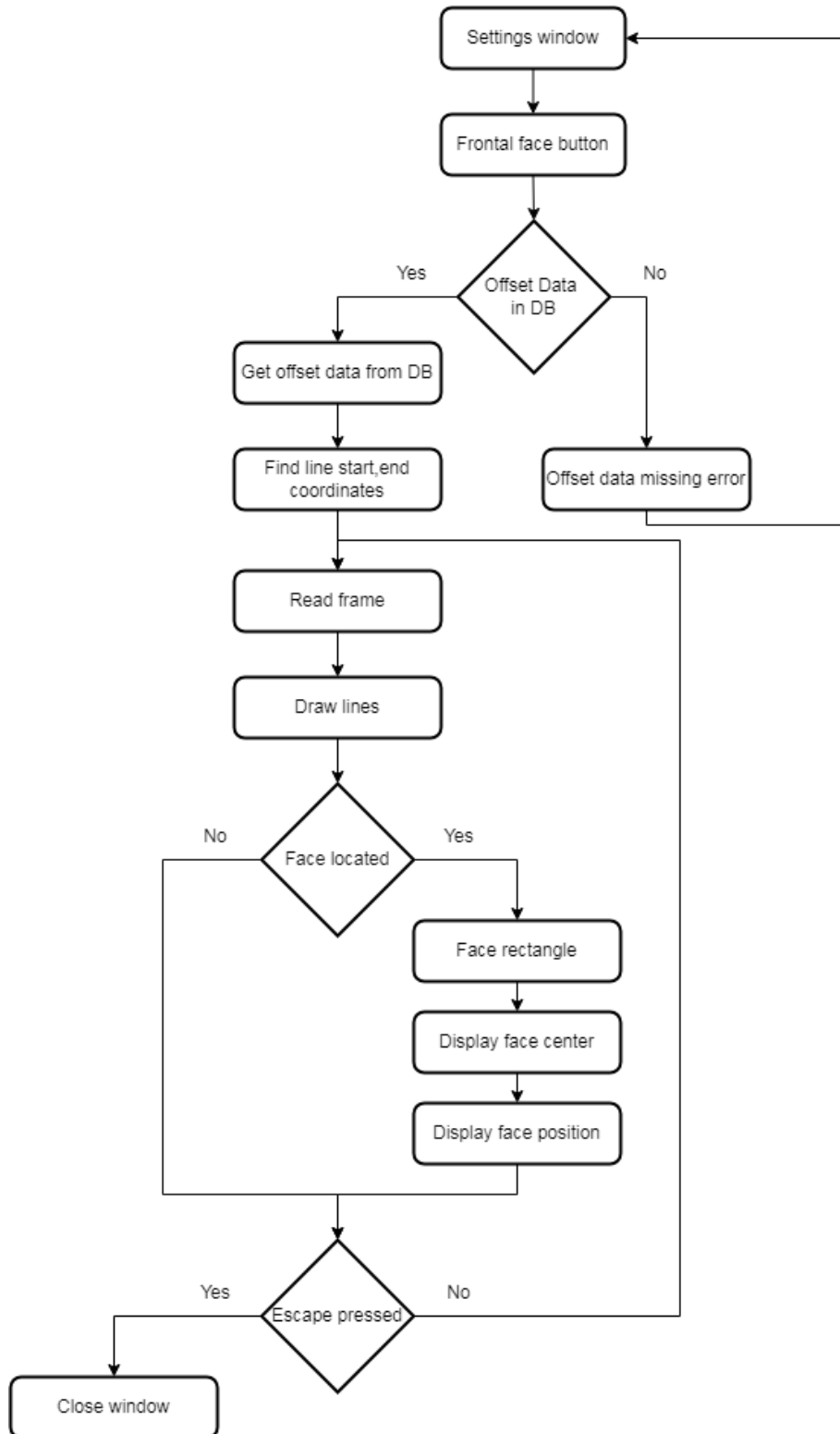
Ο κώδικας για την εμφάνιση καθώς και το διάγραμμα ροής των αποτελεσμάτων είναι τα εξής:

```

911 def frontalFace():
912     camera = cv2.VideoCapture(0)
913     #<Check if offset Data exist>
914     checkQuery = "SELECT COUNT(*) FROM offset"
915     sqlCursor.execute(checkQuery)
916     return_data = sqlCursor.fetchone()
917     if(return_data[0] == 0):
918         messagebox.showerror("OFFSET DATA ERROR","No offset data found")
919     return
920     #</Check if offset Data exist>
921     retrieveQuery = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy FROM
offset"
922     sqlCursor.execute(retrieveQuery)
923     offsetDataDB = sqlCursor.fetchall()
924     for offsetData in offsetDataDB:
925         intminx = offsetData[0]
926         intmaxx = offsetData[1]
927         intminy = offsetData[2]
928         intmaxy = offsetData[3]
929         cor_min_x_stt = (intminx,0)
930         cor_min_x_end = (intminx,frame_height)
931         cor_max_x_stt = (intmaxx,0)
932         cor_max_x_end = (intmaxx,frame_height)
933         cor_min_y_stt = (0,intminy)
934         cor_min_y_end = (frame_width,intminy)
935         cor_max_y_stt = (0,intmaxy)
936         cor_max_y_end = (frame_width,intmaxy)
937     while True:
938         gr,frame = camera.read()
939         cv2.line(frame,cor_min_x_stt,cor_min_x_end,(0,255,0),1)
940         cv2.line(frame,cor_max_x_stt,cor_max_x_end,(0,255,0),1)
941         cv2.line(frame,cor_min_y_stt,cor_min_y_end,(0,255,0),1)
942         cv2.line(frame,cor_max_y_stt,cor_max_y_end,(0,255,0),1)
943         cv2.putText(frame,(str(intminx)+" "+str(intmaxx)+" "+str(intminy)+"
"+str(intmaxy)),(10,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)
944         gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
945         faces = faceDe-
tect.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=cv2.CAS
CADE_SCALE_IMAGE)
946         for(p1,p2,m,i) in faces:
947             cv2.rectangle(frame,(p1,p2),(p1+m,p2+i),(0,255,0),2)
948             p1_m = p1+m
949             p2_i = p2+i
950             kenX = int((p1+p1_m)/2)
951             kenY = int((p2+p2_i)/2)
952             cv2.putText(frame,str("X"),(kenX,kenY),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
953             if((kenX < intminx) and (kenY < intminy)):
954                 cv2.putText(frame,str("UP &
LEFT"),(p1,p2),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
955             if((kenX >= intminx and kenX < intmaxx) and (kenY < intminy)):

```

```
956 cv2.putText(frame, str("UP"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2
)
957     if((kenX >= intmaxx) and (kenY < intminy)):
958         cv2.putText(frame, str("UP &
RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
959     if((kenX < intminx) and (kenY >= intminy and kenY < intmaxy)):
960 cv2.putText(frame, str("LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255
), 2)
961     if((kenX >= intminx and kenX < intmaxx) and (kenY >= intminy and kenY <
intmaxy)):
962 cv2.putText(frame, str("CENTER"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255,
255), 2)
963     if((kenX >= intmaxx) and (kenY >= intminy and kenY < intmaxy)):
964 cv2.putText(frame, str("RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 25
5), 2)
965     if((kenX < intminx) and (kenY >= intmaxy)):
966         cv2.putText(frame, str("DOWN &
LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
967     if((kenX >= intminx and kenX < intmaxx) and (kenY >= intmaxy)):
968 cv2.putText(frame, str("DOWN"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 25
5), 2)
969     if((kenX >= intmaxx) and (kenY >= intmaxy)):
970         cv2.putText(frame, str("DOWN &
RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
971     cv2.imshow("OFFSET FACE FRAME", frame)
972     if cv2.waitKey(1) == 27:
973         break
974     camera.release()
975     cv2.destroyAllWindows()
976
977 #<Review frontalFace>
```

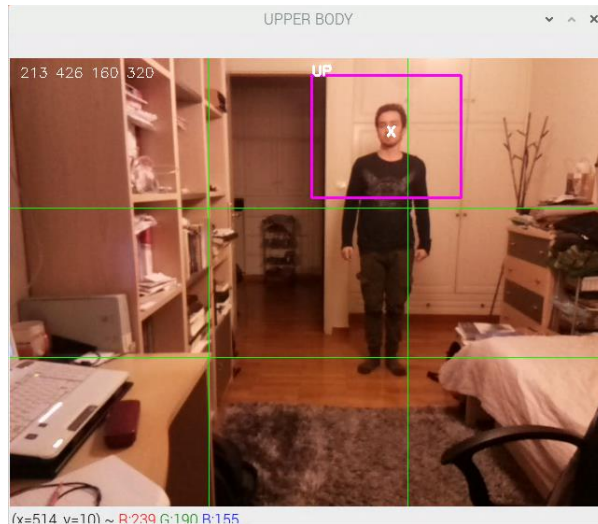


Εικόνα 6- 28: Διάγραμμα ροής παραμετροποίησης με πρόσωπα

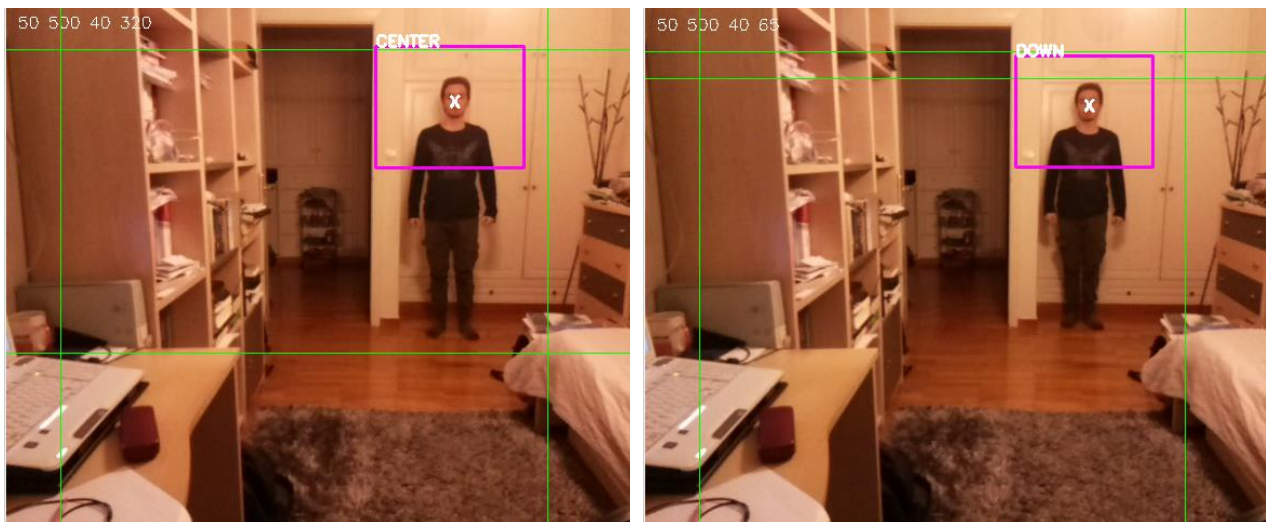
Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

#### 6.5.4: Εμφάνιση πλέγματος με εντοπισμό άνω μέρους σώματος

Με την επιλογή Review Upper Body εμφανίζεται το παράθυρο με τις γραμμές διαχωρισμού των πλαισίων. Κάθε εντοπισμός άνω μέρους σώματος εμφανίζεται με ένα μωβ παραλληλόγραμμο με ένα X στο κέντρο καθώς και ένα μήνυμα που δείχνει σε ποιο υποπλαίσιο βρίσκεται το κέντρο του.



Εικόνα 6- 29: Αναπαράσταση εντοπισμού άνω μέρους σώματος



Εικόνα 6- 30: Αναπαράσταση άνω μέρους σώματος για διαφορετικές τιμές παραμέτρων

Ακολουθούν ο κώδικας και το διάγραμμα ροής

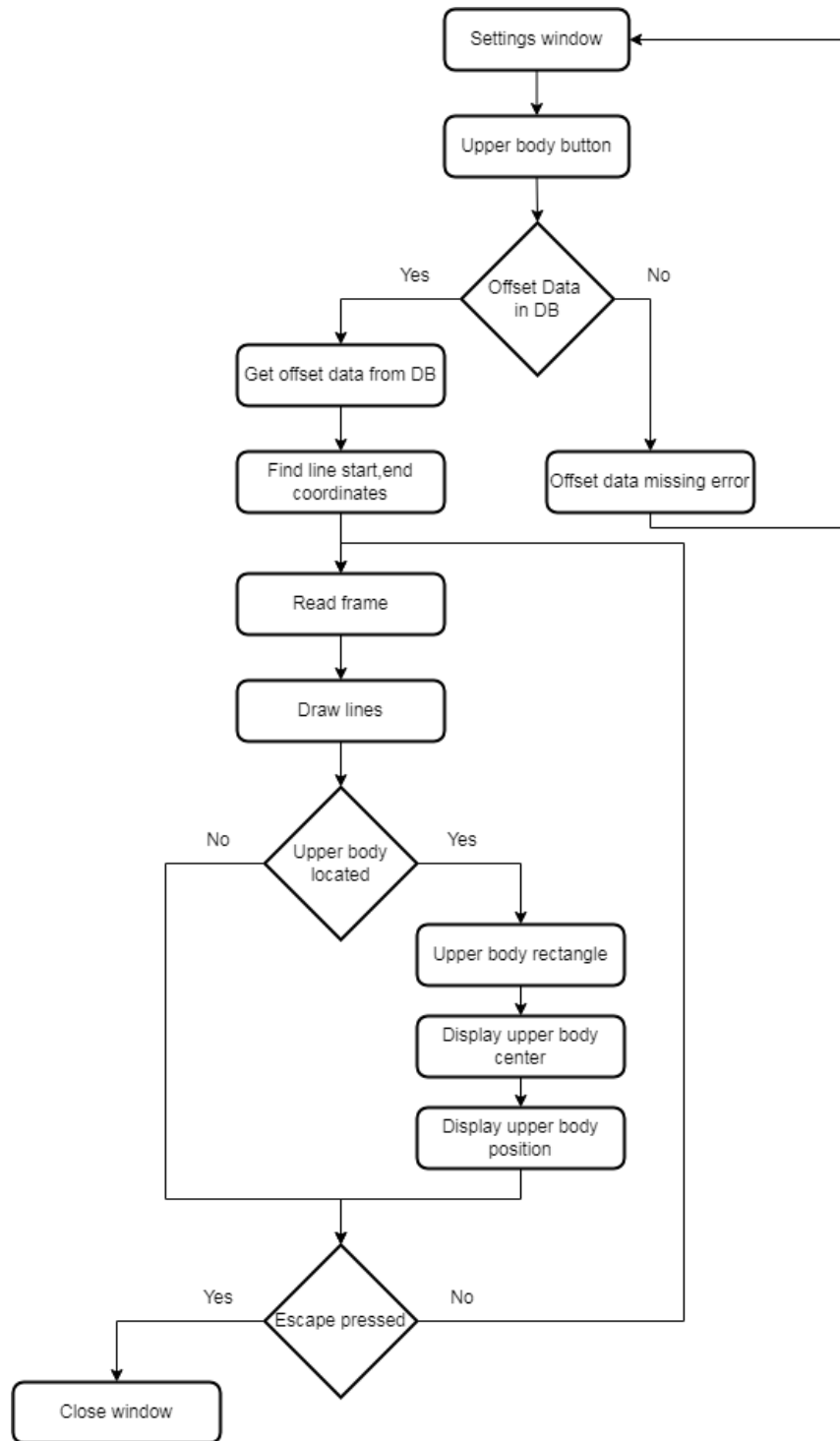
```

978
979 #<Review upperBody>
980 def upperBody():
981     #ubodyDetect
982     camera = cv2.VideoCapture(0)
983     #<Check if offset Data exist>
984     checkQuery = "SELECT COUNT(*) FROM offset"
985     sqlCursor.execute(checkQuery)
986     check_data = sqlCursor.fetchone()
987     if(check_data[0] == 0):
988         messagebox.showerror("OFFSET DATA ERROR", "No offset Data found")
989     return
990     #<Check if offset Data exist>*
991     #<Get the offset Data>
992     getDataQuery = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy
FROM offset"
993     sqlCursor.execute(getDataQuery)
994     offsetDataDB = sqlCursor.fetchall()
995     for offsetData in offsetDataDB:
996         intminX = offsetData[0]
997         intmaxX = offsetData[1]
998         intminY = offsetData[2]
999         intmaxY = offsetData[3]
1000    #</Get the offset Data>
1001    #<Create the coordinate lines>
1002    cor_min_x_stt = (intminX,0)
1003    cor_min_x_end = (intminX,frame_height)
1004    cor_max_x_stt = (intmaxX,0)
1005    cor_max_x_end = (intmaxX,frame_height)
1006    cor_min_y_stt = (0,intminY)
1007    cor_min_y_end = (frame_width,intminY)
1008    cor_max_y_stt = (0,intmaxY)
1009    cor_max_y_end = (frame_width,intmaxY)
1010    #</Create the coordinate lines>
1011    #<Display Info>
1012    while True:
1013        gr,frame = camera.read()
1014        cv2.putText(frame,(str(intminX)+" "+str(intmaxX)+" "+str(intminY)+"
"+str(intmaxY)),(10,20), cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)
1015        cv2.line(frame,cor_min_x_stt,cor_min_x_end,(0,255,0),1)
1016        cv2.line(frame,cor_max_x_stt,cor_max_x_end,(0,255,0),1)
1017        cv2.line(frame,cor_min_y_stt,cor_min_y_end,(0,255,0),1)
1018        cv2.line(frame,cor_max_y_stt,cor_max_y_end,(0,255,0),1)
1019        gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
1020        upbody =
ubodyDetect.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=
cv2.CASCADE_SCALE_IMAGE)
1021        for(p1,p2,m,i) in upbody:
1022            cv2.rectangle(frame,(p1,p2),(p1+m,p2+i),(255,0,255),2)
1023            p1_m = p1+m
1024            p2_i = p2+i

```



```
1025     kenX = int((p1+p1_m)/2)
1026     kenY = int((p2+p2_i)/2)
1027
cv2.putText(frame, str("X"), (kenX, kenY), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1028     if((kenX < intminX and (kenY < intminY)):
1029         cv2.putText(frame, str("UP &
LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1030     if((kenX >= intminX and kenX < intmaxX) and (kenY < intminY)):
1031
cv2.putText(frame, str("UP"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
)
1032     if((kenX >= intmaxX) and (kenY < intminY)):
1033         cv2.putText(frame, str("UP &
RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1034     if((kenX < intminX) and (kenY >= intminY) and kenY < intmaxY)):
1035
cv2.putText(frame, str("LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
)
1036     if((kenX >= intminX) and kenX < intmaxX) and (kenY >= intminY) and kenY < intmaxY)):
1037
cv2.putText(frame, str("CENTER"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
)
1038     if((kenX >= intmaxX) and (kenY >= intminY) and kenY < intmaxY)):
1039
cv2.putText(frame, str("RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
)
1040     if((kenX < intminX) and (kenY >= intmaxY)):
1041         cv2.putText(frame, str("DOWN &
LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1042     if((kenX >= intminX) and kenX < intmaxX) and (kenY >= intmaxY)):
1043
cv2.putText(frame, str("DOWN"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
)
1044     if((kenX >= intmaxX) and (kenY >= intmaxY)):
1045         cv2.putText(frame, str("DOWN &
RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1046     cv2.imshow("UPPER BODY", frame)
1047     if cv2.waitKey(1) == 27:
1048         break
1049     camera.release()
1050     cv2.destroyAllWindows()
1051     #</Display Info>
1052     #<Review upperBody>
```



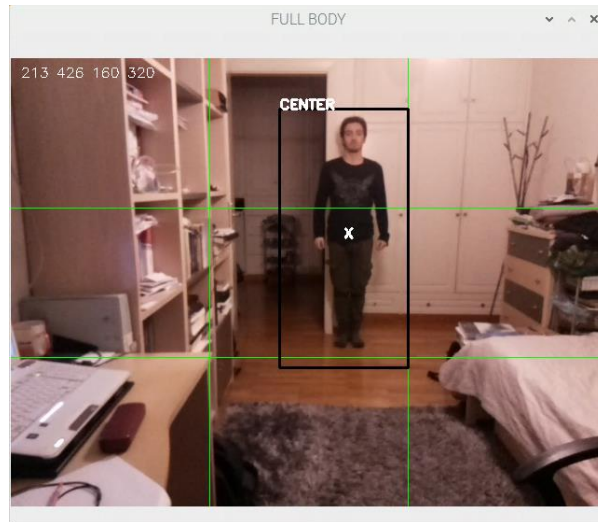
**Εικόνα 6- 31: Παραμετροποίηση άνω μέρους σώματος**



Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

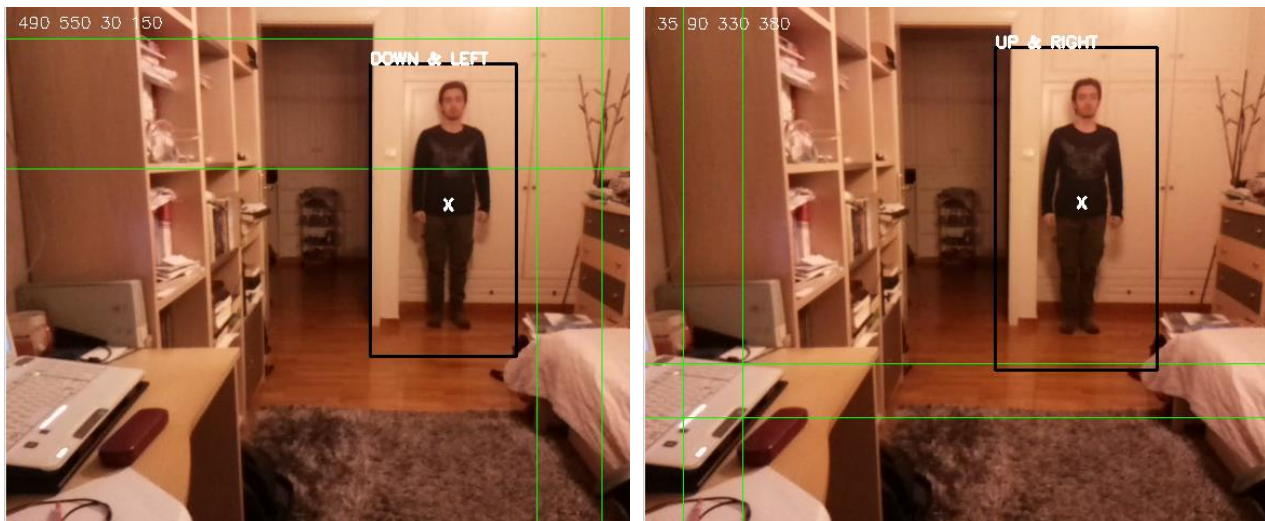
### 6.5.5: Εμφάνιση πλέγματος με εντοπισμό πλήρους σώματος

Με την επιλογή Review full body εμφανίζεται το παράθυρο με τις γραμμές που διαχωρίζουν τα πλαίσια. Κάθε πλήρες σώματος που εντοπίζεται στην οθόνη εμφανίζεται ένα μαύρο παραλληλόγραμμο, καθώς και μήνυμα που δείχνει σε ποιο υποπλαίσιο βρίσκεται το κέντρο του.



Εικόνα 6- 32: Εντοπισμός πλήρους σώματος

Με την αλλαγή των παραμέτρων οι γραμμές δείχνουν την διαίρεση της οθόνης σε πλαίσια



Εικόνα 6- 33: Παραδείγματα με διαφορετικές τιμές παραμέτρων

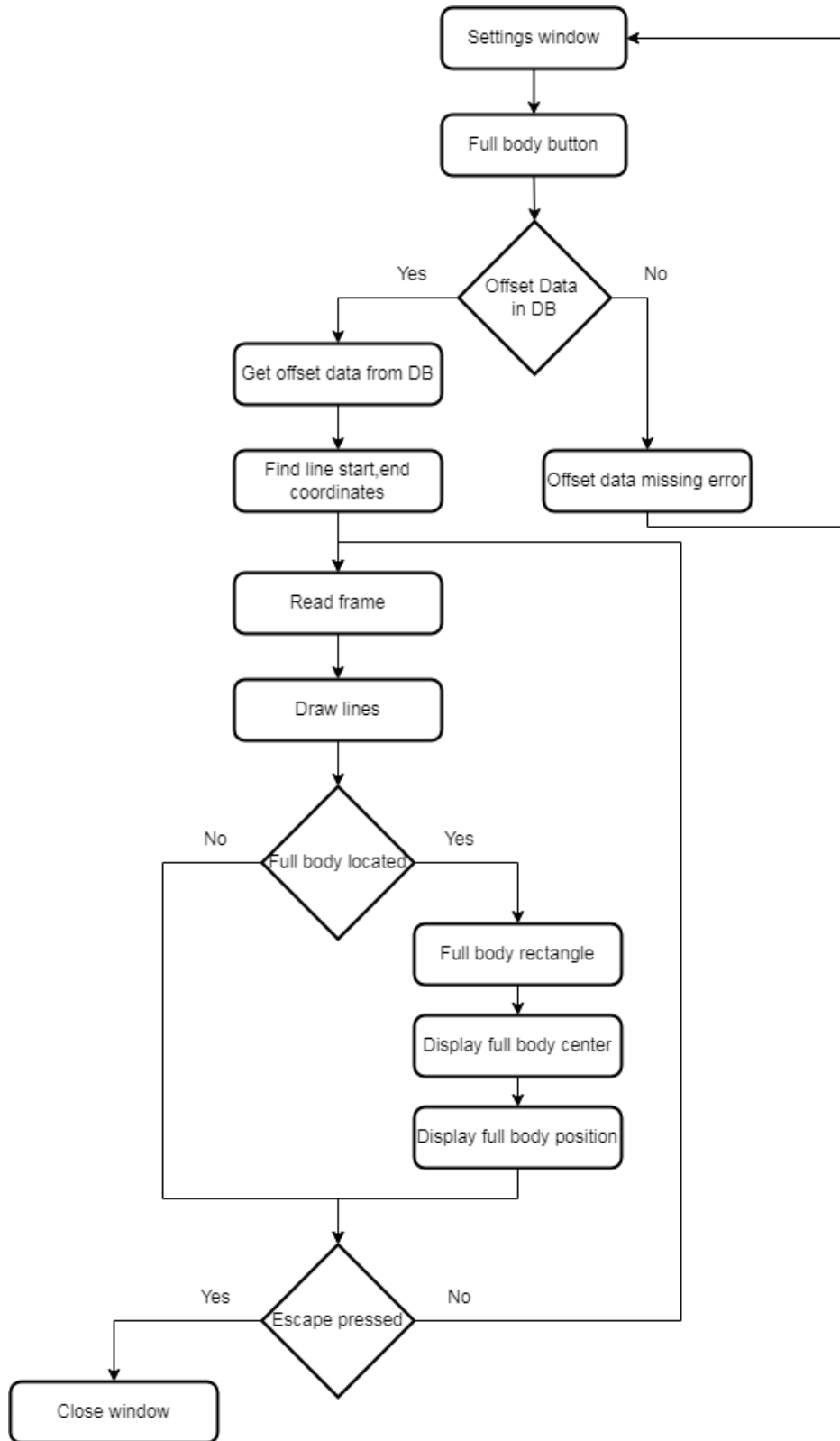
Ακλουθεί ο κώδικας και το σχετικό διάγραμμα ροής

```

1053
1054 #<Review fullBody>
1055 def fullBody():
1056     #bodyDetect
1057     camera = cv2.VideoCapture(0)
1058     #<Check if the offset data exist in the Database>
1059     checkQuery = "SELECT COUNT(*) FROM offset"
1060     sqlCursor.execute(checkQuery)
1061     return_data = sqlCursor.fetchone()
1062     if(return_data[0] == 0):
1063         messagebox.showerror("OFFSET DATA ERROR","No offset data found")
1064     return
1065 #</Check if the offset data exist in the Database>
1066 #<Get the offset Data>
1067     retrieveQuery = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy
FROM offset"
1068     sqlCursor.execute(retrieveQuery)
1069     offsetDataDB = sqlCursor.fetchall()
1070     for offsetData in offsetDataDB:
1071         intminx = offsetData[0]
1072         intmaxx = offsetData[1]
1073         intminy = offsetData[2]
1074         intmaxy = offsetData[3]
1075     #</Get the offset Data>
1076     cor_min_x_stt = (intminx,0)
1077     cor_min_x_end = (intminx,frame_width)
1078     cor_max_x_stt = (intmaxx,0)
1079     cor_max_x_end = (intmaxx,frame_height)
1080     cor_min_y_stt = (0,intminy)
1081     cor_min_y_end = (frame_width,intminy)
1082     cor_max_y_stt = (0,intmaxy)
1083     cor_max_y_end = (frame_width,intmaxy)
1084     while True:
1085         gr,frame = camera.read()
1086         cv2.putText(frame,(str(intminx)+" "+str(intmaxx)+" "+str(intminy)+"
"+str(intmaxy)),(10,20), cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)
1087         cv2.line(frame,cor_min_x_stt,cor_min_x_end,(0,255,0),1)
1088         cv2.line(frame,cor_max_x_stt,cor_max_x_end,(0,255,0),1)
1089         cv2.line(frame,cor_min_y_stt,cor_min_y_end,(0,255,0),1)
1090         cv2.line(frame,cor_max_y_stt,cor_max_y_end,(0,255,0),1)
1091         gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
1092         fbody =
fbodyDetect.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=
cv2.CASCADE_SCALE_IMAGE)
1093         for(p1,p2,m,i) in fbody:
1094             cv2.rectangle(frame,(p1,p2),(p1+m,p2+i),(0,0,0),2)
1095             p1_m = p1+m
1096             p2_i = p2+i
1097             kenX = int((p1+p1_m)/2)
1098             kenY = int((p2+p2_i)/2)

```

```
1099     cv2.putText(frame, str("X"), (kenX, kenY),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1100     if((kenX < intminx) and (kenY < intminy)):
1101         cv2.putText(frame, str("UP &
LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1102         if((kenX >= intminx and kenX < intmaxx) and (kenY < intminy)):
1103
cv2.putText(frame, str("UP"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2
)
1104         if((kenX >= intmaxx) and (kenY < intminy)):
1105             cv2.putText(frame, str("UP &
RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1106             if((kenX < intminx) and (kenY >= intminy and kenY < intmaxy)):
1107
cv2.putText(frame, str("LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255
), 2)
1108             if((kenX >= intminx and kenX < intmaxx) and (kenY >= intminy and kenY <
intmaxy)):
1109
cv2.putText(frame, str("CENTER"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255,
255), 2)
1110             if((kenX >= intmaxx) and (kenY >= intminy and kenY < intmaxy)):
1111
cv2.putText(frame, str("RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 25
5), 2)
1112             if((kenX < intminx) and (kenY >= intmaxy)):
1113                 cv2.putText(frame, str("DOWN &
LEFT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1114                 if((kenX >= intminx and kenX < intmaxx) and (kenY >= intmaxy)):
1115
cv2.putText(frame, str("DOWN"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 25
5), 2)
1116                 if((kenX >= intmaxx) and (kenY >= intmaxy)):
1117                     cv2.putText(frame, str("DOWN &
RIGHT"), (p1, p2), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)
1118
1119     cv2.imshow("FULL BODY", frame)
1120     if cv2.waitKey(1) == 27:
1121         break
1122     camera.release()
1123     cv2.destroyAllWindows()
1124     #<Review fullBody>
```

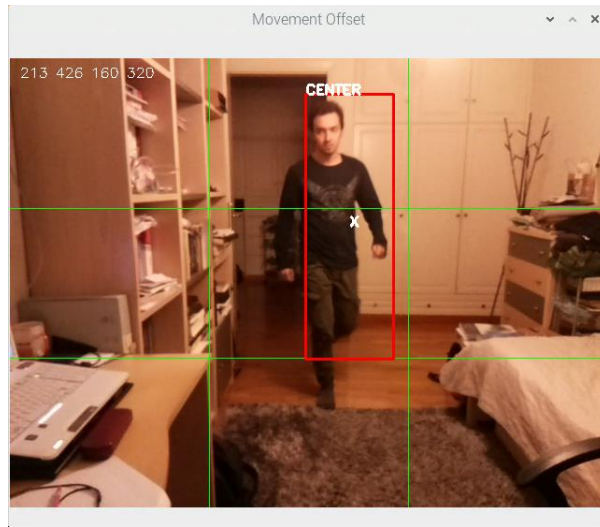


Εικόνα 6- 34: Το διάγραμμα ροής για τον εντοπισμό πλήρους σώματος

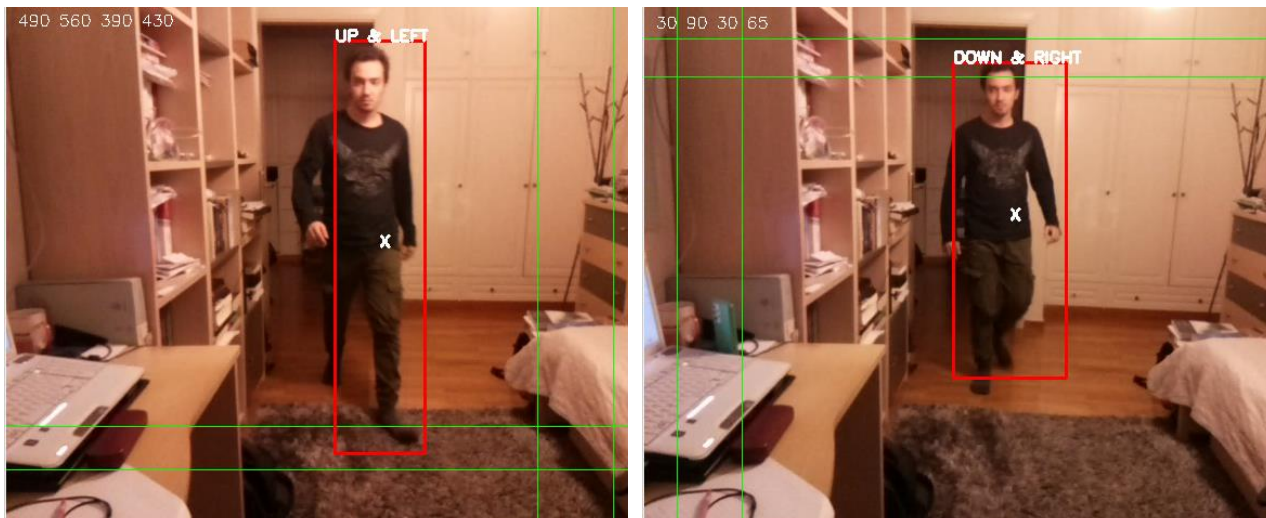
Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

### 6.5.6: Εμφάνιση πλέγματος με εντοπισμό κίνησης

Με την επιλογή Review movement εμφανίζεται το παράθυρο με τις γραμμές που διαχωρίζουν τα πλαίσια. Κάθε κίνηση που εντοπίζεται στην οθόνη εμφανίζεται ένα παραλληλόγραμμο, καθώς και μήνυμα που δείχνει σε ποιο υποπλαίσιο βρίσκεται το κέντρο του.



Εικόνα 6- 35: Εντοπισμός κίνησης



Εικόνα 6- 36: Παραδείγματα με διαφορετικές τιμές παραμέτρων

Ο κώδικας για την εμφάνιση των παραμέτρων και τον εντοπισμό κίνησης είναι ο παρακάτω

```

1126 #<Review Movement>
1127 def Movement():
1128     #<Check if offset Data exist>
1129     camera = cv2.VideoCapture(0)
1130     checkQuery = "SELECT COUNT(*) FROM offset"
1131     sqlCursor.execute(checkQuery)
1132     return_data = sqlCursor.fetchone()
1133     if(return_data[0] == 0):
1134         messagebox.showerror("OFFSET DATA ERROR","No offset data found")
1135     return
1136 #</Check if offset Data exist>
1137 #<Get the offset data>
1138 retrieveQuery = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy
FROM offset"
1139 sqlCursor.execute(retrieveQuery)
1140 offsetDataDB = sqlCursor.fetchall()
1141 for offsetData in offsetDataDB:
1142     intminX = offsetData[0]
1143     intmaxX = offsetData[1]
1144     intminY = offsetData[2]
1145     intmaxY = offsetData[3]
1146 #</Get the offset data>
1147 cor_min_x_stt = (intminX,0)
1148 cor_min_x_end = (intminX,frame_height)
1149 cor_max_x_stt = (intmaxX,0)
1150 cor_max_x_end = (intmaxX,frame_height)
1151 cor_min_y_stt = (0,intminY)
1152 cor_min_y_end = (frame_width,intminY)
1153 cor_max_y_stt = (0,intmaxY)
1154 cor_max_y_end = (frame_width,intmaxY)
1155 movement_average = None
1156 while True:
1157     (grabbed,frame) = camera.read()
1158     cv2.putText(frame,(str(intminX)+" "+str(intmaxX)+" "+str(intminY)+"
"+str(intmaxY))),(10,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)
1159     cv2.line(frame,cor_min_x_stt,cor_min_x_end,(0,255,0),1)
1160     cv2.line(frame,cor_max_x_stt,cor_max_x_end,(0,255,0),1)
1161     cv2.line(frame,cor_min_y_stt,cor_min_y_end,(0,255,0),1)
1162     cv2.line(frame,cor_max_y_stt,cor_max_y_end,(0,255,0),1)
1163     gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
1164     gray = cv2.GaussianBlur(gray,(21,21),0)
1165     if movement_average is None:
1166         movement_average = gray.copy().astype("float")
1167     continue
1168     cv2.accumulateWeighted(gray,movement_average,0.5)
1169     frame_dif = cv2.absdiff(gray,cv2.convertScaleAbs(movement_average))
1170     thresholded = cv2.threshold(frame_dif,5,255,cv2.THRESH_BINARY)[1]
1171     thresholded = cv2.dilate(thresholded,None,iterations = 3)
1172     (contured,contured2) =
cv2.findContours(thresholded.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMP
LE)[-2:]

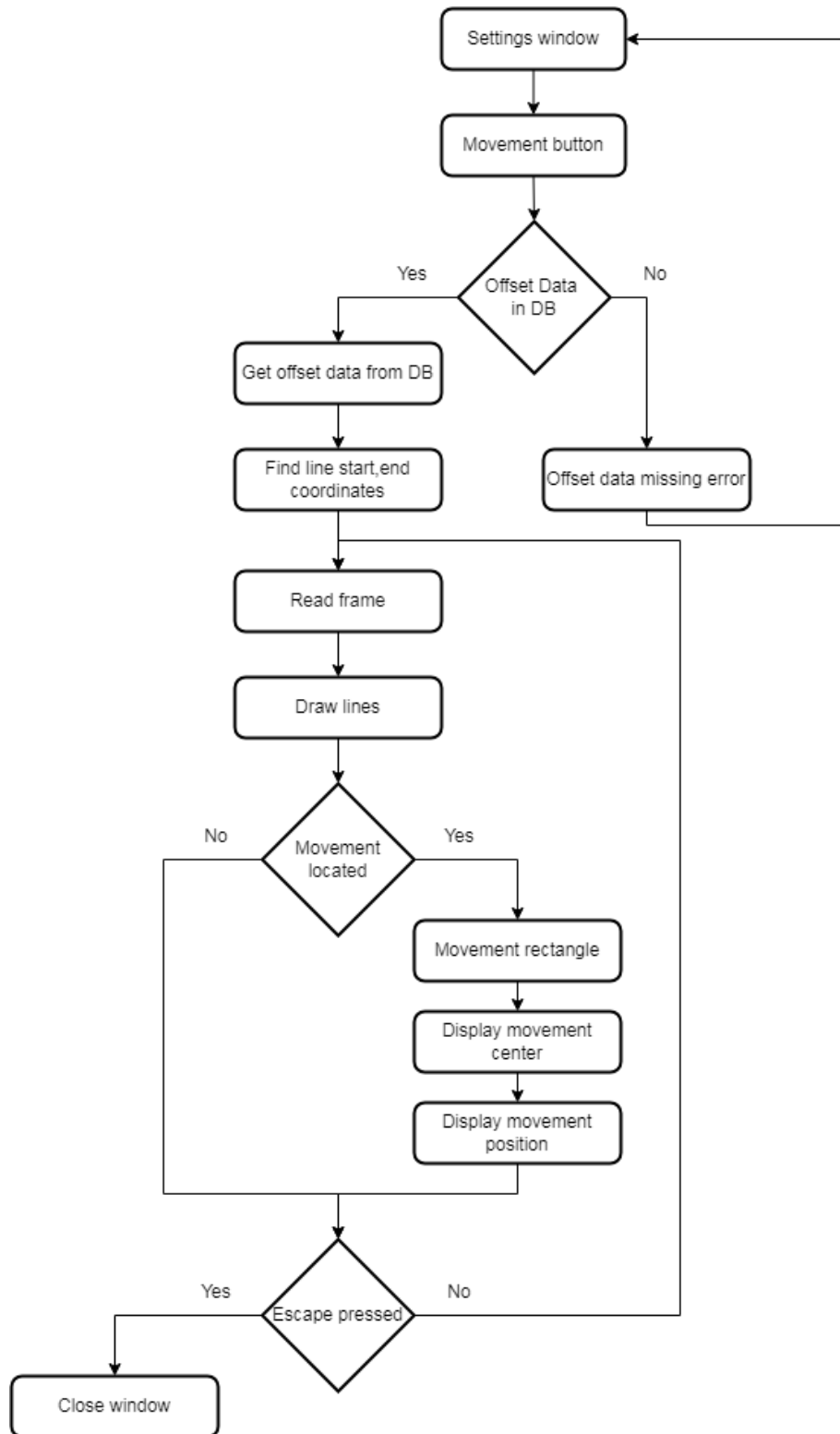
```



```

1173     cnts = sorted(contured,key = cv2.contourArea,reverse = True)[:1]
1174     for c in cnts:
1175         if cv2.contourArea(c) < 10000:
1176             continue
1177         elif cv2.contourArea(c) >= 10000 and cv2.contourArea(c) <105530:
1178             (x,y,w,h) = cv2.boundingRect(c)
1179             cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)
1180             p1_m = x+w
1181             p2_m = y+h
1182             kenX = int((x+p1_m)/2)
1183             kenY = int((y+p2_m)/2)
1184             cv2.putText(frame,str("X"),(kenX,kenY),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1185             if((kenX < intminX) and (kenY < intminY)):
1186                 cv2.putText(frame,str("UP &
LEFT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1187             if((kenX >= intminX and kenX < intmaxX) and (kenY < intminY)):
1188                 cv2.putText(frame,str("UP"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1189             if((kenX >= intmaxX) and (kenY < intminY)):
1190                 cv2.putText(frame,str("UP &
RIGHT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1191             if((kenX < intminX) and (kenY >= intminY and kenY < intmaxY)):
1192                 cv2.putText(frame,str("LEFT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1193             if((kenX >= intminX and kenX < intmaxX) and (kenY >= intminY and
kenY < intmaxY)):
1194                 cv2.putText(frame,str("CENTER"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,25
5),2)
1195             if((kenX >= intmaxX) and (kenY >= intminY and kenY < intmaxY)):
1196                 cv2.putText(frame,str("RIGHT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),
2)
1197             if((kenX < intminX) and (kenY >= intmaxY)):
1198                 cv2.putText(frame,str("DOWN &
LEFT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1199             if((kenX >= intminX and kenX < intmaxX) and (kenY >= intmaxY)):
1200                 cv2.putText(frame,str("DOWN"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),
2)
1201             if((kenX >= intmaxX) and (kenY >= intmaxY)):
1202                 cv2.putText(frame,str("DOWN &
RIGHT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1203             cv2.imshow("Movement Offset",frame)
1204             if cv2.waitKey(1) == 27:
1205                 break
1206             camera.release()
1207             cv2.destroyAllWindows()
1208             #<Review Movement>

```



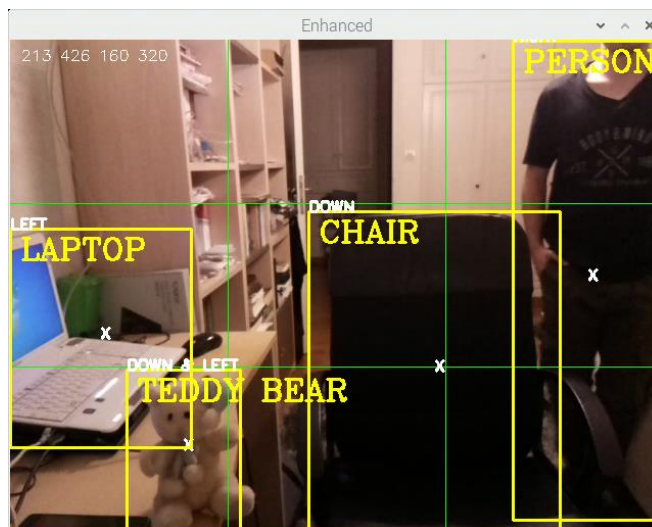
Εικόνα 6- 37: Διάγραμμα ροής για την παραμετροποίηση με εντοπισμό κίνησης



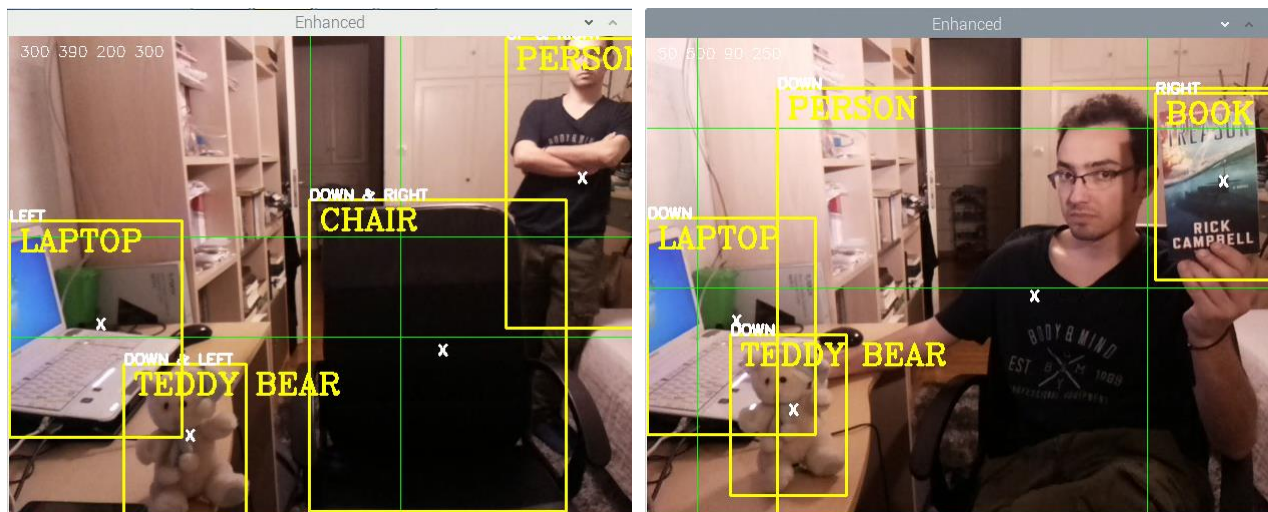
Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

### 6.5.7: Εμφάνιση πλέγματος με εντοπισμό αντικειμένων

Με την επιλογή Review Enhanced εμφανίζεται το παράθυρο με τις γραμμές που διαχωρίζουν τα πλαίσια. Κάθε αντικείμενο που εντοπίζεται από το σύνολο δεδομένων coco (Common Object in Context) εμφανίζεται με ένα κίτρινο παραλληλόγραμμο με το όνομα του αντικειμένου και την θέση του σε σχέση με τις παραμέτρους που διαιρούν την εικόνα. Περισσότερες πληροφορίες σχετικά με το σύνολο δεδομένων coco δίνεται στο επόμενο υποκεφάλαιο.



Εικόνα 6- 38: Παράδειγμα με εντοπισμό αντικειμένων



Εικόνα 6- 39: Παραδείγματα με διαφορετικές τιμές παραμέτρων

Στην συνέχεια παρουσιάζεται ο κώδικας για την εμφάνιση των απαραίτητων δεδομένων

```
1209
1210 #<Review Enhanced>
1211 def Enhanced():
1212     checkQuery = "SELECT COUNT(*) FROM offset"
1213     sqlCursor.execute(checkQuery)
1214     return_data = sqlCursor.fetchone()
1215     if(return_data[0] == 0):
1216         messagebox.showerror("OFFSET DATA ERROR","No offset data found")
1217     return
1218
1219     retrieveQuery = "SELECT offset_minx,offset_maxx,offset_miny,offset_maxy
FROM offset"
1220     sqlCursor.execute(retrieveQuery)
1221     offsetDataDB = sqlCursor.fetchall()
1222     for offsetData in offsetDataDB:
1223         intminx = offsetData[0]
1224         intmaxx = offsetData[1]
1225         intminy = offsetData[2]
1226         intmaxy = offsetData[3]
1227     camera = cv2.VideoCapture(0)
1228     thres = 0.40
1229     nms_threshold = 0.2
1230     classNames = []
1231
1232     classFile = 'Enhanced/coco.names'
1233
1234     with open(classFile,'rt') as f:
1235         classNames = f.read().strip('\n').split('\n')
1236
1237     #print(classNames)
1238
1239     configPath = 'Enhanced/ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
1240     weightsPath = 'Enhanced/frozen_inference_graph.pb'
1241
1242     net = cv2.dnn_DetectionModel(weightsPath,configPath)
1243     net.setInputSize(320,320)
1244     net.setInputScale(1.0/ 127.5)
1245     net.setInputMean((127.5, 127.5, 127.5))
1246     net.setInputSwapRB(True)
1247
1248
1249     cor_min_x_stt = (intminx,0)
1250     cor_min_x_end = (intminx,frame_width)
1251     cor_max_x_stt = (intmaxx,0)
1252     cor_max_x_end = (intmaxx,frame_height)
1253     cor_min_y_stt = (0,intminy)
1254     cor_min_y_end = (frame_width,intminy)
1255     cor_max_y_stt = (0,intmaxy)
1256     cor_max_y_end = (frame_width,intmaxy)
```

```

1257
1258
1259 while True:
1260     gr,frame = camera.read()
1261     cv2.putText(frame,(str(intminx)+" "+str(intmaxx)+" "+str(intminy)+"
"+str(intmaxy)),(10,20), cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),1)
1262     cv2.line(frame,cor_min_x_stt,cor_min_x_end,(0,255,0),1)
1263     cv2.line(frame,cor_max_x_stt,cor_max_x_end,(0,255,0),1)
1264     cv2.line(frame,cor_min_y_stt,cor_min_y_end,(0,255,0),1)
1265     cv2.line(frame,cor_max_y_stt,cor_max_y_end,(0,255,0),1)
1266     classIds, confs, bbox = net.detect(frame,confThreshold = thres)
1267     bbox = list(bbox)
1268     confs = list(np.array(confs).reshape(1,-1)[0])
1269     confs = list(map(float,confs))
1270     indices = cv2.dnn.NMSBoxes(bbox,confs,thres,nms_threshold)
1271
1272     for i in indices:
1273         i = i[0]
1274         box = bbox[i]
1275         x,y,w,h = box[0],box[1],box[2],box[3]
1276         cv2.rectangle(frame,(x,y),(x+w,y+h),color = (0,255,255),thickness = 2)
1277         cv2.putText(frame,classNames[classIds[i][0]-
1278 1].upper(),(box[0]+10,box[1]+30),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,255),2)
1279         p1_m = x+w
1280         p2_i = y+h
1281         kenX = int((x+p1_m)/2)
1282         kenY = int((y+p2_i)/2)
1283         cv2.putText(frame,str("X"),(kenX,kenY),
cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1284         if((kenX < intminx) and (kenY < intminy)):
1285             cv2.putText(frame,str("UP &
LEFT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1286         if((kenX >= intminx and kenX < intmaxx) and (kenY < intminy)):
1287             cv2.putText(frame,str("UP"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1288         if((kenX >= intmaxx) and (kenY < intminy)):
1289             cv2.putText(frame,str("UP &
RIGHT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1290         if((kenX < intminx) and (kenY >= intminy and kenY < intmaxy)):
1291             cv2.putText(frame,str("LEFT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1292         if((kenX >= intminx and kenX < intmaxx) and (kenY >= intminy and kenY <
intmaxy)):
1293             cv2.putText(frame,str("CENTER"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,25
5),2)
1294         if((kenX >= intmaxx) and (kenY >= intminy and kenY < intmaxy)):
1295             cv2.putText(frame,str("RIGHT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),
2)

```

```

1295         if((kenX < intminx) and (kenY >= intmaxy)):
1296             cv2.putText(frame, str("DOWN &
LEFT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1297         if((kenX >= intminx and kenX < intmaxx) and (kenY >= intmaxy)):
1298             cv2.putText(frame, str("DOWN"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),
2)
1299         if((kenX >= intmaxx) and (kenY >= intmaxy)):
1300             cv2.putText(frame, str("DOWN &
RIGHT"),(x,y),cv2.FONT_HERSHEY_SIMPLEX,0.5,(255,255,255),2)
1301
1302
1303         cv2.imshow("Enhanced", frame)
1304         if cv2.waitKey(1) == 27:
1305             break
1306         camera.release()
1307         cv2.destroyAllWindows()
1308     #</Review Enhacend>

```

## 6.6: Ενίσχυση βίντεο για εντοπισμό αντικειμένων

Το σύνολο δεδομένων COCO(Common object in context) είναι ένα αρκετά δημοφιλές σύνολο δεδομένων φωτογραφιών με ετικέτα μεγάλης κλίμακας διαθέσιμο για δημόσια χρήση. Περιέχει μια μεγάλη ποικιλία αντικειμένων καθημερινής χρήσης. Στην παρακάτω εικόνα διακρίνονται ορισμένες από τις κατηγορίες των κλάσεων που εντοπίζονται.

person	fire hydrant	elephant	skis	wine glass	broccoli	dining table	toaster
bicycle	stop sign	bear	snowboard	cup	carrot	toilet	sink
car	parking meter	zebra	sports ball	fork	hot dog	tv	refrigerator
motorcycle	bench	giraffe	kite	knife	pizza	laptop	book
airplane	bird	backpack	baseball bat	spoon	donut	mouse	clock
bus	cat	umbrella	baseball glove	bowl	cake	remote	vase
train	dog	handbag	skateboard	banana	chair	keyboard	scissors
truck	horse	tie	surfboard	apple	couch	cell phone	teddy bear
boat	sheep	suitcase	tennis racket	sandwich	potted plant	microwave	hair drier
traffic light	cow	frisbee	bottle	orange	bed	oven	toothbrush

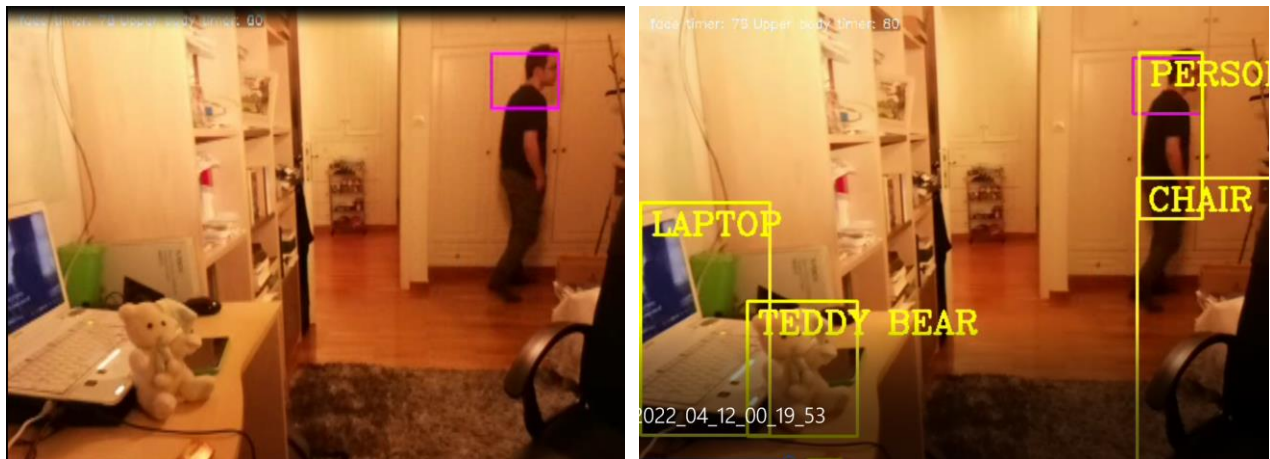
**Εικόνα 6- 40: Παραδείγματα κλάσεων του συνόλου δεδομένων**

Αυτό το σύνολο δεδομένων εκμεταλλεύεται η «μονάδα» DNN (Deep Neural Network) για τον εντοπισμό των αντικειμένων (cv2.dnn\_DetectionModel). Η διαδικασία αυτή έχει εφαρμοστεί και στην εφαρμογή. Ο εντοπισμός γίνεται στα βίντεο που έχουν ήδη αποθηκευτεί. Τα βίντεο διαβάζονται από

την εφαρμογή, εντοπίζονται οι περιοχές και στην συνέχεια αποθηκεύονται με νέο όνομα και σε διαφορετικό φάκελο.

Το πρώτο βήμα αφορά την αρχικοποίηση του DNN module. Στην συνέχεια για κάθε βίντεο που υπάρχει σε κάθε φάκελο με εντοπισμό περιοχών ενδιαφέροντος, διαβάζονται ένα-ένα τα frames και εντοπίζεται το μεγαλύτερο παραλληλόγραμμο που περικλείει το κάθε αντικείμενο. Μετά την τοποθέτηση του παραλληλογράμμου το frame καταγράφεται σε νέο βίντεο το οποίο αποθηκεύεται σε νέο βίντεο και αποθηκεύεται σε αντίστοιχο φάκελο ανάλογα με το είδος καταγραφής του αρχικού βίντεο στον φάκελο Enhanced. Στην παρακάτω εικόνα διακρίνεται το αρχικό βίντεο και το αντίστοιχο βίντεο μετά την διαδικασία που περιγράφηκε. Μετά την επιτυχή ολοκλήρωση της διαδικασίας εμφανίζεται και σχετικό μήνυμα για την ενημέρωση του χρήστη

Στις παρακάτω εικόνες διακρίνονται στιγμιότυπα από το αρχικό βίντεο και από το βίντεο μετά την διαδικασία μετατροπής



Εικόνα 6- 41: Στιγμιότυπα από τα 2 βίντεο

Στην συνέχεια παρουσιάζεται ο κώδικας που υλοποιεί την παραπάνω διαδικασία.

```
1410 def EnhanceVideos():
1411     #<Parameters init>
1412     thres = 0.40
1413     nms_threshold = 0.2
1414     classNames = []
1415     classFile = 'Enhanced/coco.names'
1416     with open(classFile,'rt') as f:
1417         classNames = f.read().strip('\n').split('\n')
1418
1419     configPath = 'Enhanced/ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
1420     weightsPath = 'Enhanced/frozen_inference_graph.pb'
1421     fourcc_code = cv2.VideoWriter_fourcc(*'DIVX')#DIVX
1422     net = cv2.dnn_DetectionModel(weightsPath,configPath)
1423     net.setInputSize(320,320)
1424     net.setInputScale(1.0/ 127.5)
```



```

1425 net.setInputMean((127.5, 127.5, 127.5))
1426 net.setInputSwapRB(True)
1427 #</Parameters init>
1428 #<Face Video enchancments>
1429 for fvideo in os.listdir("Unknown/Faces"):
1430     if fvideo.endswith(".avi"):
1431         fvid = os.path.join("Unknown/Faces",fvideo)
1432         #<Remove any existing enhanced videos with the same name>
1433         if(os.path.exists("Enhanced/Face/EN_"+str(fvid.split("/") [2] ))):
1434             os.remove("Enhanced/Face/EN_"+str(fvid.split("/") [2] ))
1435         #</Remove any existing enhanced videos with the same name>
1436         source = cv2.VideoCapture(fvid)
1437         frame_width = int(source.get(cv2.CAP_PROP_FRAME_WIDTH) + 0.5)
1438         frame_height = int(source.get(cv2.CAP_PROP_FRAME_HEIGHT) + 0.5)
1439         size = (frame_width, frame_height)
1440         result =
cv2.VideoWriter("Enhanced/Face/EN_"+str(fvid.split("/") [2] ),fourcc_code,7,size)
1441         while True:
1442             ret,frame = source.read()
1443             if not ret:
1444                 result.release()
1445                 break
1446             classIds, confs, bbox = net.detect(frame,confThreshold = thres)
1447             bbox = list(bbox)
1448             confs = list(np.array(confs).reshape(1,-1)[0])
1449             confs = list(map(float,confs))
1450             indices = cv2.dnn.NMSBoxes(bbox,confs,thres,nms_threshold)
1451
1452             for i in indices:
1453                 i = i[0]
1454                 box = bbox[i]
1455                 x,y,w,h = box[0],box[1],box[2],box[3]
1456                 cv2.rectangle(frame,(x,y),(x+w,y+h),color=(0,255,255),thickness = 2)
1457                 cv2.putText(frame,classNames[classIds[i][0]-
1458 1].upper(),(box[0]+10,box[1]+30),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,255),2)
1458                 result.write(frame)
1459                 #cv2.imshow("A",frame)
1460                 cv2.waitKey(15)
1461                 source.release()
1462                 result.release()
1463                 cv2.destroyAllWindows()
1464         #<Face Video enchancments>
1465         #<Upper body Video enchancments>
1466         for uvideo in os.listdir("Unknown/UpperBody"):
1467             if uvideo.endswith(".avi"):
1468                 uvid = os.path.join("Unknown/UpperBody",uvideo)
1469                 #<Remove any existing enhanced upper body videos>
1470                 if os.path.exists("Enhanced/UpperBody/EN_"+str(uvid.split("/") [2] )):
1471                     os.remove("Enhanced/UpperBody/EN_"+str(uvid.split("/") [2] ))
1472                 #</Remove any existing enhanced upper body videos>

```

```

1473     source = cv2.VideoCapture(uvid)
1474     frame_width = int(source.get(cv2.CAP_PROP_FRAME_WIDTH)+0.5)
1475     frame_height = int(source.get(cv2.CAP_PROP_FRAME_HEIGHT)+0.5)
1476     size = (frame_width,frame_height)
1477     result =
cv2.VideoWriter("Enhanced/UpperBody/EN_"+str(uvid.split("/")[2]),fourcc_code,7,size)
1478     while True:
1479         ret,frame = source.read()
1480         if not ret:
1481             result.release()
1482             break
1483         classIds, confs, bbox = net.detect(frame,confThreshold = thres)
1484         bbox = list(bbox)
1485         confs = list(np.array(confs).reshape(1,-1)[0])
1486         confs = list(map(float,confs))
1487         indices = cv2.dnn.NMSBoxes(bbox,confs,thres,nms_threshold)
1488
1489         for i in indices:
1490             i = i[0]
1491             box = bbox[i]
1492             x,y,w,h = box[0],box[1],box[2],box[3]
1493             cv2.rectangle(frame,(x,y),(x+w,y+h),color=(0,255,255),thickness = 2)
1494             cv2.putText(frame,classNames[classIds[i][0]-
1495             1].upper(),(box[0]+10,box[1]+30),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,255),2)
1496             result.write(frame)
1497             #cv2.imshow("B",frame)
1498             cv2.waitKey(15)
1499             source.release()
1500             result.release()
1501             cv2.destroyAllWindows()
1502     #/<Upper body Video enchancements>
1503     #<Full body video enhancements>
1504     for fvideo in os.listdir("Unknown/FullBody"):
1505         if fvideo.endswith(".avi"):
1506             fvid = os.path.join("Unknown/Fullbody",fvideo)
1507             #<Remove any existing full body enhanced videos>
1508             if(os.path.exists("Enhanced/Fullbody/EN_"+str(fvid.split("/")[2]))):
1509                 os.remove("Enhanced/Fullbody/EN_"+str(fvid.split("/")[2]))
1510             #</Remove any existing full body enhanced videos>
1511             source = cv2.VideoCapture(fvid)
1512             frame_width = int(source.get(cv2.CAP_PROP_FRAME_WIDTH)+0.5)
1513             frame_height = int(source.get(cv2.CAP_PROP_FRAME_HEIGHT)+0.5)
1514             size = (frame_width,frame_height)
1515             result =
cv2.VideoWriter("Enhanced/Fullbody/EN_"+str(fvid.split("/")[2]),fourcc_code,7,size)
1516             while True:
1517                 ret,frame = source.read()
1518                 if not ret:
1519                     result.release()
1520                     break

```

```

1520     classIds, confs, bbox = net.detect(frame,confThreshold = thres)
1521     bbox = list(bbox)
1522     confs = list(np.array(confs).reshape(1,-1)[0])
1523     confs = list(map(float,confs))
1524     indices = cv2.dnn.NMSBoxes(bbox,confs,thres,nms_threshold)
1525
1526     for i in indices:
1527         i = i[0]
1528         box = bbox[i]
1529         x,y,w,h = box[0],box[1],box[2],box[3]
1530         cv2.rectangle(frame,(x,y),(x+w,y+h),color=(0,255,255),thickness = 2)
1531         cv2.putText(frame,classNames[classIds[i][0]-
1532 1],upper(),(box[0]+10,box[1]+30),cv2.FONT_HERSHEY_COMPLEX,1,(0,255,255),2)
1532     result.write(frame)
1533     #cv2.imshow("C",frame)
1534     cv2.waitKey(15)
1535     source.release()
1536     result.release()
1537     cv2.destroyAllWindows()
1538 #</Full body video enhancements>
1539 #<Movement Videos Enhancement>
1540 for mvideo in os.listdir("Unknown/Movement"):
1541     if mvideo.endswith(".avi"):
1542         mvid = os.path.join("Unknown/Movement",mvideo)#mvid.split("/")[2]
1543         #<Remove any existing Enhanced Videos With that name>
1544         if(os.path.exists("Enhanced/Movement/EN_"+str(mvid.split("/")[2]))):
1545             os.remove("Enhanced/Movement/EN_"+str(mvid.split("/")[2]))
1546         #</Remove any existing Enhanced Videos With that name>
1547         source = cv2.VideoCapture(mvid)
1548         frame_width = int(source.get(cv2.CAP_PROP_FRAME_WIDTH) + 0.5)
1549         frame_height = int(source.get(cv2.CAP_PROP_FRAME_HEIGHT) + 0.5)
1550         size = (frame_width, frame_height)
1551         result =
cv2.VideoWriter("Enhanced/Movement/EN_"+str(mvid.split("/")[2]),fourcc_code,7,size)
1552     while True:
1553         ret,frame = source.read()
1554         if not ret:
1555             result.release()
1556             break
1557         classIds, confs, bbox = net.detect(frame,confThreshold = thres)
1558         bbox = list(bbox)
1559         confs = list(np.array(confs).reshape(1,-1)[0])
1560         confs = list(map(float,confs))
1561         indices = cv2.dnn.NMSBoxes(bbox,confs,thres,nms_threshold)
1562
1563         for i in indices:
1564             i = i[0]
1565             box = bbox[i]
1566             x,y,w,h = box[0],box[1],box[2],box[3]
1567             cv2.rectangle(frame,(x,y),(x+w,y+h),color = (0,255,255),thickness = 2)

```



Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

```
1570         #cv2.imshow("D",frame)
1571         cv2.waitKey(15)
1572         source.release()
1573         result.release()
1574         cv2.destroyAllWindows()
1575     #</Movement Videos Enhancement>
1576
1577     messagebox.showinfo("SUCCESS", "Video Enhancment Completed Successfully !!")
1578
```

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## Κεφάλαιο 7: Προβλήματα που παρουσιάστηκαν

Κατά την ανάπτυξη της εφαρμογής παρουσιάστηκαν ορισμένα προβλήματα. Ορισμένα από αυτά επιλύθηκαν ενώ άλλα παρέμειναν καθώς η επίλυση τους θα επηρέαζε την λειτουργικότητα της εφαρμογής. Σε αυτήν την ενότητα αναλύονται τα προβλήματα που εμφανίστηκαν καθώς και οι τρόποι που αντιμετωπίστηκαν.

### 7.1: Προβλήματα με το Raspberry Pi

Στα αρχικά στάδια ανάπτυξης του συστήματος ο αρχικός μικροϋπολογιστής ήταν ένα Raspberry Pi Model 3. Το model 3 είχε επεξεργαστή Broadcom BCM2837B0 quad-code A53 (ARMv8) 64-bit στα 1.4GHz και 1GB DDR2 SDRAM. Παρόλο που δεν υπήρχαν προβλήματα κατά την εκπαίδευση του συστήματος και τον εντοπισμό των περιοχών ενδιαφέροντος προσώπων, άνω μέρους σωμάτων και πλήρους σωμάτων, παρατηρήθηκαν καθυστερήσεις κατά τον εντοπισμό κινήσεων. Αυτό αποτελεί ένα σημαντικό πρόβλημα όταν η κάμερα προσαρτήθηκε στον ρομποτικό βραχίονα καθώς λόγω των καθυστερήσεων η περιοχή ενδιαφέροντος είχε απομακρυνθεί από το πεδίο λήψης της κάμερας με αποτέλεσμα την εσφαλμένη κίνηση του βραχίονα.

Τελικά ο μικροϋπολογιστής αντικαταστάθηκε με ένα Raspberry Pi Model 4. Αυτό βελτίωσε αισθητά το πρόβλημα αν και κάποιες καθυστερήσεις παρατηρούνται ακόμα

### 7.2: Προβλήματα με την σειριακή επικοινωνία

Ένα άλλο σημαντικό πρόβλημα παρουσιάστηκε κατά την προσθήκη Super Led. Ανάλογα με τις συνθήκες φωτισμού τα Leds ανάβουν με σκοπό τον καλύτερο φωτισμό του χώρου. Στην αρχική ανάπτυξη ο έλεγχος της φωτεινότητας γινόταν μέσω της κάμερας. Αρχικά βρισκόταν το πιο σκοτεινό εικονοστοιχείο στο πεδίο λήψης της κάμερας και ανάλογα με την τιμή του το Raspberry Pi μέσω σειριακής επικοινωνίας “ενημερώνει” το Arduino για να ανοίξει ή να σβήσει τα Led. Το πρόβλημα παρουσιάστηκε κατά την μετακίνηση του βραχίονα καθώς το Raspberry Pi έστελνε δεδομένα για την κίνηση του βραχίονα και για τον φωτισμό και προκαλούσε διακεκομμένη κίνηση του βραχίονα.

Το πρόβλημα λύθηκε με την αφαίρεση της παρακολούθησης της φωτεινότητας μέσω της κάμερας. Ο έλεγχος της φωτεινότητας γίνεται από έναν φωτοαντιστάτη ο οποίος είναι προσαρτημένος στον μικροελεγκτή.

### 7.3: Πρόβλημα με την καταγραφή των βίντεο

Ένα επίσης σημαντικό πρόβλημα προέκυψε κατά την καταγραφή των βίντεο στα πρώτα στάδια της ανάπτυξης. Η διαδικασία καταγραφής είναι η εξής. Αρχικά δημιουργείται ένα άδειο βίντεο. Αν βρεθεί περιοχή ενδιαφέροντος η καταγραφή γίνεται προσωρινά και όταν φύγει από το πεδίο λήψης της κάμερας και τελειώσει ο μετρητής το βίντεο αποθηκεύεται μόνιμα. Αν δεν υπάρχει εξαρχής περιοχή ενδιαφέροντος το βίντεο δημιουργείται και διαγράφεται την ίδια στιγμή. Η διαδικασία επαναλαμβάνεται όσο η κάμερα είναι ενεργή. Λόγω ενός προβλήματος στον κώδικα της αρχικής υλοποίησης όταν τελειώνει

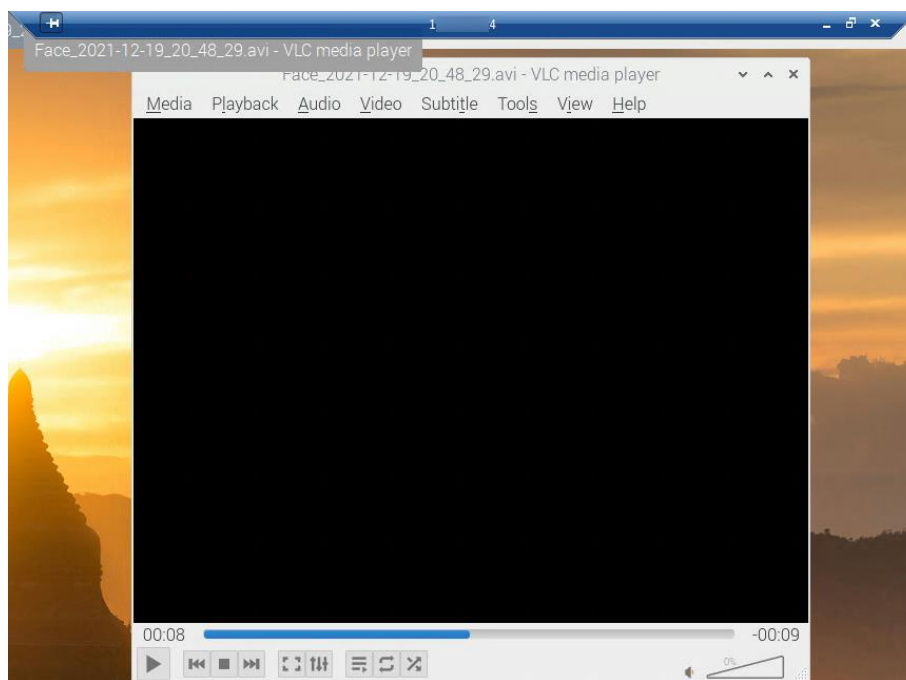
Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

η αποθήκευση των βίντεο το σύστημα συνέχιζε να δημιουργεί κενά βίντεο μέχρι να βρεθεί καινούρια περιοχή ενδιαφέροντος στο πεδίο λήψης της κάμερας. Το πρόβλημα λύθηκε προσωρινά με την διαγραφή όλων των βίντεο που είναι κενά αφού πατηθεί το πλήκτρο Escape. Το πρόβλημα λύθηκε οριστικά με τις κατάλληλες αλλαγές στον κώδικα καταγραφής των βίντεο.

#### 7.4: Πρόβλημα με την καταγραφή σε βίντεο

Κάθε βίντεο είναι αποθηκευμένο στον αντίστοιχο φάκελο ανάλογα με το είδος του συμβάντος που κατέγραψε. Τα βίντεο εμφανίζονται ανοίγοντας από το Raspberry Pi.

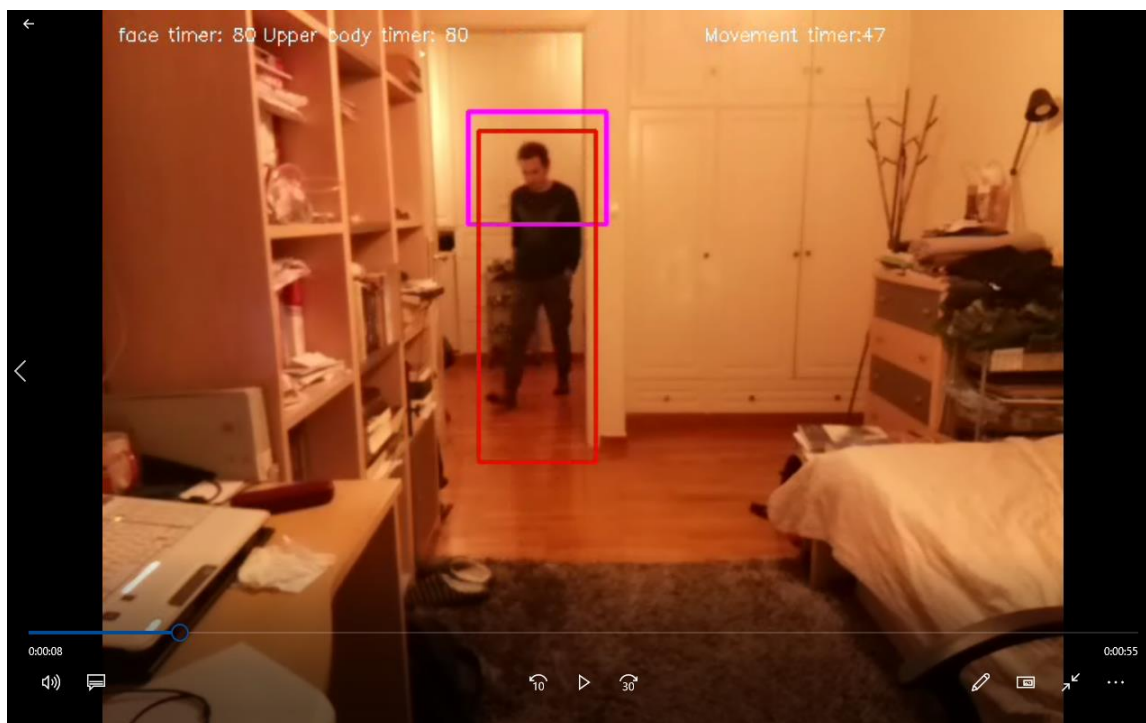
Ένα πρόβλημα προέκυψε κατά το άνοιγμα των βίντεο όταν η σύνδεση στο Raspberry Pi γίνεται μέσω απομακρυσμένης σύνδεσης. Σε αυτήν την περίπτωση τα βίντεο εμφανίζονται μαύρα.



Εικόνα 7- 1: Το πρόβλημα εμφάνισης των βίντεο μέσα από την απομακρυσμένη σύνδεση

Για αυτό το πρόβλημα δεν έχει βρεθεί λύση. Για την προβολή των βίντεο μια λύση είναι να γίνει σύνδεση στο Raspberry Pi μέσω ενός SFTP προγράμματος (π.χ. WinSCP) και να πραγματοποιηθεί μεταφορά του βίντεο από το Raspberry Pi στον υπολογιστή. Το βίντεο εμφανίζεται πλέον χωρίς προβλήματα.

Στην εικόνα που ακολουθεί εμφανίζεται το βίντεο κανονικά στον τοπικό υπολογιστή.



Εικόνα 7- 2: Προβολή του βίντεο στον τοπικό υπολογιστή

### 7.5: Πρόβλημα κατά την καταγραφή την εμφάνιση των αντικειμένων

Η αρχική ανάπτυξη της εφαρμογής ήταν ο εντοπισμός των αντικειμένων να γίνεται σε πραγματικό χρόνο, χωρίς να θεωρούνται περιοχές ενδιαφέροντος για να προκαλούν περιστροφή του βραχίονα και να αποθηκεύεται στα βίντεο με τις περιοχές ενδιαφέροντος. Όταν εφαρμόστηκε αυτή η τεχνική προκαλούσε μεγάλες καθυστερήσεις και αυτό αποτυπώθηκε στις καθυστερημένες κινήσεις του βραχίονα. Υπήρχαν περιπτώσεις όπου η περιοχή ενδιαφέροντος είχε απομακρυνθεί από το πεδίο λήψης της κάμερας και ο βραχίονας συνέχιζε την περιστροφή λόγω των καθυστερήσεων.

Για αυτό το πρόβλημα δεν βρέθηκε λύση που να διατηρεί την καταγραφή των αντικειμένων σε πραγματικό χρόνο. Η λειτουργία αυτή αφαιρέθηκε και γίνεται στα βίντεο που έχουν ήδη αποθηκευτεί ώστε να μην επιβαρύνεται το σύστημα.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## Κεφάλαιο 8: Συμπεράσματα και μελλοντικές επεκτάσεις

### 8.1: Συμπεράσματα

Το σύστημα έχει αρκετά καλή απόδοση ως προς το κόστος του εξοπλισμού (συμπεριλαμβανομένου και του βραχίονα). Η μέθοδος των τοπικών δυαδικών προτύπων παρέχει αρκετά αξιόπιστα αποτελέσματα και σε περίπτωση που το αποτέλεσμα είναι ψευδώς αρνητικό (δηλαδή ένα έμπιστο πρόσωπο εμφανιστεί ως άγνωστο λόγω κακών συνθηκών φωτισμού ή κακής γωνίας λήψης), ο χρήστης μπορεί μέσω του βίντεο να επιβεβαιώσει την ταυτότητα του.

Σε περίπτωση που το άτομο που θα εισέλθει φοράει μάσκα τότε θα εντοπιστεί από τα επόμενα στάδια ανίχνευσης (άνω μέρους σώματος, πλήρους σώματος, κίνησης) και θα καταγραφεί σε βίντεο, οπότε ο χρήστης θα γνωρίζει ότι κάποιος έχει εισέλθει στον χώρο.

Ο κύριος περιορισμός αφορά το υπολογιστικό σύστημα. Το Raspberry Pi αποτελεί την καλύτερη επιλογή ως προς την απόδοση και το κόστος. Το κυριότερο πρόβλημα του είναι η υπολογιστική ισχύς. Ο δεύτερος σημαντικός περιορισμός αφορά την κάμερα. Όσο καλύτερη ανάλυση έχει τόσο πιο αξιόπιστα θα είναι οι εντοπισμοί, ειδικά των προσώπων. Τέλος ένας παράγοντας που παίζει σημαντικό ρόλο είναι ο φωτισμός, καθώς αν δεν είναι επαρκής μπορεί να οδηγήσει σε αδυναμία αναγνώρισης έμπιστων προσώπων και περιοχών ενδιαφέροντος γενικότερα.

### 8.2: Μελλοντικές επεκτάσεις

Όπως σε κάθε project, έτσι και για το συγκεκριμένο υπάρχουν ιδέες και σχέδια για την περαιτέρω ανάπτυξη και βελτίωση. Τα περισσότερα από αυτά τα σχέδια είναι πολύ δύσκολο να αναπτυχθούν στα πλαίσια της πτυχιακής εργασίας, και απαιτούν πιο ισχυρά υπολογιστικά συστήματα. Ορισμένες από τις επεκτάσεις περιγράφονται στις επόμενες παραγράφους.

Η πρώτη μελλοντική επέκταση αφορά την ενημέρωση του χρήστη. Στο Raspberry Pi μπορεί να συνδεθεί ένα GSM Module με μια κάρτα SIM και να αποστέλλει SMS στα κινητά τηλέφωνα των εμπιστων χρηστών. Οι αριθμοί των χρηστών μπορούν να αποθηκευτούν στην βάση δεδομένων. Επιπλέον θα μπορούσε να γίνει επέκταση πέρα από το SMS να αποστέλλεται και e-mail με συνημμένα τα βίντεο. Όπως και με τα τηλέφωνα τα e-mail μπορούν να αποθηκευτούν στην βάση δεδομένων.

Μια βελτίωση της παραπάνω πρότασης αφορά την ανάπτυξη μιας εφαρμογής για κινητό που να επιτρέπει την άμεση παρακολούθηση του χώρου με την εμφάνιση των δεδομένων της κάμερας. Ο χρήστης θα μπορεί να συνδέεται δίνοντας ένα συνηματικό και να παρακολουθεί ζωντανά και οποιαδήποτε ώρα τον χώρο.

Η τρίτη επέκταση αφορά την πλήρη ρύθμιση των παραμέτρων του συστήματος από τους χρήστες. Συγκεκριμένα οι χρήστες θα είναι σε θέση να ρυθμίζουν τις παρακάτω τιμές

- Τον αριθμό των φωτογραφιών που συλλέγονται για την εκπαίδευση του συστήματος
- Την τιμή του κατωφλίου για την επιτυχή αναγνώριση προσώπων
- Τις τιμές για τον διάρκεια του χρόνου καταγραφής του κάθε συμβάντος
- Τις τιμές παραμετροποίησης των Haar ταξινομητών
- Την τιμή κατωφλίου για τον εντοπισμό κίνησης
- Την τιμή του περιγράμματος

Οι τιμές μπορούν να αποθηκευτούν στην βάση δεδομένων σε έναν πίνακα με δύο στήλες. Η μια στήλη διατηρεί το όνομα της παραμέτρου και η δεύτερη την τιμή. Κατά την εκτέλεση της εφαρμογής θα μπορούσαν να ενημερώνονται οι μεταβλητές από την βάση δεδομένων.

Επίσης η μέθοδος ταυτοποίησης θα μπορούσε να βελτιωθεί με τον συνδυασμό μιας επιπλέον τεχνικής πέρα από τα τοπικά δυαδικά πρότυπα όπως τα βιομετρικά δεδομένα.

Μια προσθήκη είναι η εγκατάσταση μιας μονάδας UPS. Με αυτόν τον τρόπο το σύστημα θα μπορεί να λειτουργεί ακόμα και όταν γίνει διακοπή ρεύματος στον χώρο. Δεδομένου ότι ο μικροϋπολογιστής και ο μικροελεγκτής δεν έχουν μεγάλες απαιτήσεις σε ρεύμα με μια μέση UPS το σύστημα θα μπορεί να λειτουργεί εκτεταμένα.

Η τελευταία επέκταση αφορά την υποστήριξη πολλαπλών γλωσσών ώστε να μπορεί να χρησιμοποιηθεί από πολλαπλούς χρήστες. Η γλώσσα θα μπορούσε να αποθηκευτεί ως επιπλέον πεδίο στον πίνακα με τα πεδία των χρηστών και να εμφανίζονται οι οδηγίες στην αντίστοιχη γλώσσα.



## Κεφάλαιο 9: Ανάλυση SWOT

Με την ανάλυση SWOT(Strength,Weakness,Oppportunities,Threats) εντοπίζονται τα δυνατά σημεία, οι αδυναμίες καθώς και οι ευκαιρίες και οι απειλές κατά τον προγραμματισμό ενός έργου.

Η συγκεκριμένη τεχνική είναι σχεδιασμένη για να χρησιμοποιείται στα αρχικά στάδια για την ορθή λήψη αποφάσεων. Μέσω της ανάλυσης μπορούν να εντοπιστούν οι ευνοϊκοί και δυσμενείς παράγοντες, τόσο οι εσωτερικοί καθώς και οι εξωτερικοί. Παρόλο που ο πίνακας SWOT θεωρείται ένα αρκετά δοκιμασμένο εργαλείο στρατηγικής, έχει και περιορισμούς και έχει επικριθεί για αυτό. Στον επόμενο πίνακα φαίνεται η ανάλυση SWOT του συστήματος

Strengths	Weaknesses
<ol style="list-style-type: none"><li>1. Χαμηλό κόστος</li><li>2. Αξιόπιστοι εντοπισμοί λόγω ελέγχου πολλαπλών περιοχών ενδιαφέροντος</li><li>3. Καταγραφή συμβάντων</li><li>4. Open Source προγράμματα</li></ol>	<ol style="list-style-type: none"><li>1. Πιθανά προβλήματα λόγω υπολογιστικής ισχύος του Raspberry Pi</li><li>2. Προβλήματα κατά τον εντοπισμό των περιοχών ενδιαφέροντος σε κακές συνθήκες φωτισμού</li><li>3. Πιθανά προβλήματα αν η περιοχή ενδιαφέροντος κινείται γρήγορα</li></ol>
Oppportunities	Threats
<ol style="list-style-type: none"><li>1. Παρακολούθηση απομακρυσμένων περιοχών</li><li>2. Προστασία από ανεπιθύμητα άτομα</li><li>3. Εύκολη κατασκευή με 3D printer και υλικά που βρίσκονται εύκολα στο εμπόριο</li></ol>	<ol style="list-style-type: none"><li>1. Ορισμένα ή όλα τα προγράμματα αλλάζουν κατάσταση από δωρεάν</li><li>2. Άλλες μέθοδοι αυθεντικοποίησης ενδέχεται να είναι πιο αξιόπιστες</li><li>3. Διακοπή ρεύματος</li></ol>

**Πίνακας 9- 1: Ανάλυση SWOT της εφαρμογής**

Πέρα από τις αρχικές απαιτήσεις τέθηκαν δύο επιπλέον στόχοι, το χαμηλό κόστος και η υψηλή αξιοπιστία. Αυτά τα δύο χαρακτηριστικά αποτελούν τα δυνατά στοιχεία της εφαρμογής που αναπτύχθηκε. Το χαμηλό κόστος έχει προκύψει από το γεγονός ότι ο μικροϋπολογιστής είναι χαμηλού κόστους. Επιπλέον τα πακέτα υπολογιστικής όρασης και τα προγράμματα είναι open source. Για το ζήτημα της αξιοπιστίας παρέχονται εντοπισμοί πολλαπλών περιοχών ενδιαφέροντος αυξάνοντας την δυνατότητα εντοπισμού ατόμου. Επιπλέον ο χρήστης έχει την δυνατότητα να δει τα βίντεο με τους εντοπισμούς που έχουν γίνει, και μέσω του ονόματος του βίντεο είναι γνωστή η ημερομηνία και η ώρα του εντοπισμού.

Αν και το Raspberry Pi εκτελεί τους πολύπλοκους υπολογισμούς υπολογιστικής όρασης, παρατηρούνται καθυστερήσεις. Επίσης η κάμερα του Raspberry pi παρόλο που είναι αρκετά μικρή και ελαφριά για να προσαρμοστεί στον βραχίονα, παρουσιάζει δυσκολίες στην αναγνώριση προσώπων σε μεγάλες αποστάσεις. Αυτά τα προβλήματα είναι δυνατόν να λυθούν αν ο προϋπολογισμός επιτρέπει την απόκτηση ισχυρότερου μικροϋπολογιστή και καλύτερης κάμερας.

Όσον αφορά την χρήση του συστήματος, μπορεί να χρησιμοποιηθεί σε απομακρυσμένες κατοικίες για την παρακολούθηση του χώρου. Εδώ χρειάζεται προσοχή καθώς η επιτήρηση πρέπει να γίνεται εντός του χώρου της κατοικίας. Τα αρχεία για τους ταξινομητές βρίσκονται εύκολα στο διαδίκτυο.

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

Το μεγαλύτερο θέμα που μπορεί να επηρεάσει το σύστημα είναι η βιβλιοθήκη υπολογιστικής όρασης. Αν σταματήσει να είναι δωρεάν θα πρέπει να βρεθεί άλλο πακέτο ή να γίνει αγορά. Επίσης ένα δυνητικό πρόβλημα είναι η διακοπή της παροχής ισχύος, αν και μπορεί να αντιμετωπιστεί με σχετικά μικρό κόστος.

## Παραρτήματα

### Παράρτημα 1

Με βάση τον νόμο 2472/1997 και συγκεκριμένα το άρθρο 6 σε περίπτωση που εγκατασταθεί ή μεταβληθεί κάποιο σύστημα βιντεοεπιτήρησης είναι απαραίτητη η γνωστοποίηση στην αρχή προστασίας δεδομένων. Σε περίπτωση μη ενημέρωσης μπορεί να επιβληθούν ποινές σύμφωνα με τα άρθρα 20 και 22 του νόμου.

Όταν το σύστημα βιντεοεπιτήρησης καταγράφει εικόνες και βίντεο από ιδιωτικό μη επαγγελματικό χώρο τότε δεν είναι υποχρεωτικό να δηλωθεί στην αρχή προστασίας δεδομένων. Οι κάμερες μπορεί να βρίσκονται σε οποιοδήποτε σημείο εντός του σπιτιού. Αξίζει να σημειωθεί ότι απαγορεύεται ρητά να γίνεται καταγραφή από εξωτερικούς δημόσιους χώρους όπως πεζοδρόμια και διπλανά κτήρια.

Υπεύθυνος επεξεργασίας ορίζεται όποιος έχει τοποθετήσει την κάμερα και είναι υπεύθυνος για όλες τις νομικές υποχρεώσεις των νόμων 2472/1997 και 1/2011. Ορισμένες από αυτές είναι η υποχρέωση υποβολής στην αρχή προστασίας δεδομένων (όταν απαιτείται) καθώς και των δικαιωμάτων αντίρρησης, ενημέρωσης και τήρησης απορρήτου και ασφάλειας της επεξεργασίας.[36]

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...

## Παράρτημα 2

Στο παράρτημα Β αναλύονται όλες οι εντολές του OpenCV που έχουν χρησιμοποιηθεί για την ανάπτυξη του συστήματος επιτήρησης χώρου

faceDetect = cv2.CascadeClassifier(fileName)[36]	
Περιγραφή	Κλήση του ταξινομητή Haar
Παράμετροι	fileName: το αρχείο του ταξινομητή Haar

**Πίνακας Π-2- 1: Εντολή cv2.CascadeClassifier**

Camera = cv2.VideoCapture(index)[37]	
Περιγραφή	Άνοιγμα αρχείου η κάμερας
Παράμετροι	<u>Index</u> : Η παράμετρος index μπορεί να είναι βίντεο, εικόνες η από κάμερα

**Πίνακας Π-2- 2: Εντολή cv2.VideoCapture**

Gray = cv2.cvtColor(src,code)[38]	
Περιγραφή	Μετατροπή εικόνας ή βίντεο σε διαφορετική χρωματική κλίμακα
Παράμετροι	<u>Src</u> : Η εικόνα ή το βίντεο προς μετατροπή <u>code</u> : ο κωδικός μετατροπής

Παράδειγμα με κωδικό μετατροπής cv2.COLOR\_BGR2GRAY



**Πίνακας Π-2- 3: Η εντολή cv2.cvtColor**

ubodyDetect.detectMultiScale(Frane,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)[39]	
Περιγραφή:	Εντοπισμός αντικειμένων με βάση τον haar ταξινομητή. Τα αντικείμενα που έχουν εντοπιστεί επιστρέφονται ως λίστα συντεταγμένων
Παράμετροι	<u>Frame</u> : η εικόνα ή το βίντεο που θα γίνει η αναγνώριση των αντικειμένων
	<u>scaleFactor</u> : παράμετρος που καθορίζει την μείωση της εικόνας σε κάθε κλίμακα
	<u>minNeighbors</u> : για κάθε υποψήφιο προς εντοπισμό ορθογώνιο πόσους γείτονες θα πρέπει να διατηρήσει. Όσο υψηλότερη είναι η τιμή που θα έχει, τόσο λιγότεροι εντοπισμοί αλλά με μεγαλύτερη ποιότητα.
	<u>minSize</u> : καθορισμός του μικρότερου/μεγαλύτερου μεγέθους του αντικειμένου. Μικρότερα αντικείμενα αγνοούνται. Προαιρετική παράμετρος

**Πίνακας Π-2- 4: Εντολή detectMultiScale**

cv2.rectangle(frame,(p1,p2),(p3,p4),(B,G,R),thickness)[40]	
Περιγραφή	Σχεδιασμός και εμφάνιση ενός παραλληλογράμμου στην οθόνη
Παράμετροι	<u>frame</u> : η εικόνα ή το βίντεο που θα σχεδιαστεί το παραλληλόγραμμο
	<u>p1,p2</u> : οι συντεταγμένες της άνω αριστερής γωνίας του παραλληλογράμμου
	<u>p3,p4</u> : Το δεύτερο σημείο του παραλληλογράμμου, απέναντι από το p1,p2
	<u>B,G,R</u> : το χρώμα του παραλληλογράμμου Οι τιμές αφορούν τα χρώματα κόκκινο,πράσινο και μπλε και είναι ακέραιοι από το 0 έως το 255
	<u>Thickness</u> : Το πάχος της γραμμής

**Πίνακας Π-2- 5: Εντολή cv2.rectangle**

cv2.imwrite(path, image[p1:p2,p3:p4])[41]	
Περιγραφή	Αποθήκευση της εικόνας που έχει ληφθεί από την κάμερα σε συγκεκριμένο φάκελο
Παράμετροι	<u>path</u> : ο φάκελος που θα αποθηκευτεί η εικόνα. Αν είναι σε διαφορετικό φάκελο από το πρόγραμμα τότε το τελευταίο κομμάτι του μονοπατιού θα είναι το όνομα της εικόνας.
	<u>image</u> : η εικόνα που θα αποθηκευτεί.
	<u>[p1:p2,p3:p4]</u> : Προαιρετικές παράμετροι που αφορούν την απομόνωση συγκεκριμένου μέρους της εικόνας

**Πίνακας Π-2- 6: Εντολή cv2.imwrite**

frame_dif = cv2.absdiff(scr1,scr2)[42]	
Περιγραφή	Υπολογισμός απόλυτης διαφοράς ανάμεσα σε δύο εικόνες. Υπολογίζεται στοιχείο προς στοιχείο
Παράμετροι	<u>scr1,scr2</u> : οι δυο εικόνες εισόδου

**Πίνακας Π-2- 7: Η εντολή cv2.absdiff**

<code>gray = cv2.GaussianBlur(frame,(X,Y),border)[43]</code>	
Περιγραφή	Εφαρμογή Γκαουσιανού φίλτρου στην εικόνα
Παράμετροι	<u>frame</u> : Η εικόνα στην οποία θα εφαρμοστεί το φίλτρο <u>X, Y</u> : Καθορισμός τυπικής απόκλισης στον οριζόντιο και κατακόρυφο άξονα. Και οι δύο τιμές είναι θετικοί ακέραιοι μονοί αριθμοί <u>border</u> : Μέγεθος προβολής του εικονοστοιχείου
Στο παρακάτω παράδειγμα εφαρμόζεται φίλτρο Gauss με αποκλίσεις 15,15 και border 0	
	

**Πίνακας Π-2- 8: Η εντολή GaussianBlur**

<code>faceRecognizer = cv2.face.LBPHFaceRecognizer_create()[44]</code>	
Περιγραφή	Δημιουργία του αναγνωριστή τοπικού δυαδικού μοτίβου

**Πίνακας Π-2- 9: cv2.face.LBPHFaceRecognizer\_create**

<code>faceRecognizer.train(InputArray,labels)[45]</code>	
Περιγραφή	Εκπαίδευση του αναγνωριστή με συγκεκριμένες φωτογραφίες και ετικέτες
Παράμετροι	<u>InputArray</u> : Οι εικόνες με τις οποίες θα γίνει η εκπαίδευση. <u>labels</u> : Τα αναγνωριστικά για τις φωτογραφίες

**Πίνακας Π-2- 10: Η εντολή faceRecognizer.train**

<code>faceRecognizer.save(Path)</code>	
Περιγραφή	αποθήκευση των δεδομένων του αναγνωριστή σε συγκεκριμένο φάκελο
Παράμετροι	<u>Path</u> : Ο φάκελος που θα αποθηκευτούν οι φωτογραφίες. Το τελευταίο μέρος είναι το όνομα του αρχείου που πρέπει να περιλαμβάνει και την κατάληξη.

**Πίνακας Π-2- 11: Η εντολή faceRecognizer.save**

idTag,sure=faceRecognizer.predict(img[p2:p2+i,p1:p1+m])[46]	
Περιγραφή	Πρόβλεψη σε μια εικόνα στα σημεία που εισάγονται ως παράμετροι. Επιστρέφει το αναγνωριστικό καθώς και το πόσο ταιριάζει η εικόνα στα αποθηκευμένα δεδομένα
Παράμετροι	<u>img</u> : Η εικόνα που περιέχει το πρόσωπο
	<u>[p2:p2+i,p1:p1+m]</u> : Η περιοχή μέσα στην εικόνα που βρίσκεται το πρόσωπο

**Πίνακας Π-2- 12: Η εντολή faceRecognizer.predict**

thresholded = cv2.dilate(image,dilation_Element,no_of_interations)[47]	
Περιγραφή	Διαστολή της εικόνας. Η διαστολή επιτυγχάνεται μέσω συνέλιξης της εικόνας με έναν σταθερό πυρήνα σταθερού μεγέθους και σχήματος. Τα πιο συνηθισμένα σχήματα είναι ο κυκλικός ή ο τετράγωνος πυρήνας. Ο πυρήνας έχει ένα κεντρικό σημείο που είναι συνήθως το κέντρο του. Καθώς ο πυρήνας περνάει από κάθε εικονοστοιχείο της εικόνας υπολογίζεται η μέγιστη τιμή και αντικαθιστά το εικονοστοιχείο της εικόνας
Παράμετροι	<u>image</u> : Η εικόνα εισόδου
	<u>dilation_Element</u> : ο πυρήνας (πίνακας NxN) όπου θα γίνει η συνέλιξη
	<u>no_of_interations</u> : Ο αριθμός των επαναλήψεων

**Πίνακας Π-2- 13: Η εντολή cv2.dilate**

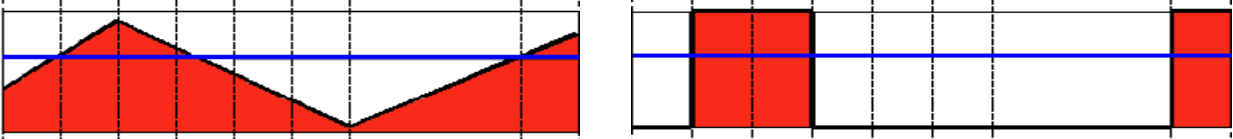
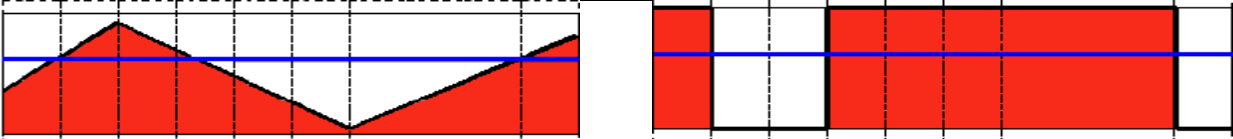
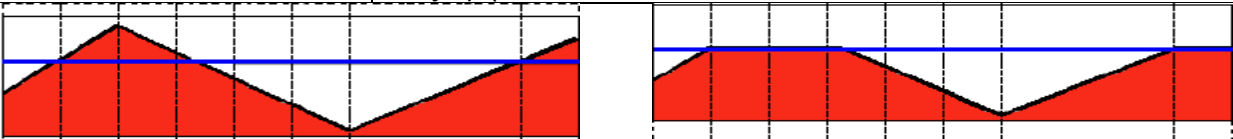
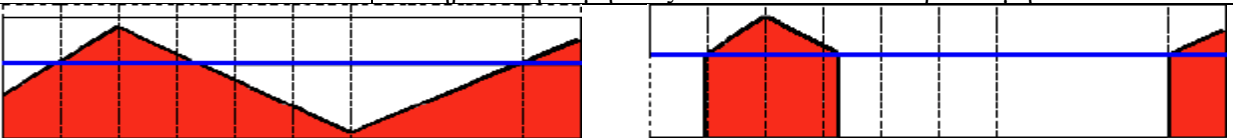
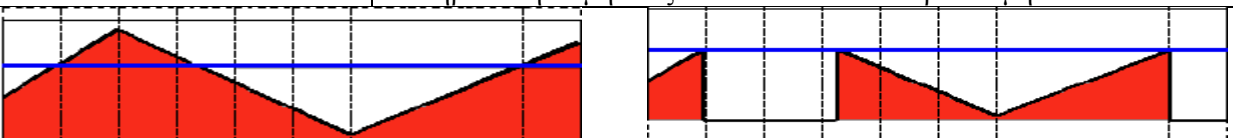
cv2.drawContours(src_image,contour,contour_index,(B,G,R),thickness)[48]	
Περιγραφή	Σχεδιασμός περιγραμμάτων σε μια εικόνα
Παράμετροι	<u>Src_image</u> : Η εικόνα στην οποία θα σχεδιαστεί το περίγραμμα
	<u>contour</u> : Το περίγραμμα που έχει εντοπιστεί από την εντολή findContour
	<u>contour_index</u> : Παράμετρος που καθορίζει το περίγραμμα που θα σχεδιαστεί. Αν είναι αρνητικός αριθμός σχεδιάζονται όλα τα περιγράμματα.
	<u>(B,G,R)</u> : το χρώμα που θα σχεδιαστεί το περίγραμμα
	<u>thickness</u> : Το πάχος της γραμμής του περιγράμματος

**Πίνακας Π-2- 14: Η εντολή drawContours**



(contoured,contoured2) = cv2.findContours(src,mode,method)[49]	
Περιγραφή	Εντοπισμός περιγραμμάτων σε δυαδική εικόνα. Να η εικόνα δεν είναι δυαδική τότε όλα τα στοιχεία με φωτεινότητα μεγαλύτερη του μηδενός θεωρούνται ένα και όλα τα στοιχεία με φωτεινότητα μηδέν ως μηδέν
Παράμετροι	<p><u>src</u>: Η εικόνα εισόδου</p> <p><u>Mode</u>: Μέθοδος ανάκτησης του περιγράμματος. Έχει τέσσερις δυνατές τιμές</p> <p><u>method</u>: Μέθοδος προσέγγισης περιγράμματος. Έχει δύο δυνατές τιμές</p>
Τιμές της παραμέτρου mode	<p><u>CV_RETR_EXTERNAL</u>: Ανάκτηση μόνο των ακραίων εξωτερικών περιγραμμάτων</p> <p><u>CV_RETR_LIST</u>: Ανάκτηση των περιγραμμάτων χωρίς την δημιουργία ιεραρχικών σχέσεων</p> <p><u>CV_RETR_COMP</u>: Ανάκτηση όλων των περιγραμμάτων και οργάνωση του σε δομή διπλού επιπέδου. Στο άνω επίπεδο υπάρχουν τα εξωτερικά όρια των στοιχείων και στο άλλο επίπεδο τα όρια των οπών</p> <p><u>CV_RETR_TREE</u>: Ανάκτηση όλων των περιγραμμάτων και κατασκευή πλήρους ιεραρχίας περιγραμμάτων</p>
Τιμές παραμέτρου method	<p><u>CV_CHAIN_APPROX_NONE</u>: Αποθήκευση όλων των σημείων του περιγράμματος</p> <p><u>CV_CHAIN_APPROX_SIMPLE</u>: Οριζόντια, κάθετη και διαγώνια συμπίεση με σκοπό να μείνουν μόνο τα τελικά σημεία. Για παράδειγμα τα παραλληλόγραμμα κωδικοποιούνται με τέσσερα σημεία</p>

Πίνακας Π-2- 15: Η εντολή cv2.findContours

thresholded = cv2.threshold(frame,thresh, maxVal,Type)[50]	
Περιγραφή	Εφαρμογή κατωφλίου σε εικόνα
Παράμετροι	<b>frame</b> : Η εικόνα που θα εφαρμοστεί το κατώφλι
	<b>thresh</b> : Η τιμή του κατωφλίου
	<b>maxVal</b> : Η μέγιστη τιμή που θα αποδοθεί στα στοιχεία της εικόνας.
cv2.THRESH_BINARY	Όσα εικονοστοιχεία έχουν τιμή μεγαλύτερη από την τιμή του κατωφλίου παίρνουν την τιμή maxVal. Τα υπόλοιπα παίρνουν μηδέν
	
$dest(x, y) = \begin{cases} maxVal & , src(x, y) > thresh \\ 0 & , src(x, y) \leq thresh \end{cases}$	
cv2.THRESH_BINARY_INV	Όσα εικονοστοιχεία έχουν μικρότερη τιμή από την τιμή του κατωφλίου παίρνουν την τιμή μηδέν. Τα υπόλοιπα παίρνουν την τιμή maxVal
	
$dest(x, y) = \begin{cases} 0 & , src(x, y) > thresh \\ maxVal & , src(x, y) \leq thresh \end{cases}$	
cv2.THRESH_TRUNC	Τα στοιχεία έχουν τιμή μεγαλύτερη από την τιμή του κατωφλίου παίρνουν την τιμή του κατωφλίου. Τα υπόλοιπα διατηρούν την αρχική τους τιμή
	
$dest(x, y) = \begin{cases} thresh & , src(x, y) > thresh \\ src(x, y) & , src(x, y) \leq thresh \end{cases}$	
cv2.THRESH_TOZERO	Όσα εικονοστοιχεία έχουν τιμή μεγαλύτερη από την τιμή του κατωφλίου διατηρούν την τιμή τους ενώ τα υπόλοιπα παίρνουν μηδέν
	
$dest(x, y) = \begin{cases} thresh & , src(x, y) > thresh \\ src(x, y) & , src(x, y) \leq thresh \end{cases}$	
cv2.THRESH_TOZERO	Όσα εικονοστοιχεία έχουν τιμή μεγαλύτερη από την τιμή του κατωφλίου διατηρούν την τιμή τους ενώ τα υπόλοιπα παίρνουν μηδέν
	
$dest(x, y) = \begin{cases} thresh & , src(x, y) > thresh \\ src(x, y) & , src(x, y) \leq thresh \end{cases}$	

Πίνακας Π-2- 16: Η εντολή cv2.threshold

## Βιβλιογραφία

1. Yang M., Kriegman D. (2002) 'Detecting Faces in Images: A Survey'
2. De Carrera F., Marquez I. (2010) 'Face Recognition Algorithms'
3. Viola M., Jones P. 'Robust Real-Time Face Detection' (2001)
4. Lienhart R., Maydt J. (2002) 'An Extended Set of Haar-like Features for Rapid Object Detection'
5. Cerna L, Cámara-Chávez G., Menotti D.,(2013) 'Face Detection: Histogram of Oriented Gradients and Bag of Feature Method'
6. Γκαγκαράς Σ., Τερζακης Π. (2018) 'Εφαρμογές Ρομποτικής Όρασης'
7. Introduction to Face Recognition Part 1 Available at <http://what-when-how.com/face-recognition/introduction-to-face-recognition-part-1/>
8. Starovoitov V., Samal D (1999) 'A geometric approach to face recognition'
9. Divyarajsinh P., Brijesh M. (2014) 'Face Recognition Methods & Application'
10. Liton P., Abdulla Al S., (2012) 'Face Recognition using principal component analysis'
11. Bhattacharyya K. S., Rahul K. (2013) 'Face Recognition by linear discriminant analysis'
12. Parth Singh (2021) Understanding Face Recognition using PBPH algorithm available at <https://www.analyticsvidhya.com/blog/2021/07/understanding-face-recognition-using-lbph-algorithm/>
13. Kelvin Salton do Prado (2017) Face Recognition: Understanding LBPH Algorithm available at <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>
14. Ahonen T., Hadid A., Pietikäinen M., (2004) Face Recognition with Local Binary Patterns
15. Song Zhou & Sheng Xiao (2018) 3D face recognition: a survey available at <https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0157-2#Sec15>
16. Wikipedia (2022) Foreground detection available at [https://en.wikipedia.org/wiki/Foreground\\_detection](https://en.wikipedia.org/wiki/Foreground_detection)
17. Wikipedia (2022) Optical flow available at [https://en.wikipedia.org/wiki/Optical\\_flow](https://en.wikipedia.org/wiki/Optical_flow)
18. Nanonets (2021) Introduction to Motion Estimation with Optical Flow available at <https://nanonets.com/blog/optical-flow/>
19. NVIDIA (2022) Getting started with jetson Nano developer kit available at <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>
20. NVIDIA (2022) Jetson Nano available at <https://developer.nvidia.com/embedded/jetson-nano>
21. Github (2022) Linux for Terga (L4T) First Steps available at <https://znmeb.github.io/edgyR/articles/cc-14t-first-steps.html>
22. Wikipedia (2022) Raspberry Pi available at [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi)
23. Raspberry Pi (2022) Raspberry Pi 4 Tech Specs available at <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>
24. Raspberry Pi (2022) Raspberry Pi OS available at <https://www.raspberrypi.com/software/>
25. Advanced IP Scanner (2022) available at <https://www.advanced-ip-scanner.com/>
26. Raspberry Pi Recourses (2022) MySQL available at [https://raspberrypi-projects.com/pi/software\\_utilities/web-servers/mysql](https://raspberrypi-projects.com/pi/software_utilities/web-servers/mysql)
27. The Robotics Back-End (2022) available at <https://roboticsbackend.com/raspberry-pi-arduino-serial-communication/>
28. Electronic Wings (2022) Raspberry Pi UART Communication using python and C available at <https://www.electronicwings.com/raspberry-pi/raspberry-pi-uart-communication-using-python-and-c>

29. Raspberry Pi (2022) Camera available at <https://www.raspberrypi.com/documentation/accessories/camera.html>
30. Adafruit (2022) PIR Motion Sensor available at <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor>
31. DS04-NFC 360 degrees dataset
32. SG90 datasheet
33. KY-040 datasheet
34. Computer Vision (2022) The 12 Most Popular Computer Vision Tools in 2022 available at <https://viso.ai/computer-vision/the-most-popular-computer-vision-tools/>
35. Αρχή προστασίας δεδομένων (2022) Συστήματα βιντεοεπιτήρησης σε οικίες και πολυκατοικίες available at [https://www.dpa.gr/el/enimerwtiko/thematikes\\_enotites/eisagwgi\\_videoepitirisi/oikies\\_videoepitirisi](https://www.dpa.gr/el/enimerwtiko/thematikes_enotites/eisagwgi_videoepitirisi/oikies_videoepitirisi)
36. OpenCV (2022) CascadeClassifier Class Reference available at [https://docs.opencv.org/4.x/d1/de5/classcv\\_1\\_1CascadeClassifier.html](https://docs.opencv.org/4.x/d1/de5/classcv_1_1CascadeClassifier.html)
37. OpenCV (2022) VideoCapture Class Reference available at [https://docs.opencv.org/3.4/d8/dfc/classcv\\_1\\_1VideoCapture.html](https://docs.opencv.org/3.4/d8/dfc/classcv_1_1VideoCapture.html)
38. OpenCV (2022) Color Space Conversions available at [https://docs.opencv.org/3.4/d8/d01/group\\_imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/d8/d01/group_imgproc_color_conversions.html)
39. OpenCV (2022) CascadeClassifier Class Reference available at [https://docs.opencv.org/3.4/d1/de5/classcv\\_1\\_1CascadeClassifier.html](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html)
40. OpenCV (2022) Drawing Functions in OpenCV available at [https://docs.opencv.org/4.x/dc/da5/tutorial\\_py\\_drawing\\_functions.html](https://docs.opencv.org/4.x/dc/da5/tutorial_py_drawing_functions.html)
41. OpenCV (2022) Image file reading and writing available at [https://docs.opencv.org/3.4/d4/da8/group\\_imgcodecs.html](https://docs.opencv.org/3.4/d4/da8/group_imgcodecs.html)
42. OpenCV (2022) Operations on arrays available at [https://docs.opencv.org/3.4/d2/de8/group\\_core\\_array.html](https://docs.opencv.org/3.4/d2/de8/group_core_array.html)
43. OpenCV (2022) Image Filtering available at [https://docs.opencv.org/4.x/d4/d86/group\\_\\_imgproc\\_\\_filter.html](https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html)
44. OpenCV (2022) LBPHFaceRecognizer Class Reference available at [https://docs.opencv.org/3.4/df/d25/classcv\\_1\\_1face\\_1\\_1LBPHFaceRecognizer.html](https://docs.opencv.org/3.4/df/d25/classcv_1_1face_1_1LBPHFaceRecognizer.html)
45. OpenCV (2022) LBPHFaceRecognizer Class Reference (training) available at [https://docs.opencv.org/3.4/dd/d65/classcv\\_1\\_1face\\_1\\_1FaceRecognizer.html#ac8680c2aa9649ad3f55e27761165c0d6](https://docs.opencv.org/3.4/dd/d65/classcv_1_1face_1_1FaceRecognizer.html#ac8680c2aa9649ad3f55e27761165c0d6)
46. OpenCV (2022) LBPHFaceRecognizer Class Reference (Predict) available at [https://docs.opencv.org/3.4/dd/d65/classcv\\_1\\_1face\\_1\\_1FaceRecognizer.html#aa2d2f02faffab1bf01317ae6502fb631](https://docs.opencv.org/3.4/dd/d65/classcv_1_1face_1_1FaceRecognizer.html#aa2d2f02faffab1bf01317ae6502fb631)
47. OpenCV (2022) Eroding and Dilating available at [https://docs.opencv.org/3.4/db/df6/tutorial\\_erosion\\_dilatation.html](https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html)
48. OpenCV (2022) Drawing Functions (contours) available at [https://docs.opencv.org/4.x/d6/d6e/group\\_imgproc\\_draw.html#ga746c0625f1781f1ffc9056259103edbc](https://docs.opencv.org/4.x/d6/d6e/group_imgproc_draw.html#ga746c0625f1781f1ffc9056259103edbc)
49. OpenCV (2022) Contours : Getting Started available at [https://docs.opencv.org/4.x/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html)
50. OpenCV (2022) Miscellaneous Image Transformations available at [https://docs.opencv.org/4.x/d7/d1b/group\\_\\_imgproc\\_\\_misc.html#ggaa9e58d2860d4afa658ef70a9b1115576a147222a96556ebc1d948b372bcd7ac59](https://docs.opencv.org/4.x/d7/d1b/group__imgproc__misc.html#ggaa9e58d2860d4afa658ef70a9b1115576a147222a96556ebc1d948b372bcd7ac59)

Ανάπτυξη λογισμικού αναγνώρισης και παρακολούθησης προσώπων μέσω κάμερας...