ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κίνηση οχήματος με Raspberry Pi 3 με αναγνώριση γραμμής πορείας και σημάτων οδικής κυκλοφορίας

Βορδάκης Εμμανουήλ

Ομέρ Μουσταφά

Εισηγητής: Ιωάννης Ν. Έλληνας, Καθηγητής

ΑΙΓΑΛΕΩ

Ιούνιος 2017

(Κενό φύλλο)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κίνηση οχήματος με Raspberry Pi 3 με αναγνώριση γραμμής πορείας και σημάτων οδικής κυκλοφορίας

Βορδάκης Εμμανουήλ
Α.Μ. 42876

Ομέρ Μουσταφά
Α.Μ. 42927

Εισηγητής: Ιωάννης Ν. Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης:

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Βορδάκης Εμμανουήλ**, του **Γεωργίου**, με αριθμό μητρώου **42876,** φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

“Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.”

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Ομέρ Μουσταφά**, του **Ιμπραήμ**, με αριθμό μητρώου **42927**, φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

“Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.”

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θέλουμε να ευχαριστήσουμε τον καθηγητή μας κ. Έλληνα για την ευκαιρία που μας έδωσε να ασχοληθούμε με ένα τόσο ενδιαφέρον, επίκαιρο, καθώς και καινοτόμο θέμα. Μπορεί αρκετές φορές να βρεθήκαμε σε “αδιέξοδο”, όμως καταφέραμε με πείσμα και υπομονή να ανταπεξέλθουμε πλήρως στις ανάγκες της εργασίας. Μας δόθηκαν αρκετές στιγμές δημιουργίας και ενασχόλησης, τόσο με την κατασκευή ενός οχήματος, όσο και με τον προγραμματισμό του, ώστε να εκτελεί τις λειτουργίες που επιθυμούμε.

Στην ολοκλήρωση του έργου μας, επίσης, βοήθησαν και όλοι οι υπόλοιποι καθηγητές και εργαστηριακοί συνεργάτες του τμηματός μας, οι οποίοι συνέβαλαν μέσω των μαθημάτων και ιδιαίτερα των εργαστηρίων, με το να μας μυήσουν σε μικρότερα, όμως πολύ σημαντικά, παρόμοια πειράματα.

Τέλος, ευχαριστούμε θερμά τις οικογένειές μας που μας στήριξαν, οικονομικά και ηθικά, καθόλη τη διάρκεια της πτυχιακής.

(Κενό φύλλο)

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την κίνηση ενός οχήματος με το Raspberry Pi 3 Model B, ικανό να αναγνωρίζει, μέσω κάμερας, μια μαύρη γραμμή που σηματοδοτεί την πορεία του, καθώς και ορισμένα σήματα του Κώδικα Οδικής Κυκλοφορίας. Το όχημά μας αποτελείται από 2 κινητήρες συνεχούς ρεύματος (DC) που επικοινωνούν σειριακά με τις ακίδες του Raspberry Pi 3. Η κάμερα λαμβάνει καρέ προς καρέ την εικόνα που βρίσκεται μπροστά της. Ο ανιχνευτής των σημάτων εκπαιδεύεται μέσω της βιβλιοθήκης OpenCV, ώστε αργότερα να αναγνωρίζει, σε πραγματικό χρόνο, το εκάστοτε σήμα που η κάμερα θα συναντήσει. Το όχημα εκτελεί την αντίστοιχη εντολή (πχ. στροφή, στοπ) ενεργοποιώντας για καθεμιά ένα διαφορετικού χρώματος LED. Παράλληλα, αναπτύσσεται πρόγραμμα σε γλώσσα Python, για την υλοποίηση αυτής της εφαρμογής.

Επιστημονική περιοχή: Επεξεργασία Εικόνας

Λέξεις-κλειδιά: Raspberry Pi, Κινητήρες, Ανίχνευση αντικειμένου, Αναγνώριση αντικειμένου, Σήμα, OpenCV

(Κενό φύλλο)

ABSTRACT

This thesis deals with the movement of a vehicle with Raspberry Pi 3 Model B, capable of recognizing, through a camera, a black line that marks its route, as well as some signals of the Highway Code. Our vehicle consists of two DC motors that communicate serially with the Raspberry Pi 3 pins. The camera takes frame-by-frame the image in front of it. The signal detector is being trained via the OpenCV library, so later it will recognize, in real time, each sign the camera will meet. The vehicle performs the corresponding order (eg turn, stop) by activating a different LED color for each. At the same time, a Python program is being developed for the implementation of this application.

Scientific Area: Image Processing

Keywords: Raspberry Pi, Motors, Object Detection, Object Recognition, Signal, OpenCV

(Κενό φύλλο)

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Adaboost	Adaptive Boosting
ARM	Advanced RISC Machine
ASCII	American Standard Code for Information Interchange
BGR	Blue Green Red
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central Processing Unit
DC	Direct Current
Esc	Escape
GAB	Gentle AdaBoost
GB	GigaByte
GPIO	General Purpose Input/Output
HDMI	High-Definition Multimedia Interface
IP	Internet Protocol
LAN	Local Area Network
LED	Light Emitting Diode
microSD	micro Secure Digital
OpenCV	Open Source Computer Vision Library
PWM	Pulse Width Modulation
RAM	Random-access memory
RPi	Raspberry Pi
rpm	Revolution Per Minute
sec	Second
SSH	Secure Shell
TTL	Transistor–Transistor Logic
USB	Universe Serial Bus
xml	Extensible Markup Language
ΛΣ	Λειτουργικό Σύστημα

ΟΡΟΛΟΓΙΑ

adaptive boosting	προσαρμοστική ενίσχυση
autostart	απευθείας εκτέλεση
cascade	διάταξη καταρράκτη
cascade of classifiers	καταρράκτης ταξινομητών
classification	ταξινόμηση
decision tree	δένδρο απόφασης
ethernet	πρότυπο δίκτυο υπολογιστών για ενσύρματα τοπικά δίκτυα
Fritzing	πρόγραμμα σχεδίασης κυκλωμάτων
powerbank	φορτιστής
pulse width modulation	διαμόρφωση εύρους παλμών
Raspbian Jessie	εικόνα που αποθηκεύεται στη microSD του RPi
Ubuntu	λειτουργικό σύστημα
Windows	λειτουργικό σύστημα

ΚΕΦΑΛΑΙΟ 1

Σε αυτό το κεφάλαιο γίνεται μια ιστορική αναδρομή γύρω από το Raspberry Pi. Επίσης, αναφέρονται τα λειτουργικά συστήματα που υποστηρίζει και η διαδικασία της εγκατάστασης της εικόνας Raspbian Jessie.

1.1 Raspberry Pi

Το Raspberry Pi (για συντομία RPi) είναι ένας προσωπικός υπολογιστής “τσέπης”, βάρους 45 γραμμαρίων, βασισμένος στο λειτουργικό σύστημα Linux. Πρόκειται για μια χαμηλού κόστους, στην ουσία, πλακέτα με την οποία οι ενδιαφερόμενοι, κυρίως μαθητές και φοιτητές, εξασκούνται σε θέματα προγραμματισμού και ηλεκτρονικών. Άλλωστε, αυτή ήταν και η αφορμή για την κατασκευή της από εργαζόμενους του τμήματος Computer Laboratory του πανεπιστημίου Cambridge. Συγκεκριμένα, ήταν επιθυμητή η κατασκευή ενός μικροϋπολογιστή που θα προσελκύει τους μαθητές, μυώντας τους στον κλάδο της πληροφορικής. Με το RPi μπορούν να δημιουργηθούν σημαντικές εφαρμογές για την εκπαίδευση (εκμάθηση προγραμματισμού), την ψυχαγωγία (αναμετάδοση (streaming) μουσικών κομματιών, δημιουργία παιχνιδιών, λήψη φωτογραφιών ανά τακτά χρονικά διαστήματα), την ασφάλεια (συναγερμός σπιτιού, προστασία υπολογιστή από ανεπιθύμητους εισβολείς, παρακολούθηση μέσω κάμερας, έλεγχος παράβασης του ΚΟΚ για τα οχήματα), την επικοινωνία (ρύθμιση του RPi ως σημείο ασύρματης πρόσβασης, ασύρματος χειρισμός ρομπότ) κτλ.

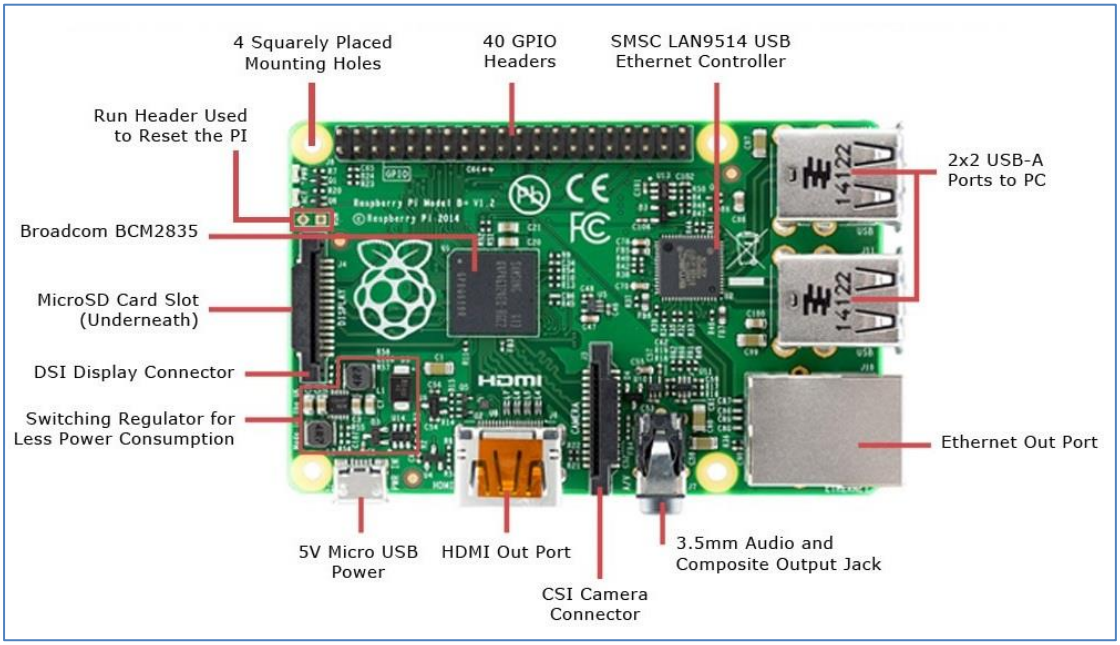
Το πρώτο RPi βγήκε στην αγορά στις 29 Φεβρουαρίου 2012 με επεξεργαστή ARMv6k στα 700 MHz και 256MB RAM, ενώ το τελευταίο μοντέλο τρίτης γενιάς (Raspberry Pi Generation 3 Model B) (Εικόνα 1.1) -μέχρι στιγμής- λανσαρίστηκε 4 χρόνια μετά, δηλαδή στις 29 Φεβρουαρίου 2016 αντικαθιστώντας το Raspberry Pi Generation 2 Model B που είχε κυκλοφορήσει το Φεβρουάριο του 2015.

Εικόνα 1.1: Raspberry Pi Generation 3 Model B και κάρτα microSD

Το πρώτο μέρος του ονόματός του (Raspberry) έχει επιλεγεί λόγω μιας παράδοσης, κατά την οποία καινούργιες πλατφόρμες υπολογιστών ονοματοδοτούνται με φρούτα. Το δεύτερο συνθετικό του (Pi) προκύπτει από την γλώσσα προγραμματισμού Python.

Αυτή τη στιγμή στην Ελλάδα, η μέση τιμή του κυμαίνεται στα 40 ευρώ.

Όπως και το Raspberry Pi 2 Model B, το 3ης γενιάς μοντέλο (Εικόνα 1.2) διαθέτει:



Εικόνα 1.2: Raspberry Pi 3

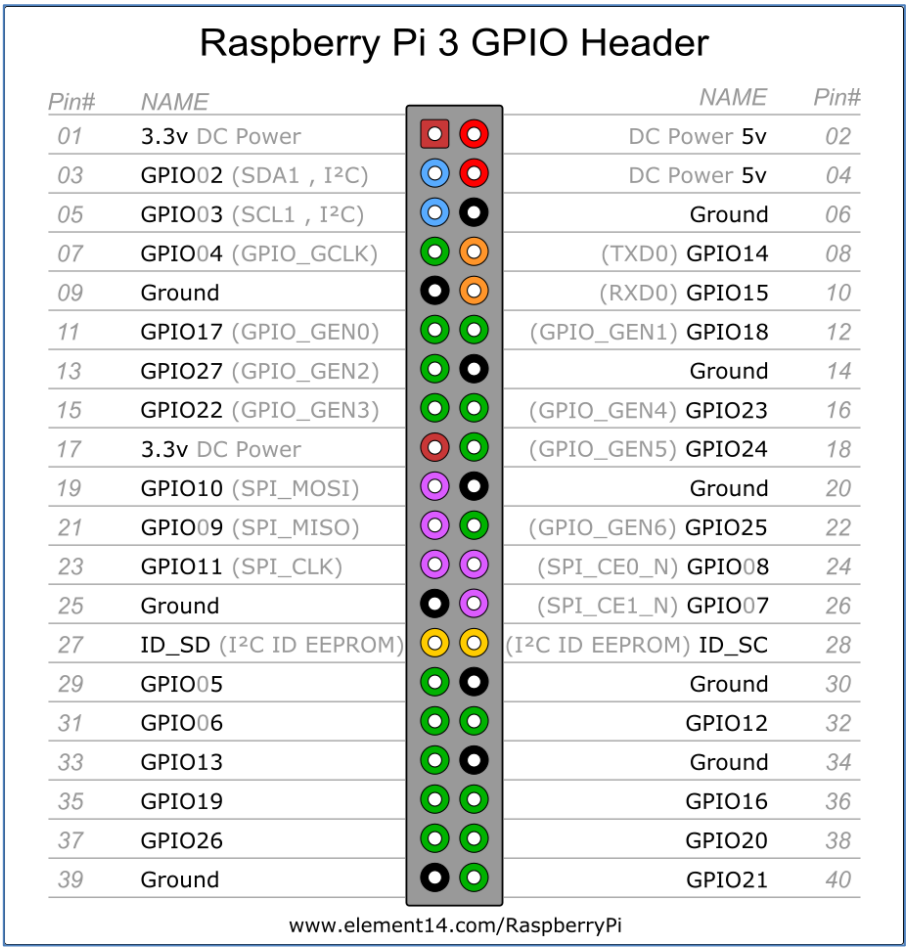
- 1GB μνήμη RAM (δε δέχεται επέκταση)
- Θύρα Full HDMI για εικόνα υψηλής ανάλυσης (χρησιμοποιεί ρεύμα 50mA)
- 4 θύρες USB για την επικοινωνία του RPi 3 με περιφερειακές συσκευές, όπως πχ. ποντίκι, πληκτρολόγιο (καταναλώνουν ρεύμα από 100-1000mA)
- Υποδοχή micro-USB, από την οποία τροφοδοτείται με 5V
- 40 ακίδες (pins) GPIO γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά (καταναλώνουν μέχρι 50mA)
- Θύρα Ethernet για ενσύρματη τοπική δικτύωση
- Βύσμα με υποδοχή 3.5mm με υποστήριξη ήχου και βίντεο (Composite Video)
- Σύνδεση με κάμερα (Camera interface - CSI) (η κάμερα απαιτεί ρεύμα 250mA)
- Σύνδεση με οθόνη (Display interface - DSI)

Επιπλέον, το νέο RPi περιλαμβάνει:

- Μία 64-bit τετραπύρηνη ARMv8 CPU στα 1.2GHz
- Διπύρηνη κάρτα γραφικών Broadcom VideoCore IV 3D στα 250MHz
- 802.11n Wireless LAN και Bluetooth 4.1 για ασύρματη σύνδεση
- Μέγιστη ταχύτητα μεταφοράς USB: 60 MB/s
- Bluetooth Low Energy (BLE) για αισθητή μείωση κατανάλωσης ενέργειας κατά τη διάρκεια της επικοινωνίας

Η μέγιστη ένταση ρεύματος του RPi 3 είναι 1200mA (1.2A) από τα οποία μένουν μόνο τα 800mA (μέγιστη ισχύς 4W για τάση 5V), όπου αρκούν για την ομαλή λειτουργία μεγάλης γκάμας εφαρμογών. Τα υπόλοιπα 400mA καταναλώνονται από το ίδιο το RPi για τη λειτουργία του. Δεν προτείνεται η τροφοδοσία του μέσω μπαταριών, διότι ακόμα και αν η συνολική τάση τους είναι αρκετά μεγάλη (που όμως δεν πρέπει να ξεπερνά τα 5V), κάποια στιγμή η τάση αυτή θα μειωθεί αισθητά και θα υπάρξει πρόβλημα με τη σωστή λειτουργία του RPi.

1.2 Ακίδες GPIO (General Purpose Input/Output)



Εικόνα 1.3: Λειτουργίες των GPIO

Στο πάνω αριστερό μέρος του RPi 3 υπάρχει μια κεφαλή (header) όπου βρίσκονται οι ακίδες GPIO, οι οποίες είναι ψηφιακές είσοδοι και έξοδοι που (συνήθως) προγραμματίζονται για επικοινωνία με περιφερειακές συσκευές. Το RPi 3 διαθέτει 40 τέτοια pins (Εικόνα 1.3) που κατανέμονται σε 2 σειρές των 20 και για να χρησιμοποιηθούν ως είσοδοι ή έξοδοι χρειάζεται να τα ορίσουμε μέσω του κώδικα Python.

2 pins (1 και 17) είναι για τροφοδοσία 3.3V (μέχρι 50mA).

2 pins (2 και 4) είναι για τροφοδοσία 5V (παίρνουν τροφοδοσία από το micro-USB).

8 pins (6, 9, 14, 20, 25, 30, 34 και 39) είναι για γείωση.

28 pins (τα υπόλοιπα) είναι για χρήση GPIO σαν είσοδοι ή έξοδοι.

ΠΡΟΣΟΧΗ Όλα τα pins δέχονται τροφοδοσία μέχρι 3.3V διαφορετικά υπάρχει κίνδυνος να καούν.

1.3 Λειτουργικά συστήματα που υποστηρίζει το Raspberry Pi 3

- Noobs
- Raspbian
- Ubuntu Mate
- Snappy Ubuntu Core
- Windows 10 IoT
- PiNet
- Risc OS
- Weather Station
- Arch Linux ARM

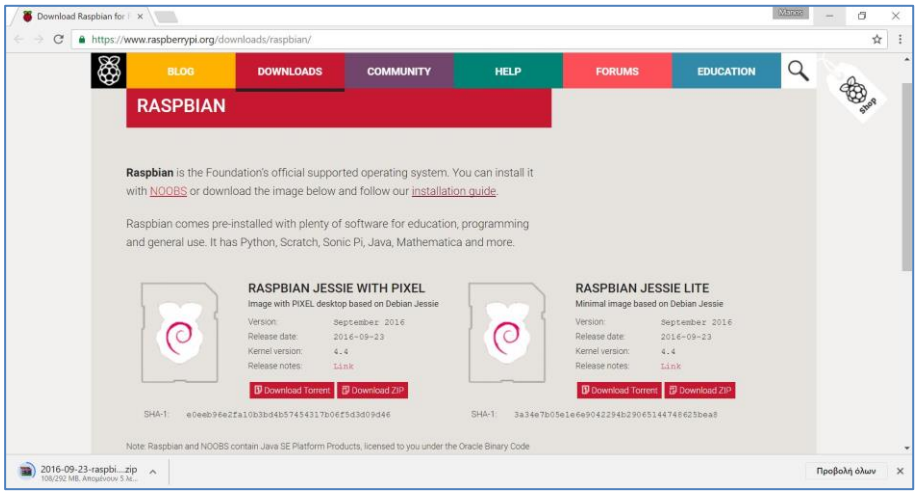
Στην εφαρμογή μας χρησιμοποιούμε το Raspbian και συγκεκριμένα την έκδοση Jessie, μιας και είναι το “επίσημο” λειτουργικό σύστημα του RPi 3. Βασίζεται στο Debian και είναι σχεδιασμένο ειδικά για το RPi 3. Παρέχει, μάλλον, τις περισσότερες δυνατότητες για εκείνο, αφού έχει ποικίλες προεγκατεστημένες εφαρμογές που αφορούν καθημερινές εργασίες, όπως φυλλομετρητή, προγράμματα επεξεργασίας κειμένου και ηλεκτρονικού ταχυδρομείου, καθώς και προγράμματα εκμάθησης προγραμματισμού που στο σύνολό τους στοχεύουν στο να βοηθούν το χρήστη να διεκπεραιώνει διάφορες εργασίες.

Τα υπόλοιπα λειτουργικά συστήματα δεν είναι τόσο διαδεδομένα και η χρήση τους είναι ωφέλιμη κάτω από συγκεκριμένες περιπτώσεις. Πιο συγκεκριμένα, το Windows 10 IoT αφορά σε προγραμματιστές και είναι συμβατό μόνο με συστήματα που έχουν Windows 10. Το Arch Linux ARM αφορά και αυτό σε έμπειρους χρήστες. Τέλος, το RISC OS δεν είναι τόσο φιλικό προς το χρήστη, αφού δεν σχετίζεται με τη γνωστή μορφή των λειτουργικών συστημάτων. Για παράδειγμα, για να χρησιμοποιήσουμε τον επεξεργαστή κειμένου (text editor) χρειάζεται να κλικάρουμε στην προεγκατεστημένη εφαρμογή για τον text editor (iStrongED), και αφού εμφανιστεί στη γραμμή εργασιών, πατώντας πάνω του, αυτό ανοίγει στη συνέχεια με μορφή παράθυρου.

1.4 Εγκατάσταση εικόνας (image) Raspbian Jessie

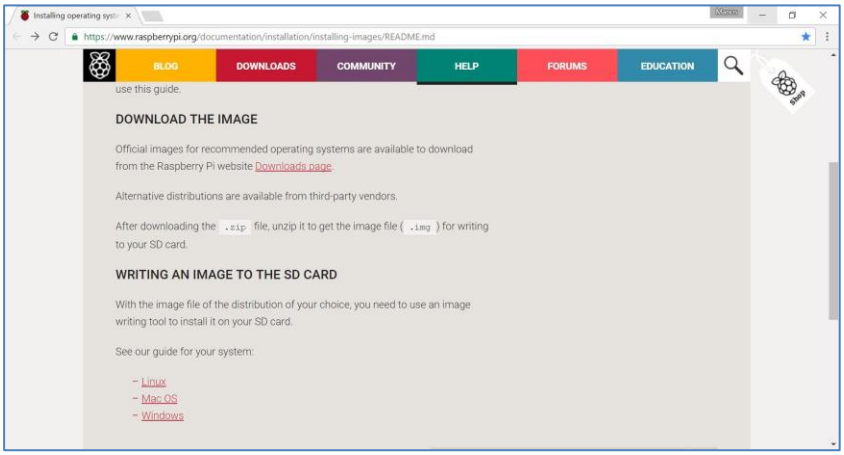
Για να μπορέσουμε να χρησιμοποιήσουμε το Raspberry Pi 3, χρειάζεται πρώτα να εγκαταστήσουμε μία εικόνα στην κάρτα microSD. Επομένως, εκτελούμε τις παρακάτω οδηγίες:

1. Κάνουμε λήψη της τελευταίας έκδοσης του Raspbian Jessie (Εικόνα 1.5), είτε απευθείας από την ιστοσελίδα (https://www.raspberrypi.org/downloads/raspbian/), είτε μέσω Torrent (Εικόνα 1.4).



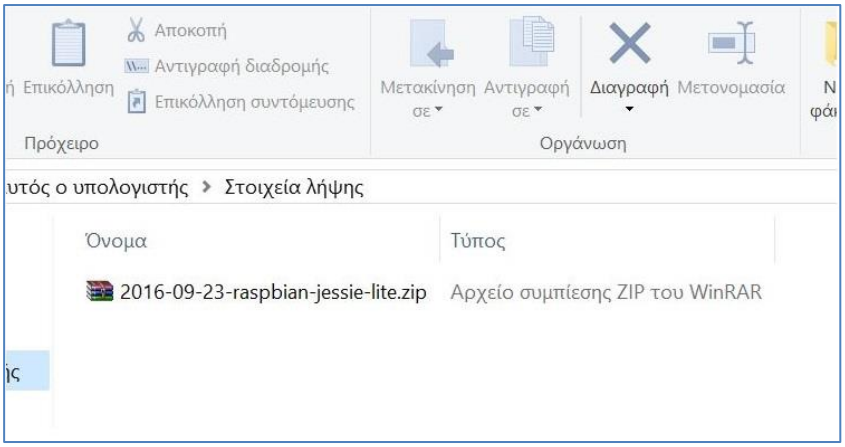
Εικόνα 1.4: Επιλογή εικόνας Raspbian Jessie

(Αναλυτικές οδηγίες εγκατάστασης υπάρχουν στο σύνδεσμο: https://www.raspberrypi.org/documentation/installation/installing-images/README.md)



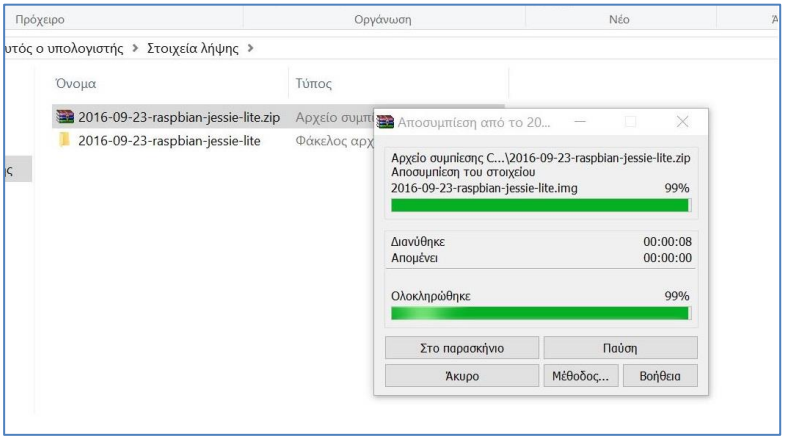
Εικόνα 1.5: Λήψη εικόνας Raspbian Jessie

2. Το αρχείο βρίσκεται σε μορφή .zip (Εικόνα 1.6),…



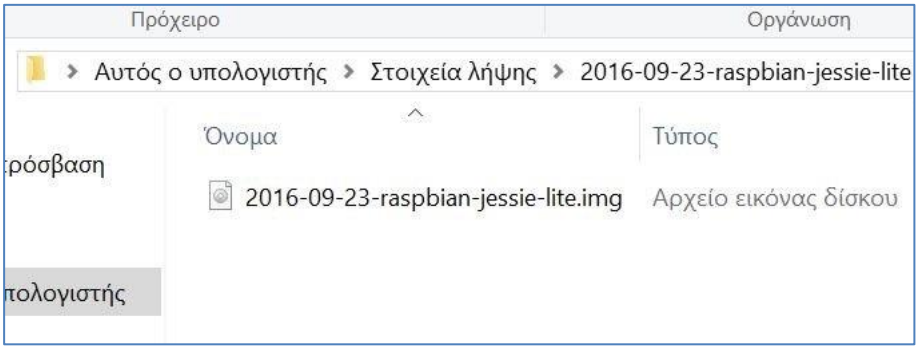
Εικόνα 1.6: Αρχείο .zip εικόνας

3. …όπου με την αποσυμπίεσή του (Εικόνα 1.7)…



Εικόνα 1.7: Αποσυμπίεση της εικόνας

4. …προκύπτει το αρχείο εικόνας (.img) (Εικόνα 1.8).



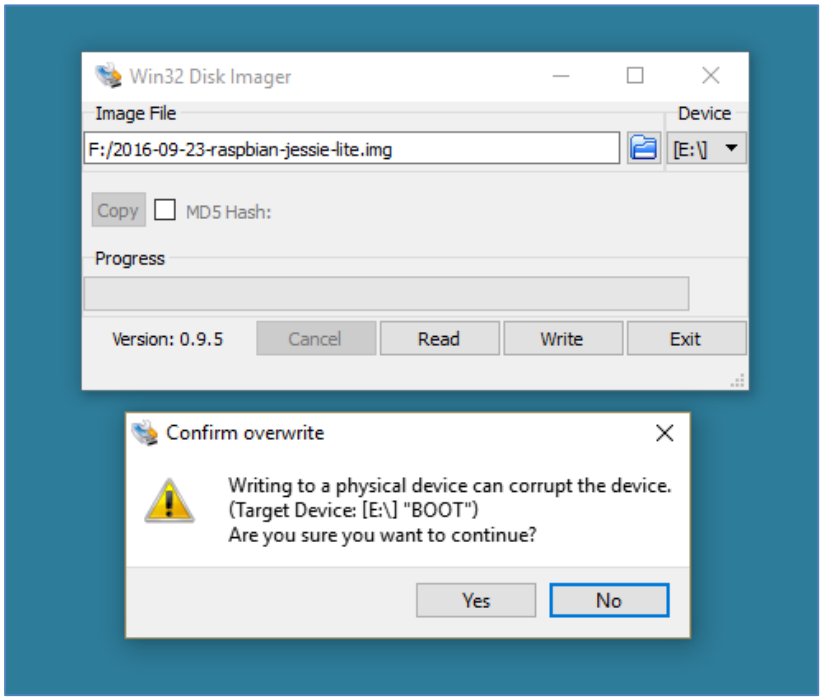
Εικόνα 1.8: Αρχείο εικόνας .img

5. Μέσω του προγράμματος Win32 Disk Imager (https://sourceforge.net/projects/win32diskimager/) (Εικόνα 1.9) ...

Εικόνα 1.9: Λήψη προγράμματος Win32 Disk Imager

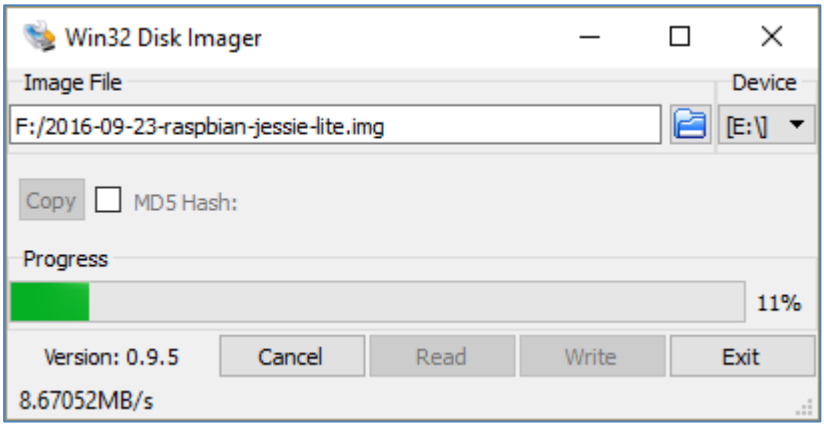
…εγκαθιστούμε σε μια κάρτα microSD το αρχείο εικόνας.

6. Αφού επιλέξουμε το αρχείο εικόνας (Image File) από το μπλε φάκελο στα δεξιά, επιλέγουμε τη συσκευή (Device) όπου θα κάνουμε εγγραφή την εικόνα. Τέλος, πατάμε την επιλογή “**Write**”, όπου ζητείται η επιβεβαίωση για την εγγραφή της εικόνας στην κάρτα SD (Εικόνα 1.10).



Εικόνα 1.10: Επιλογή εικόνας για εγγραφή στην microSD

Με την επιλογή “**Yes**” ξεκινά η εγγραφή (Εικόνα 1.11).



Εικόνα 1.11: Εγγραφή εικόνας στην microSD

7. Μόλις ολοκληρωθεί η εγγραφή, εκτελούμε ασφαλή κατάργηση της κάρτας από τον υπολογιστή και την τοποθετούμε στο Raspberry Pi 3.

ΚΕΦΑΛΑΙΟ 2

Σε αυτό το κεφάλαιο εξηγείται αρχικά η διαδικασία της σύνδεσης του Raspberry Pi μέσω Ethernet στα Windows. Στη συνέχεια, αναφέρεται το πώς γίνεται η απευθείας εκτέλεση του προγράμματος αμέσως μόλις συνδεθεί η τροφοδοσία.

2.1 Σύνδεση του Raspberry Pi μέσω Ethernet στα Windows

Ένας τρόπος πρόσβασης στο περιβάλλον του Raspberry Pi 3 αποτελεί η σύνδεση με καλώδιο Ethernet.

1. Αρχικά, πρέπει να βρούμε την IP διεύθυνσή του RPi, οπότε το συνδέουμε μέσω καλωδίου Ethernet στο router. Ανοίγουμε τη σελίδα διαμόρφωσης του router (http://192.168.1.1) και βάζοντας το κατάλληλο ψευδώνυμο και κωδικό έχουμε πρόσβαση στην IP του RPi.

2. Για τη σύνδεση του RPi με τα Windows, αφαιρούμε τη μία άκρη του Ethernet από το router και τη συνδέουμε με τον υπολογιστή.

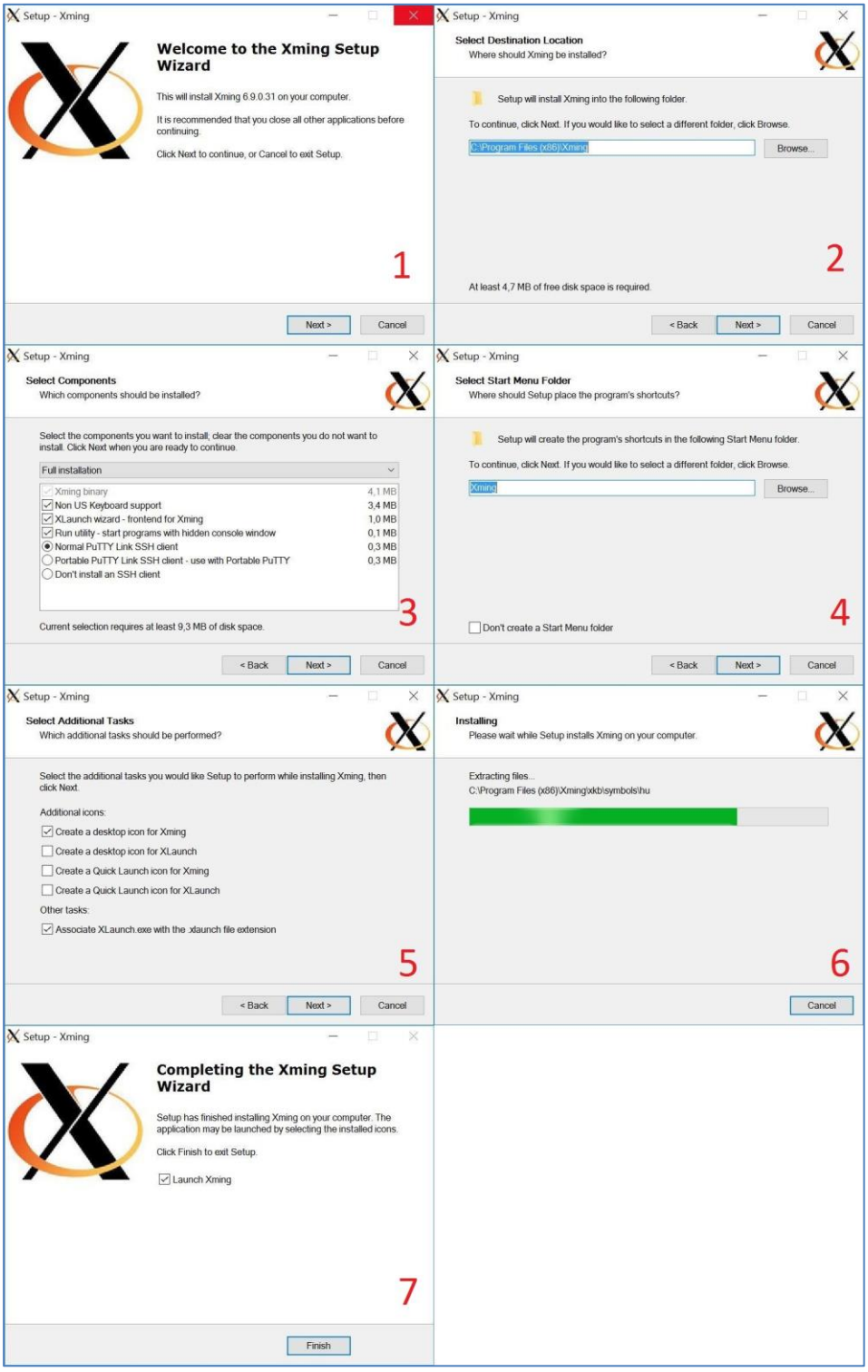
3. Για τη χρήση του SSH είναι απαραίτητο το **Putty** (https://the.earth.li/~sgtatham/putty/latest/x86/putty.exe), γι' αυτό και το εγκαθιστούμε στον υπολογιστή.

Το SSH (Secure Shell) είναι ένα δικτυακό πρωτόκολλο βασισμένο σε Unix, το οποίο προορίζεται για τα Windows και στοχεύει στην ασφαλή απομακρυσμένη μεταφορά δεδομένων και εκτέλεση εντολών ανάμεσα στον υπολογιστή και στο RPi.

4. Το **Xming** είναι απαραίτητο για την απομακρυσμένη πρόσβαση του γραφικού περιβάλλοντος του RPi 3, οπότε εκτελούμε και εδώ την εγκατάστασή του.

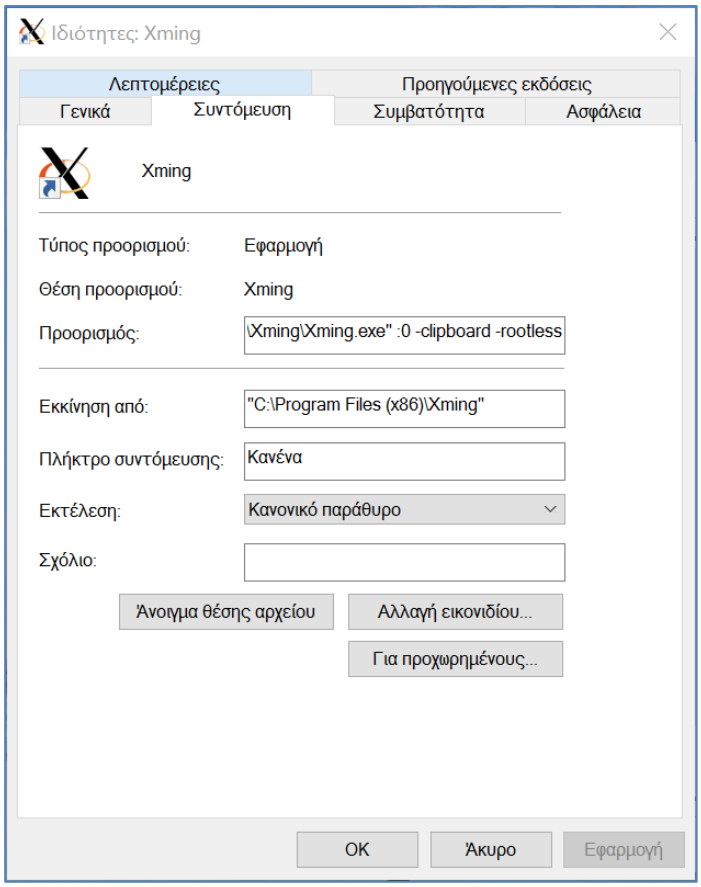
(http://downloads.sourceforge.net/project/xming/Xming/6.9.0.31/Xming-6-9-0-31-setup.exe)

5. Αφού εκτελέσουμε τα βήματα της εγκατάστασης του Xming με τη σειρά που φαίνονται στις παρακάτω εικόνες (Εικόνα 2.1)…



Εικόνα 2.1: Εγκατάσταση Xming

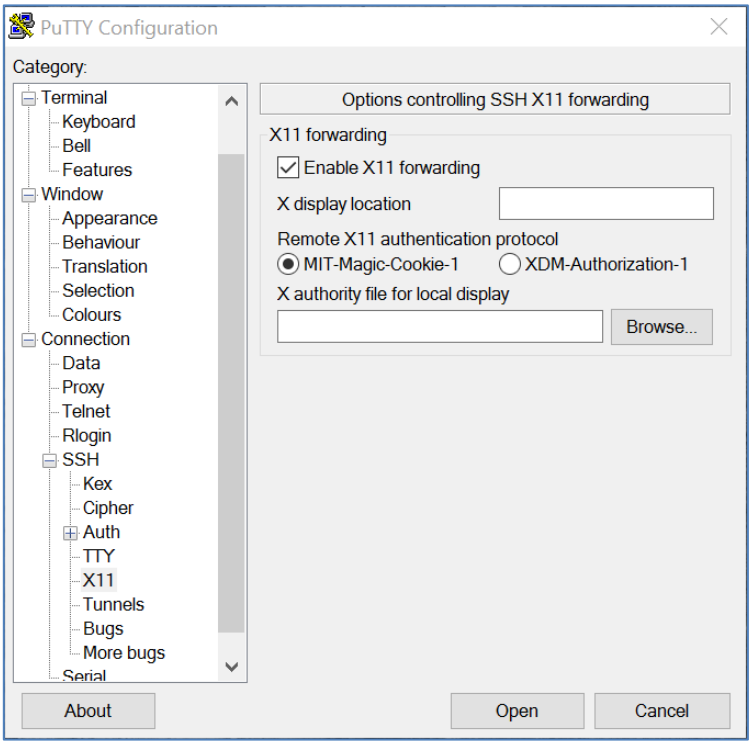
6. …αλλάζουμε τον προορισμό της συντόμευσης του Xming σε: "C:\Program Files (x86)\Xming\Xming.exe" :0 -clipboard -rootless" (Εικόνα 2.2).



Εικόνα 2.2: Ιδιότητες Xming

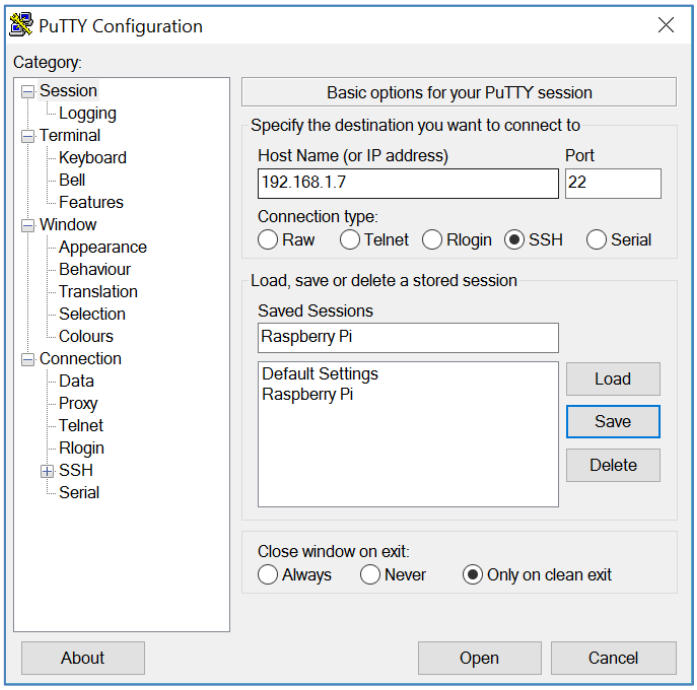
7. Ανοίγουμε με διπλό κλικ το Putty και το διαμορφώνουμε κατάλληλα, ώστε να συνδεθούμε με το RPi 3, ακολουθώντας τα παρακάτω βήματα (Εικόνα 2.3):

    a. Επιλέγουμε από την αριστερή στήλη με τη σειρά:
    Connection->SSH->X11
    και στη συνέχεια επιλέγουμε την ενεργοποίηση του X11 forwarding



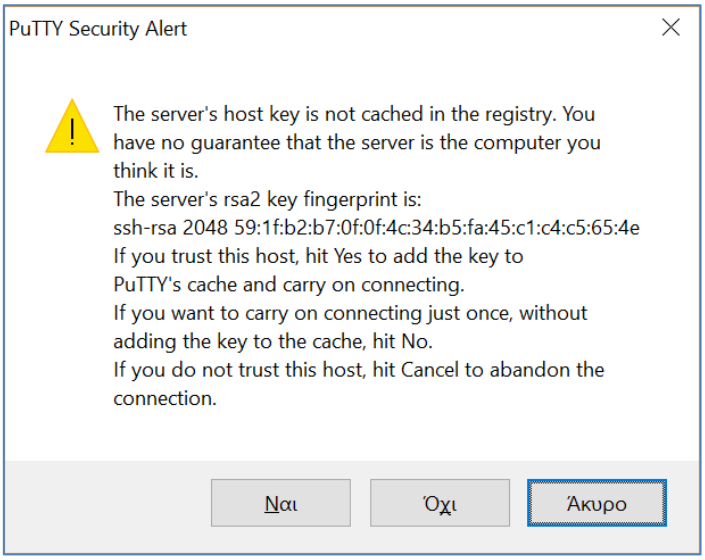
Εικόνα 2.3: Ενεργοποίηση του X11 forwarding

b. Σημειώνουμε τη θύρα 22 για το SSH στη σύνδεση του υπολογιστή μας με το RPi 3, καθώς και την IP του RPi 3 (Εικόνα 2.4).



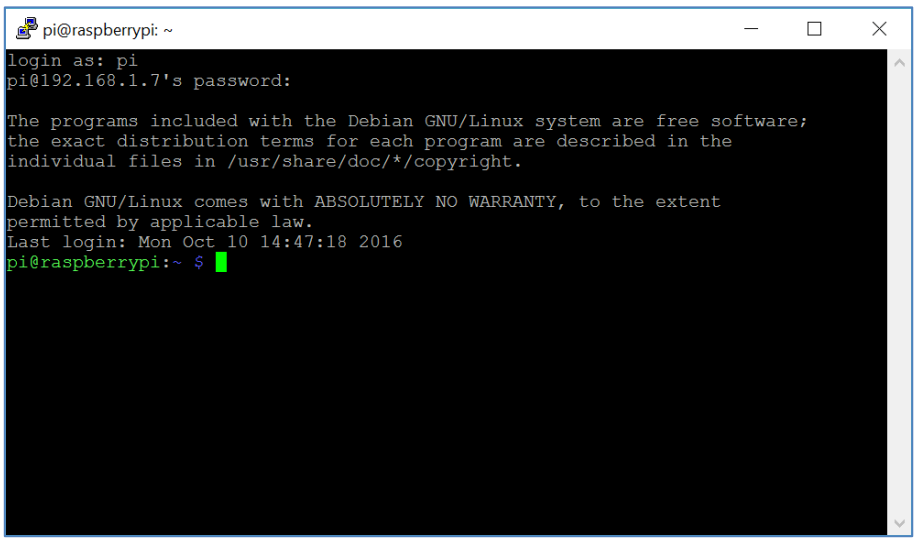
Εικόνα 2.4: Ορισμός IP διεύθυνσης και πόρτας

c. Μόλις επιλέξουμε “**Open**” εμφανίζεται ένα μήνυμα που αφορά στο κλειδί ασφαλείας ανάμεσα στις 2 συσκευές που θα συνδέσουμε. Πατάμε την επιλογή “**Ναι**” για να συνεχίσουμε (Εικόνα 2.5).



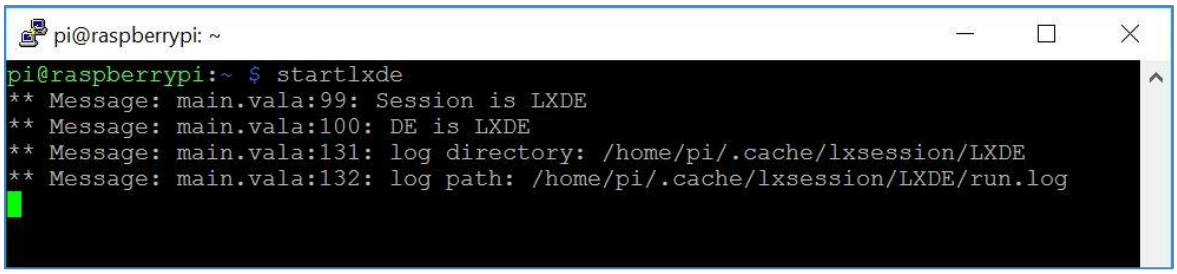
Εικόνα 2.5: Μήνυμα ασφαλείας για σύνδεση με το RPi 3

8. Εντέλει, πληκτρολογώντας το όνομα χρήστη και τον κωδικό για το RPi 3 συνδεόμαστε απευθείας με τη συσκευή (Εικόνα 2.6).



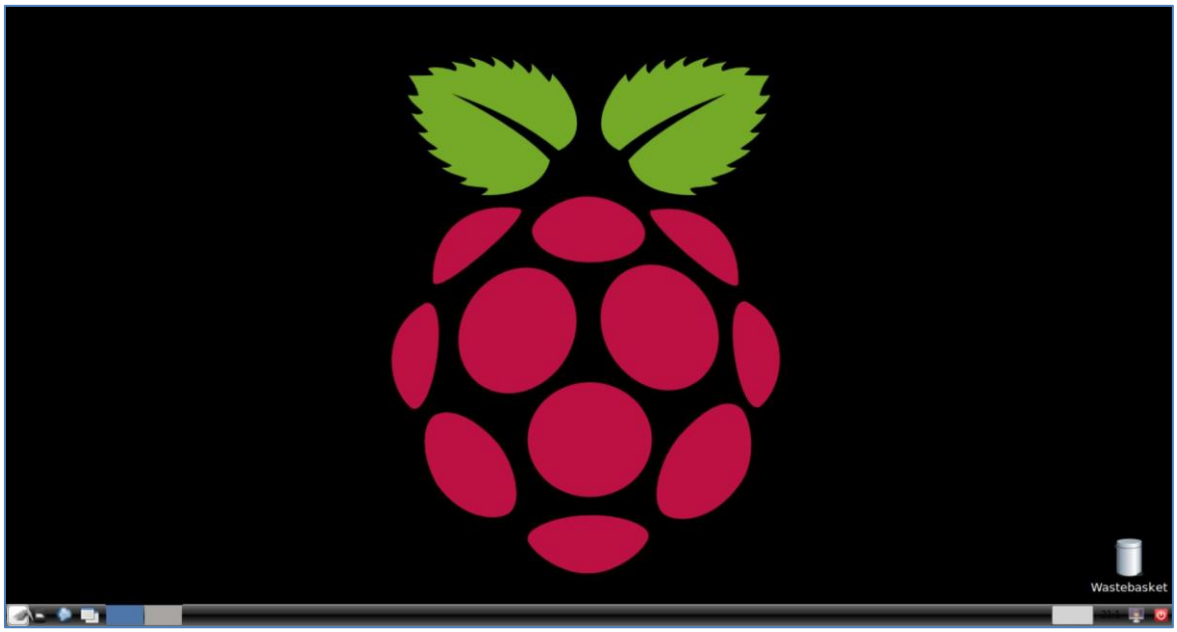
Εικόνα 2.6: Σύνδεση με το RPi 3

9. Έπειτα, αφού ανοίξουμε (με δεξί κλικ) τη συντόμευση Xming με δικαιώματα διαχειριστή, εφόσον το επιθυμούμε, πληκτρολογούμε στο τερματικό του Putty την εντολή “startlxde” (Εικόνα 2.7)…



Εικόνα 2.7: Εντολή “startlxde”

10. …και σε λίγα δευτερόλεπτα έχουμε πρόσβαση στο γραφικό περιβάλλον του Raspberry Pi 3 (Εικόνα 2.8).



Εικόνα 2.8: Γραφικό περιβάλλον του RPi 3

<u>ΠΡΟΣΟΧΗ</u>: Για διασφάλιση των δεδομένων της κάρτας SD, καλό είναι να γίνει ένα εφεδρικό (backup) της κάρτας, στην περίπτωση όπου για οποιοδήποτε λόγο δημιουργηθεί πρόβλημα με την κάρτα. Αυτό μπορεί να επιτευχθεί μέσω του ίδιου προγράμματος με το οποίο έγινε προηγουμένως η εγγραφή της εικόνας στην κάρτα. Εν ολίγοις, μόλις εκτελέσουμε το Win32 Disk Imager, επιλέγουμε το μονοπάτι της εικόνας, καθώς και την συσκευή που έχει την εικόνα, επιλέγουμε “**Read”** και με την ολοκλήρωση της διαδικασίας έχουμε το αρχείο εικόνας.

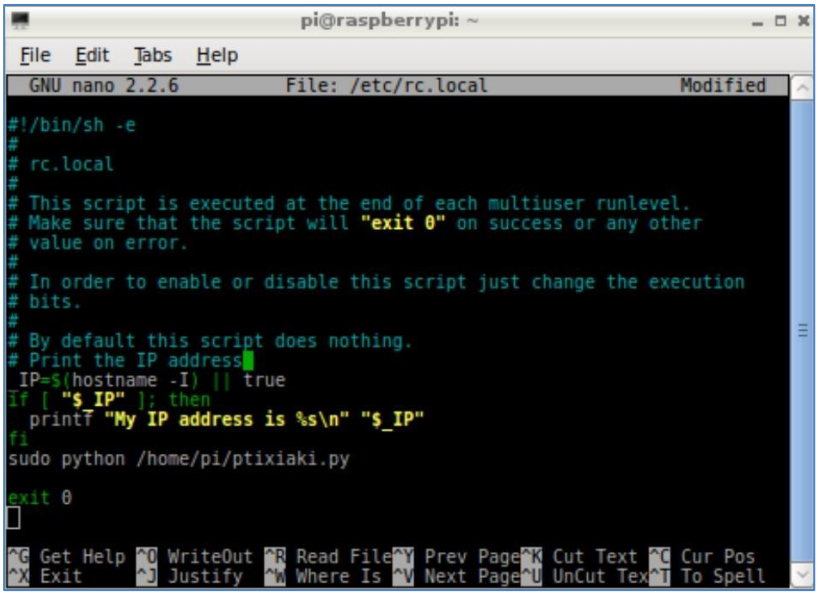
2.2 Απευθείας εκτέλεση της εφαρμογής με τη σύνδεση powerbank

Στην προηγούμενη ενότητα αναφερθήκαμε στη σύνδεση του RPi 3 μέσω Ethernet για να μπορέσουμε να εκτελέσουμε την εφαρμογή μας. Όμως, αυτό μπορεί να αποτελεί πρόβλημα, ειδικά στην περίπτωση όπου δεν είναι εφικτή ή απαραίτητη η χρήση ενός υπολογιστή. Για παράδειγμα, για μία απλή επίδειξη της εφαρμογής σε εξωτερικό περιβάλλον η ύπαρξη Η/Υ είναι μάλλον περιττή. Άλλη περίπτωση είναι η εφαρμογή να πρόκειται να εφαρμοστεί από άτομα μη εξοικειωμένα ή μη αρεστά στη χρήση ενός Η/Υ.

Για τους λόγους αυτούς, προσπαθήσαμε να επιλύσουμε το θέμα με τον υπολογιστή καταφέρνοντας να εκτελέσουμε την εφαρμογή κατευθείαν μόλις συνδέσουμε το κύκλωμα με το powerbank.

Δεδομένου ότι το RPi 3 δεν έχει, μέχρι στιγμής, τη δυνατότητα αυτόματης έναρξης προγραμμάτων κατά την έναρξή του, χρειάζεται να κάνουμε ορισμένες ενέργειες:

Για την αυτόματη εκτέλεση χωρίς τη χρήση του γραφικού περιβάλλοντος Raspbian Jessie επεξεργαζόμαστε το αρχείο “rc.local” πληκτρολογώντας στο τερματικό την εντολή “sudo nano /etc/rc.local”. Με αυτή την εντολή θα ανοίξει το αρχείο rc.local, στο οποίο προσθέτουμε πάνω από το “exit 0” την εντολή “sudo python /home/pi/ptixiaki.py”, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 2.9):



Εικόνα 2.9: Autostart

Αποθηκεύουμε το αρχείο με “Ctrl+X”, έπειτα πατάμε “Y” για Yes και τέλος “Enter”. Για να δοκιμάσουμε ότι το παραπάνω λειτουργεί, βγάζουμε από την τροφοδοσία το RPi, το συνδέουμε ξανά και παρατηρούμε ότι έπειτα από μερικά δευτερόλεπτα το όχημα αρχίζει να κινείται.

Δυστυχώς, δεν υπάρχει διακόπτης ενεργοποίησης/απενεργοποίησης (on/off) του RPi. Για να το θέσουμε σε λειτουργία απλά το συνδέουμε στην τροφοδοσία. Για να το απενεργοποιήσουμε (αν είμαστε στο γραφικό περιβάλλον) πληκτρολογούμε στο τερματικό την εντολή “sudo halt -h” περιμένοντας μέχρις ότου σβήσουν όλα τα LED, εκτός αυτού της τροφοδοσίας ή “sudo shutdown -h now”.

ΚΕΦΑΛΑΙΟ 3

Στην αρχή του κεφαλαίου παρουσιάζονται τα εξαρτήματα και οι προδιαγραφές τους. Μετά, γίνεται λόγος για το ολοκληρωμένο L293D και για τη λειτουργία του. Τέλος, περιγράφουμε πως μπορούμε να ρυθμίσουμε την ταχύτητα του οχήματος με παλμούς PWM.

3.1 Τα εξαρτήματα και οι προδιαγραφές τους

- Το λειτουργικό σύστημα είναι εγκατεστημένο σε μια κάρτα microSD (Sandisk Ultra microSDHC 8GB Class 10), η οποία έχει μέγεθος 8GB (δέχεται και πάνω από 32GB) ώστε να υπάρχει επαρκής αποθηκευτικός χώρος (απαιτείται χώρος τουλάχιστον 4GB) και είναι κλάσης 10 (απαιτείται τουλάχιστον κλάση 4) για να έχουμε γρήγορες ταχύτητες εγγραφής και ανάγνωσης της κάρτας (πχ. γρήγορη εγκατάσταση λειτουργικού συστήματος). Είναι προφανές ότι είναι προαπαιτούμενο να υπάρχει υποδοχή για την microSD στον υπολογιστή (Εικόνα 3.1).

  Κόστος **10 ευρώ**.

Εικόνα 3.1: MicroSD

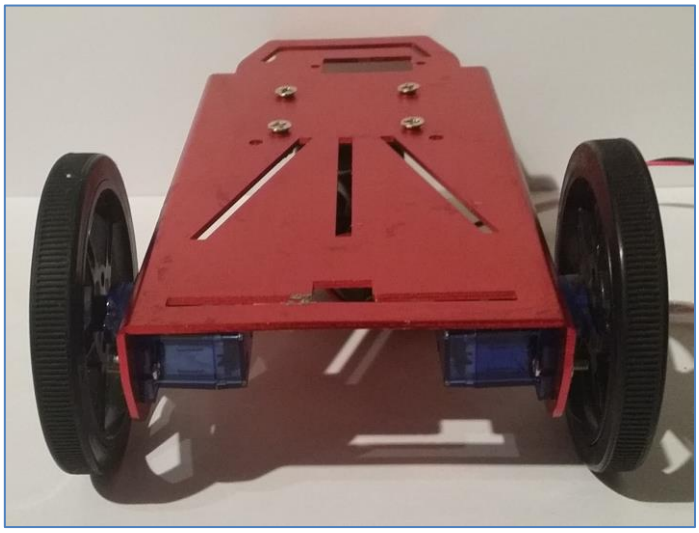
- Για την προστασία του RPi 3 χρησιμοποιούμε την ειδική θήκη για να αποφύγουμε βλάβες από πιθανά ατυχήματα, όπως βραχυκύκλωμα ή πτώση του Raspberry (Εικόνα 3.2).

  Κόστος **5 ευρώ**.

Εικόνα 3.2: Θήκη RPi 3

- Το όχημα που χρησιμοποιούμε μαζί με τους 2 DC κινητήρες είναι το “Mini Robot Rover Chassis Kit - 2WD with DC Motors” (Εικόνα 3.3).

  Κόστος **18 ευρώ**.



Εικόνα 3.3: 2WD Mini Robot Rover Chassis Kit

Οι κινητήρες που χρησιμοποιούμε έχουν τα εξής χαρακτηριστικά:

Διαστάσεις: 32.3 x 12.3 x 29.9 χιλ.

Βάρος: 8.4γρ.

Μέγιστη ταχύτητα χωρίς φορτίο: 110RPM(στροφές/λεπτό) (4.8v) / 130RPM (6v)

Μέγιστο ρεύμα (χωρίς φορτίο): 100mA (4.8v) / 120mA (6v)

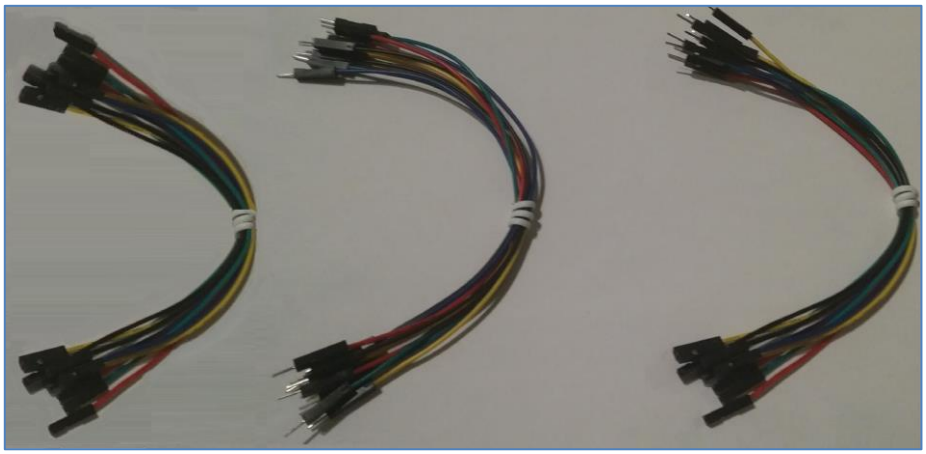
Μέγιστη ροπή (4.8v): 1.3 kg/cm

Μέγιστη ροπή (6v): 1.5 kg/cm

Μέγιστο ρεύμα (για μέγιστη ροπή κινητήρα): 550mA (4.8v) / 650mA (6v)

- Για τη σύνδεση των συσκευών και των εξαρτημάτων χρειαστήκαμε καλώδια: “Jumper Wires Female/Female”, “Jumper Wires Male/Male” και “Jumper Wires Female/Male” (Εικόνα 3.4).

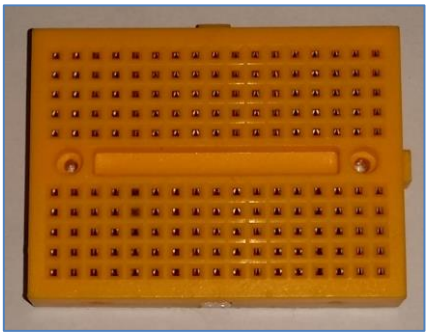
  Κόστος **5 ευρώ.**



Εικόνα 3.4: Καλώδια

- Η σύνδεση έγινε με τη βοήθεια ενός breadboard (Εικόνα 3.5).

  Κόστος **1.6 ευρώ.**



Εικόνα 3.5: Breadboard

- Επίσης χρησιμοποιούμε το ολοκληρωμένο L293D (Εικόνα 3.6).

  Κόστος **4 ευρώ**.

Εικόνα 3.6: L293D

- Χρησιμοποιήσαμε 4 LED για την ένδειξη της ευθείας πορείας και για τα τρία σήματα (left, notturn και stop) (Εικόνα 3.7)…

  Κόστος **0,80 ευρώ.**

Εικόνα 3.7: LED

- …και 4 αντιστάσεις 330Ω για το κάθε LED (Εικόνα 3.8).

  Κόστος **0,10 ευρώ.**

Εικόνα 3.8: Αντιστάσεις 330Ω

- Η κάμερα που χρησιμοποιήσουμε για την εφαρμογή μας είναι η “Web Camera Crypto Budget V” (Εικόνα 3.9).

  Κόστος **10 ευρώ.**

Εικόνα 3.9: Κάμερα

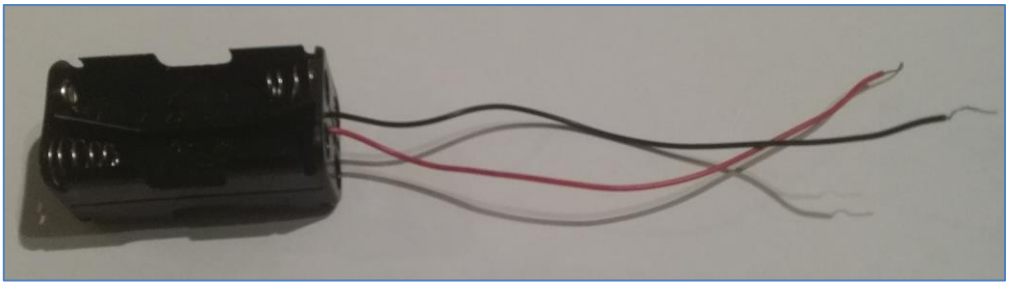
- Για τη τροφοδοσία των κινητήρων χρειαστήκαμε 4 μπαταρίες τύπου AA (Εικόνα 3.10).

  Κόστος **2.5 ευρώ.**

Εικόνα 3.10: Μπαταρίες

- Οι μπαταρίες τοποθετήθηκαν σε μια βάση (Εικόνα 3.11).

  Κόστος **0.5 ευρώ.**



Εικόνα 3.11: Θήκη μπαταριών

- Η τροφοδοσία για το RPi 3 γίνεται μέσω ενός φορητού φορτιστή (powerbank). Το μοντέλο είναι το “TP LINK TL-PB5200 Power Bank 5200mAh”. Είναι αρκετά μικρό (92.7 x 53.5 x 22.0 χιλ.) και ελαφρύ (135γρ.) και παρέχει ρεύμα εξόδου 2.4A και τάση τροφοδοσίας 5V (Εικόνα 3.12).

  Κόστος **15 ευρώ**.

Εικόνα 3.12: Powerbank

3.2 Ολοκληρωμένο L293D

Το ολοκληρωμένο L293D είναι κατάλληλο για την οδήγηση 2 DC κινητήρων, ταυτόχρονα και ανεξάρτητα, προς όλες τις κατευθύνσεις. Διατίθεται σε διάφορες εκδόσεις με την ικανότητα οδήγησης με (μη επαναλαμβανόμενο -για να μην καούν τα τρανζίστορ που περιλαμβάνει εσωτερικά) ρεύμα έως 1.2A και μέγιστο ρεύμα ανά κανάλι 600mA. Έχει τάση λειτουργίας από 4.5V έως 36V και δέχεται είσοδο μέχρι 7V. Επίσης, είναι συμβατό με TTL και CMOS μπορεί να λειτουργεί σε συνθήκες θερμοκρασίας από 0 - 70^{O} C.

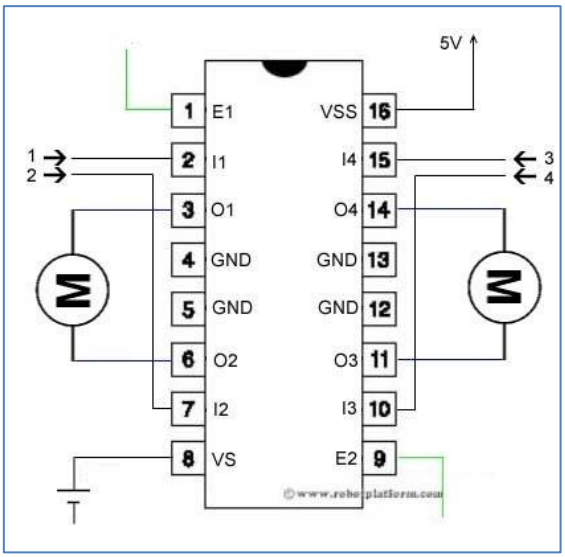
Έχει 16 pins:

- Τα Pins 1 και 9 είναι για την ενεργοποίηση των 2 κινητήρων
- Το pin 8 είναι για την τροφοδοσία των 2 κινητήρων
- Το pin 16 είναι για την τροφοδοσία του L293D από το Raspberry Pi 3
- Τα pins 4, 5, 12 και 13 συνδέονται με τη γείωση του Raspberry Pi 3
- Τα pins 2 και 7 είναι λογικές είσοδοι για έλεγχο του αριστερού κινητήρα, ενώ τα pins 15 και 10 του δεξιού κινητήρα
- Τα pins 3 και 6 είναι λογικές έξοδοι του αριστερού κινητήρα, ενώ τα pins 14 και 11 του δεξιού κινητήρα

Οι λόγοι που χρησιμοποιούμε το συγκεκριμένο ολοκληρωμένο είναι ότι πρώτον, τα pins του Raspberry Pi 3 δεν είναι ικανά να στείλουν το ρεύμα που χρειάζονται οι 2 κινητήρες για να κινηθούν, και δεύτερον ότι θέλουμε να δώσουμε στους κινητήρες οποιαδήποτε ταχύτητα και κατεύθυνση, μέσω αντιστροφής της φοράς του ρεύματος (που επιτυγχάνεται με το L293D).

3.3 Λειτουργία του ολοκληρωμένου L293D

Για την λειτουργία του ολοκληρωμένου χρειάζεται να ρυθμίσουμε κατάλληλα τους ακροδέκτες INPUT και OUTPUT του μικροελεγκτή για τον έλεγχο του κάθε κινητήρα.



Εικόνα 3.13: Ολοκληρωμένο L293D

- Τροφοδοσία L293D: Ο ακροδέκτης 16 (Vss) τροφοδοτείται με 5V από τα pins του Raspberry Pi 3 (ή γενικότερα μέχρι 36V και **όχι παραπάνω** για να μην καεί), αποκλειστικά και μόνο για τη λειτουργία του.

- Τροφοδοσία κινητήρων: Ο ακροδέκτης 8 (Vs) πρέπει να τροφοδοτείται από μπαταριές και **όχι** από τα pins του Raspberry Pi 3. Το L293D με το pin 8 παρέχει τροφοδοσία στους κινητήρες από 4.5V έως 36V. Άρα, για να μπορεί να λειτουργήσει ένας κινητήρας αρκεί να συνδέσουμε την κατάλληλη τροφοδοσία στο pin 8.

- Έλεγχος Ταχύτητας: Μέσω των ακροδεκτών E1 (pin 1) και E2 (pin 9) δίνεται σήμα PWM. Η χρήση PWM μας δίνει τη δυνατότητα ελέγχου της ταχύτητας περιστροφής του κινητήρα. Αν κάποιο Enable έχει ενεργοποιηθεί, τότε η έξοδος εξαρτάται από τα inputs όσον αφορά στην κίνηση του κινητήρα. Εάν είναι

απενεργοποιημένο, τότε δεν έχει σημασία σε τι κατάσταση βρίσκονται οι είσοδοι, αφού ο κινητήρας θα παραμείνει ακίνητος.

- Έλεγχος Φοράς: Μέσω των 1,2 και 3,4 (Εικόνα 3.13) επιλέγεται η φορά περιστροφής. Πιο συγκεκριμένα, οι ακροδέκτες 2 και 7 ελέγχουν την κίνηση του κινητήρα που θα συνδεθεί στα αριστερά του L293D, ενώ οι ακροδέκτες 15 και 10 ελέγχουν την κίνηση του κινητήρα που θα συνδεθεί στα δεξιά του.

3.4 Κινητήρες DC

Ο ηλεκτρικός κινητήρας ή ηλεκτροκινητήρας, αποτελεί διάταξη για τη μετατροπή της ηλεκτρικής ενέργειας σε μηχανική ενέργεια, που χρησιμοποιείται ευρέως σε σύγχρονες βιομηχανικές εγκαταστάσεις. Οι Συνεχούς Ρεύματος κινητήρες (DC motor) αποτελούν υποκατηγορία των ηλεκτρικών κινητήρων. Τα κυριότερα πλεονεκτήματά τους είναι η ευκολία ελέγχου της ταχύτητάς τους και η εξαιρετική δυναμική συμπεριφοράς τους. Η αρχή λειτουργίας ενός ηλεκτρικού κινητήρα στηρίζεται κυρίως στην ανάπτυξη δύναμης Laplace. Εν συντομία, όταν ένας αγωγός διαρρέεται από ηλεκτρικό ρεύμα έντασης I και ταυτόχρονα βρίσκεται σε μαγνητικό πεδίο έντασης B, αναπτύσσεται πάνω σε αυτόν δύναμη Laplace ίση με:

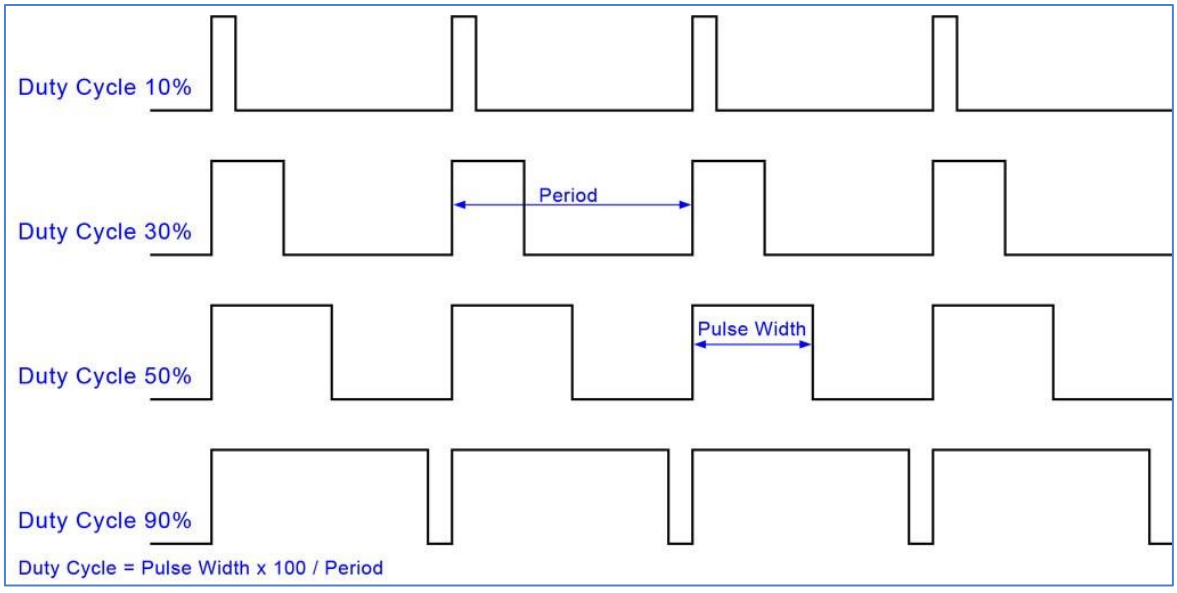
$F=B*I*L*\sin\Phi$, όπου:

- I = Ένταση Ρεύματος
- L = Μήκος Αγωγού
- B = Ένταση Μαγνητικού πεδίου
- φ = Η γωνία που σχηματίζει ο αγωγός με τη διεύθυνση των δυναμικών γραμμών

Οι κινητήρες του οχήματος ενεργοποιούνται βάσει της πορείας της γραμμής, αλλά και των σημάτων.

3.5 PWM (Pulse Width Modulation)

Είδαμε παραπάνω ότι για να ελέγξουμε την ισχύ που εφαρμόζεται σ' έναν κινητήρα ή καλύτερα, την ταχύτητα περιστροφής του κινητήρα, εφαρμόζουμε παλμούς PWM στην τροφοδοσία του. Πρόκειται για μια τεχνική διαμόρφωσης εύρους παλμών ή αλλιώς για μια περιοδική κυματομορφή κατά την οποία εναλλάσσονται ανά τακτά χρονικά διαστήματα δύο καταστάσεις: η κατάσταση ON και η κατάσταση OFF. Κατά την κατάσταση ON, έχουμε τη μέγιστη τιμή παλμού (5V), ενώ κατά την OFF έχουμε μηδενική τιμή (0V). Ένας παλμός PWM μπορεί να πάρει 256 τιμές (0 - 255). Αν θέλουμε, για παράδειγμα, να κινήσουμε έναν κινητήρα με τη μισή της μέγιστης ταχύτητάς του θα επιλέξουμε την τιμή 126. Το Duty Cycle εκφράζει τη χρονική διάρκεια του παλμού, ενόσω είναι σε κατάσταση ON, σε ποσοστό επί τοις εκατό (%) επί της περιόδου ή σε μονάδες χρόνου (sec). Τα παραπάνω σημαίνουν ότι όσο αυξάνονται οι παλμοί (pulse width), τόσο αυξάνεται η ταχύτητα (Εικόνα 3.14).



Εικόνα 3.14: Παλμοί PWM

(Κενό φύλλο)

ΚΕΦΑΛΑΙΟ 4

Σε αυτό το κεφάλαιο γίνεται η συνδεσμολογία του κυκλώματος και στη συνέχεια ακολουθεί η σχεδίασή του με το πρόγραμμα Fritzing.

4.1 Συνδεσμολογία κυκλώματος

Ακολουθεί αναλυτικά όλη η διαδικασία συνδεσμολογίας του κυκλώματος με το Raspberry Pi 3:

- Γειώνουμε τα pins 4, 5, 12, 13 του L293D συνδέοντάς τα (μέσω του breadboard) με το ground (pin 6) του GPIO και τον αρνητικό πόλο των μπαταριών.

- Συνδέουμε τον ακροδέκτη 8 (Vs) του L293D (που είναι η τροφοδοσία για τους κινητήρες) με το I του διακόπτη και το θετικό πόλο των μπαταριών με το O.

- Για τον κινητήρα 1 (δεξιός κινητήρας) συνδέουμε:

  Το ένα καλώδιο στο Pin 11 του L293D (Motor1A)
  Το άλλο καλώδιο στο Pin 14 του L293D (Motor1B)

- Προσθέτουμε τρία καλώδια για τον κινητήρα 1 από τις ακίδες GPIO στο L293D (2 για τη φορά περιστροφής και 1 για την παροχή Enable σήματος στο L293D):

  Το GPIO 17 (Pin 11) στο Pin 15 του L293D
  Το GPIO 27 (Pin 13) στο Pin 10 του L293D
  Το GPIO 22 (Pin 15) στο Pin 9 του L293D (PWM – Motor1E)

- Για τον κινητήρα 2 (αριστερός κινητήρας) συνδέουμε:

Το ένα καλώδιο στο Pin 3 του L293D (Motor2A)
Το άλλο καλώδιο στο Pin 6 του L293D (Motor2B)

- Προσθέτουμε τρία καλώδια για τον κινητήρα 2 από τις ακίδες GPIO στο L293D:

Το GPIO 3 (Pin 5) στο Pin 7 του L293D

Το GPIO 2 (Pin 3) στο Pin 2 του L293D

Το GPIO 4 (Pin 7) στο Pin 1 του L293D (PWM – Motor2E)

- Συνδέουμε τα 5V του GPIO (Pin 2) στο Pin 16 του L293D (που είναι για την τροφοδοσία του L293D).

- Συνδέουμε τα 4 LED ως εξής:

  - Πράσινο: Το αρνητικό άκρο της αντίστασης με το GPIO (Pin 8), το θετικό άκρο του με τη μία άκρη της αντίστασης, και το αρνητικό με το 14 (ground).

  - Μπλε: Το αρνητικό άκρο της αντίστασης με το GPIO (Pin 12), το θετικό άκρο του με τη μία άκρη της αντίστασης, και το αρνητικό με το 14 (ground).

  - Πορτοκαλί: Το αρνητικό άκρο της αντίστασης με το GPIO (Pin 16), το θετικό άκρο του με τη μία άκρη της αντίστασης, και το αρνητικό με το 14 (ground).

  - Κόκκινο: Το αρνητικό άκρο της αντίστασης με το GPIO (Pin 18), το θετικό άκρο του με τη μία άκρη της αντίστασης, και το αρνητικό με το 14 (ground).

- Συνδέουμε την κάμερα σε μία θύρα από τις θύρες USB.

- Τέλος, για την εκκίνηση της εφαρμογής μας χρειάζεται να συνδέσουμε το καλώδιο τροφοδοσίας του φορτιστή με την αντίστοιχη υποδοχή στο RPi 3.

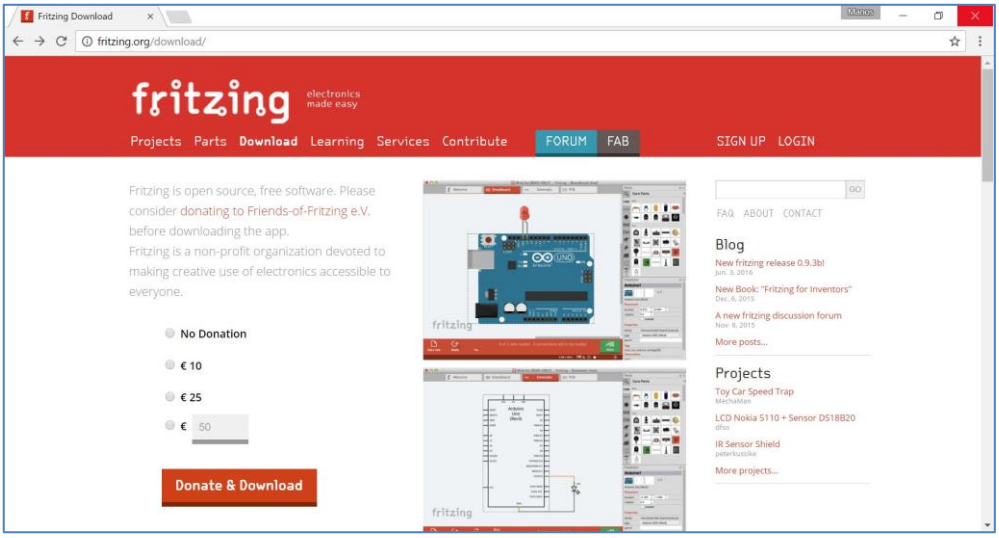
- Τα παραπάνω συνοψίζονται εδώ:

  - L293D Pin 1 –> RPi Pin 7
  - L293D Pin 2 –> RPi Pin 3
  - L293D Pin 3 –> Motor2A (αριστερός κινητήρας)
  - L293D Pin 4 –> RPi Pin 6 (Ground)
  - L293D Pin 5 –> RPi Pin 6 (Ground)
  - L293D Pin 6 –> Motor2B (αριστερός κινητήρας)
  - L293D Pin 7 –> RPi Pin 5
  - L293D Pin 8 –> I του διακόπτη
  - L293D Pin 9 –> RPi Pin 15
  - L293D Pin 10 –> RPi Pin 13
  - L293D Pin 11 –> Motor1A (δεξιός κινητήρας)
  - L293D Pin 12 –> RPi Pin 6 (Ground)
  - L293D Pin 13 –> RPi Pin 6 (Ground)
  - L293D Pin 14 –> Motor1B (δεξιός κινητήρας)
  - L293D Pin 15 –> RPi Pin 11
  - L293D Pin 16 –> RPi Pin 2 (+5V)
  - Battery +V –> O του διακόπτη
  - + πράσινου LED –> RPi Pin 8
  - + μπλε LED –> RPi Pin 12
  - + πορτοκαλί LED –> RPi Pin 16
  - + κόκκινου LED –> RPi Pin 18
  - - πράσινου LED –> RPi Pin 14 (Ground)
  - - μπλε LED –> RPi Pin 14 (Ground)
  - - πορτοκαλί LED –> RPi Pin 14 (Ground)
  - - κόκκινου LED –> RPi Pin 14 (Ground)

4.2 Σχεδίαση κυκλώματος με το πρόγραμμα Fritzing

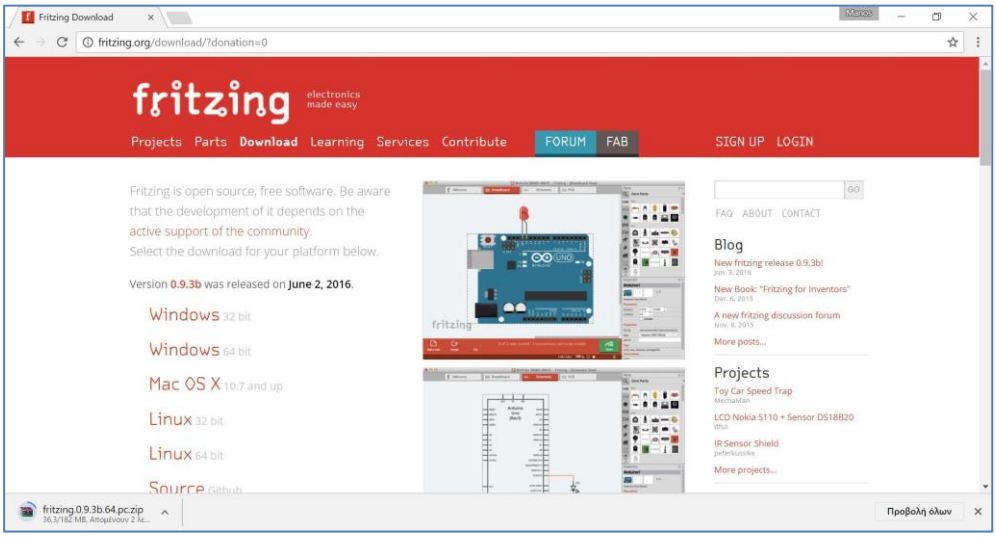
Ένας εύκολος τρόπος για την εικονική σχεδίαση του κυκλώματός μας είναι η κατασκευή του με το πρόγραμμα ανοιχτού κώδικα “**Fritzing**”.

- Αρχικά, για να κάνουμε λήψη του προγράμματος επισκεπτόμαστε τον αντίστοιχο ιστότοπο (http://fritzing.org/download/) και επιλέγοντας (εάν το επιθυμούμε) ένα ποσό δωρεάς (Εικόνα 4.1)…



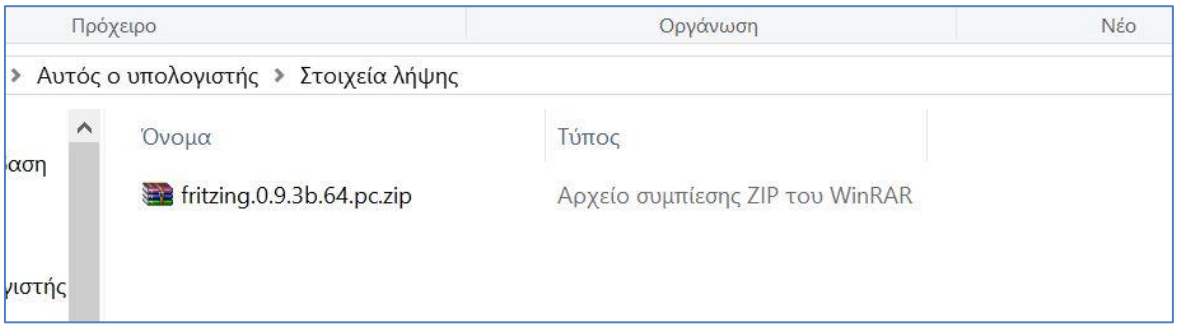
Εικόνα 4.1: Ιστότοπος Fritzing

- …επιλέγουμε, στη συνέχεια, το λειτουργικό μας σύστημα και πατώντας “**Download**” ξεκινά η λήψη του Fritzing (Εικόνα 4.2).



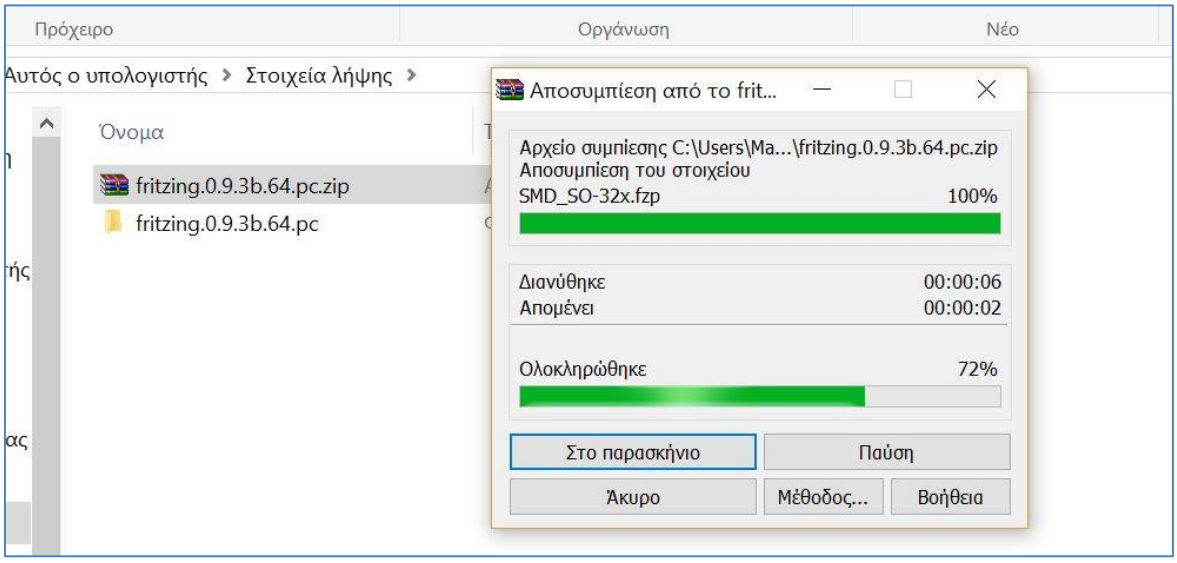
Εικόνα 4.2: Λήψη Fritzing

- Με την ολοκλήρωση της λήψης, το πρόγραμμα βρίσκεται πλέον σε μορφή **.zip** (Εικόνα 4.3)**,…**



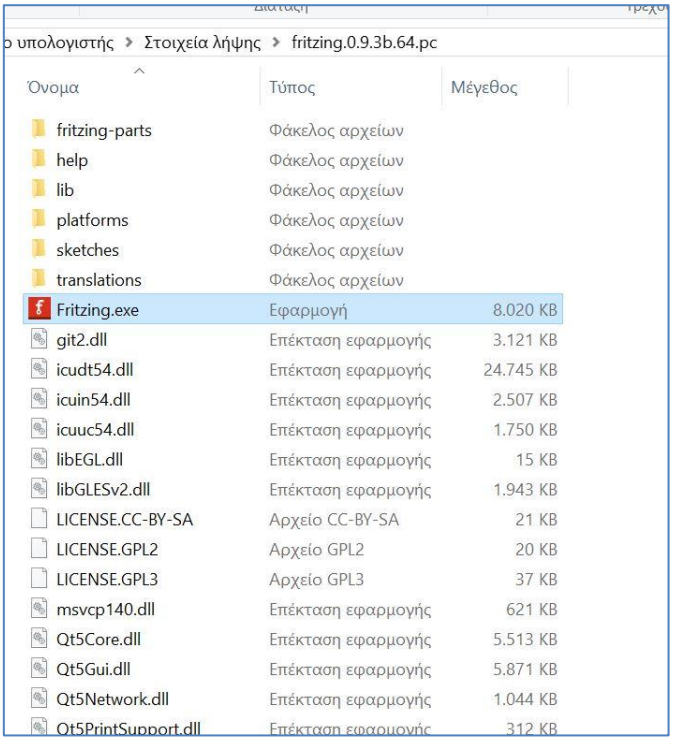
Εικόνα 4.3: Αρχείο .zip του Fritzing

- …όπου με αποσυμπίεση έχουμε τον φάκελο με τα αναγκαία αρχεία του προγράμματος (Εικόνα 4.4)…



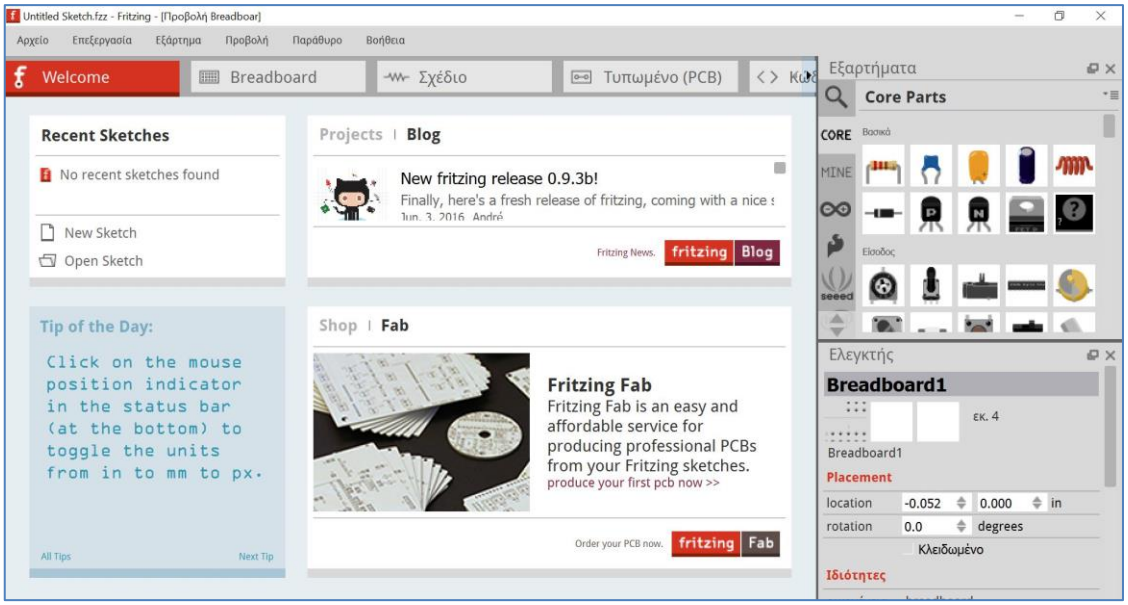
Εικόνα 4.4: Αποσυμπίεση του αρχείου .zip

- Επιλέγοντας με διπλό κλικ, μέσα στα περιεχόμενα του φακέλου, την εφαρμογή “**Fritzing.exe**” (Εικόνα 4.5)…



Εικόνα 4.5: Εκτέλεση του Fritzing

- …αποκτάμε πρόσβαση στο περιβάλλον του Fritzing (Εικόνα 4.6).



Εικόνα 4.6: Περιβάλλον του Fritzing

Ακολουθεί παρακάτω η αναλυτική διαδικασία σχεδίασης του κυκλώματος:

- Επιλέγουμε (δεξιά του Welcome) τη σχεδίαση σε breadboard.

- Σβήνουμε το breadboard που θα εμφανιστεί, και το αντικαθιστούμε με το “mini breadboard” από την καρτέλα “Εξαρτήματα”.

- Από την ίδια καρτέλα αναζητούμε το L293D και το τοποθετούμε πάνω στο breadboard.

- Κάνουμε αναζήτηση το εικονίδιο του RPi 3 και το σέρνουμε στο σχέδιό μας.

- Με δεξί κλικ στο breadboard επιλέγουμε να το περιστρέψουμε κατά 135^{o} δεξιόστροφα, ώστε το L293D να είναι με την εγκοπή του προς τα πάνω, ενώ στο RPi 3 επιλέγουμε 45^{o} αριστερόστροφα.

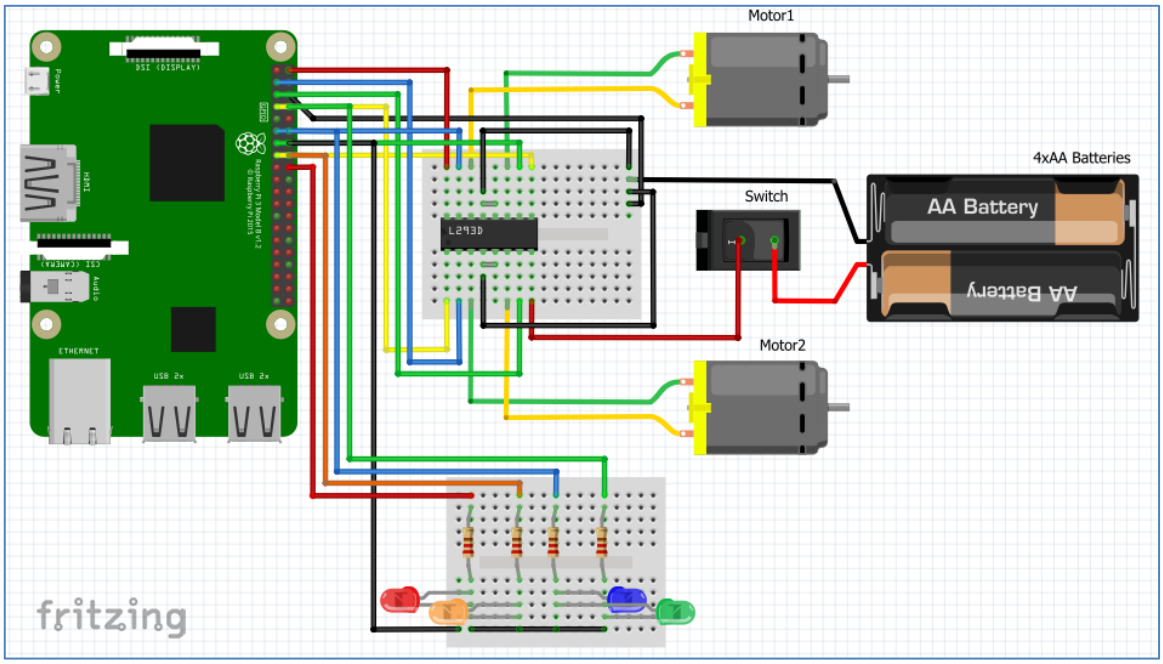
- Δημιουργούμε ένα διπλότυπο του mini breadboard με δεξί κλικ πάνω του και προσθέτουμε σε αυτό τα 4 LED και τις 4 αντιστάσεις γράφοντας “LED” και “resistor” αντίστοιχα, διαμορφώνοντας τα χρώματα των LED και τις τιμές των αντιστάσεων κατάλληλα.

- Τέλος, ψάχνουμε για τις μπαταρίες γράφοντας “battery”, για τους 2 κινητήρες γράφοντας “DC motor” και για τον διακόπτη γράφοντας “SWITCH-SPST-2” σέρνοντάς τα και αυτά πάνω στο σχέδιο.

- Δυστυχώς, δεν υπήρχε δυνατότητα προσθήκης της κάμερας σε υποδοχή USB με το πρόγραμμα Fritzing.

- Για τη σύνδεση των εξαρτημάτων ισχύουν όσα ειπώθηκαν στην ενότητα “4.1”. Η σύνδεση γίνεται απλά επιλέγοντας το άκρο κάποιου εξαρτήματος ή ακροδέκτη και τραβώντας από κει καλώδιο με το ποντίκι μέχρι το άλλο άκρο, του οποίου επιθυμούμε τη σύνδεση.

- Με βάση όλα αυτά, προκύπτει το αποτέλεσμα που φαίνεται στην “Εικόνα 4.7”:



Εικόνα 4.7: Συνδεσμολογία κυκλώματος

- Τέλος, αποθηκεύουμε το αρχείο σε μορφή **.fzz** (πατώντας Ctrl+S) δίνοντάς του και ένα όνομα.

ΚΕΦΑΛΑΙΟ 5

Σε αυτό το κεφάλαιο αρχικά εξηγείται η επιλογή της γλώσσας Python για το πρόγραμμά μας. Ακολουθούν λίγα λόγια για τη βιβλιοθήκη OpenCV, καθώς και η εγκατάστασή της στο RPi 3. Ακόμη, γίνεται αναφορά στη βιβλιοθήκη NumPy, η οποία χρησιμοποιείται από την Python και επιλύει μαθηματικές πράξεις.

5.1 Γλώσσα προγραμματισμού του προγράμματος

Επιλέξαμε Python διότι είναι μια ανοιχτού λογισμικού και υψηλού επιπέδου γλώσσα προγραμματισμού με πολλές δυνατότητες για τον προγραμματιστή. Έχει μεγάλη ευκολία στη χρήση, αφού το συντακτικό και η δομή της είναι πιο απλά σε σχέση με άλλες γλώσσες, όπως η C++ και η Java. Για παράδειγμα, δεν απαιτούνται αγκύλες στις συναρτήσεις, ούτε ερωτηματικά στο τέλος των εντολών. Επιπλέον, περιέχει πολλές βιβλιοθήκες και έτσι τα πράγματα είναι αρκετές φορές πιο εύκολα για το χρήστη.

5.2 Τι είναι η OpenCV

Η OpenCV είναι μια βιβλιοθήκη συναρτήσεων ανοιχτού λογισμικού και αναπτύχθηκε από την Intel το 1999 και αφορά στην επεξεργασία εικόνας. Σε πρώτη φάση, οι βιβλιοθήκες OpenCV άρχισαν να αναπτύσσονται με τη γλώσσα προγραμματισμού C και έπειτα οι περισσότεροι αλγόριθμοι αναπτύχθηκαν με τη γλώσσα C++. Η OpenCV λειτουργεί σε Linux, Windows και Mac OS και υποστηρίζει γλώσσες προγραμματισμού όπως Python, C#, Visual Basic.NET, Rubby.

Η βιβλιοθήκη περιέχει πάνω από 500 συναρτήσεις, οι οποίες εκτείνονται σε πολλούς τομείς της όρασης. Ένας από τους στόχους της βιβλιοθήκης OpenCV είναι να προσφέρει στο χρήστη την ανάπτυξη επικοινωνίας του ανθρώπου με τον υπολογιστή. Επιπλέον, είναι χρήσιμη για την ανίχνευση και αναγνώριση αντικειμένων και προσώπων, καθώς και για την κατανόηση και παρακολούθηση κίνησης.

Εικόνα 5.1: Λογότυπο OpenCV

Στο πρόγραμμά μας γίνεται χρήση αυτής της βιβλιοθήκης για την ενεργοποίηση της κάμερας και τη λήψη εικόνας, αλλά και για την επεξεργασία των εικόνων αυτών. Επίσης, χρησιμοποιείται για τη δημιουργία και την εκπαίδευση του ανιχνευτή των αντικειμένων που ψάχνουμε.

5.3 Εγκατάσταση της OpenCV στο Raspberry Pi 3

Ακολουθούν τα βήματα εγκατάστασης του OpenCV στο Raspberry Pi 3:

- Αρχικά, έγιναν οι ανάλογες αναβαθμίσεις του λειτουργικού συστήματος με τις εντολές:

```
sudo apt-get update
sudo apt-get upgrade
```

- Στη συνέχεια, εγκαταστάθηκαν οι απαραίτητες βιβλιοθήκες για τη χρήση της OpenCV.

```
sudo apt-get install python-numpy python-scipy python-matplotlib
sudo apt-get install build-essential cmake pkg-config
sudo apt-get install default-jdk ant
sudo apt-get install libgtkglext1-dev
sudo apt-get install v4l-utils
sudo apt-get install libjpeg8 \
libjpeg8-dev \
libjpeg8-dbg \
libjpeg-progs \
```

libavcodec-dev \
libavformat-dev \
libgstreamer0.10-0-dbg \
libgstreamer0.10-0 \
libgstreamer0.10-dev \
libxine2-dev \
libunicap2 \
libunicap2-dev \
swig \
libv4l-0 \
libv4l-dev \
python-numpy \
libpython2.7 \
python-dev \
python2.7-dev \
libgtk2.0-dev \
libjasper-dev \
libpng12-dev \
libswscale-dev

- Από την ακόλουθη διεύθυνση κάνουμε λήψη της βιβλιοθήκης OpenCV

Wget http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/3.0.0/opencv-3.0.0.zip

- Έπειτα, κάνουμε αποσυμπίεση του φακέλου

unzip opencv-3.0.0.zip

- Μεταφερόμαστε στο φάκελο opencv-3.0.0

cd opencv-3.0.0

- Δημιουργούμε φάκελο με όνομα build, ώστε να κάνουμε μέσα σ' αυτόν εγκατάσταση τα πιο κάτω προγράμματα

mkdir build

- Μεταφερόμαστε στο φάκελο build

```
cd build
```

- Κάνουμε εγκατάσταση τα παρακάτω:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D INSTALL_C_EXAMPLES=ON \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D BUILD_EXAMPLES=ON \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D WITH_V4L=ON ..
sudo make
sudo make install
```

- Εκτελούμε άμεσο τερματισμό του Raspberry Pi 3, ώστε να ολοκληρωθεί η εγκατάσταση της βιβλιοθήκης OpenCV

```
sudo shutdown -r now
```

5.4 NumPy

Το NumPy είναι μια βιβλιοθήκη για τη γλώσσα Python που (μεταξύ άλλων) επιλύει μαθηματικά προβλήματα παρέχοντας υποστήριξη για μεγάλους και πολυδιάστατους πίνακες. Χρησιμοποιώντας το NumPy, μπορούμε να εκφράσουμε τις εικόνες ως πολυδιάστατους πίνακες. Πολλές βιβλιοθήκες επεξεργασίας εικόνας και μηχανικής μάθησης χρησιμοποιούν αναπαραστάσεις πινάκων NumPy. Επιπλέον, χρησιμοποιώντας τις ενσωματωμένες μαθηματικές λειτουργίες υψηλού επιπέδου του NumPy, μπορούμε γρήγορα να εκτελέσουμε αριθμητική ανάλυση σε μια εικόνα.

ΚΕΦΑΛΑΙΟ 6

Σε αυτό το κεφάλαιο παρουσιάζεται η διαδικασία της δημιουργίας ενός ανιχνευτή για την εύρεση των σημάτων. Η ανίχνευση βασίζεται σε μια μέθοδο των Paul Viola και Michael Jones, η οποία αναπτύχθηκε από τον Rainer Lienhart και κάνει χρήση ταξινομητών καταρράκτη με χαρακτηριστικά τύπου Haar.

6.1 Ανίχνευση αντικειμένων

Ανίχνευση είναι η διαδικασία αναζήτησης στοιχείων που υπάρχουν στο περιβάλλον και έχει στόχο την επιβεβαίωση ότι υπάρχει κάτι σε αυτό.

Ο άνθρωπος είναι εξοικειωμένος στην αναγνώριση αντικειμένων έστω και αν δεν γνωρίζει όλα τα αντικείμενα που υπάρχουν αρκεί να είναι της ίδιας κατηγορίας. Για παράδειγμα, γνωρίζει την πέτρα σαν αντικείμενο όμως δεν έχει δει όλες τις πέτρες που υπάρχουν. Παρόλα αυτά είναι ικανός να αναγνωρίσει μια πέτρα που δεν έχει ξαναδεί ξέροντας απλά τα γενικά χαρακτηριστικά της.

Ένας υπολογιστής από την άλλη χρειάζεται να κάνει ανάλυση του κάθε αντικειμένου μιας και δεν είναι σε θέση να το αναγνωρίσει κατευθείαν. Αυτή η ανάλυση βέβαια απαιτεί αρκετό χρόνο ειδικά αν πρόκειται για πληθώρα αντικειμένων. Όμως, είναι πολύ χρήσιμη, διότι σε περιπτώσεις όπου δεν είναι εφικτή η βοήθεια του ανθρώπου, ένας υπολογιστής συμβάλλει σημαντικά στη δουλειά αυτή. Για παράδειγμα, σε μεγάλους δρόμους όπου θέλουμε να μετρήσουμε την ταχύτητα των αυτοκινήτων, αυτό είναι αρκετά πιο εύκολο (και με μεγαλύτερη ακρίβεια) να επιτευχθεί μέσω ραντάρ παρά μέσω ενός υπαλλήλου.

Κατά την ανίχνευση αντικειμένων ένας υπολογιστής παίρνει ως είσοδο μια εικόνα και βγάζει ως έξοδο τις περιοχές της εικόνας στις οποίες εμφανίζεται κάποιο συγκεκριμένο αντικείμενο. Το αποτέλεσμα της διαδικασίας είναι να βρίσκει τη θέση και τη διάστασή του αντικειμένου καθώς και την κατηγορία που ανήκει (πχ. κατά την ανίχνευση ματιών σε ανθρώπους, η κατηγορία είναι τα μάτια).

Η ανίχνευση αντικειμένων σε εικόνες διαφοροποιείται από την ταξινόμηση εικόνων. Στην ταξινόμηση εικόνων, η έξοδος του υπολογιστικού συστήματος περιλαμβάνει μόνο την πληροφορία αν η εικόνα απεικονίζει ή όχι ένα συγκεκριμένο αντικείμενο και όχι τη θέση και τις διαστάσεις του αντικειμένου. Αντίθετα, στην

ανίχνευση αντικειμένων, χρειάζεται να ελεγχθεί αν εμφανίζεται κάποιο αντικείμενο σε οποιαδήποτε θέση και με οποιοδήποτε μέγεθος μέσα σε μια εικόνα. Βλέπουμε, λοιπόν, ότι η ανίχνευση αντικειμένων είναι μια διαδικασία αρκετά σύνθετη και πιο πολύπλοκη από τη διαδικασία της ταξινόμησης εικόνων [31].

6.2 Χαρακτηριστικά Haar

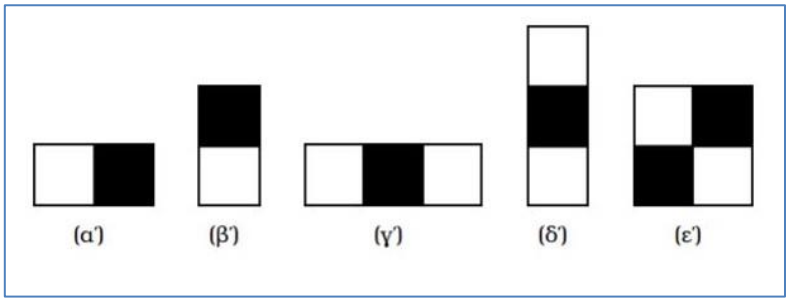
Μια εικόνα αποτελείται από ένα σύνολο εικονοστοιχείων, που το καθένα παίρνει μια τιμή που αντιστοιχεί στη φωτεινότητα ή στην τιμή του χρώματος στη συγκεκριμένη θέση της εικόνας. Η τιμή της φωτεινότητας κάθε εικονοστοιχείων επηρεάζεται έντονα από τις αλλαγές στο φωτισμό της σκηνής. Αυτή η αλλαγή όμως, επηρεάζει αρκετά ομοιόμορφα όλα τα εικονοστοιχεία της εικόνας. Έτσι, η τιμή μιας συνάρτησης που εξετάζει τη μέση διαφορά ανάμεσα σε δύο ή τρεις περιοχές της ίδιας εικόνας, θα παραμένει σε μεγάλο βαθμό ανεπηρέαστη. Χρησιμοποιώντας, λοιπόν, τις συναρτήσεις Haar, η διαδικασία της ανίχνευσης αντικειμένων δε θα επηρεάζεται από τις διαφορές στη φωτεινότητα από εικόνα σε εικόνα. Οι συναρτήσεις Haar υπολογίζουν τη διαφορά ανάμεσα στους μέσους όρους των τιμών των εικονοστοιχείων δύο (ή τριών) περιοχών [31].

Επομένως καταλήγουμε ότι για να αναπαραστήσουμε τις πληροφορίες που περιέχονται στην εικόνα και να γίνει η ανίχνευση των αντικειμένων που ψάχνουμε γίνεται χρήση των χαρακτηριστικών τύπου Haar έχοντας αρκετά καλά αποτελέσματα ανίχνευσης.

Αν ο ανιχνευτής σάρωνε κάθε ένα pixel ξεχωριστό, τότε το υπολογιστικό κόστος θα ήταν πολύ μεγάλο. Αντίθετα, η χρησιμοποίηση ενός συνόλου χαρακτηριστικών (feature set) των τιμών των pixels κάνει τον αλγόριθμο πολύ ταχύτερο. Έτσι, το σύνολο χαρακτηριστικών που χρησιμοποιείται χωρίζει την εικόνα σε ορθογώνιες περιοχές και υπολογίζει το άθροισμα των pixels των τετραγώνων μέσω της ολοκληρωτικής εικόνας (integral image). Μάλιστα αυτή η τακτική είναι ιδιαίτερα συμφέρουσα αν συνυπολογιστεί το γεγονός ότι τα χαρακτηριστικά περιέχουν μεγαλύτερη πληροφορία για την περιοχή σε σύγκριση με τα μεμονωμένα pixels.

Τα χαρακτηριστικά που πρότειναν οι Viola και Jones είναι ιδιαίτερα απλά ορθογώνια σχήματα όπως φαίνεται και στην Εικόνα 6.1. Η τιμή του κάθε χαρακτηριστικού ορίζεται ως η διαφορά του αθροίσματος των pixels της λευκής

περιοχής από αυτά της μαύρης περιοχής. Φυσικά η θέση και το μέγεθος των χαρακτηριστικών κάθε φορά διαφέρουν και εξαρτώνται από το παράθυρο που σαρώνει την εικόνα εισόδου.



Εικόνα 6.1: Viola-Jones Χαρακτηριστικά που μοιάζουν με Haar

Κάθε χαρακτηριστικό συνοδεύεται από κάποιες παραμέτρους. Για παράδειγμα, το χαρακτηριστικό της Εικόνας 6.1(γ') έχει τέσσερις παραμέτρους: τη θέση μέσα στο παράθυρο σάρωσης, το μέγεθος της άσπρης (ή θετικής) περιοχής, το μέγεθος της μαύρης (ή αρνητικής) περιοχής και το ύψος [32].

Η ανάλυση που χρησιμοποιούμε για τον ανιχνευτή είναι 22x22 pixels. Με τον υπολογισμό των χαρακτηριστικών haar για όλες τις περιπτώσεις της Εικόνας 6.1, προκύπτουν 115.000 περίπου διαφορετικά χαρακτηριστικά (Εικόνα 6.2), τα οποία έχουν πολύ καλά αποτελέσματα με ελάχιστο κόστος υπολογισμού.

Επομένως, το πλήθος των ορθογώνιων χαρακτηριστικών που δημιουργούνται είναι υπερβολικά μεγάλο, ώστε να απαιτείται η επιλογή των πλέον αποτελεσματικών χαρακτηριστικών που θα χρησιμοποιηθούν στον τελικό ταξινομητή [33].

Για τον υπολογισμό του πλήθους των Haar χαρακτηριστικών σε κάποιο παράθυρο εικόνας πλάτους W και ύψους H, ακολουθούμε την παρακάτω διαδικασία. Έστω ότι w και h είναι το πλάτος και ύψος του ορθογωνίου της συνάρτησης Haar που εξετάζουμε. Το μέγεθος του ορθογωνίου θα αυξάνεται κατά ένα σε κάθε βήμα. Άρα, οι μέγιστοι συντελεστές μεγέθυνσης των ορθογωνίων σε πλάτος και ύψος θα είναι X=W/w και Y=H/h, αντίστοιχα. Το πλήθος των χαρακτηριστικών που προκύπτουν από την εφαρμογή ενός κατακόρυφου Haar χαρακτηριστικού στο παράθυρο εικόνας, είναι [31]:

$$XY \cdot \left(W + 1 - w\frac{X+1}{2}\right) \cdot \left(H + 1 - h\frac{Y+1}{2}\right)$$

Τύπος Χαρακτηριστικού	w	h	X	Ψ	Χαρακτηριστικά
	2	1	11	22	30.613
	1	2	22	11	30.613
	3	1	7	22	19.481
	1	3	22	7	19.481
	2	2	11	11	14.641

Εικόνα 6.2: Πλήθος χαρακτηριστικών τύπου Haar σε παράθυρο 22x22

6.3 Δημιουργία Haar ανιχνευτή

Για την ανίχνευση των σημάτων θα χρησιμοποιηθεί η βιβλιοθήκη OpenCV, η οποία περιλαμβάνει την εφαρμογή traincascade. Αυτή η εφαρμογή δημιουργεί ένα αρχείο .xml για να έχουμε πρόσβαση στις πληροφορίες των εικόνων, στο οποίο τα στοιχεία βρίσκονται σε ετικέτες βοηθώντας με αυτό τον τρόπο την εύκολη μεταφορά και αναγνώριση των δεδομένων στα διάφορα συστήματα.

Εικόνα 6.3: Σήματα left, stop και notturn

Για να δημιουργήσουμε τον Haar ανιχνευτή θα χρειαστούμε τυχαίες αρνητικές εικόνες, οι οποίες χρησιμεύουν ως εικόνες παρασκηνίου. Πρέπει να είναι όσο το δυνατόν περισσότερες (από 1500 και πάνω) για να έχουμε πιο εύκολη ανίχνευση. Αν σε αυτές τις αρνητικές εικόνες προσθέσουμε σε καθεμία το αντικείμενο προς ανίχνευση (δηλάδή ένα από τα σήματα της Εικόνας 6.3) θα προκύψουν οι θετικές εικόνες. Οι θετικές εικόνες δημιουργούνται μέσω του προγράμματος creat_samples (πιο αναλυτικά στην ενότητα 7.1.2).

Στην περίπτωση που θέλουμε να βρούμε μόνο ένα αντικείμενο, η ανίχνευση είναι μία σχετικά απλή υπόθεση. Όμως, αν έχουμε περισσότερα αντικείμενα να ανιχνεύσουμε τότε, αν αυτά τα αντικείμενα δεν είναι τόσο διαφορετικά μεταξύ τους (πχ. βέλος για αριστερη και δεξιά στροφή, τα οποία μοιάζουν), υπάρχει μεγάλη πιθανότητα να υπάρξουν λανθασμένα αποτελέσματα.

Όλες οι εικόνες (θετικές και αρνητικές) πρέπει να έχουν μικρό μέγεθος (πχ. 100x100) για γρήγορη επεξεργασία και το αντικείμενο που ψάχνουμε να έχει αρκετά μικρότερο μέγεθος από αυτές (πχ. 50x50), ώστε να υπάρχει αρκετό φόντο.

Οι θετικές εικόνες χρησιμεύουν για τη δημιουργία την θετικών δειγμάτων και πρέπει να είναι αρκετές σε πλήθος.

Η εύρεση τόσων θετικών παραδειγμάτων, ειδικά όταν θέλουμε να εκπαιδεύσουμε έναν ταξινομητή για αντικείμενα όχι τόσο κοινά όσο οι άνθρωποι και τα πρόσωπα, δεν είναι πάντα εφικτή. Το χωρικό μέγεθος των θετικών παραδειγμάτων θα πρέπει να είναι αρκετά μικρό ώστε να μπορούν να γίνουν σε λογικό χρόνο οι υπολογισμοί που χρειάζονται. Αντίθετα, όσο μεγαλύτερο είναι το μέγεθός τους, τόσο καλύτερη ανάλυση του αντικειμένου θα μπορέσουμε να κάνουμε. Τα θετικά παραδείγματα προκύπτουν από εικόνες οι οποίες να περιέχουν το αντικείμενο για το οποίο θέλουμε να εκπαιδεύσουμε τον ανιχνευτή μας. Σε αυτές τις εικόνες το αντικείμενο θα πρέπει να φαίνεται ολόκληρο, χωρίς επικαλύψεις από άλλα αντικείμενα. Η μέθοδος που χρησιμοποιούμε ψάχνει για γεωμετρικά σχήματα που χαρακτηρίζουν ένα αντικείμενο σε όλα τα θετικά παραδείγματα, σε συγκεκριμένες θέσεις [31].

Συγκεκριμένα, μέσω της εφαρμογής create_samples συλλέγονται τα θετικά δείγματα των εικόνων από κάθε θετική εικόνα επιλέγοντας τυχαία ένα κομμάτι του με ίδιες διαστασεις για κάθε μία από αυτές. Τα αρνητικά δείγματα έχουν διαστάσεις ίδιες με αυτές των θετικών δειγμάτων και δεν περιέχουν το αντικείμενο προς ανίχνευση. Μετά τη δημιουργία των θετικών και αρνητικών δειγμάτων φτάνουμε στο σημείο της εκπαίδευσης του ανιχνευτή, η διαδικασία του οποίου εξηγείται σε επόμενη ενότητα.

6.4 Αλγόριθμος AdaBoost

Για την ανίχνευση ενός αντικειμένου πρέπει πρώτα να ελέγξουμε αν αυτό το αντικείμενο βρίσκεται στην κατηγορία που θέλουμε. Ο έλεγχος αυτός θα πραγματοποιηθεί με τον αλγόριθμο AdaBoost (Εικόνα 6.4), ο οποίος θα είναι υπεύθυνος και για την εκπαίδευση του ταξινομητή από τις πληροφορίες των σημάτων.

Ο αλγόριθμος εκμάθησης AdaBoost ανήκει στην κατηγορία των αλγορίθμων ενδυνάμωσης (boosting) και χρησιμοποιείται για να αυξήσει την απόδοση ενός οποιουδήποτε απλού αλγορίθμου ταξινόμησης. Ο απλός αλγόριθμος ταξινόμησης λέγεται και ασθενής αλγόριθμος ταξινόμησης, καθώς ακόμα και η καλύτερη συνάρτηση ταξινόμησης που μπορεί να προκύψει από αυτόν, δεν αναμένεται να ταξινομεί καλά τα δεδομένα. Συγκεκριμένα, αρκεί η συνάρτηση ταξινόμησης να έχει απόδοση ελαφρά καλύτερη από την τυχαία ταξινόμηση (50%). Για να αυξήσει, λοιπόν, την απόδοση ενός ασθενούς αλγορίθμου ταξινόμησης, ο AdaBoost συνδυάζει μια συλλογή ασθενών συναρτήσεων ταξινόμησης χρησιμοποιώντας άπληστο αλγόριθμο, ώστε να σχηματίσει από αυτούς έναν ισχυρότερο ταξινομητή. Η βελτίωση του ασθενούς αλγορίθμου ταξινόμησης πραγματοποιείται, καλώντας τον αλγόριθμο να επιλύσει μια αλληλουχία προβλημάτων ταξινόμησης. Αρχικά, όλα τα παραδείγματα (θετικά και αρνητικά) παίρνουν μια τιμή βάρους, η οποία είναι ίδια για όλα. Δίνονται στον αλγόριθμο τα παραδείγματα και πραγματοποιείται ο πρώτος κύκλος εκμάθησης, όπου ο αλγόριθμος ταξινομεί όλα τα παραδείγματα με κάθε διαθέσιμη συνάρτηση ταξινόμησης. Έπειτα, οι συναρτήσεις ταξινόμησης διατάσσονται σύμφωνα με τα αποτελέσματά τους, λαμβάνοντας υπόψη το βάρος κάθε παραδείγματος. Επιλέγεται ένας μικρός αριθμός συναρτήσεων ταξινόμησης, από αυτές με τα καλύτερα αποτελέσματα, που αποτελούν τον πρώτο ασθενή ταξινομητή. Ο πρώτος κύκλος εκμάθησης ολοκληρώνεται και τα βάρη των παραδειγμάτων ισοσταθμίζονται, δίνοντας μεγαλύτερο βάρος στα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον πρώτο ασθενή ταξινομητή. Έτσι, στον δεύτερο κύκλο εκμάθησης ο αλγόριθμος ταξινόμησης θα θεωρήσει πιο σημαντικά τα παραδείγματα που ταξινομήθηκαν λανθασμένα από τον προηγούμενο ταξινομητή. Τα βήματα επαναλαμβάνονται διαδοχικά, μέχρι να φτάσουμε στο επίπεδο του συνολικού λόγου λανθασμένης ταξινόμησης που επιθυμούμε. Τελικά,

ο ισχυρός ταξινομητής προκύπτει από τον συνδυασμό των ασθενών ταξινομητών που επιλέχθηκαν και ένα κατώφλι. Κατά την διαδικασία της ταξινόμησης ενός υποπαραθύρου εικόνας από τον ισχυρό ταξινομητή, εφαρμόζονται στο υποπαράθυρο όλοι οι ασθενείς ταξινομητές. Τα αποτελέσματα των ασθενών ταξινομητών αθροίζονται, και αν το άθροισμα ξεπερνά το κατώφλι του ταξινομητή, το υπό εξέταση αντικείμενο ταξινομείται ως θετικό, αλλιώς ως αρνητικό.

Υπάρχουν τέσσερις εκδοχές του αλγόριθμου AdaBoost. Η εκδοχή του δικού μας αλγορίθμου είναι ο Ήπιος AdaBoost (Gentle AdaBoost – GAB), ο οποίος βασίζεται στον Πραγματικό AdaBoost, όπου η συνάρτηση ταξινόμησης κάθε ασθενή ταξινομητή παίρνει όλες τις πραγματικές τιμές στο διάστημα [0,1]. Ο GAB από την άλλη χρησιμοποιεί βήματα της μεθόδου Newton αντί για ακριβή υπολογισμό.

Στη μέθοδο ανίχνευσης αντικειμένων που χρησιμοποιούμε, κάθε ασθενής αλγόριθμος εκμάθησης περιορίζεται στο σύνολο των συναρτήσεων ταξινόμησης που αποτελούνται από ένα μόνο χαρακτηριστικό τύπου Haar. Προφανώς, από ένα μόνο χαρακτηριστικό δε μπορούμε να περιμένουμε ιδιαίτερα χαμηλό λόγο σφάλματος. Σε κάθε στάδιο του αλγορίθμου AdaBoost επιλέγεται το χαρακτηριστικό που διαχωρίζει καλύτερα τα θετικά από τα αρνητικά δείγματα. Για κάθε χαρακτηριστικό, ο ασθενής αλγόριθμος εκμάθησης προσδιορίζει ένα κατώφλι της τιμής του χαρακτηριστικού, που ελέγχοντάς το περιορίζονται οι λανθασμένες ταξινομήσεις από το συγκεκριμένο χαρακτηριστικό στις ελάχιστες δυνατές. Έπειτα, επιλέγεται ως ασθενής ταξινομητής το χαρακτηριστικό τύπου Haar, που, για το δεδομένο κατώφλι του, κάνει τη συνολικά καλύτερη ταξινόμηση. Ο AdaBoost συνεχίζει εκπαιδεύοντας όλους τους ασθενείς ταξινομητές, μέχρι το σημείο που ο ισχυρός συνολικός ταξινομητής επιτυγχάνει το επίπεδο ταξινόμησης που ζητάμε. Ο αλγόριθμος AdaBoost παρέχει αρκετά ισχυρές εγγυήσεις για την ορθότητά του. Έχει αποδειχθεί, ότι το σφάλμα ταξινόμησης του ισχυρού ταξινομητή που προκύπτει από την εφαρμογή του αλγορίθμου, τείνει προς το μηδέν εκθετικά ως προς τον αριθμό των κύκλων εκπαίδευσης. Επίσης, η όλη διαδικασία της εκμάθησης πραγματοποιείται με μεγάλη ταχύτητα. Ας θεωρήσουμε ότι έχουμε στη διάθεσή μας K χαρακτηριστικά τύπου Haar και εικόνες-παραδείγματα. Για να κατασκευαστεί ένας ισχυρός ταξινομητής από τον αλγόριθμο AdaBoost, που αποτελείται από M ασθενείς ταξινομητές, χρειάζονται O (MNK) βήματα, σε αντίθεση με άλλους αλγορίθμους που χρειάζονται O (MNKN) βήματα [31].

Συμπεραίνεται ότι η μέθοδος AdaBoost στοχεύει στην επίλυση των παρακάτω 3 θεμελιωδών προβλημάτων:

(1) εκμάθηση των πιο αποτελεσματικών χαρακτηριστικών από ένα μεγάλο σύνολο χαρακτηριστικών,

(2) κατασκευή αδύναμων ταξινομητών, καθένας από τους οποίους στηρίζεται σε ένα μόνο από τα δημιουργηθέντα χαρακτηριστικά, και

(3) συνδυασμός των αδύναμων ταξινομητών για την κατασκευή ενός ισχυρού ταξινομητή [33].

Αλγόριθμος AdaBoost

Input: Εικόνες εκπαίδευσης $(x_1, \ldots, x_n)$ και οι αντίστοιχες ετικέτες $(y_1, \ldots, y_n)$, όπου $y_i \in \{0, 1\}$ για αρνητικά και θετικά δείγματα αντίστοιχα. Ο αριθμός των αρνητικών δειγμάτων είναι m και αυτός των θετικών δειγμάτων $l = n - m$

1: Αρχικοποίηση των n βαρών ως $(2m)^{-1}$ για $y_i = 0$ και $(2l)^{-1}$ για $y_i = 1$

2: **for** $t = 1, \ldots, T$ **do**

3: Κανονικοποίηση των βαρών $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$

4: Επιλογή του καλύτερου αδύναμου ταξινομητή ως προς το σφάλμα με βάρη $\epsilon_t = \min_{f,p,\theta} \sum_i w_i \left| h(x_i, f, p, \theta) - y_i \right|$

5: Ορισμός της $h_t(x) = h(x, f_t, p_t, \theta_t)$, όπου f_t, p_t και θ_t οι τιμές των μεταβλητών που ελαχιστοποιούν το ϵ_t

6: Ανανέωση των βαρών $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ όπου $e_i = 0$ αν το δείγμα x_i ταξινομείται σωστά, $e_i = 1$ αν το δείγμα x_i ταξινομείται λανθασμένα και $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$

7: **end for**

8: Ο δυνατός ταξινομητής με $\alpha_t = \log \frac{1}{\beta_t}$ ορίζεται ως

$$h_x = \begin{cases} 1, & \text{αν } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, & \text{αλλιώς} \end{cases}$$

Εικόνα 6.4: Αλγόριθμος AdaBoost

Σύμφωνα με τους Viola & Jones κατά την εφαρμογή του AdaBoost:

- Σε κάθε πέρασμα t εκπαιδεύεται ένας νέος αδύναμος ταξινομητής ht που προστίθεται στο σύνολο με μεγαλύτερο συντελεστή αt όσο μικρότερο σφάλμα ταξινόμησης ε δίνει.
- Το βάρος κάθε δείγματος ενημερώνεται έτσι ώστε σε κάθε επόμενο πέρασμα τα δείγματα που ταξινομούνται σωστά να έχουν μικρότερη βαρύτητα κατά βt.

- Ο τελικός ισχυρός ταξινομητής αποτελείται από Τ αδύναμους ταξινομητές που αντιστοιχούν στα ισχυρότερα χαρακτηριστικά, με βάρη αντιστρόφως ανάλογα προς το σφάλμα ταξινόμησης.
- Όπως αποδεικνύεται από τους Freund & Schapire ο ισχυρός ταξινομητής μπορεί να πετύχει αυθαίρετα υψηλό ρυθμό ορθών ταξινομήσεων με αυθαίρετα χαμηλό ρυθμό εσφαλμένων ταξινομήσεων, αρκεί το πλήθος των αδύναμων ταξινομητών να ναι αρκετά μεγάλο [33].

6.5 Κατασκευή του Αδύναμου Ταξινομητή

Ο απλούστερος τύπος ενός αδύναμου ταξινομητή είναι η "ρίζα" ("stump") ενός δένδρου απόφασης (decision tree). Όταν το χαρακτηριστικό παίρνει πραγματικές τιμές, μπορεί να κατασκευαστεί μία ρίζα απόφασης, συγκρίνοντας απλά την τιμή του επιλεγμένου χαρακτηριστικού με μια συγκεκριμένη τιμή κατωφλίου. Έτσι ο αδύναμος αλγόριθμος μάθησης σχεδιάζεται ώστε να μπορεί να επιλέγει εκείνο το μοναδικό χαρακτηριστικό που διαχωρίζει καλύτερα τα θετικά από τα αρνητικά δείγματα. Για κάθε χαρακτηριστικό, ο αδύναμος ταξινομητής καθορίζει το ιδανικό κατώφλι λειτουργίας της ταξινόμησης, έτσι ώστε να ελαχιστοποιείται ο αριθμός δειγμάτων που ταξινομείται εσφαλμένα [33].

6.6 Ταξινόμηση με έναν Καταρράκτη Ταξινομητών

Για να αποφύγουν την σύγκλιση του AdaBoost που απαιτεί μεγάλο αριθμό δειγμάτων και καταλήγει σε μεγάλο πλήθος αδύναμων ταξινομητών που απαιτούν μεγάλο υπολογιστικό κόστος, οι Viola & Jones εισήγαγαν την έννοια του καταρράκτη ταξινομητών (cascade of classifiers). Με τη μέθοδο αυτή κατασκευάζεται μία διαδοχή ταξινομητών στη μορφή καταρράκτη (cascade) που επιτυγχάνει αυξημένη απόδοση στην ανίχνευση και ριζικά τον χρόνο υπολογισμού. Η ιδέα είναι ότι μπορούν να κατασκευαστούν μικροί και ωστόσο αποτελεσματικοί, συνδυασμένοι ταξινομητές που απορρίπτουν πολλά από τα αρνητικά, ενώ ανιχνεύουν σχεδόν όλα τα θετικά περιστατικά. Η μέθοδος του καταρράκτη ταξινομητών στηρίζεται στο γεγονός οτι σε μία οποιαδήποτε εικόνα η πλειονότητα των παραθύρων ανίχνευσης δεν περιλαμβάνει πρόσωπα. Έτσι πιο απλοποιημένοι

και λιγότερο χρονοβόροι ταξινομητές χρησιμοποιούνται για να απορρίψουν την πλειονότητα των παραθύρων ανίχνευσης ως αρνητικά, προτού χρησιμοποιηθούν οι πιο σύνθετοι και περισσότερο χρονοβόροι ταξινομητές που θα επεξεργαστούν τις πιο πολύπλοκες περιπτώσεις και θα επιτύχουν χαμηλά επίπεδα εσφαλμένων θετικών ανιχνεύσεων [33].

6.7 Σύνοψη του Συστήματος Ανίχνευσης Προσώπου

- Υπολογίζονται απλά βαθμωτά χαρακτηριστικά. Υπάρχει ένας μεγάλος αριθμός από υποψήφια χαρακτηριστικά τύπου Haar.

- Επιλέγεται ένα μικρό υποσύνολο από αυτά και οι αντίστοιχοι αδύναμοι ταξινομητές από δένδρα απόφασης εκπαιδεύονται χρησιμοποιώντας AdaBoost.

- Ο ισχυρός ταξινομητής κατασκευάζεται σαν ένας γραμμικός συνδυασμός των αδύναμων και είναι το αποτέλεσμα του αλγόριθμου AdaBoost .

- Ο τελικός ανιχνευτής δημιουργείται από έναν ή από μία διαδοχή, σε διάταξη καταρράκτη, ισχυρών ταξινομητών που χρησιμοποιούν διαρκώς περισσότερα χαρακτηριστικά.

- Η προεπεξεργασία προετοιμάζει την εικόνα που θα ανιχνευτεί.

- Η μετεπεξεργασία συγχωνεύει ή απορρίπτει τις ανιχνεύσεις [33].

(Κενό φύλλο)

ΚΕΦΑΛΑΙΟ 7

Σε αυτό το κεφάλαιο εξηγείται η διαδικασία της εκπαίδευσης του ταξινομητή, η οποία γίνεται μέσω εντολών, ώστε να προκύψει το τελικό .xml αρχείο με τα ταξινομημένα αποτελέσματα των haar χαρακτηριστικών.

7.1 Εκπαίδευση ταξινομητή

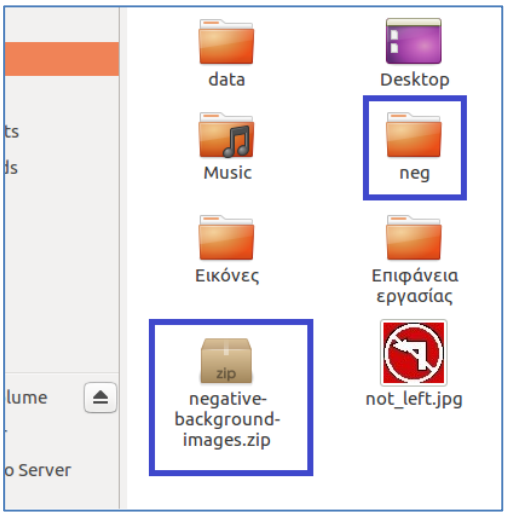
Για να έχουμε όσο το δυνατόν καλύτερη εύρεση του αντικειμένου χρησιμοποιούμε τουλάχιστον 15 στάδια (εμείς χρησιμοποιούμε 18). Αυτό όμως χρειάζεται έναν ισχυρό υπολογιστή καθώς και πολύ χρόνο για την εκπαίδευση του ταξινομητή. Ο χρόνος αυτός εξαρτάται επιπλέον και από το πλήθος των αρνητικών και θετικών δειγμάτων. Στην περίπτωσή μας, χρειάστηκαν περίπου 3 ώρες για 1950 αρνητικά και 1800 θετικά δείγματα, για το κάθε αντικείμενο ξεχωριστά (Εικόνα 6.3).

Η διαδικασία έχει ως εξής:

7.1.1 Αρνητικές εικόνες

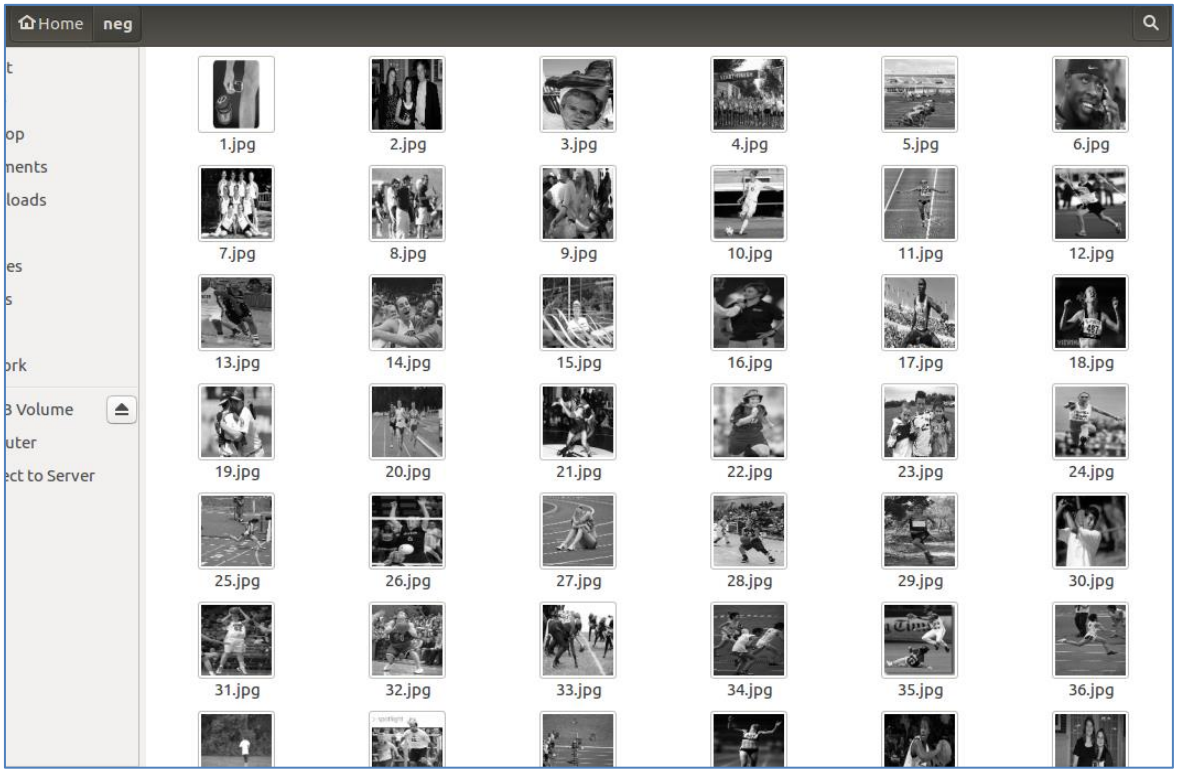
7.1.1.1 Κατέβασμα και αποσυμπίεση αρνητικών εικόνων

Κατεβάζουμε τις αρνητικές εικόνες από αυτόν το σύνδεσμο: “https://pythonprogramming.net/static/images/opencv/negative-background-images.zip“. Αποσυμπιέζουμε τον φάκελο “negative-background-images.zip”, ο οποίος περιέχει το φάκελο neg (Εικόνα 7.1)...



Εικόνα 7.1: Αρνητικές εικόνες σε .zip

...με τις αρνητικές εικόνες σε μορφή .jpg (Εικόνα 7.2).

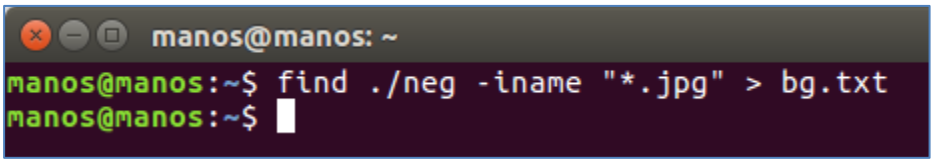


Εικόνα 7.2: Αρνητικές εικόνες

7.1.1.2 Εύρεση μονοπατιού αρνητικών εικόνων

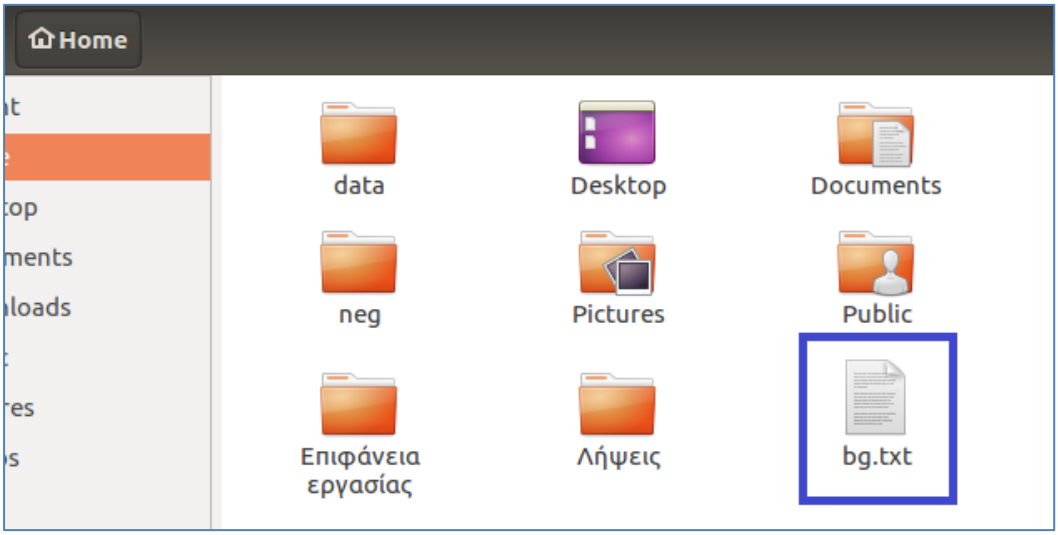
Εντοπίζουμε πού βρίσκονται οι αρνητικές εικόνες (Εικόνα 7.3)...

find ./neg -iname "*.jpg" > bg.txt



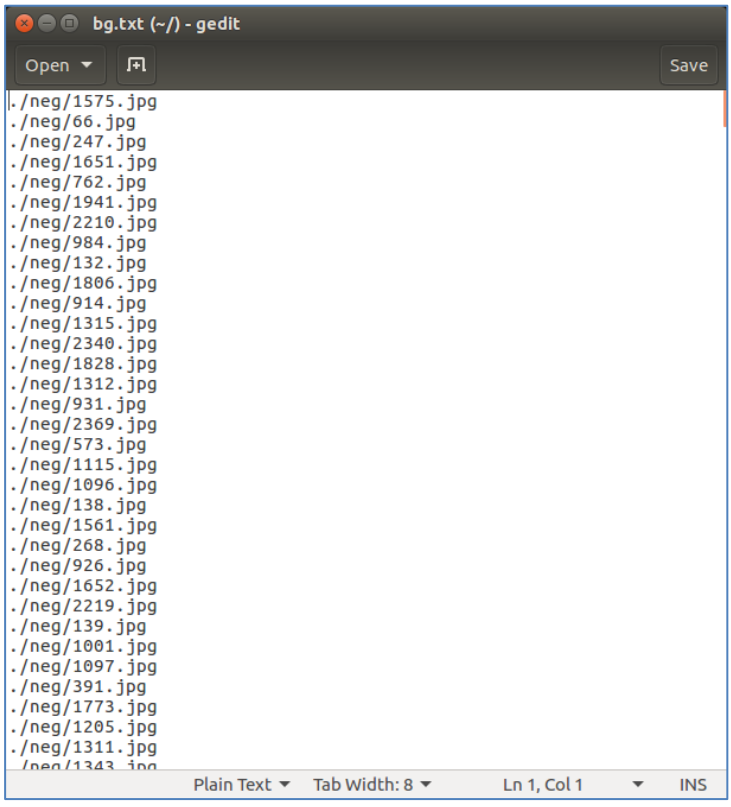
Εικόνα 7.3: Εντολή find

...και γράφουμε σ' ένα αρχείο .txt το μονοπάτι (path) των εικόνων αυτών (Εικόνα 7.4)...



Εικόνα 7.4: Αρχείο bg.txt

...του οποίου τα δεδομένα φαίνονται παρακάτω (Εικόνα 7.5):



Εικόνα 7.5: Δεδομένα bg.txt

7.1.2 Δημιουργία θετικών εικόνων

Δημιουργούμε τις θετικές εικόνες και το αρχείο info.lst, το οποίο περιέχει τις πληροφορίες της κάθε εικόνας. Αυτό επιτυγχάνεται προσθέτοντας την εικόνα προς ανίχνευση σε όλες τις αρνητικές εικόνες.

Τα παραπάνω πραγματοποιούνται από την επόμενη εντολή:

opencv_createsamples -img stop.jpg -bg bg.txt -info info/info.lst -pngoutput info -maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -num 1950

Οι παράμετροι της παραπάνω εντολής:

-img stop.jpg : το αντικείμενο προς ανίχνευση

-bg bg.txt : το αρχείο με το μονοπάτι (path) των αρνητικών εικόνων

-info info/info.lst : Το αρχειο info.lst περιέχει πληροφορίες για τις θετικές εικόνες

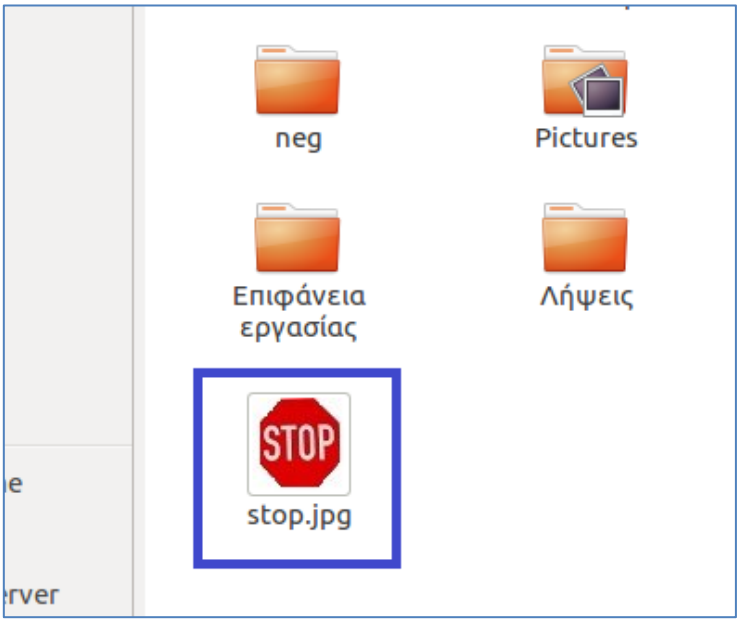
-pngoutput info : στελνει τις θετικές εικόνες να αποθηκευτουν στο φάκελο info

-maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 : μέγιστη γωνία περιστροφής που μπορεί να έχει η εικόνα

-num 1950 : αριθμός των αρνητικών εικόνων

7.1.2.1 Επιλογή αντικειμένου προς ανίχνευση

Αρχικά, γίνεται η επιλογή του σήματος που θα προστεθεί στις αρνητικές εικόνες (Εικόνα 7.6)...



Εικόνα 7.6: Επιλογή αντικειμένου για τις αρνητικές εικόνες

7.1.2.2 Προσθήκη αντικειμένου προς ανίχνευση

Έπειτα, κάθε αρνητική εικόνα ανοίγει και προστίθεται σε αυτή το αντικείμενο προς ανίχνευση (Εικόνα 7.7)...

```
manos@manos:~$ opencv_createsamples -img stop.jpg -bg bg.txt -info info/info.lst -pngoutput info -maxxangle 0.5 -maxyangle 0.5 -maxza
ngle 0.5 -num 1950
Info file name: info/info.lst
Img file name: stop.jpg
Vec file name: (NULL)
BG  file name: bg.txt
Num: 1950
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 0.5
Max y angle: 0.5
Max z angle: 0.5
Show samples: FALSE
Width: 24
Height: 24
Create test samples from single image applying distortions...
Open background image: ./neg/1575.jpg
Open background image: ./neg/1768.jpg
Open background image: ./neg/1327.jpg
Open background image: ./neg/1361.jpg
Open background image: ./neg/1642.jpg
Open background image: ./neg/588.jpg
Open background image: ./neg/1439.jpg
Open background image: ./neg/1852.jpg
Open background image: ./neg/1039.jpg
Open background image: ./neg/1065.jpg
Open background image: ./neg/1215.jpg
Open background image: ./neg/242.jpg
Open background image: ./neg/2211.jpg
Open background image: ./neg/414.jpg
Open background image: ./neg/1394.jpg
Open background image: ./neg/1738.jpg
Open background image: ./neg/1919.jpg
Open background image: ./neg/192.jpg
Open background image: ./neg/1604.jpg
Open background image: ./neg/699.jpg
Open background image: ./neg/1469.jpg
```

Εικόνα 7.7: Προσθήκη αντικειμένου στις αρνητικές εικόνες

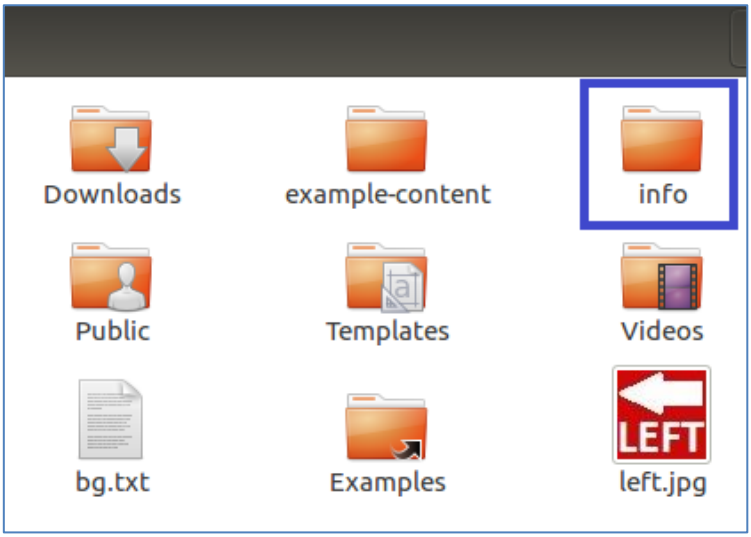
...και η διαδικασία αυτή ολοκληρώνεται με την ένδειξη “Done” στο τερματικό (Εικόνα 7.8).

```
Open background image: ./neg/1070.jpg
Open background image: ./neg/1131.jpg
Open background image: ./neg/1134.jpg
Open background image: ./neg/778.jpg
Open background image: ./neg/2105.jpg
Open background image: ./neg/1439.jpg
Open background image: ./neg/761.jpg
Open background image: ./neg/2136.jpg
Open background image: ./neg/33.jpg
Open background image: ./neg/93.jpg
Open background image: ./neg/537.jpg
Open background image: ./neg/1973.jpg
Open background image: ./neg/360.jpg
Open background image: ./neg/1008.jpg
Open background image: ./neg/777.jpg
Open background image: ./neg/1008.jpg
Open background image: ./neg/1262.jpg
Open background image: ./neg/1411.jpg
Open background image: ./neg/376.jpg
Open background image: ./neg/1790.jpg
Open background image: ./neg/537.jpg
Done
```

Εικόνα 7.8: Ολοκλήρωση προσθήκης αντικειμένου

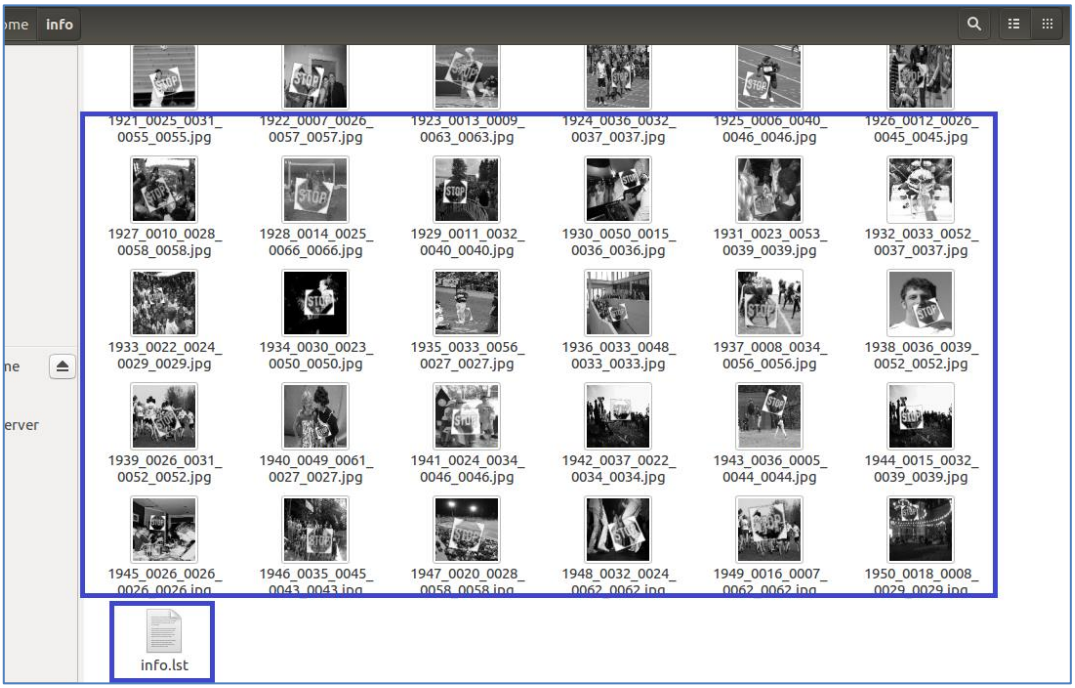
7.1.2.3 Δημιουργία φακέλου info και αρχείου info.lst

Επιπλέον, δημιουργείται ο φάκελος info (Εικόνα 7.9)...



Εικόνα 7.9: Δημιουργία φακέλου info

...και μέσα σε αυτόν δημιουργούνται οι θετικές εικόνες και το αρχείο info.lst (Εικόνα 7.10).



Εικόνα 7.10: Θετικές εικόνες και info.lst

Ακολουθεί ένα κομμάτι του info.lst (Εικόνα 7.11):

info.lst

```
0001_0025_0031_0049_0049.jpg 1 25 31 49 49
0002_0056_0057_0035_0035.jpg 1 56 57 35 35
0003_0027_0010_0056_0056.jpg 1 27 10 56 56
0004_0037_0019_0055_0055.jpg 1 37 19 55 55
0005_0021_0041_0037_0037.jpg 1 21 41 37 37
0006_0014_0010_0060_0060.jpg 1 14 10 60 60
0007_0047_0010_0041_0041.jpg 1 47 10 41 41
0008_0035_0019_0036_0036.jpg 1 35 19 36 36
0009_0017_0034_0057_0057.jpg 1 17 34 57 57
0010_0050_0022_0038_0038.jpg 1 50 22 38 38
0011_0043_0047_0034_0034.jpg 1 43 47 34 34
0012_0023_0020_0057_0057.jpg 1 23 20 57 57
0013_0032_0009_0036_0036.jpg 1 32 9 36 36
0014_0012_0011_0063_0063.jpg 1 12 11 63 63
0015_0030_0049_0036_0036.jpg 1 30 49 36 36
0016_0040_0021_0042_0042.jpg 1 40 21 42 42
0017_0039_0054_0030_0030.jpg 1 39 54 30 30
0018_0018_0044_0040_0040.jpg 1 18 44 40 40
0019_0051_0006_0041_0041.jpg 1 51 6 41 41
0020_0030_0017_0057_0057.jpg 1 30 17 57 57
0021_0028_0044_0044_0044.jpg 1 28 44 44 44
0022_0036_0015_0043_0043.jpg 1 36 15 43 43
0023_0027_0013_0026_0026.jpg 1 27 13 26 26
0024_0013_0028_0028_0028.jpg 1 13 28 28 28
0025_0047_0015_0044_0044.jpg 1 47 15 44 44
0026_0034_0040_0033_0033.jpg 1 34 40 33 33
```

Εικόνα 7.11: Αρχείο info.lst

Για παράδειγμα, η παραπάνω πληροφορία στο κόκκινο πλαίσιο σημαίνει ότι:

0006_0014_0010_0060_0060.jpg : ονομασία εικόνας

1 : πλήθος εμφανίσεων στην θετική εικόνα

14 : τετμημένη x του πάνω αριστερά pixel της εικόνας προς ανίχνευση

10 : τεταγμένη y του πάνω αριστερά pixel της εικόνας προς ανίχνευση

60 : μήκος σε pixel

60 : πλάτος σε pixel

7.1.3 Δημιουργία θετικών δειγμάτων

Η παρακάτω εντολή παράγει το αρχείο positives.vec, το οποίο περιέχει τις διαστάσεις του κάθε θετικού δείγματος και το πλήθος αυτών, σε δυαδική μορφή. Για να κατασκευάσουμε τον ταξινομητή, τα θετικά δείγματα πρέπει να έχουν όλα το ίδιο ακριβώς μέγεθος. Για το λόγο αυτό, δίνουμε ως παράμετρο στην createsamples το μέγεθος (πλάτος και ύψος) που θέλουμε να έχουν.

opencv_createsamples -info info/info.lst -num 1950 -w 22 -h 22 -vec positives.vec

Οι παράμετροι της παραπάνω εντολής:

-info info/info.lst : Το αρχειο info.lst περιέχει πληροφορίες για τις θετικές εικόνες

-w 22 -h 22 : πλάτος και ύψος δειγμάτων σε pixels

-num 1950 : αριθμος θετικών δειγμάτων

-vec positives.vec : Το αρχειο positives.vec περιέχει πληροφορίες για τα θετικά δείγματα.

7.1.3.1 Επιλογή κομματιού από θετικά δείγματα

Το πρόγραμμα γνωρίζει πλέον μέσω του info.lst τις συντεταγμένες του αντικειμένου που ψάχνουμε, οπότε από κάθε θετική εικόνα επιλέγει τυχαία ένα κομμάτι του με διαστασεις 22x22 για κάθε μία από τις 1950 αρνητικές εικόνες (Εικόνα 7.12).

Εικόνα 7.12: Επιλογή κομματιού 22x22

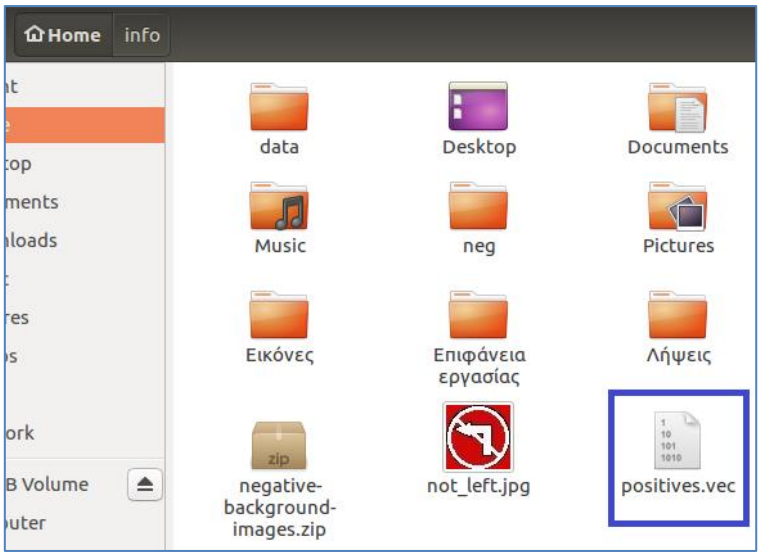
7.1.3.2 Δημιουργία .vec αρχείου

Στη συνέχεια, η createsamples λαμβάνει το σύνολο των 1950 κομματιών (Εικόνα 7.13)…

```
manos@manos: ~
manos@manos:~$ opencv_createsamples -info info/info.lst -num 1950 -w 22 -h 22 -vec positives
.vec
Info file name: info/info.lst
Img file name: (NULL)
Vec file name: positives.vec
BG  file name: (NULL)
Num: 1950
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 22
Height: 22
Create training samples from images collection...
Done. Created 1950 samples
```

Εικόνα 7.13: Δημιουργία αρχείου positives.vec

…και τα συγκεντρώνει μαζί σε ένα .vec αρχείο, ώστε να μην έχουμε πολλά αρχεία εικόνων ξεχωριστά (Εικόνα 7.14).



Εικόνα 7.14: Αρχείο positives.vec

7.1.4 Εκπαίδευση ταξινομητή

Εκπαιδεύουμε τον ταξινομητή, χρησιμοποιώντας τα θετικά και τα αρνητικά δείγματα που έχουμε δημιουργήσει με την επόμενη εντολή (Εικόνα 7.15).

opencv_traincascade -data data -vec positives.vec -bg bg.txt -numPos 1800 -numNeg 1950 -numStages 18 -w 22 -h 22

Οι παράμετροι της παραπάνω εντολής:

-data data : στο φάκελο data θα αποθηκευτεί ο ανιχνευτής με καταληξη .xml

-vec positives.vec : το αρχειο positives.vec περιέχει πληροφορίες για τα θετικά δείγματα.

-bg bg.txt : το αρχείο με το μονοπάτι (path) των αρνητικών εικόνων

-numPos 1800 : πλήθος θετικών δειγμάτων (πρέπει να είναι λιγότερα από τα αρνητικά δείγματα)

-numNeg 1950 : πλήθος αρνητικών δειγμάτων όπου χρησιμοποιούνται στο κάθε ν-στάδιο. Συλλέγονται τυχαία και δημιουργούνται μετά από επεξεργασία των αντίστοιχων αρνητικών εικόνων (πχ. περικοπή ή αλλαγή κλίμακας). Μπορούν να είναι περισσότερα ή λιγότερα από το πλήθος των αρνητικών εικόνων.

-numStages 18 : πλήθος σταδίων

-w 22 -h 22 : πλάτος και ύψος δειγμάτων

```
manos@manos:~$ opencv_traincascade -data data -vec positives.vec -bg bg.txt -numPos 1800 -numNeg 1950 -numStages 18 -w 22 -h 22
PARAMETERS:
cascadeDirName: data
vecFileName: positives.vec
bgFileName: bg.txt
numPos: 1800
numNeg: 1950
numStages: 18
precalcValBufSize[Mb] : 1024
precalcIdxBufSize[Mb] : 1024
acceptanceRatioBreakValue : -1
stageType: BOOST
featureType: HAAR
sampleWidth: 22
sampleHeight: 22
boostType: GAB
minHitRate: 0.995
maxFalseAlarmRate: 0.5
weightTrimRate: 0.95
maxDepth: 1
maxWeakCount: 100
mode: BASIC
```

Εικόνα 7.15: Δημιουργία του αρχείου .xml

Τα δεδομένα εξόδου της εντολής είναι:

- **cascadeDirName**: το όνομα του καταλόγου που θα αποθηκευτεί ο ταξινομητής
- **vecFileName**: το αρχείο .vec που δημιουργήσαμε
- **bgFileName**: το αρχείο .txt που περιλαμβάνει τα αρνητικά δείγματα
- **numPos**: ο αριθμός θετικών δειγμάτων
- **numNeg**: ο αριθμός αρνητικών δειγμάτων
- **numStages**: ο αριθμός σταδίων
- **precalcValBufSize[Mb]** και **precalcIdxBufSize[Mb]**: μέγιστη χρησιμοποιούμενη RAM
- **acceptanceRatioBreakValue**: ακρίβεια εκπαίδευσης του ταξινομητή και όριο παύσης της εκπαίδευσης. Με το -1 απενεργοποιείται η λειτουργία αυτή.
- **stageType**: ο τύπος των σταδίων (πρέπει οπωσδήποτε να είναι τύπου Boost)
- **featureType**: τύπος χαρακτηριστικών

- **sampleWidth**: πλάτος δειγμάτων

- **sampleHeight**: ύψος δειγμάτων

- **boostType**: επιλογή τύπου του αλγορίθμου AdaBoost (GAB - Gentle AdaBoost)

- **minHitRate**: ελάχιστος λόγος θετικών ανιχνεύσεων, δηλαδή το επιθυμητό ποσοστό επιτυχίας κάθε σταδίου.
  Για να έχουμε καλή ταξινόμηση, η τιμή αυτή θα πρέπει να είναι πολύ κοντά στη μονάδα. Η προεπιλεγμένη τιμή είναι 0,995 [31].

- **maxFalseAlarmRate**: μέγιστος λόγος εσφαλμένων ανιχνεύσεων κάθε σταδίου, δηλαδή το πόσα χαρακτηριστικά πρέπει να προστεθούν.
  Με τιμή κοντά στο 0,5 (αλλά πάντα κάτω από αυτό), κατασκευάζεται ένας γρήγορος ταξινομητής που σε κάθε στάδιο απορρίπτει περίπου τα μισά αρνητικά παράθυρα. Αν επιλέξουμε τιμή πολύ χαμηλότερη (στην περιοχή του 0,05), ο ταξινομητής θα απορρίπτει πολύ περισσότερα αρνητικά παράθυρα σε κάθε στάδιο. Έτσι, για τον ίδιο αριθμό σταδίων με το προηγούμενο παράδειγμα, θα έχει πολύ χαμηλότερο συνολικό λόγο λανθασμένων ανιχνεύσεων. Ωστόσο, ο ταξινομητής θα είναι πιο αργός και η εκπαίδευσή του πιο χρονοβόρα. Η προεπιλεγμένη τιμή είναι 0,5 [31].

- **weightTrimRate**: Η παράμετρος αυτή δηλώνει το συντελεστή ρύθμισης βαρών wf. Κατά την εκπαίδευση του ταξινομητή με τη χρήση του αλγορίθμου AdaBoost, τα παραδείγματα που χρησιμοποιούνται έχουν μια τιμή βάρους η οποία μεταβάλλεται σε κάθε κύκλο εκπαίδευσης. Αν ένα παράδειγμα ταξινομήθηκε σωστά σε έναν κύκλο εκπαίδευσης, τότε το βάρος του μειώνεται, ώστε ο AdaBoost να θεωρήσει πιο σημαντικά τα παραδείγματα που ταξινομήθηκαν λανθασμένα. Το βάρος των παραδειγμάτων αυτών μειώνεται πολλαπλασιάζοντάς το με τον αριθμό 1 - wf , άρα, όσο μεγαλύτερη είναι η τιμή της παραμέτρου weightTrimRate, τόσο πιο πολύ μειώνεται η τιμή του βάρους των παραδειγμάτων που ταξινομήθηκαν σωστά [31].

- **maxDepth**: Μέγιστο βάθος ενός αδύναμου δέντρου

- **maxWeakCount**: Μέγιστος αριθμός των αδύναμων δέντρων για κάθε στάδιο ταξινομητή καταρράκτη.

- **mode**: επιλογή τύπου των χαρακτηριστικών Haar που χρησιμοποιούνται στην εκπαίδευση

7.1.4.1 Υπολογισμός χαρακτηριστικών Haar

Αρχικά, γίνεται η διαδικασία υπολογισμού των haar χαρακτηριστικών από τα θετικά δείγματα. Αφού χωρίσουμε σε 2 ή 3 περιοχές το κάθε θετικό δείγμα, δηλαδή σε μία λευκή και μία μαύρη ή σε 2 λευκές και μία μαύρη, αντίστοιχα (Εικόνα 6.1), υπολογίζουμε το άθροισμα των λευκών και το αντίστοιχο των μαύρων ορθογωνίων για καθεμιά από τις 5 εικόνες (Εικόνα 7.16). Το αποτέλεσμα που προκύπτει από τη διαφορά του αθροίσματος των μαύρων από αυτό των λευκών αντιστοιχεί στην τιμή του haar χαρακτηριστικού.

Εικόνα 7.16: Υπολογισμός χαρακτηριστικών Haar

7.1.4.2 Κατασκευή Αδύναμου Ταξινομητή (classifier)

Έπειτα, επιλέγεται μια ομάδα 18 καλών χαρακτηριστικών τύπου Haar, τα οποία διαφέρουν μεταξύ τους. Αυτή η ομάδα αντιπροσωπεύει τους 18 αδύναμους ταξινομητές (από τα 18 στάδια ταξινόμησης), που θα συμβάλλουν στην καλύτερη ταξινόμηση και τελικά, στη δημιουργία ενός ισχυρού ταξινομητή.

7.1.4.3 Κατασκευή Ισχυρού Ταξινομητή με Adaboost

Για να δημιουργήσουμε τον ισχυρό ταξινομητή χρησιμοποιούμε έναν αλγόριθμο ενίσχυσης, ώστε από τους 18 αδύναμους που δημιουργήθηκαν πριν, να προκύψει ο τελικός ισχυρός ταξινομητής. Ο αλγόριθμος ενίσχυσης που εκτελεί την παραπάνω λειτουργία είναι ο Adaboost και ο τρόπος με τον οποίο θα ταξινομηθούν τα χαρακτηριστικά αυτά είναι με τη μέθοδο GAB του Adaboost.

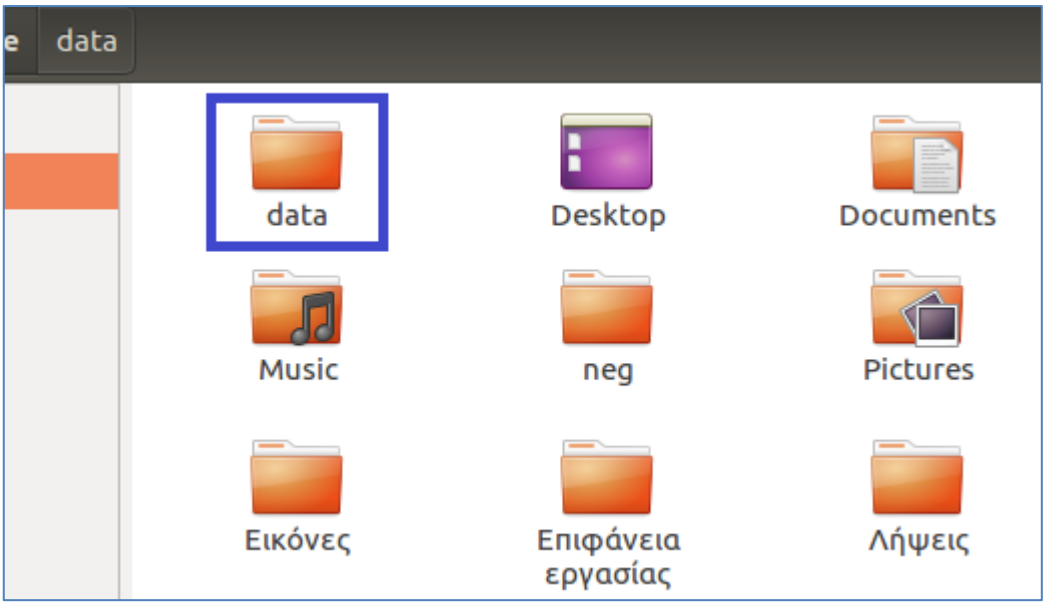
7.1.4.4 Δημιουργία ενός ταξινομημένου καταρράκτη (cascade classification)

Η εκπαίδευση του ταξινομητή καταρράκτη γίνεται χρησιμοποιώντας τον AdaBoost, και καθορίζει τον αριθμό των επιπέδων του καταρράκτη ταξινομητή, τον αριθμό των χαρακτηριστικών σε κάθε επίπεδο και το κατώφλι σε κάθε επίπεδο, ώστε να ελαχιστοποιείται ο αριθμός των χρησιμοποιούμενων χαρακτηριστικών.

Η διάταξη καταρράκτη βελτιώνει σημαντικά την ταχύτητα ανίχνευσης και μειώνει αποτελεσματικά τα σφάλματα με μικρό κόστος στους χρόνους ανίχνευσης [33].

Επομένως, έχοντας δημιουργήσει τον τελικό ανιχνευτή (cascade.xml), το Raspberry Pi χρησιμοποιεί αυτές τις πληροφορίες για την ανίχνευση των σημάτων. Συγκεκριμένα, συγκρίνει τις πληροφορίες αυτές με τις περιοχές ενδιαφέροντος της εικόνας που λαμβάνει η κάμερα κάθε στιγμή και εντοπίζει αν υπάρχει το σήμα ή όχι.

Τα αποτελέσματα της εντολής μετά την εκπαίδευση, αποθηκεύονται σε ένα φάκελο με όνομα “data” σε μορφή .xml (Εικόνα 7.17).



Εικόνα 7.17: Δημιουργία του φακέλου data

Αμέσως μόλις τρέξουμε την traincascade δημιουργείται μέσα στο data το αρχείο params.xml με τις παραμέτρους της εντολής (Εικόνα 7.18)...

```
−<opencv_storage>
 −<params>
    <stageType>BOOST</stageType>
    <featureType>HAAR</featureType>
    <height>22</height>
    <width>22</width>
   −<stageParams>
      <boostType>GAB</boostType>
      <minHitRate>9.9500000476837158e-01</minHitRate>
      <maxFalseAlarm>5.0000000000000000e-01</maxFalseAlarm>
      <weightTrimRate>9.4999999999999996e-01</weightTrimRate>
      <maxDepth>1</maxDepth>
      <maxWeakCount>100</maxWeakCount>
    </stageParams>
   −<featureParams>
      <maxCatCount>0</maxCatCount>
      <featSize>1</featSize>
      <mode>BASIC</mode>
    </featureParams>
  </params>
</opencv_storage>
```

Εικόνα 7.18: Δημιουργία του φακέλου data

...και ξεκινάει η εκπαίδευση με το πρώτο στάδιο (0-stage) παράγοντας το παρακάτω αποτέλεσμα στο τερματικό (Εικόνα 7.19):

```
===== TRAINING 0-stage =====
<BEGIN
POS count : consumed   1800 : 1800
NEG count : acceptanceRatio    1950 : 1
Precalculation time: 9
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|        1|        1|
+----+---------+---------+
|   4| 0.997222| 0.449744|
+----+---------+---------+
END>
Training until now has taken 0 days 0 hours 1 minutes 28 seconds.
```

Εικόνα 7.19: Πρώτο στάδιο εκπαίδευσης

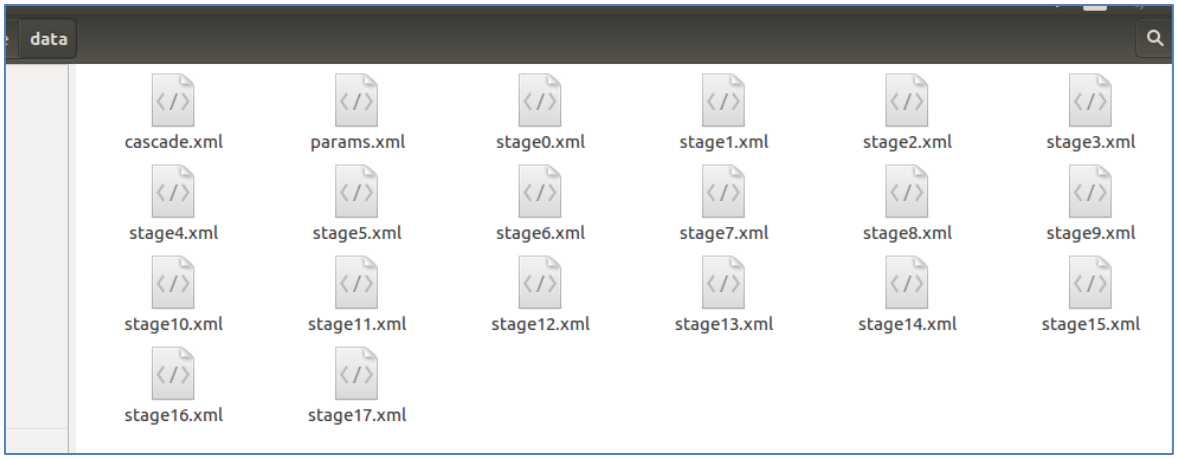
Η εκπαίδευση των σημάτων ολοκληρώνεται σε 18 στάδια, των οποίων το τελευταίο (stage-17) φαίνεται στην Εικόνα 7.20:

```
===== TRAINING 17-stage =====
<BEGIN
POS count : consumed   1800 : 1948
NEG count : acceptanceRatio    1950 : 5.78633e-06
Precalculation time: 8
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3|        1|        1|
+----+---------+---------+
|   4| 0.999444| 0.975385|
+----+---------+---------+
|   5| 0.995556| 0.912821|
+----+---------+---------+
|   6| 0.996111| 0.917436|
+----+---------+---------+
|   7| 0.998333| 0.961026|
+----+---------+---------+
|   8| 0.995556| 0.866154|
+----+---------+---------+
|   9| 0.995556| 0.890256|
+----+---------+---------+
|  10| 0.995556| 0.851282|
+----+---------+---------+
|  11| 0.995556| 0.828205|
+----+---------+---------+
|  12| 0.996667| 0.718974|
+----+---------+---------+
|  13| 0.996667| 0.724103|
+----+---------+---------+
|  14| 0.995556| 0.644615|
+----+---------+---------+
|  15| 0.995556| 0.666667|
+----+---------+---------+
|  16| 0.995556| 0.631282|
+----+---------+---------+
|  17| 0.995556| 0.550769|
+----+---------+---------+
|  18| 0.995556| 0.537949|
+----+---------+---------+
|  19| 0.995556| 0.494872|
+----+---------+---------+
END>
Training until now has taken 0 days 2 hours 58 minutes 53 seconds.
```

Εικόνα 7.20: Τελευταίο στάδιο εκπαίδευσης

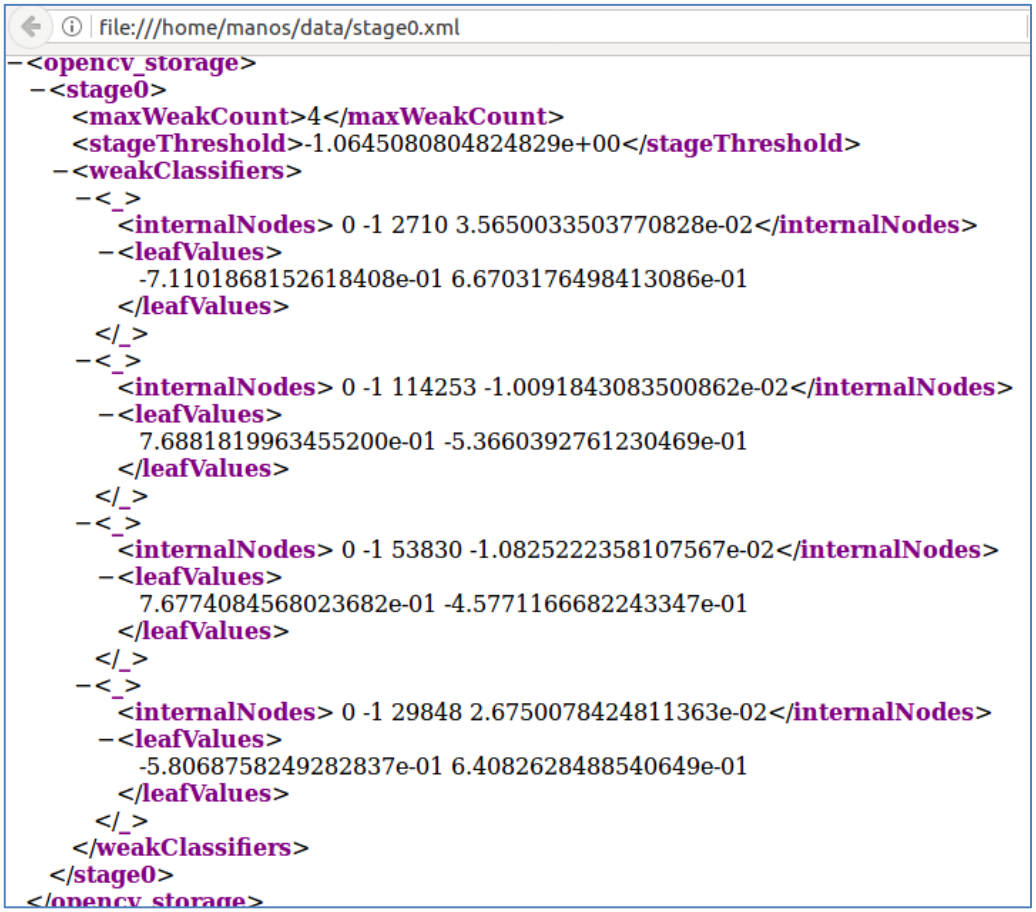
Το acceptanceRatio δηλώνει την ακρίβεια ταξινόμησης. Είναι το πηλίκο του αριθμού των θετικών δειγμάτων που έχουν ταξινομηθεί σωστά ως θετικά προς τον αριθμό των αρνητικών δειγμάτων που έχουν ταξινομηθεί λανθασμένα ως θετικά. Οι γραμμές των σταδίων αντιπροσωπεύουν η καθεμιά ένα εκπαιδευόμενο χαρακτηριστικό. Η 2η στήλη είναι το HitRatio που αντιπροσωπεύει το ποσοστό των θετικών δειγμάτων που έχουν ταξινομηθεί σωστά ως θετικά, ενώ η 3η το FalseAlarm Ratio που αντιπροσωπεύει το ποσοστό των αρνητικών δειγμάτων που ταξινομήθηκαν λανθασμένα ως θετικά.

Με την ολοκλήρωση των σταδίων δημιουργούνται μέσα στον data τα αρχεία .xml (Εικόνα 7.21)...



Εικόνα 7.21: Αρχεία .xml του φακέλου data

...με τα αποτελέσματα του κάθε σταδίου (Εικόνα 7.22)...



Εικόνα 7.22: Αρχείο stage0.xml

...και το αρχείο “cascade.xml” που περιέχει τις πληροφορίες που αφορούν τον ταξινομητή και το οποίο θα χρησιμοποιηθεί αργότερα για την ανίχνευση των σημάτων μέσω της κάμερας (Εικόνα 7.23).

Εικόνα 7.23: Αρχείο cascade.xml

Η ίδια διαδικασία πρέπει στη συνέχεια να γίνει και για τα άλλα 2 σήματα (left και notturn) για να προκύψουν και οι τρεις ανιχνευτές, ώστε μετά να μπορούμε να προχωρήσουμε στην ανίχνευση και έπειτα αναγνώριση των σημάτων.

Σημείωση: Όταν ολοκληρώνεται το κάθε στάδιο, αποθηκεύεται σε ένα αρχείο και η διαδικασία μπορεί να σταματήσει και να ξεκινήσει άλλη φορά από το στάδιο στο οποίο σταμάτησε.

ΚΕΦΑΛΑΙΟ 8

Στην προηγούμενη ενότητα έγινε η δημιουργία των τριών .xml αρχείων. Επομένως, μπορούμε πλέον να προχωρήσουμε στην αξιοποίηση αυτών των δεδομένων κατά τη διάρκεια της κίνησης του οχήματος με σκοπό την αναγνώριση των ευρισκόμενων σημάτων. Η όλη διαδικασία έχει ως εξής:

1. Ενεργοποιείται η κάμερα με επιτυχία
2. Φορτώνονται στο πρόγραμμα τα αρχεία .xml
3. Γίνεται επεξεργασία της εικόνας για κάθε στιγμιότυπο
4. Ανίχνευση της γραμμής και των σημάτων εκτελώντας τις αντίστοιχες οδηγίες των τελευταίων (στροφή, σταμάτημα)

8.1 Ενεργοποίηση κάμερας και λήψη εικόνας

Όμως, για να μπορούμε να επεξεργαστούμε μια εικόνα χρειάζεται πρώτα να την αποκτήσουμε. Αυτό θα πραγματοποιηθεί με τη χρήση μια κάμερας USB. Μέσω της κάμερας λοιπόν, η εικόνα συλλαμβάνεται σε πραγματικό χρόνο, φιλτράρεται και μετατρέπεται από κύματα φωτός σε αναλογικό σήμα. Έπειτα, ψηφιοποιείται μέσω ψηφιακού μετατροπέα και βελτιώνεται μέσω διαφόρων φίλτρων. Τέλος, η εικόνα συμπιέζεται και αποθηκεύεται προσωρινά στη μνήμη buffer (εφόσον είναι άδεια) μέχρι να χρησιμοποιηθεί. Η επεξεργασία αυτή θα γίνει μέσω κώδικα, όπου σταδιακά οι πιο σημαντικές εντολές του θα εξηγούνται παρακάτω.

Αρχικά, με την εντολή **cv2.VideoCapture(0)** ενεργοποιείται η κάμερα και "διαβάζει" frames ανά δευτερόλεπτο. Το "0" δηλώνει την πρώτη κάμερα που έχει οριστεί και είναι αυτή που χρησιμοποιούμε. Για όσο η κάμερα παραμένει συνδεδεμένη, διαβάζεται συνεχώς το επόμενο frame και, αν δεν μπορέσει να διαβαστεί επιτυχώς, το πρόγραμμα τερματίζει.

8.2 Φόρτωση πληροφοριών του .xml στο πρόγραμμα

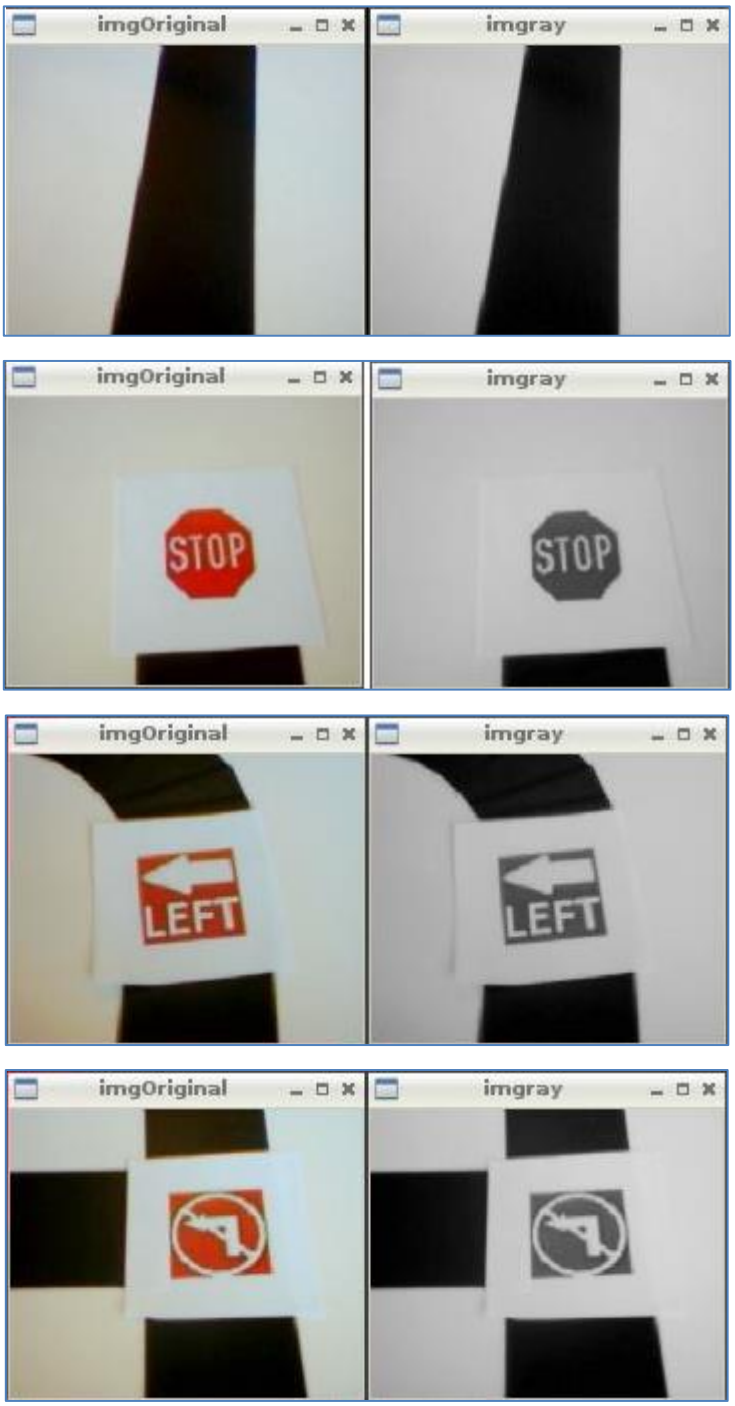
```
left_cascade = cv2.CascadeClassifier('/home/pi/left.xml')
notturn_cascade = cv2.CascadeClassifier('/home/pi/ notturn.xml')
stop_cascade = cv2.CascadeClassifier('/home/pi/stop.xml')
```

8.3 Εντολές για την επεξεργασία εικόνας

Παρακάτω θα αναλυθούν οι τρόποι εκμετάλλευσης της εικόνας, τους οποίους θα χρησιμοποιήσουμε στην εργασία μας. Η εικόνα λαμβάνεται σε έγχρωμη μορφή. Όμως για να γίνει η γραμμή εύκολα ανιχνεύσιμη από την κάμερα πρέπει πρώτα η εικόνα να περάσει από κάποια στάδια μετατροπών. Η έγχρωμη εικόνα μετατρέπεται σε αποχρώσεις του γκρι, ώστε έπειτα να εφαρμοστεί κατωφλίωση και να προκύψει ασπρόμαυρη εικόνα. Στη συνέχεια, μέσω της πράξης not σε όλα τα pixels της εικόνας, η μαύρη γραμμή γίνεται λευκή για να μπορέσουν τελικά να βρεθούν τα όρια της γραμμής. Ακολουθεί αναλυτικά αυτή η διαδικασία:

8.3.1 CvtColor

Με την εντολή **cv2.cvtColor(imgOriginal,cv2.COLOR_BGR2GRAY)** διαβάζουμε τις εικόνες τύπου BGR (Εικόνα 8.1α) (εικόνες με χρωματικούς πίνακες που συνδυάζουν τα τρία χρώματα) που έχει αποθηκεύσει προηγουμένως η κάμερα, για να αναπαράγει τα υπόλοιπα χρώματα. Αυτοί οι πίνακες διαθέτουν τιμές από 0-255 (BGR: B→0-255,G→0-255,R→0-255) και για να μετατρέψουμε σε grayscale χρησιμοποιούμε τον τύπο Gray= (B+G+R)/3 παράγοντας την τελική εικόνα με αποχρώσεις γκρι (Εικόνα 8.1β)).



Εικόνα 8.1: (α) BGR - (β) Grayscale

8.3.2 Threshold

Με την εντολή **cv2.threshold(imgray,127,255,cv2.THRESH_BINARY)** εφαρμόζουμε κατωφλίωση σε μια εικόνα για να διαχωρίζουμε τα αντικείμενα από το φόντο. Στη συγκεκριμένη περίπτωση το αντικείμενο θα είναι μια γραμμή επάνω σε ένα άσπρο φόντο. Για να είναι πιο αποτελεσματικός ο διαχωρισμός τους χρειάζεται να έχουμε μετατρέψει την εικόνα από χρωματικό πίνακα (BGR: B→0-255,G→0-255,R→0-255) σε αποχρώσεις του γκρι (grayscale→0-255 ή αλλιώς 0-1), ώστε να έχουμε μόνο έναν πίνακα (του γκρι). Έπειτα εφαρμόζουμε την κατωφλίωση μετατρέποντας την εικόνα σε δυαδική μορφή (Binary→0/1) (Εικόνα 8.2γ). Ορίζουμε την τιμή της κατωφλίωσης να είναι 127, δηλαδή κατά τη μετατροπή της εικόνας από grayscale σε δυαδική μορφή τα pixels που έχουν τιμή κάτω από 127 (πίνακας χρωμάτων του γκρι) θα γίνουν μαύρα (με τιμή 0) και αυτά που έχουν τιμή πάνω από 127 θα γίνουν άσπρα (με τιμή 1).

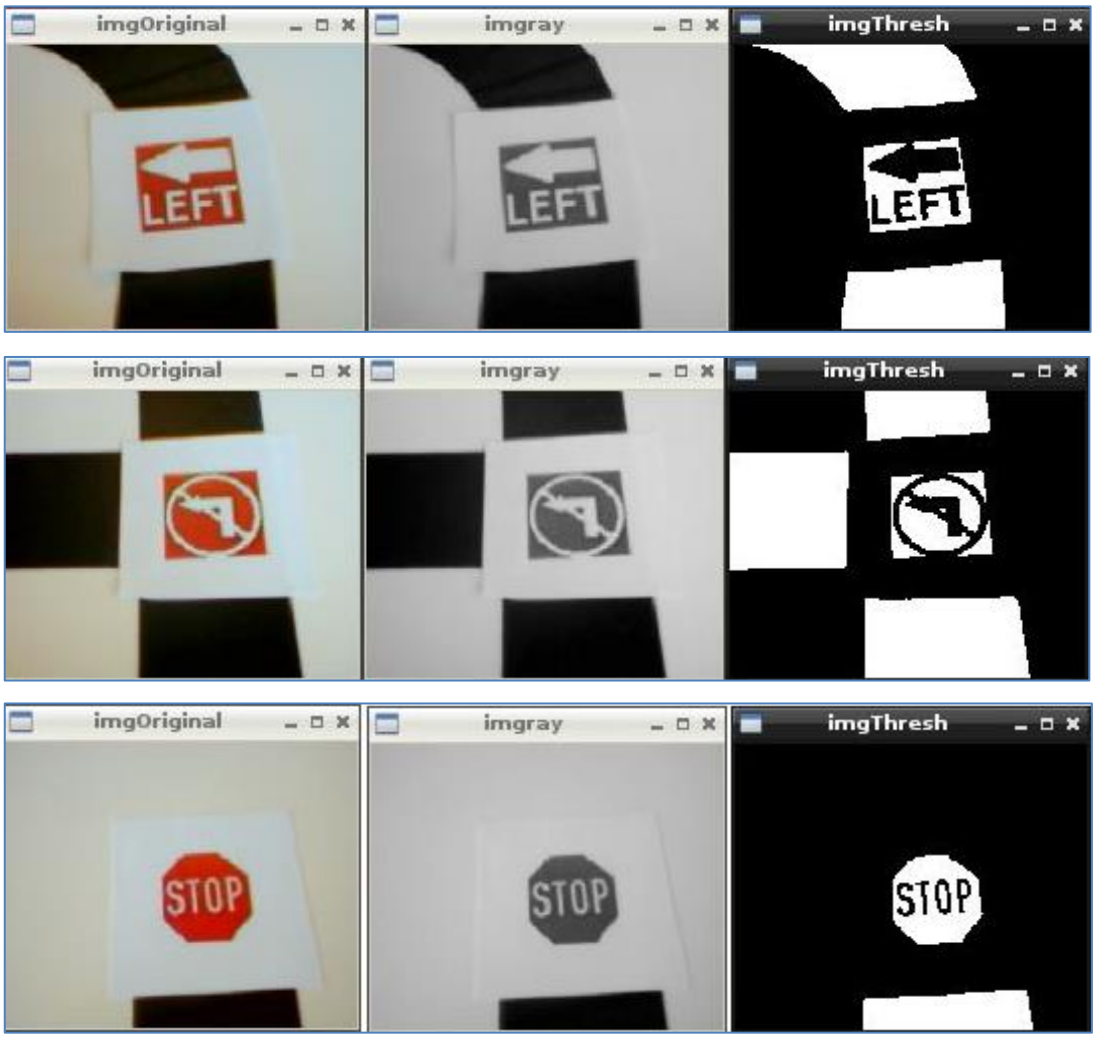
Τα ορίσματα της εντολής:

imgray: Η εικόνα εισόδου σε μορφή ασπρόμαυρη.

127: Η τιμή του ορίου (κατώφλι). Όσο μεγαλύτερη είναι η τιμή του ορίου, τόσο μικρότερη ευαισθησία υπάρχει ανάμεσα στα πλαίσια. Η τιμή που επιλέχτηκε είναι το 127.

255: Η μέγιστη τιμή που μπορεί να χρησιμοποιηθεί στην τελική εικόνα. Έχει οριστεί το 255 δηλαδή το χρώμα μαύρο

cv2.THRESH_BINARY: Ο τύπος κατωφλίωσης όπου μετατρέπει την εικόνα από αποχρώσεις του γκρι σε δυαδική

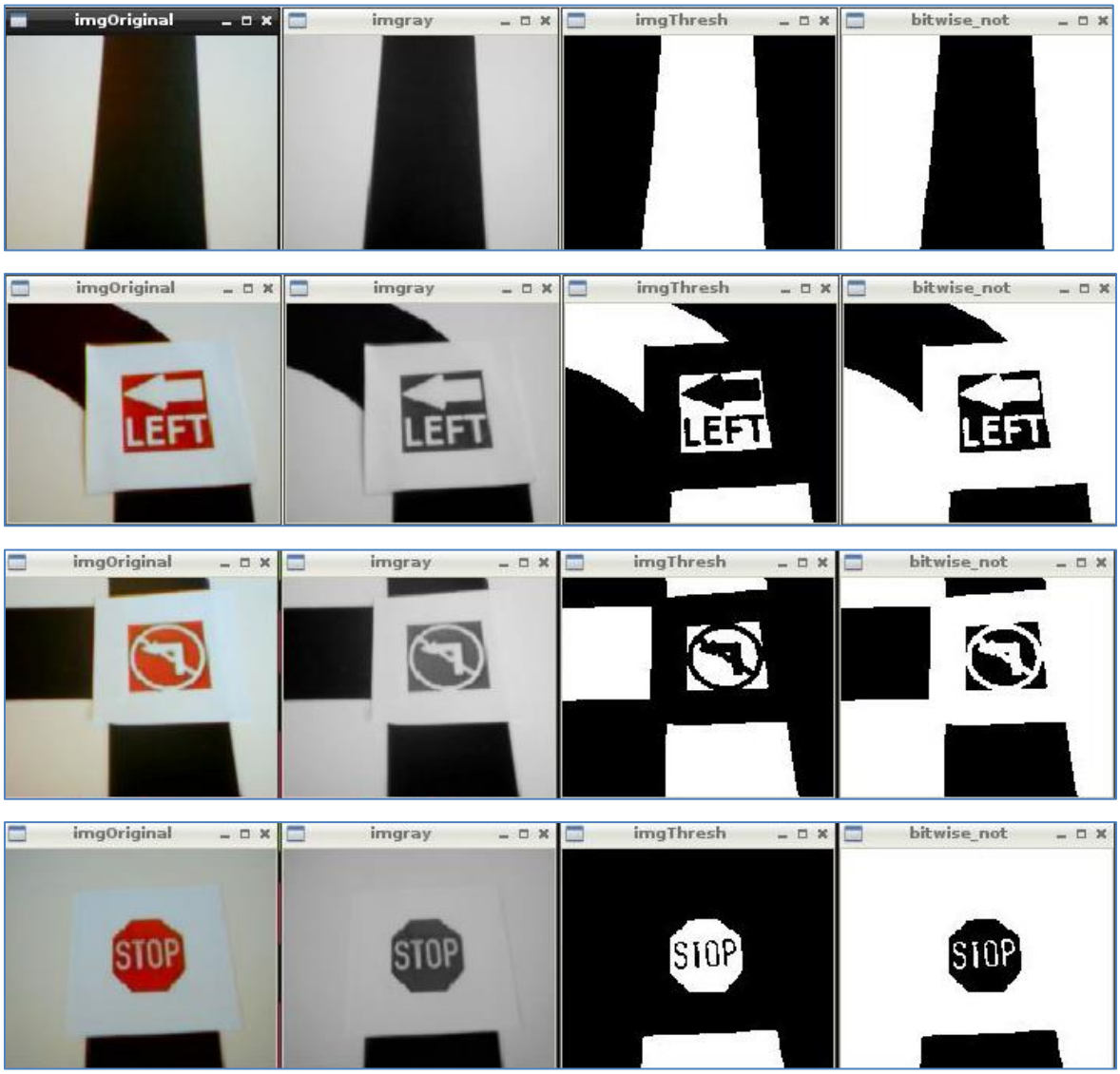


Εικόνα 8.2: (α) BGR - (β) Grayscale – (γ) Threshold

8.3.3 Bitwise_not

Με την εντολή **cv2.bitwise_not(thresh)** αντιστρέφουμε τα pixels της εικόνας για να προκύψει η λευκή γραμμή της διαδρομής (Εικόνα 8.3δ), ώστε αργότερα να είναι δυνατός ο εντοπισμός των ορίων της μέσω της εντολής FindContours. Η τελευταία εντολή βρίσκει τα όρια σε λευκά αντικείμενα και άρα με την αλλαγή του χρώματος της γραμμής σε λευκή θα μπορέσουν να βρεθούν τα όρια.

Το όρισμα thresh είναι η εικόνα που αντιστρέφεται.



Εικόνα 8.3: (α) BGR - (β) Grayscale – (γ) Threshold – (δ) Bitwise not

8.3.4 DetectMultiScale

Με την εντολή **stop_cascade.detectMultiScale(imgray, 1.3, 5)** ανιχνεύουμε τις σκαναρισμένες περιοχές όπου μπορεί να βρίσκονται τα αντικείμενα που έχουν προηγουμένως εκπαιδεύσει τον ταξινομητή και έχουν αποθηκευτεί στο αρχείο .xml (Εικόνα 8.4). Η εικόνα σαρώνεται συνεχώς από την detectMultiScale ελέγχοντας κάθε φορά αν έχει βρεθεί κάποιο αντικείμενο χρησιμοποιώντας διαφορετικές κλίμακες.

Οι παράμετροι της εντολής αντιστοιχούν στην ασπρόμαυρη εικόνα (imgray), το min μέγεθος του αντικειμένου (1.3) και max μέγεθος του αντικειμένου (5).

Εικόνα 8.4: Ανίχνευση σημάτων

8.3.5 FindContours

Με την εντολή **cv2.findContours(thresh.copy(),cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)** προσδιορίζουμε τα όρια του αντικειμένου. Συγκεκριμένα, διαχωρίζεται το αντικείμενο από το φόντο με την εισαγωγή περιγράμματος γύρω απ' το αντικείμενο (Εικόνα 8.5).

Έχει 3 ορίσματα: το πρώτο αποτελεί τη δυαδική εικόνα, το 2ο είναι μια σημαία (flag), η οποία κρατάει μόνο τα εξωτερικά στοιχεία, και το 3ο αφαιρεί όλα τα περιττά σημεία και συμπιέζει το περίγραμμα, εξοικονομώντας έτσι τη μνήμη.

8.3.6 ContourArea

Η cv2.contourArea υπολογίζει την περιοχή των ορίων του αντικειμένου, δηλαδή τις συντεταγμένες που έχει στην εικόνα

8.3.7 BoundingRect

Η cv2.boundingRect(c) δημιουργεί ένα τετράγωνο γύρω από το αντικείμενο με διαστάσεις που παίρνει από τις συντεταγμένες που βρέθηκαν πριν.

Στην παράμετρο c αποθηκεύονται οι διαστάσεις.

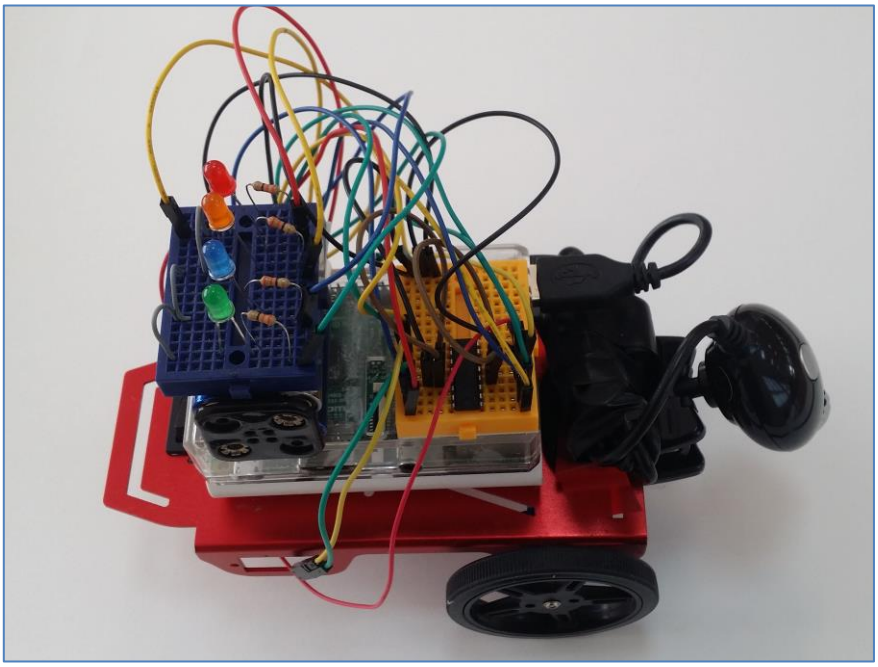
Εικόνα 8.5: Περίγραμμα αντικειμένου

Στη συνέχεια, αναλύεται ο τρόπος που γίνεται η αναγνώριση της γραμμής και των σημάτων.

8.4 Αναγνώριση γραμμής

Αφού έχει ενεργοποιηθεί η κάμερα με επιτυχία και φορτωθούν οι πληροφορίες του .xml στο πρόγραμμα, αυτό που μας ενδιαφέρει στη συνέχεια είναι η αναγνώριση της μαύρης γραμμής (Εικόνα 8.6) που θα ακολουθήσει το όχημα (Εικόνα 8.7).

Εικόνα 8.6: Πίστα γραμμής πορείας



Εικόνα 8.7: Όχημα

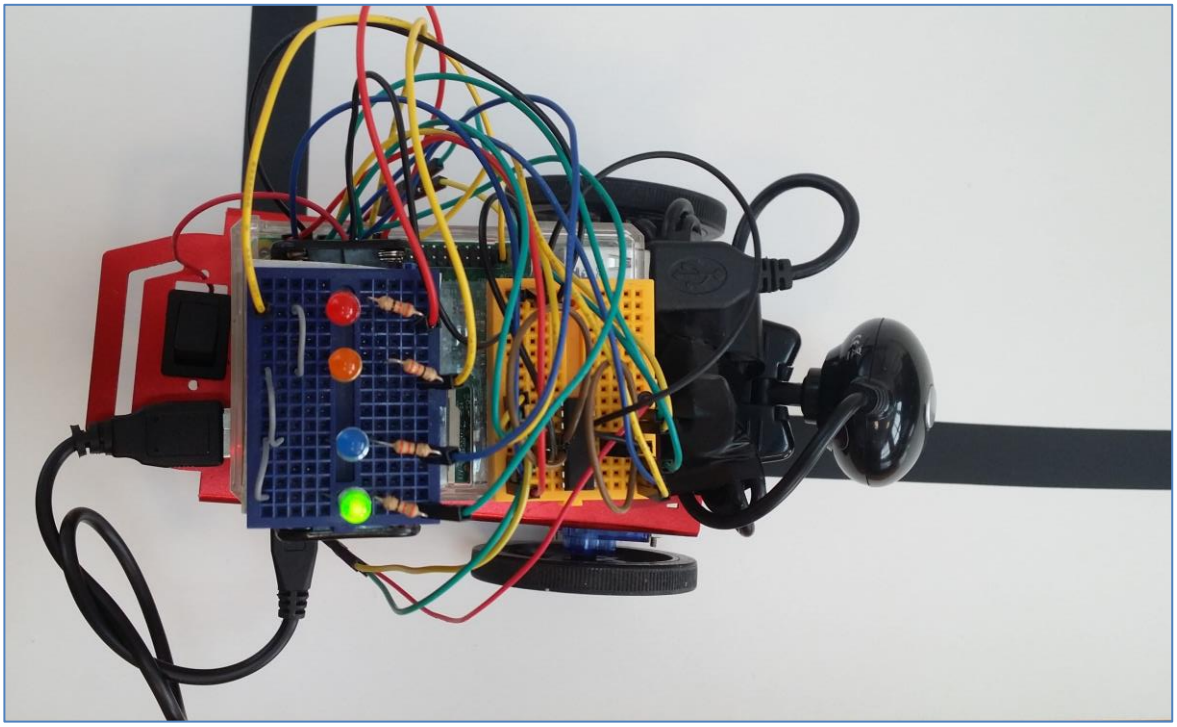
Η εικόνα που τραβάει η κάμερα έχει μέγεθος 160 * 120 (μήκος * ύψος). Στην περίπτωσή μας το ύψος της εικόνας δε μας αφορά. Αυτό που μας ενδιαφέρει είναι η πρώτη διάσταση (δηλαδή το μήκος), διότι το όχημα θα ανιχνεύει μια γραμμή, η οποία δε θα έχει κάποιο ύψος και θα είναι σχεδιασμένη σε χαρτί.

Αρχικά, η κάμερα που βρίσκεται πάνω του σκανάρει το χώρο για να βρει τη γραμμή. Αυτό επιτυγχάνεται εξετάζοντας το εμβαδόν του κάθε αντικειμένου που θα βρει η κάμερα. Αν αυτό είναι μεγαλύτερο ή ίσο του 10 αυτό σημαίνει ότι το αντικείμενο που βρέθηκε αντιστοιχεί στη γραμμή. Στη συνέχεια, η διαδικασία έχει ως εξής:

Διαιρείται το μήκος του παραθύρου δια δύο (160/2=80), ώστε να βρεθεί το κέντρο του. Αν η κάμερα βρει τη γραμμή τότε υπολογίζεται και αυτής το κέντρο (cx=x+((w)/2)). Έπειτα, για να έρθει σε ταύτιση το κέντρο του παραθύρου (=80) με το κέντρο της γραμμής εκτελείται το cx-=80, και κατά συνέπεια το όχημα βρίσκεται πάνω στη γραμμή. Από τη στιγμή που θα τη βρει, τότε θα ανάψει το πράσινο LED και το αυτοκίνητο θα ξεκινήσει να προχωράει για όσο θα βρίσκεται πάνω σε αυτή. Εάν όπως προχωράει χάσει τη γραμμή ή δεν υπάρχει άλλος δρόμος μπροστά του να διανύσει, τότε το όχημα θα σταματήσει και το LED θα σβήσει.

Είπαμε πριν ότι, αν η κάμερα εντοπίσει τη γραμμή, τότε προσπαθεί να την “φέρει” στο κέντρο του παραθύρου. Συγκεκριμένα, για να το πετύχει αυτό ελέγχει την απόσταση του κέντρου της γραμμής από το κέντρο του παραθύρου. Αν αυτή είναι μεταξύ -20 και 20 προχωράει ευθεία για 0.00003125 δευτερόλεπτα, ενεργοποιώντας τους 2 κινητήρες να πηγαίνουν προς τα μπροστά. Αν η τιμή της απόστασης είναι μικρότερη από -20 τότε στρίβει αριστερά για 0.025 δευτερόλεπτα, ενεργοποιώντας μόνο τον δεξιό κινητήρα, ενώ εάν είναι μεγαλύτερη από 20 τότε στρίβει δεξιά για 0.025 δευτερόλεπτα, ενεργοποιώντας μόνο τον αριστερό κινητήρα. Για να μην έχουμε απότομες κινήσεις και ξεφύγει από την πορεία του το όχημα, προσθέτουμε μια χρονοκαθυστέρηση για 0.00625 δευτερόλεπτα, η οποία θα το φρενάρει ελάχιστα σε κάθε στροφή.

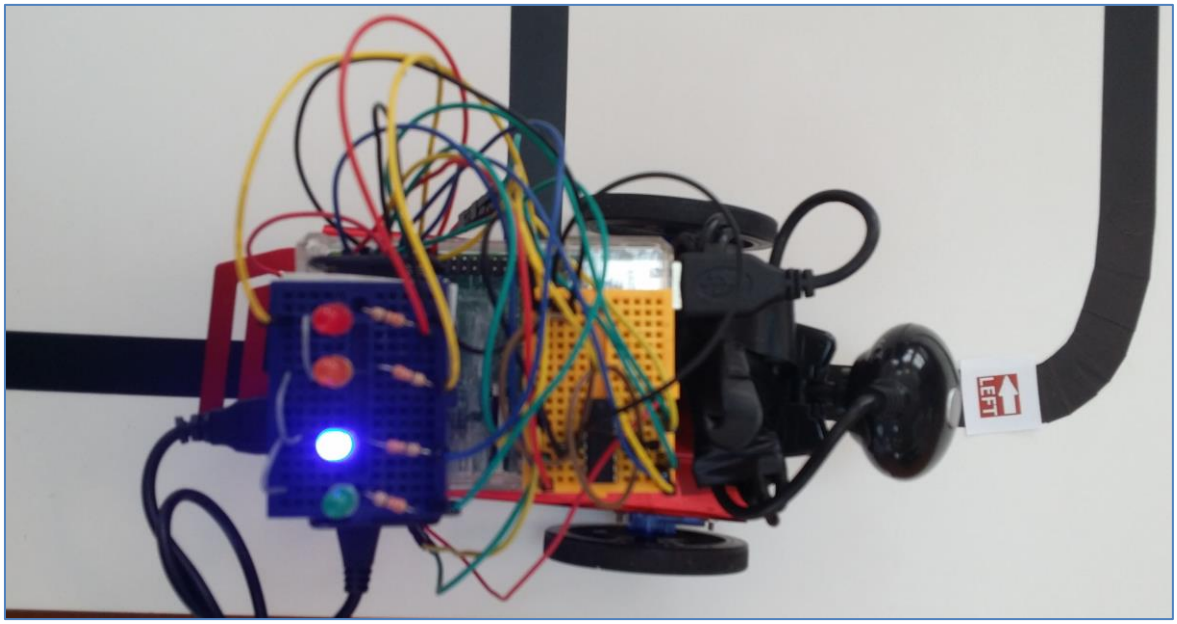
Στην επόμενη εικόνα βλέπουμε την ενεργοποίηση του πράσινου LED όταν αναγνωρίζεται η γραμμή.



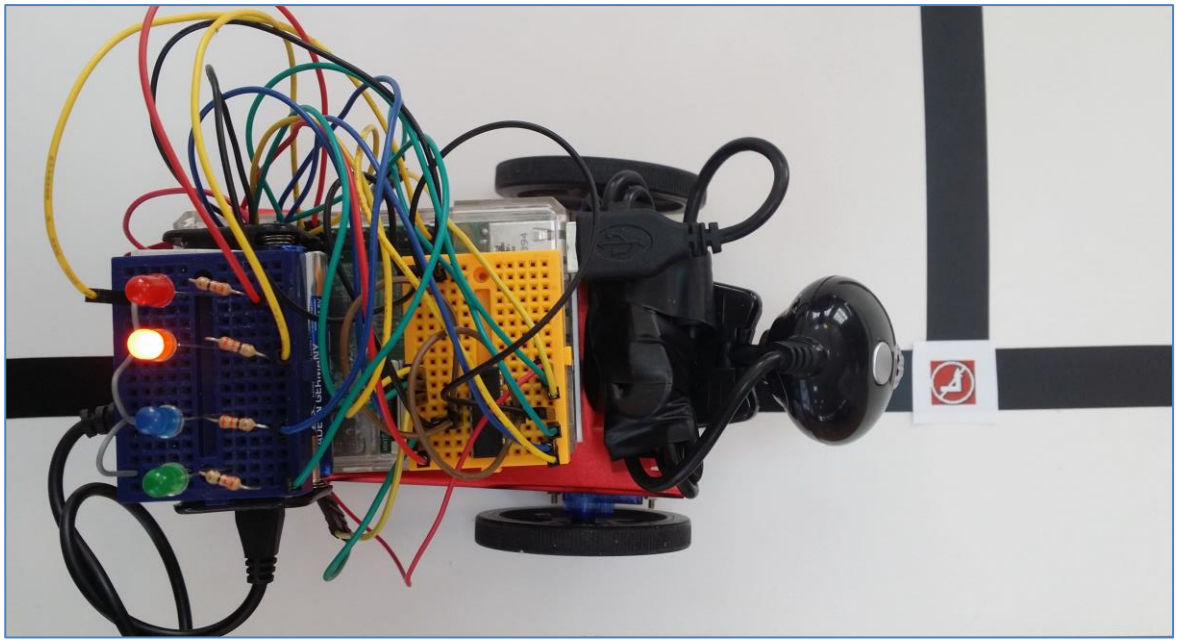
Εικόνα 8.8: Αναγνώριση γραμμής (πράσινο LED)

8.5 Αναγνώριση σημάτων

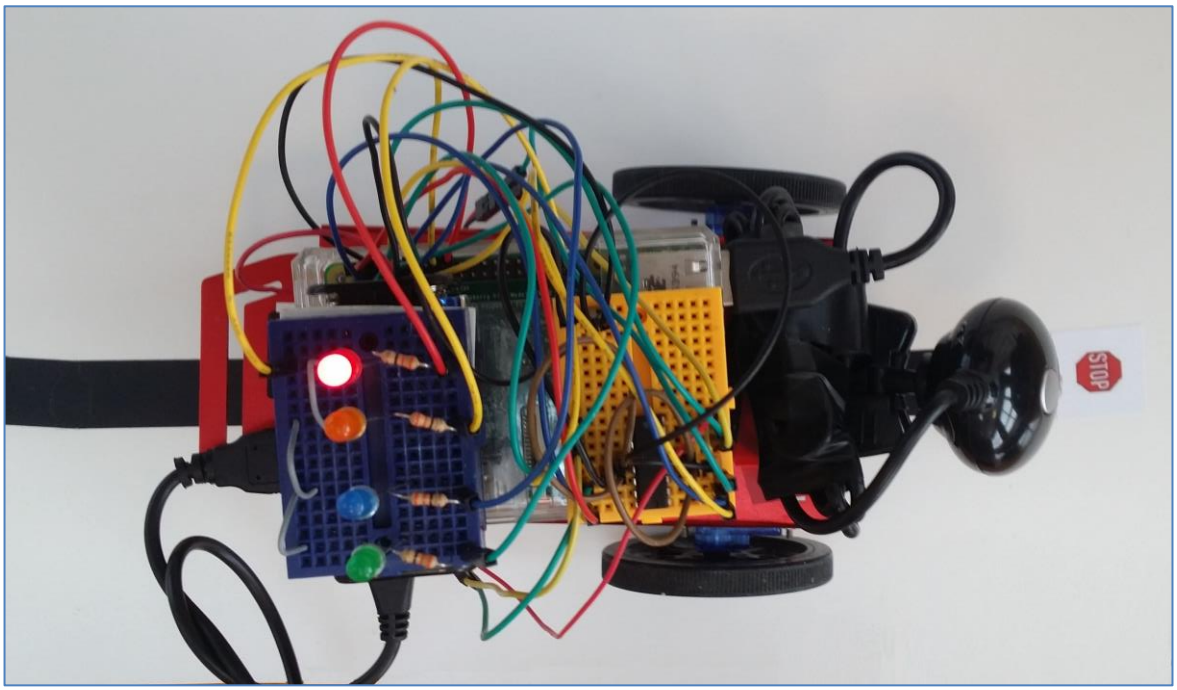
Όσον αφορά στην αναγνώριση σημάτων, αρχικά φορτώνονται οι πληροφορίες για τα σήματα από τα αρχεία .xml στο πρόγραμμα. Στη συνέχεια, η κάμερα ψάχνει ένα από τα τρία αντικείμενα (left, notturn και stop) στο χώρο αναλύοντας το κάθε frame που αντικρίζει και έπειτα γίνεται σύγκριση των πληροφοριών της συγκεκριμένης εικόνας με όλες τις πληροφορίες που έχουν από πριν αποθηκευτεί στα αρχεία .xml. Το όχημα κινείται συνεχώς στην πορεία της γραμμής μέχρι τη στιγμή που θα βρεθεί κάποιο σήμα και θα χρειαστεί να εκτελεστεί η αντίστοιχη λειτουργία του. Για παράδειγμα, για αριστερή στροφή (left) ενεργοποιείται μόνο ο δεξιός κινητήρας προς τα μπροστά για 0.25 sec, για την αποφυγή της δεξιάς στροφής (notturn) ενεργοποιούνται και οι 2 κινητήρες προς τα μπροστά για 0.03125 sec, ενώ για το σταμάτημα του οχήματος (stop) απενεργοποιούνται και οι 2 κινητήρες για 3 sec. Με την αναγνώριση του κάθε σήματος από τα παραπάνω, ανάβει το μπλε (Εικόνα 8.9), το πορτοκαλί (Εικόνα 8.10) ή το κόκκινο LED (Εικόνα 8.11), αντίστοιχα.



Εικόνα 8.9: Αναγνώριση left (μπλε LED)



Εικόνα 8.10: Αναγνώριση notturn (πορτοκαλί LED)



Εικόνα 8.11: Αναγνώριση stop (κόκκινο LED)

ΠΑΡΑΤΗΡΗΣΕΙΣ

1. Η εγκατάσταση της βιβλιοθήκης OpenCV απαιτεί αρκετή ώρα για να ολοκληρωθεί και χρειάστηκε να γίνει η διαδικασία αυτή ξανά μερικές φορές λόγω ανεξήγητων διακοπών της. Το χειρότερο είναι ότι δεν είναι δυνατή η συνέχιση της διαδικασίας από το σημείο που σταμάτησε πριν τη διακοπή και έτσι η εγκατάσταση χρειάζεται να ξεκινήσει πάλι από την αρχή.

2. Για την τροφοδοσία του Raspberry Pi χρησιμοποιήσαμε αρχικά τον αυθεντικό φορτιστή του, κάτι που στην πορεία καταλάβαμε ότι δεν ήταν βολικό, αφού κατά τη διαδρομή του οχήματος το καλώδιο δεν ήταν αρκετό (τέντωνε). Επιπλέον, σκεφτήκαμε ότι δε θα είναι πάντα εφικτή η ύπαρξη πρίζας (πχ. σε εξωτερικό χώρο). Έτσι επιλέξαμε την τροφοδοσία με powerbank, όπου με μια απλή φόρτισή του καταφέρνουμε την ελευθερία μας από καλώδια και πρίζες.

3. Μόλις το πρόγραμμα φορτωθεί στο RPi, η κάμερα ξεκινάει την ανίχνευση και το όχημα προχωράει. Για να ρυθμίσουμε την εκκίνηση και το σταμάτημα του οχήματος τοποθετήσαμε ένα διακόπτη, ούτως ώστε να έχουμε τον έλεγχο των κινητήρων.

4. Δεν μπορούμε να βάλουμε μεγάλη ταχύτητα στους κινητήρες, γιατί αλλιώς η κάμερα δεν προλαβαίνει να αναγνωρίσει τα αντικείμενα (γραμμή και σήματα).

5. Η ανίχνευση των σημάτων δεν είναι πάντα εφικτή όταν ο φωτισμός δεν είναι ο ιδανικός. Για παράδειγμα, σε συνθήκες χαμηλού φωτισμού τα σήματα δεν διακρίνονται αρκετά, ενώ στον υψηλό φωτισμό η μαύρη γραμμή γυαλίζει.

6. Ο κακός φωτισμός επηρεάζει και την αναγνώριση της γραμμής, αφού εξαιτίας του ορισμένες φορές δημιουργούνται σκιές στην πίστα, τις οποίες η κάμερα τις ανιχνεύει σαν την μαύρη γραμμή. Αυτό συμβαίνει όταν η σκιά είναι μεγάλη, δηλαδή έχει ξεπεράσει το όριο εμβαδού, ώστε τελικά να αναγνωρίζεται σαν μαύρη γραμμή (Ενότητα 8.4).

7. Ορισμένες φορές η κάμερα που έχει αναγνωρίσει μία φορά ένα σήμα, τυχαίνει και εξακολουθεί να το αναγνωρίζει για παραπάνω από μία φορές κάτι που κάνει το όχημα να στρίβει συνεχώς (αν πχ. το σήμα που “βλέπει” είναι σήμα στροφής). Άλλη περίπτωση είναι το όχημα να εκτελεί την εντολή του σήματος πριν έχει φτάσει η ώρα για να γίνει αυτό. Πιο συγκεκριμένα, η κάμερα αναγνωρίζει το σήμα στροφής που βρίσκεται εκείνη τη στιγμή στο οπτικό της πεδίο, όμως το όχημα μπορεί να βρίσκεται σε ευθεία πορεία και να μην πρέπει να στρίψει ακόμα. Με την ανίχνευση, όμως, του σήματος, το όχημα εκτελεί την οδηγία που έχει μπροστά του και φυσικά βρίσκεται αμέσως εκτός πορείας.

8. Όσον αφορά στην επιλογή των σημάτων, η πρώτη μας επιλογή περιλάμβανε το σήμα της δεξιάς στροφής αντί αυτού της απαγορευτικής δεξιάς στροφής (notturn). Στην πράξη, όμως, διαπιστώθηκε ότι η κάμερα μερικές φορές το αναγνώριζε ως σήμα left λόγω της ομοιότητας των 2 σημάτων. Αρχικά σκεφτήκαμε να προσθέσουμε έναν κύκλο γύρω από το βέλος, αλλά και πάλι το πρόβλημα “επέμενε”. Έτσι καταλήξαμε στη λύση της προσθήκης της γραμμής απαγόρευσης για να το διαφοροποιήσουμε περισσότερο από το σήμα αριστερής στροφής.

9. Κάθε αντικείμενο έχει ένα ιδιαίτερο χαρακτηριστικό, το οποίο το κάνει αναγνωρίσιμο, ούτως ώστε ο ανιχνευτής να μπορεί να καταλάβει για ποιο πρόκειται. Για παράδειγμα, στο σήμα του <STOP> η ομώνυμη λέξη αντιπροσωπεύει ένα τέτοιο χαρακτηριστικό. Για να γίνει το σήμα αυτό αντιληπτό από τον ανιχνευτή, η λέξη αυτή πρέπει να είναι ευδιάκριτη στα σήματα της πίστας. Αυτό το πετυχαίνουμε χρωματίζοντας τη λέξη με άσπρο χρώμα, και οτιδήποτε άλλο, όχι τόσο σημαντικό, με διαφορετικό χρώμα.

10. Οι διαστάσεις που ορίζουμε για την κάμερα (160 * 120) ισχύουν για τη συγκεκριμένη κάμερα. Δοκιμάσαμε, για παράδειγμα, σε διαφορετική κάμερα από τη δική μας και όταν τρέξαμε το πρόγραμμα οι διαστάσεις της πήραν διαφορετική τιμή. Αυτό είχε σαν αποτέλεσμα το κέντρο του παραθύρου της κάμερας να είναι διαφορετικό από αυτό που έχουμε ορίσει (με τιμή 80) και κατά συνέπεια το όχημα να χάνει την πορεία του (χάνει τη γραμμή).

11. Ο λόγος που επιλέγουμε να χρησιμοποιήσουμε τα 1800 από τα 1950 θετικά δείγματα είναι για να προτρέψουμε τον ταξινομητή να απορρίψει εύκολα τις περιοχές που δεν περιέχουν το προς ανίχνευση αντικείμενο και έτσι να μειωθεί το σφάλμα ανίχνευσης και να αυξηθεί η υπολογιστική ικανότητα. Με δοκιμές (για 1949 και 1900) παρατηρήσαμε ότι όντως αυτό ισχύει, αλλιώς δεν προχωράει η διαδικασία με τα στάδια.

```
sampleHeight: 22
boostType: GAB
minHitRate: 0.995
maxFalseAlarmRate: 0.5
weightTrimRate: 0.95
maxDepth: 1
maxWeakCount: 100
mode: BASIC

===== TRAINING 0-stage =====
<BEGIN
POS count : consumed   1949 : 1949
NEG count : acceptanceRatio    1950 : 1
Precalculation time: 9
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3| 0.998974| 0.337436|
+----+---------+---------+
END>
Training until now has taken 0 days 0 hours 1 minutes 23 seconds.

===== TRAINING 1-stage =====
<BEGIN
POpenCV Error: Bad argument (Can not get new positive sample. The most possible reason is insufficient count of samples in given vec-
file.
) in get, file /home/manos/opencv-3.1.0/apps/traincascade/imagestorage.cpp, line 157
terminate called after throwing an instance of 'cv::Exception'
  what():  /home/manos/opencv-3.1.0/apps/traincascade/imagestorage.cpp:157: error: (-5) Can not get new positive sample. The most pos
sible reason is insufficient count of samples in given vec-file.
 in function get

Aborted (core dumped)1948
```

Εικόνα 8.12: Δοκιμή με 1949 θετικά δείγματα

```
===== TRAINING 7-stage =====
<BEGIN
POS count : consumed   1900 : 1947
NEG count : acceptanceRatio    1950 : 0.00243868
Precalculation time: 10
+----+---------+---------+
|  N |    HR   |    FA   |
+----+---------+---------+
|   1|        1|        1|
+----+---------+---------+
|   2|        1|        1|
+----+---------+---------+
|   3| 0.998947| 0.857436|
+----+---------+---------+
|   4| 0.997368| 0.794359|
+----+---------+---------+
|   5| 0.996316| 0.720513|
+----+---------+---------+
|   6| 0.995263| 0.639487|
+----+---------+---------+
|   7| 0.995263| 0.607179|
+----+---------+---------+
|   8| 0.995263|  0.49641|
+----+---------+---------+
END>
Training until now has taken 0 days 0 hours 18 minutes 50 seconds.

===== TRAINING 8-stage =====
<BEGIN
POS current samplOpenCV Error: Bad argument (Can not get new positive sample. The most possible reason is insufficient count of sampl
es in given vec-file.
) in get, file /home/manos/opencv-3.1.0/apps/traincascade/imagestorage.cpp, line 157
terminate called after throwing an instance of 'cv::Exception'
  what():  /home/manos/opencv-3.1.0/apps/traincascade/imagestorage.cpp:157: error: (-5) Can not get new positive sample. The most pos
sible reason is insufficient count of samples in given vec-file.
 in function get

Aborted (core dumped)
```

Εικόνα 8.13: Δοκιμή με 1900 θετικά δείγματα

12. Αν δώσουμε στα θετικά δείγματα μικρό μέγεθος και θέσουμε μεγαλύτερο acceptanceRatio, τότε ο ταξινομητής θα χρειαστεί περισσότερα στάδια να ολοκληρωθεί. Παρατηρούμε ότι η τιμή του μειώνεται κατά πολύ κάθε φορά στο κάθε στάδιο μέχρι να φτάσει σε ένα αποτέλεσμα το οποίο είναι αρκετά καλό, ώστε να έχει επιτυχία η διαδικασία του αλγορίθμου. Αν όμως φτάσει αρκετά κοντά στο 0, τότε σημαίνει ότι ο ταξινομητής έχει εκπαιδευτεί παραπάνω από όσο πρέπει με αποτέλεσμα να χάσει την ικανότητα ανίχνευσης.

13. Η δημιουργία του αδύναμου ταξινομητή δεν έγινε στο Raspberry Pi, αλλά σε υπολογιστή με δυνατό επεξεργαστή και μνήμη RAM σε περιβάλλον Ubuntu, ώστε να ολοκληρωθεί γρήγορα και σωστά η διαδικασία.

ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

Μια προτεινόμενη λύση στην παρατήρηση 4 είναι η χρήση μιας κάμερας με δυνατότητα σύλληψης περισσότερων καρέ ανά δευτερόλεπτο, ώστε να προλαβαίνει η κάμερα να λαμβάνει γρηγορότερα την εικόνα που θα βρίσκεται στο οπτικό πεδίο του οχήματος την κάθε στιγμή και να μην κολλάει η εικόνα στα προηγούμενα καρέ.

Μια πιθανή λύση στην παρατήρηση 5 είναι να βρούμε περισσότερα αρνητικά και θετικά δείγματα, ώστε να έχουμε ακόμα καλύτερα αποτελέσματα ανίχνευσης.

Για την παρατήρηση 6, θα μπορούσε μελλοντικά να υπάρχει δυνατότητα αποφυγής της αναγνώρισης των σκιών με τροποποίηση του κώδικα. Για παράδειγμα, να απορρίπτονται από την ανίχνευση τα αντικείμενα που θα εντοπίζει η κάμερα και θα έχουν μέγεθος πολύ μεγαλύτερο από 10.

Μια προτεινόμενη λύση στην 7η παρατήρηση είναι να υπάρχει μια χρονοκαθυστέρηση μετά την “αντίληψη” του σήματος από την κάμερα. Κάτι τέτοιο, όμως, δεν είναι εύκολο να επιτευχθεί, αφού η αντίληψη αυτή εξαρτάται κάθε φορά από το φωτισμό, τη θέση και τον προσανατολισμό του σήματος.

Με τον υπολογισμό περισσότερων haar χαρακτηριστικών από αυτά που ήδη έχουμε βρει, θα έχουμε ένα μεγαλύτερο πλήθος από αυτά, με αποτέλεσμα οι πληροφορίες των εικόνων να είναι περισσότερες προσφέροντας έναν πιο ισχυρό ταξινομητή και πιθανόν να λυνόταν και το θέμα της χρήσης του σήματος της δεξιάς στροφής (παρατήρηση 8).

Στην παρατήρηση 12 μια πρόταση για να μην έχουμε πρόβλημα με την ανίχνευση είναι να μειώσουμε τον αριθμό των σταδίων.

Η δημιουργία μιας πίστας με πιο ανοιχτές στροφές ίσως μείωνε τις περιπτώσεις πιθανής εκτροπής του οχήματος από την πορεία του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Raspberry Pi

[1] https://www.pcsteps.gr/52828-τι-είναι-το-raspberry-pi/

[2] https://www.cl.cam.ac.uk/projects/raspberrypi/

[3] https://www.raspberrypi.org/help/faqs/#generalDifference

[4] http://coolweb.gr/digital-photo-camera/

[5] https://el.wikipedia.org/wiki/Web_camera

[6] https://en.wikipedia.org/wiki/Bluetooth_low_energy

[7] https://el.wikipedia.org/wiki/USB

[8] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[9] http://www.e-shop.gr/mitriki-raspberry-pi-3-model-b-p-PER.911869?gclid=CIeTgY2Mzs8CFeQy0wodMpgHCA

[10] http://lifehacker.com/the-best-operating-systems-for-your-raspberry-pi-projec-1774669829

[11] http://raspberrypi.stackexchange.com/questions/40318/raspberry-pi-2-can-gpio-pins-29-40-be-used-gpio-gen-input-output-configurable-in

Putty, SSH, XMing

[12] https://en.wikipedia.org/wiki/PuTTY

[13] http://blog.e-wireless.gr/τι-είναι-το-ssh/

[14] http://www.cs.uoi.gr/~gkappes/files/tutorials/ssh.pdf

[15] http://wiki.grid.auth.gr/wiki/bin/view/Groups/ALL/WindowsXForwarding

L293D, Κινητήρες, PWM

[16] http://www.ti.com/lit/ds/symlink/l293.pdf

[17] http://users.sch.gr/asal1/material/seminaria/NEASMYRNH/efarmoges_arduino%20(2).pdf

[18] https://www.adafruit.com/product/2941

[19] https://learn.adafruit.com/adafruit-raspberry-pi-lesson-9-controlling-a-dc-motor

[20] https://eclass.teiath.gr/modules/document/file.php/NAFP100/Εργαστήριο%20Μαθήματος /Άσκηση%205.pdf

[21] http://www.rakeshmondal.info/L293D-Motor-Driver

Κίνηση οχήματος με Raspberry Pi 3 με αναγνώριση γραμμής πορείας και σημάτων οδικής κυκλοφορίας

[22] http://www.robotplatform.com/howto/L293/motor_driver_1.html

[23] https://en.wikipedia.org/wiki/Duty_cycle

OpenCV, Python, Numpy

[24] https://en.wikipedia.org/wiki/OpenCV

[25] https://el.wikipedia.org/wiki/Python

[26] http://www.pyimagesearch.com/wp-content/uploads/2014/10/computer_vision_pthon_resource_guide.pdf

Επεξεργασία εικόνας

[27] http://docs.opencv.org/2.4/modules/highgui/doc/reading_and_writing_images_and_video.html

Haar, Traincascade, Adaboost

[28] http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html

[29] http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

[30] http://docs.opencv.org/2.4.13.2/doc/user_guide/ug_traincascade.html

[31] Χρήστος Ι. Βαρυτιμίδης, “Ανίχνευση αντικειμένων και ημιαυτόματος χαρακτηρισμός εικόνων”, Διπλωματική εργασία, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο, Ιούνιος 2008

[32] Επαμεινώνδα Π. Αντωνάκου, “Οπτική Μοντελοποίηση Ανθρώπινου Προσώπου με Εφαρμογές σε Αναγνώριση”, Διπλωματική εργασία, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Σημάτων, Ελέγχου και Ρομποτικής, Εθνικό Μετσόβιο Πολυτεχνείο, Ιούλιος 2011

[33] Παναγιώτης Β. Περάκης, “Ανίχνευση Προσώπων και Σημείων Ενδιαφέροντος σε Πρόσωπα”, Διπλωματική εργασία, Σχολή Θετικών Επιστημών, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Μάιος 2008

[34] https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/