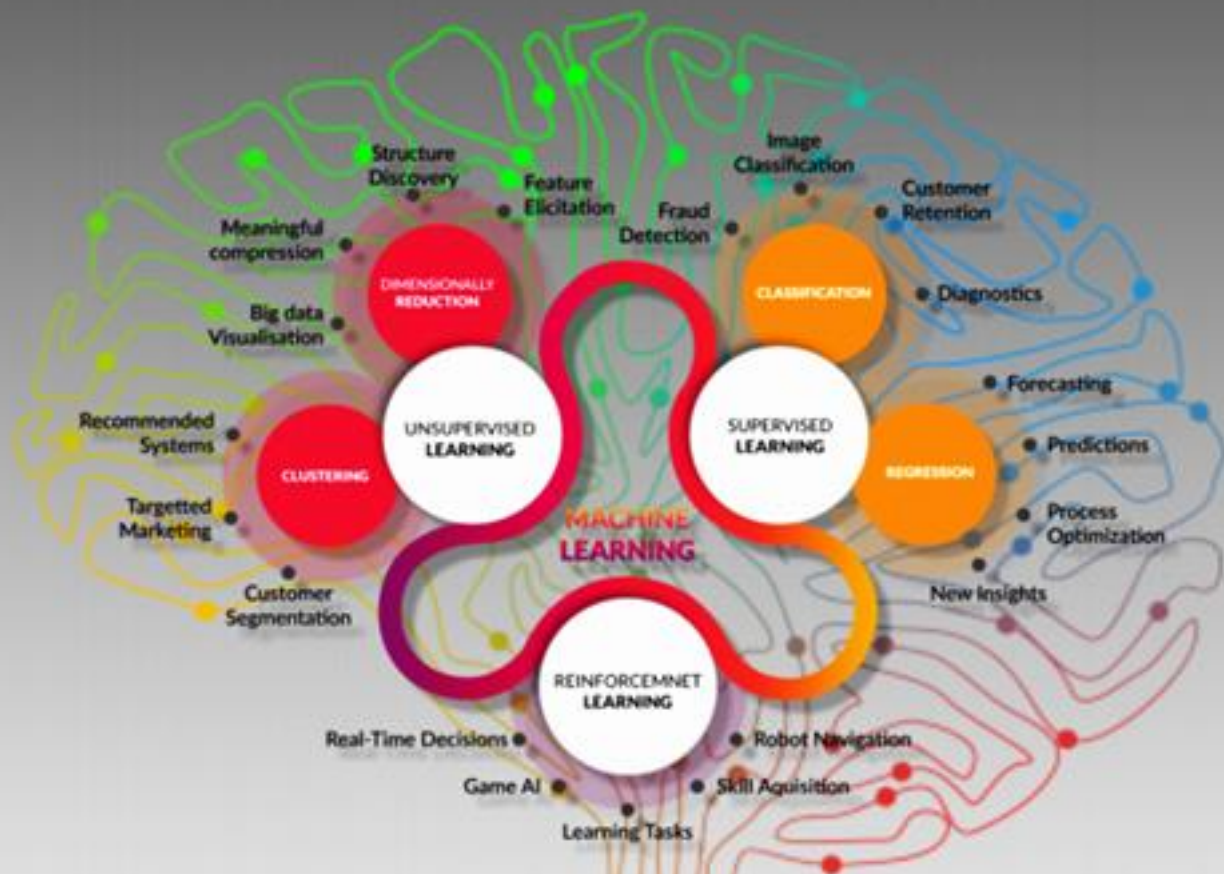


Αλέξανδρος Σπουρνιάς

Διπλωματική Εργασία

Framework Design and Development targeting accurate object identification, focusing on Door/Windows, optimized for Ambient Assisted Living Environment



TensorFlow

Keras

OpenCV

ROS





Πρόγραμμα Μεταπτυχιακών Σπουδών

“Framework Design and Development targeting accurate object identification, focusing on Door/Windows, optimized for Ambient Assisted Living Environment”

Αλέξανδρος Αθ. Σπουρνιάς Α.Μ. : 26

Εισηγητές :

Καθ. Κος Νικόλαος Βώρος Καθ. Κος Χρήστος Αντωνόπουλος

Πάτρα 2021

© Αλέξανδρος Αθ. Σπουρνιάς

© Πανεπιστήμιο Πελοποννήσου – Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

© Εργαστήριο Σχεδιασμού Ενσωματωμένων Συστημάτων & Εφαρμογών



Framework Design and Development targeting accurate object identification, focusing on Door/Windows, optimized for Ambient Assisted Living Environment platform based on open machine learning technologies

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Βώρος Νικόλαος

Αντωνόπουλος Χρήστος

Καθηγητής

Επ. Καθηγητής



Πίνακας περιεχομένων

Πρόλογος	6
Abstract	7
Εισαγωγή	8
1. Νευρωνικά Δίκτυα και Μηχανική Μάθηση	9
1.1 Νευρωνικά Δίκτυα	9
1.2 Ιστορική Αναδρομή	10
1.3 Τι είναι Μηχανική Μάθηση	10
1.4 Ιστορική Αναδρομή – Ορισμός.....	11
1.5 Στόχοι Επίλυσης Προβλημάτων	13
1.6 Κατηγορίες και Διαχωρισμοί Μηχανικής Μάθησης	13
1.6.1 Βασικές Κατηγορίες	14
1.6.1.1 Επιβλεπόμενη Μάθηση.....	14
1.6.1.2 Μη Επιβλεπόμενη Μάθηση	15
1.6.1.3 Μερικώς Επιβλεπόμενη Μάθηση.....	15
1.6.1.4 Ενισχυτική Μάθηση.....	16
1.6.2 Βασικοί Διαχωρισμοί Αλγορίθμων Μηχανικής Μάθησης.....	17
1.6.2.1 Ταξινόμηση - Classification.....	17
1.6.2.2 Παλινδρόμηση – Regression.....	18
1.6.2.3 Συσταδοποίηση - Clustering	19
1.6.2.4 Δένδρα Απόφασης.....	20
1.6.2.5 Μάθηση βασισμένη σε Στιγμιότυπα	22
2. Μηχανική όραση	23
2.1 Ορισμός	23
2.2 Χρήση.....	24
2.3 Επεξεργασία Εικόνας	25
2.3.1 Εικόνα.....	25
2.3.2 Ψηφιακή Επεξεργασία Εικόνας.....	26
2.4 Εφαρμογές και Συστήματα	27
3. Λογισμικό & Βιβλιοθήκες μηχανικής όρασης	29



2.1	Βιβλιοθήκες	29
2.2	Λογισμικό.....	31
4.	Βιβλιοθήκη OpenCV.....	35
3.1	Γενικά.....	35
3.2	Χρήση της OpenCV.....	36
3.3	Αρχιτεκτονική της OpenCV.....	37
5.	Χαρακτηριστικά Εικόνας	38
2.5	Χρώμα Εικόνων.....	39
2.6	Χρωματικοί χώροι	40
2.6.1	Ο χρωματικός χώρος RGB	41
2.6.2	Ο χρωματικός χώρος YUV.....	42
2.6.3	Ο χρωματικός χώρος YCbCr.....	43
2.6.4	Οι χρωματικοί χώροι LUV, LAB.....	43
2.6.5	Ο χρωματικός χώρος HSV.....	44
2.6.6	Οι χρωματικοί χώροι HIS & HLS.....	46
2.6.7	Χρωματικός χώρος CMY.....	48
2.6.8	Περίληπτική σύγκριση των χρωματικών χώρων	49
2.7	Σχήμα εικόνας (Shape).....	49
3.	Χαρακτηριστικά χρώματος.....	50
3.1	Ιστογράμμα Χρώματος Εικόνας	50
3.2	Φωτεινότητα (Brightness).....	51
3.3	Κορεσμός (Saturation).....	51
4.	Τεχνικές Ανάλυσης Εικόνας.....	52
4.1	Τεχνικές Τμηματοποίησης Εικόνων	52
4.1.1	Τεχνική Κατωφλίωσης Ιστογράμματος.....	53
4.1.2	Τεχνική με βάση την ομαδοποίηση.....	54
4.1.3	Τεχνικές Τεχνητών νευρωνικών δικτύων – Neural Networks.....	55
4.1.4	Αναγνώριση Ακμών	56
4.1.4.1	Μέθοδος Sobel.....	58
4.1.4.2	Μέθοδος Laplacian of Gaussian (LoG).....	58



4.1.4.3	Μέθοδος Zero-Cross.....	59
4.1.4.4	Μέθοδος Canny.....	60
4.1.5	Μεγέθυνση Περιοχής.....	61
4.1.6	Τεχνικές ασαφούς λογικής - Fuzzy Techniques	63
4.1.7	Τεχνικές βασισμένες στη φυσική – Physics based	64
4.2	Τμηματοποίηση για την ανάπτυξη λογισμικού αναγνώρισης αντικειμένων	64
5.	Robotic Operating System (ROS).....	67
5.1	Συστατικά στοιχεία του ROS	69
5.2	Μια γενική ιδέα των επιπέδων του ROS	69
5.2.1	Επίπεδο συστήματος Αρχείων.....	70
5.3	Επίπεδο γραφικής παράστασης υπολογισμού	70
5.4	Επίπεδο Κοινότητας ROS.....	71
6.	Τεχνική Εκπαίδευσης Μοντέλου με Μηχανική Μάθηση.....	72
6.1	Εγκατάσταση TensorFlow και Anaconda Frameworks	72
6.2	Συλλογή και επεξεργασία φωτογραφιών	74
6.3	Παραγωγή Δεδομένων Εκπαίδευσης	76
6.4	Δημιουργία ετικετών και παραμετροποίηση εκπαίδευσης	77
6.4.1	Ετικέτες αντικειμένων	77
6.4.2	Παραμετροποίηση εκπαίδευσης.....	78
6.5	Εκτέλεση εκπαίδευσης.....	79
6.6	Εξαγωγή Inference graph	79
	Βιβλιογραφία – Πηγές	80



Πρόλογος

Η παρούσα διπλωματική εργασία με θέμα "Σχεδίαση και ανάπτυξη συστήματος που στοχεύει στην ακριβή αναγνώριση αντικειμένων, εστιάζοντας στα αντικείμενα πόρτες / παράθυρα και βελτιστοποιημένη για την ρομποτική πλατφόρμα στα πλαίσια περιβάλλοντος υποβοηθούμενης διαβίωσης, με βάση τεχνολογίες ανοιχτού λογισμικού μηχανικής μάθησης", εκπονήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος σπουδών του τμήματος Μηχανικών Πληροφορικής ΤΕ, του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Δυτικής Ελλάδας.

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω τους καθηγητές και μέντορές μου, κ.κ. Βώρο Νικόλαο και Αντωνόπουλο Χρήστο, για την πολυτιμότερη βοήθεια και υποστήριξη που μου παρείχαν.

Λέξεις Κλειδιά: Machine Learning, TensorFlow, Keras, OpenCV, Python, Όραση Μηχανής, Αλγόριθμοι Μηχανικής Μάθησης, Επεξεργασία Ψηφιακής Εικόνας





Abstract

This diploma thesis with title "Framework Design and Development targeting accurate object identification, focusing on Door/Windows, optimized for Ambient Assisted Living Environment platform based on open machine learning technologies", written by postgraduate student Alexander Spournias as a part of the postgraduate Master's program of Engineering Department of Informatics TE, of the Technological Educational Institute of Western Greece.

At this point, I would like to thank my teachers and mentors, Dr. Voros Nikolaos and Dr. Antonopoulos Christos, for the valuable help and support they have given to me.



Εισαγωγή

Η εργασία ασχολείται με την αναγνώριση θυρών και παραθύρων εντός μιας οικίας που έχει σχεδιαστεί να παρέχει μια αυτόνομη και υποβοηθούμενη διαβίωση των ενοίκων της, μέσω της μεθοδολογίας μηχανικής μάθησης.

Η εργασία, θα προσεγγίσει και θα καλύψει θέματα που άπτονται καταρχάς της μηχανικής μάθησης, κατά δεύτερον της ρομποτικής όρασης, καθώς και της ρομποτικής πλατφόρμας που θα χρησιμοποιηθεί. Επίσης, θα γίνει ανάλυση της μεθοδολογίας εκπαίδευσης των αλγορίθμων μηχανικής μάθησης, έτσι ώστε να γίνει κατανοητός ο τρόπος λειτουργίας τους, καθώς και ο ρόλος της μηχανικής όρασης όπου συμβάλει στην καλύτερη εκπαίδευση των αλγορίθμων αυτών.

Με τον όρο Μηχανική Μάθηση, αναφερόμαστε στις μεθοδολογίες εκείνες όπου σκοπό έχουν τη δημιουργία μηχανών ικανών να μαθαίνουν μέσω των αντίστοιχων αλγορίθμων, ώστε να βελτιώνουν την απόδοσή τους σε κάποιους τομείς εργασίας που καλούνται να μετέχουν, μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας.

Ο όρος «μηχανική όραση» ή «όραση υπολογιστών» (Computer Vision), αναφέρεται στην τεχνολογία και τις μεθοδολογίες, όπου ουσιαστικά, επιτρέπουν στις μηχανές να έχουν μια τεχνητή όραση, καθώς με την αναπαράσταση μιας εικόνας με ψηφιακό τρόπο, είναι δυνατή η αλγοριθμική επεξεργασία και η εξαγωγή πληροφορίας.

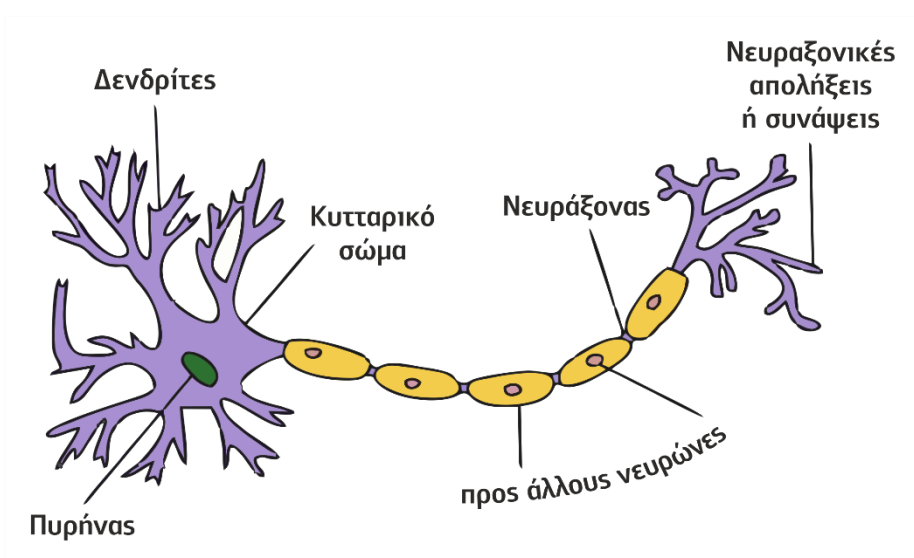
Εφαρμογές μηχανικής μάθησης σε συνδυασμό με την μηχανική όραση, μπορούν να χρησιμοποιηθούν σε πληθώρα προβλημάτων των περισσότερων επιστημών, της καθημερινής ζωής, καθώς και τις βιομηχανικής παραγωγής.

1. Νευρωνικά Δίκτυα και Μηχανική Μάθηση

1.1 Νευρωνικά Δίκτυα

Τα τεχνητά Νευρωνικά δίκτυα, έχουν εμπνευστεί από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου, με το βασικό του στοιχείο να είναι οι νευρώνες, (Εικόνα 1), αποτελώντας μια πυκνή επικοινωνιακή δομή από νευρικά κύτταρα.

Τα συνηθισμένα Τεχνητά Νευρωνικά Δίκτυα, χρησιμοποιούν αρκετά απλοποιημένα μοντέλα νευρώνων, έτσι ώστε να διατηρούνται μόνο τα αδρά χαρακτηριστικά των λεπτομερών μοντέλων που χρησιμοποιούνται στην ανθρώπινη Νευροφυσιολογία, με αποτέλεσμα να έχουν ελάχιστη σχέση αναμεταξύ τους. Παρόλα αυτά, ακόμα και αυτά τα απλά μοντέλα τεχνητών νευρωνικών δικτύων, έχουν την ικανότητα να δημιουργήσουν ιδιαίτερα ενδιαφέροντα δίκτυα, με την αίρεση όμως ότι θα τηρούνται δύο βασικά χαρακτηριστικά. Το πρώτο χαρακτηριστικό που πρέπει να τηρεί ένα τεχνητό νευρωνικό δίκτυο, αφορά τους νευρώνες του όπου πρέπει να έχουν “Πλαστικότητα”, δηλαδή οι παράμετροί τους να είναι ρυθμιζόμενες, ώστε να επιτυγχάνεται ευκολότερη μάθηση. Το δεύτερο χαρακτηριστικό, αφορά το πλήθος των νευρώνων όπου πρέπει να είναι μεγάλο, έτσι ώστε να επιτυγχάνεται “Παραλληλοποίηση” της επεξεργασίας, καθώς και καλύτερη “Κατανομή” της πληροφορίας.



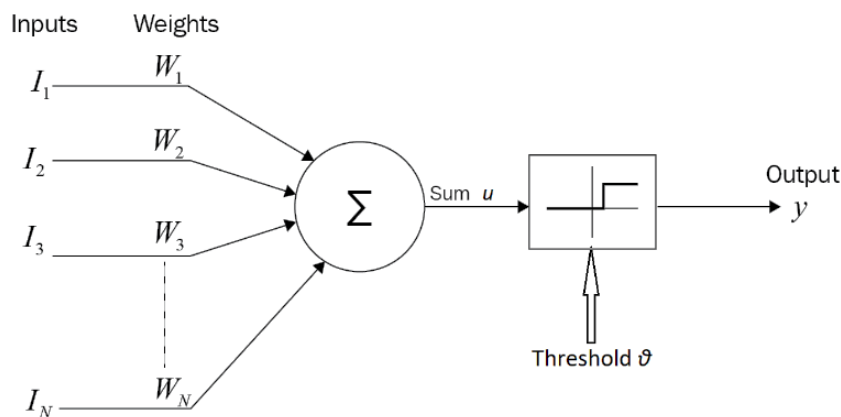
Εικόνα 1. Βιολογικός Νευρώνας

1.2 Ιστορική Αναδρομή

Η μελέτη των βιολογικών νευρωνικών δικτύων και της μαθηματικής μοντελοποίησής τους, ξεκίνησε από τους Αμερικανούς ερευνητές McCulloch και Pitts το 1940 [1], περιγράφοντας μαθηματικά, (Εικόνα 2), ένα απλό μοντέλο δραστηριότητας του νευρώνα. Συγκεκριμένα, η κατάσταση του νευρώνα περιγράφεται δυαδικά ως εξής:

- $\gamma=0$, όπου ο νευρώνας είναι αδρανής
- $\gamma=1$, όπου ο νευρώνας είναι ενεργοποιημένος.

Οι συνάψεις τώρα του νευρώνα, περιγράφονται από τα “συναπτικά βάρη” (Synaptic weights) w_i , όπου είναι πραγματικοί αριθμοί, έχοντας δύο καταστάσεις, τους θετικούς πραγματικούς για τις ενισχυτικές συνάψεις και τους αρνητικούς πραγματικούς για τις ανασταλτικές συνάψεις.



Εικόνα 2. Μοντέλο McCulloch and Pitts

Έτσι, εάν x_1, x_2, \dots, x_n είναι οι εισοδοι του νευρώνα, τότε το άθροισμα u του φορτίου που δέχεται ο νευρώνας, δίνεται από τον μαθηματικό τύπο:

$$u = \sum_{i=1}^n w_i X_i$$

Από την εικόνα 2 και από τον μαθηματικό τύπο, καταλαβαίνουμε ότι αν το άθροισμα u είναι μεγαλύτερο από το κατώφλι θ τότε ισχύει $\gamma=1$, ενώ σε αντίθετη περίπτωση, $\gamma=0$.

1.3 Τι είναι Μηχανική Μάθηση

Σύμφωνα με τον Arthur L. Samuel η μηχανική μάθηση είναι ο κλάδος της επιστήμης των υπολογιστών που δίνει την δυνατότητα σε υπολογιστικά συστήματα να ‘μαθαίνουν’, δηλαδή να βελτιώνουν προοδευτικά την επίδοσή τους σε μια διεργασία) χωρίς να έχουν



ρητά προγραμματιστεί. Αυτό επιτυγχάνεται μέσω της μελέτης και κατασκευής αλγορίθμων που μπορούν να χτίσουν μοντέλα βασισμένα σε δεδομένα και να κάνουν προβλέψεις ή να πάρουν αποφάσεις βασισμένοι σε αυτά τα δεδομένα. Η λογική τους βασίζεται στην προσομοίωση των νευρώνων και των νευρωνικών δικτύων.

1.4 Ιστορική Αναδρομή – Ορισμός

Παρά το γεγονός πως τα τελευταία χρόνια, η Μηχανική Μάθηση αποτελεί ένα ιδιαίτερα ενδιαφέρον επιστημονικό ερευνητικό πεδίο έχοντας καταφέρει να τραβήξει το ενδιαφέρον πολλών επιστημόνων και επιχειρηματιών, η ουσιαστική δημιουργία αυτού του τομέα, έγινε περίπου στο 1700, και στηρίχθηκε στο θεώρημά των πιθανοτήτων του Bayes [2]. Τα επόμενα αξιοσημείωτα επιστημονικά επιτεύγματα που αξιοποιούνται πλέον από τις εφαρμογές Μηχανικής μάθησης, ήταν η τεχνική της Παλινδρόμησης (Regression) το 1920 [3], καθώς και τα Νευρωνικά Δίκτυα και οι ταξινομητές όπως ο ταξινομητής πλησιέστερου-ων γείτονα-ων, (K Nearest Neighbors) [4], όπου έκαναν την εμφάνισή τους περί τα μέσα του 20ου αιώνα. Ακολούθως, τις αρχές του 1960 ξεκινάει και πρακτικά η άνθηση της Μηχανικής Μάθησης, όπου εξαιτίας πλέον της τεράστιας ύπαρξης όγκου δεδομένων, καθώς και με τη ραγδαία ανάπτυξη της επιστήμης των υπολογιστών - αλγορίθμων που εμφανίζονται τη σημερινή εποχή, μπορούμε να πούμε, πως έφτασε στο αποκορύφωμά της. Οι πιο σημαντικές μεθοδολογίες, ήταν τα Δένδρα Αποφάσεων (Decision Tree) [5], η Συσταδοποίηση (Clustering) και οι Γενετικοί Αλγόριθμοι (Genetic Algorithms) έως τέλος του 1960, καθώς και οι Μηχανές Διανυσματικής Στήριξης (Support Vector Machines) γύρω στο 1990.

Η Μηχανική Μάθηση, (machine learning), είναι ένα υποπεδίο της τεχνητής νοημοσύνης, όπου ασχολείται με τη σχεδίαση αυτόματων διαδικασιών ικανών να μαθαίνουν μέσω δοσμένων παραδειγμάτων. Η μάθηση μέσω παραδειγμάτων, αφορά στην παραγωγή μιας λογικής περιγραφής των αναγκαίων και ικανών συνθηκών που αντιστοιχούν σε μια κλάση αντικειμένων.

Έτσι η μηχανική μάθηση, έχει ως σκοπό τη δημιουργία αλγοριθμικών διαδικασιών, ικανών να μαθαίνουν, δηλαδή ικανών να βελτιώνουν την απόδοση τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας. Αν και απέχουμε πάρα πολύ από τη



μίμηση της μάθησης που επιτυγχάνει ο άνθρωπος, έχουν αναπτυχθεί αλγόριθμοι για συγκεκριμένες περιοχές μάθησης, οι οποίοι έχουν επιτρέψει την εμφάνιση εμπορικών εφαρμογών με σημαντική επιτυχία. Επίσης, Τα τελευταία χρόνια, οι εφαρμογές της μηχανικής μάθησης πληθαίνουν ολοένα και περισσότερο, καθώς μέσω των τεχνικών όπως οι γενετικοί αλγόριθμοι, τα νευρωνικά δίκτυα, η ασαφής λογική και άλλες τεχνικές από τους τομείς της επεξεργασίας σημάτων και από την αναγνώριση προτύπων, αντιμετωπίζουν με μεγάλη επιτυχία τη δύσκολη φύση αρκετών προβλημάτων.

Αναφορικά με τον ορισμό του όρου Μηχανική Μάθηση, αρκετοί ερευνητές στο πεδίο αυτό, έχουν εισάγει τον δικό τους ορισμό. Μερικοί από αυτούς, είναι ο **Carbonell**, όπου το 1987, διατύπωσε τον ορισμό ότι “η μηχανική μάθηση είναι η μελέτη υπολογιστικών μεθόδων για την απόκτηση νέας γνώσης, νέων δεξιοτήτων και νέων τρόπων οργάνωσης της υπάρχουσας γνώση”. Μεταγενέστερα, το 1997 έρχεται ο **Mitchell**, όπου διατυπώνει τον ορισμό ότι “Ένα πρόγραμμα υπολογιστή θεωρείται ότι μαθαίνει από την εμπειρία E σε σχέση με μια κατηγορία εργασιών T και μια μετρική απόδοσης P , αν απόδοσή του σε εργασίες της T , όπως μετρούνται από την P , βελτιώνονται με την εμπειρία E ”. Ακολούθως, το 2000, οι **Witten & Frank**, όρισαν ότι “μια μηχανή μαθαίνει κάτι όταν αλλάζει η συμπεριφορά της κατά τέτοιο τρόπο, ώστε να αποδίδει καλύτερα στο μέλλον”

Εφαρμογές για την ανίχνευση θυρών – παραθύρων σε εσωτερικό περιβάλλον, έχουν προσεγγισθεί κατά το παρελθόν, από αρκετούς ερευνητές και πανεπιστημιακά ιδρύματα. Μερικές από αυτές, ήταν των **Kragic et al** το 2002 [6], όπου προσπαθήσαν να ανιχνεύσουν την πόρτα μέσα σε μια φωτογραφία και μέσω της ανίχνευσης τετράγωνων σχημάτων, των **Munoz – Salinas et al** [7] το 2005 και βασιζόμενοι στον αλγόριθμο Canny Edge Detector και Hough Transform, καθώς και άλλων ερευνητών με παρόμοιες τεχνικές που όμως δεν ανήκουν στο πεδίο της Μηχανικής Μάθησης. Μία τεχνική ανίχνευσης θυρών και παραθύρων που δεν είναι διαδεδομένη, είναι η ανίχνευση μέσω αλγορίθμων Μηχανικής Μάθησης και νευρωνικών δικτύων, κάτι που θα αναλύσουμε σε επόμενο κεφάλαιο.

1.5 Στόχοι Επίλυσης Προβλημάτων

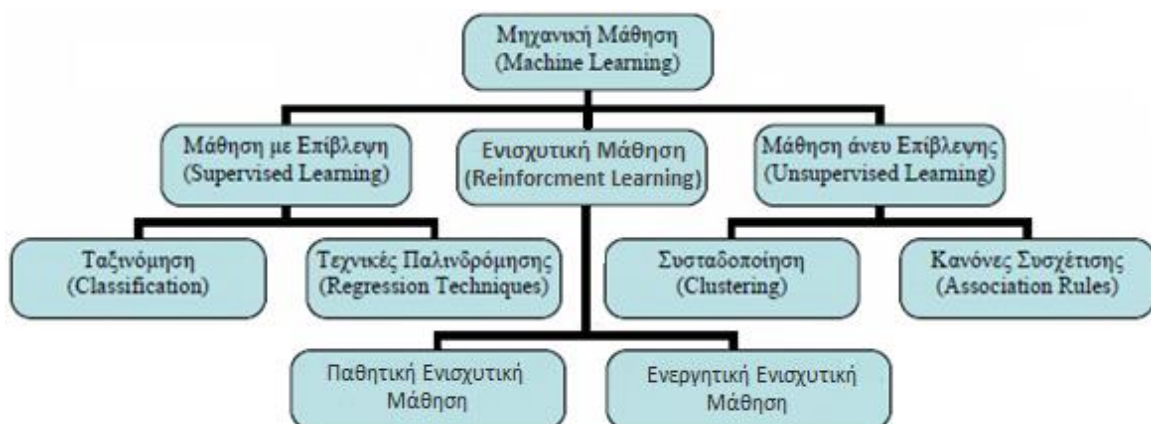
Οι δύο στόχοι των προβλημάτων με τα οποία ασχολείται η μηχανική μάθηση είναι η πρόβλεψη και η ανακάλυψη νέας γνώσης.

Η πρόβλεψη, εμπλέκει κάποιες μεταβλητές ή κάποια πεδία δεδομένων, έτσι ώστε να προβλεφθούν άγνωστες ή μελλοντικές τιμές ή και άλλες μεταβλητές ενδιαφέροντος. Πάνω στη πρόβλεψη στηρίζονται εφαρμογές σε διάφορα πεδία, όπως αυτό της ιατρικής για πρόβλεψη καρκίνου, για χρηματοοικονομικές προβλέψεις, στην μετεωρολογία και σε πολλές άλλες εφαρμογές.

Από την άλλη μεριά, η ανακάλυψη νέας γνώσης από δεδομένα, είναι μία σύνθετη διαδικασία για τον προσδιορισμό έγκυρων, νέων, χρήσιμων και κατανοητών σχέσεων-προτύπων σε δεδομένα. Αποτελεί μια σημαντική εφαρμογή σε πραγματικές συνθήκες και σε μεγάλη κλίμακα των ερευνητικών αποτελεσμάτων της Στατιστικής, των Βάσεων δεδομένων και της Μηχανικής μάθησης.

1.6 Κατηγορίες και Διαχωρισμοί Μηχανικής Μάθησης

Τα είδη μηχανικής μάθησης (Εικόνα 1), σύμφωνα με την ανατροφοδότηση που παρέχεται στο σύστημα, διακρίνονται σε τέσσερις βασικές κατηγορίες, όπου είναι η Επιβλεπόμενη, η Μερικώς Επιβλεπόμενη, η Μη Επιβλεπόμενη, καθώς και η Ενισχυτική Μηχανική Μάθηση. Επίσης, υπάρχουν τέσσερις βασικοί διαχωρισμοί Μηχανικής μάθησης, όπου αφορούν μόνο την έξοδο του συστήματος. Αυτοί είναι, η Ταξινόμηση, η Παλινδρόμηση, η Συσταδοποίηση, καθώς και οι Κανόνες Συσχέτισης.



Εικόνα 1.

1.6.1 Βασικές Κατηγορίες

1.6.1.1 Επιβλεπόμενη Μάθηση

Στην επιβλεπόμενη μάθηση (Supervised learning), το σύνολο δεδομένων αποτελείται από ένα πεπερασμένο αριθμό παραδειγμάτων (Instance), όπου στη δική μας περίπτωση είναι οι φωτογραφίες, (Εικόνα 2). Το κάθε παράδειγμα, αναπαρίσταται από ένα n - διάστατο διάνυσμα (που προκύπτει από την εξαγωγή των χαρακτηριστικών κατά το annotation), του οποίου τα στοιχεία ονομάζονται χαρακτηριστικά (attributes). Επιπλέον, σε κάθε παράδειγμα, αντιστοιχεί μία κλάση (class). Το σύνολο των παραδειγμάτων που χρησιμοποιείται για την εκπαίδευση του αλγόριθμου ονομάζεται σύνολο εκπαίδευσης (training set). Ο αλγόριθμος κατασκευάζει ένα μοντέλο που στην ουσία είναι μια συνάρτηση, όπου κατατάσσει κάθε παράδειγμα σε μια κλάση. Σκοπός του συγκεκριμένου μοντέλου είναι, όταν του δοθεί ως είσοδος ένα καινούριο παράδειγμα του οποίου δεν γνωρίζει εκ των προτέρων την κλάση, να προβλέψει (να κατηγοριοποιήσει) σωστά την κλάση στην οποία ανήκει.



Εικόνα 2. Επιβλεπόμενη Μάθηση

Για να εξετάσουμε αν το μοντέλο πραγματοποιεί σωστές προβλέψεις, χρησιμοποιούμε ένα τμήμα παραδειγμάτων για έλεγχο (test set), το οποίο αποτελείται από παραδείγματα τα οποία δεν χρησιμοποιήθηκαν στο σύνολο εκπαίδευσης. Το σύνολο παραδειγμάτων ελέγχου δίνεται ως είσοδος στο μοντέλο, το οποίο σε κάθε φωτογραφία του συνόλου ελέγχου, αντιστοιχίζει μία από τις οριζόμενες κλάσεις. Το ποσοστό των σωστά κατηγοριοποιημένων παραδειγμάτων του συνόλου ελέγχου, είναι και το ποσοστό επιτυχίας των προβλέψεων του μοντέλου.

Η επιβλεπόμενη Μάθηση, χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης, πρόγνωσης, καθώς και διερμηνείας.

1.6.1.2 Μη Επιβλεπόμενη Μάθηση

Στις μη επιβλεπόμενες τεχνικές μάθησης (Unsupervised learning), (Εικόνα 3), τα παραδείγματα, δεν περιέχουν κάποια πληροφορία για το αν ανήκουν σε κάποια κατηγορία, αλλά ούτε και γνωρίζουμε ποιες είναι οι κατηγορίες στις οποίες μπορεί να χωρίζονται. Έτσι, οι αλγόριθμοι μη επιβλεπόμενης μάθησης, προσπαθούν να βρουν λανθάνουσα πληροφορία, τέτοια που θα καταδείξει την ύπαρξη διαφορετικών ομάδων. Η Μη-Επιβλεπόμενη Μάθηση, χρησιμοποιείται κατά κόρον από αλγόριθμους ομαδοποίησης, καθώς και στην ανάλυση συσχετισμών. Επίσης, χρησιμοποιείται στην εκπαίδευση πρακτόρων μέσω ανταμοιβών και τιμωριών κάτι που θα αναλύσουμε σε επόμενη ενότητα.



Εικόνα 3. Μη επιβλεπόμενη Μάθηση

1.6.1.3 Μερικώς Επιβλεπόμενη Μάθηση

Η μερικώς επιβλεπόμενη μάθηση (Semi Supervised learning), (Εικόνα 4), είναι ένας συνδυασμός των προηγούμενων δύο κατηγοριών, όπου η μάθηση γίνεται με ένα συνδυασμό προ-κατηγοριοποιημένων παραδειγμάτων, καθώς και παραδειγμάτων χωρίς τέτοια πληροφορία. Στηρίζονται στην ιδέα ότι τα προ-κατηγοριοποιημένα δεδομένα, είναι πολλές φορές δύσκολο να βρεθούν. Το πλεονέκτημα της, είναι πως εκμεταλλεύεται τα δυνατά σημεία των δύο άλλων τεχνικών. Πολλές φορές τα αποτελέσματα που δίνουν είναι καλύτερα τόσο από τις τεχνικές επιβλεπόμενης όσο και από τις τεχνικές μη επιβλεπόμενης μάθησης.

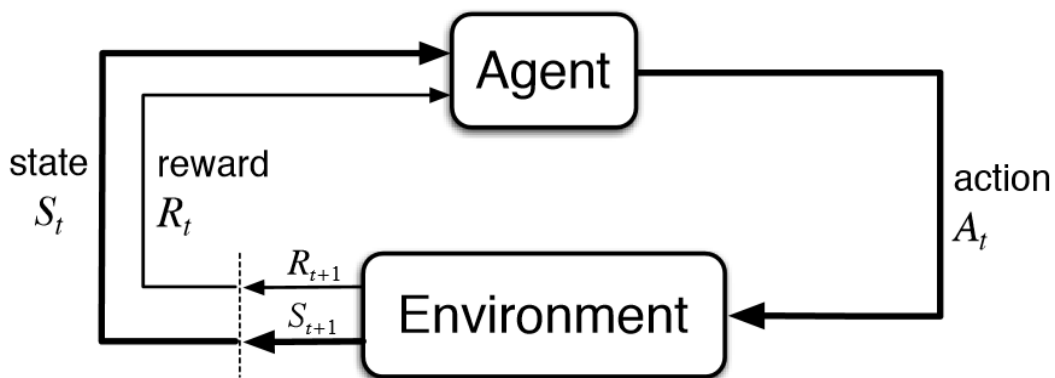


Εικόνα 4. Μερικώς Επιβλεπόμενη Μάθηση

1.6.1.4 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση (Reinforcement Learning), (Εικόνα 5), αποτελεί μία διαδικασία μάθησης η οποία είναι εμπνευσμένη από τις θεωρίες της ψυχολογίας, όπου εξηγούν με ποιο τρόπο τα έμβια όντα μαθαίνουν διάφορες συμπεριφορές. Πιο συγκεκριμένα, η ενισχυτική μάθηση έχει να κάνει με το πώς ένα υπολογιστικό σύστημα (ή πράκτορας) μπορεί να μάθει μια συγκεκριμένη συμπεριφορά μέσω της αλληλεπίδρασής του με το περιβάλλον στο οποίο ενεργεί. Σκοπός του πράκτορα, είναι να μεγιστοποιήσει ένα σήμα ενίσχυσης που ονομάζεται ανταμοιβή (reward), το οποίο λαμβάνεται από το περιβάλλον ως αποτέλεσμα των ενεργειών του. Ο πράκτορας δεν λαμβάνει κάποια άλλη απόκριση από το περιβάλλον, εκτός από τις ανταμοιβές, που σημαίνει ότι θα πρέπει να μάθει πως να συμπεριφέρεται, με την περιορισμένη πληροφορία που του παρέχεται.

Η ενισχυτική μάθηση, διαφέρει από τη μάθηση με επίβλεψη όπου υπάρχει κάποιος εξωτερικός επιβλέπων ο οποίος παρέχει για κάθε ενέργεια του πράκτορα την επιθυμητή ενέργεια. Τα προβλήματα ενισχυτικής μάθησης είναι δυσκολότερα καθώς η μόνη πληροφορία την οποία λαμβάνει ο πράκτορας είναι μια αριθμητική ανταμοιβή.



Εικόνα 5. Ενισχυτική μάθηση

Η Ενισχυτική Μάθηση, χωρίζεται σε δύο υποκατηγορίες που βασίζονται στην εκάστοτε πολιτική που ακολουθεί ένας πράκτορας. Η μία υποκατηγορία είναι η Παθητική Μάθηση (passive learning), όπου αναφέρεται στη μάθηση χρησιμότητας καταστάσεων, ή ζευγών καταστάσεων-ενεργειών, συμπεριλαμβανομένων και της μάθησης μοντέλων περιβάλλοντος. Η δεύτερη υποκατηγορία, είναι η Ενεργητική Μάθηση (active learning), όπου αναφέρεται στη εξερεύνηση (exploration), δηλαδή στη διαδικασία όπου οι



πράκτορες πρέπει να μάθουν τι κάνουν, με απώτερο σκοπό να μάθουν, πώς να συμπεριφέρονται μέσα σε αυτό.

1.6.2 Βασικοί Διαχωρισμοί Αλγορίθμων Μηχανικής Μάθησης

Στη μηχανική μάθηση, συναντάμε μια μεγάλη ποικιλία από αλγόριθμους, οι οποίοι μπορούν να χωριστούν σε κατηγορίες, με βασικό κριτήριο τα είδη των προβλημάτων που χειρίζονται. Ο κάθε αλγόριθμος βασίζεται σε ξεχωριστά ποσοτικά μέτρα, έτσι ώστε να αναλύει διαφορετικά το σύνολο δεδομένων που δέχεται και να κατασκευάζονται διαφορετικά μοντέλα ταξινόμησης.

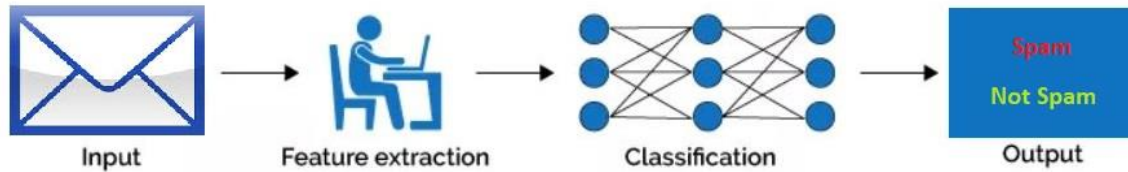
Στόχος της έννοιας της μάθησης, είναι να “μάθει” ο αλγόριθμος μέσα από μια συνάρτηση και να εκφράσει τα δεδομένα ως ένα μοντέλο. Τα δεδομένα μάθησης, όπως προ-είπαμε αποτελούνται από τη μάθηση με επίβλεψη, από τη μάθηση με μερική επίβλεψη, καθώς και από τη μάθηση χωρίς επίβλεψη. Στην μάθηση με επίβλεψη, διαπιστώνονται δύο είδη προβλημάτων, με το πρώτο να είναι η ταξινόμηση, στο οποίο δημιουργούνται μοντέλα πρόβλεψης διακριτών κλάσεων. Στο δεύτερο πρόβλημα το οποίο είναι η παλινδρόμηση, δημιουργούνται μοντέλα πρόβλεψης αριθμητικών τιμών. Παρακάτω αναλύονται οι βασικοί αλγόριθμοι μάθησης.

1.6.2.1 Ταξινόμηση - Classification

Στη μηχανική μάθηση, με τον όρο ταξινόμηση, αναφερόμαστε στο πρόβλημα του προσδιορισμού που δείχνει σε ποιο σύνολο κλάσεων ανήκει μια νέα παρατήρηση, με βάση ένα σετ εκπαίδευσης των παραδειγμάτων που περιέχει τις παρατηρήσεις, των οποίων όμως η κατηγορία κλάσης είναι γνωστή. Ένα παράδειγμα, θα ήταν η ταξινόμηση ηλεκτρονικού ταχυδρομείου σε "Spam" ή "Not-Spam" κλάσεις (Εικόνα 6), ή μια ταξινόμηση δοθέντων εικόνων που απεικονίζουν πόρτες και παράθυρα, διαχωρίζοντας αυτές σε δύο κλάσεις τις α) open και β) closed. Καταλαβαίνουμε λοιπόν, ότι η ταξινόμηση είναι στην ουσία ένα παράδειγμα αναγνώρισης μοτίβων.

Ένας αλγόριθμος που υλοποιεί την ταξινόμηση, ειδικά σε μια συγκεκριμένη εφαρμογή, είναι γνωστός ως ταξινομητής. Επίσης, μερικές φορές ο όρος «ταξινομητής», αναφέρεται στην μαθηματική συνάρτηση που υλοποιείται από έναν αλγόριθμο ταξινόμησης, όπου χαρτογραφεί την εισαγωγή δεδομένων σε μια κατηγορία.

Η ταξινόμηση συνήθως προσεγγίζεται με μεθόδους επιβλεπόμενης μηχανικής μάθησης.

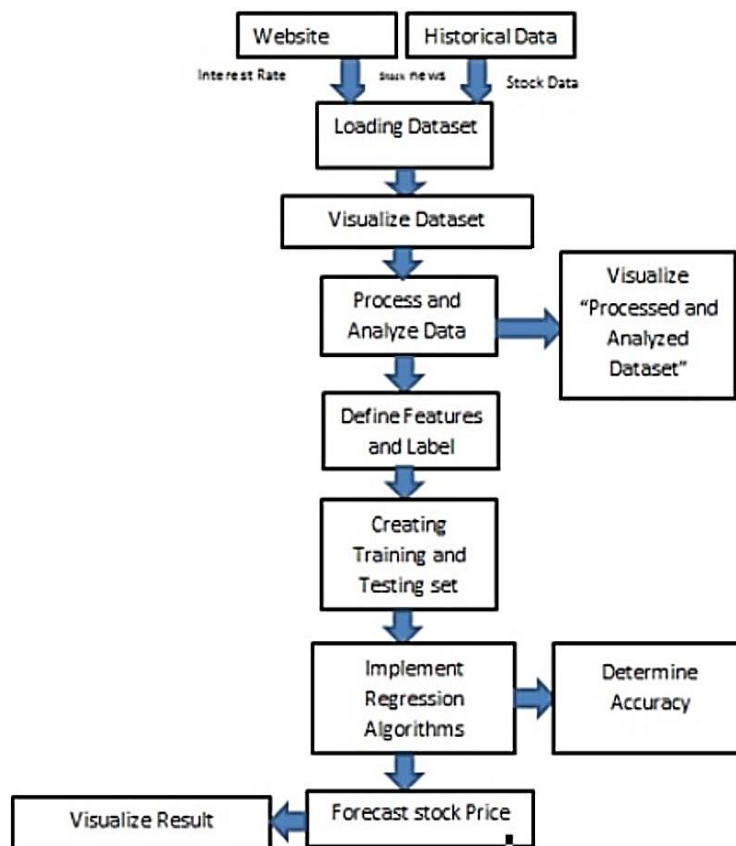


Εικόνα 6. Παράδειγμα ταξινομητή

1.6.2.2 Παλινδρόμηση – Regression

Η Παλινδρόμηση, είναι βασικά μια στατιστική προσέγγιση ώστε να μπορεί να βρεθεί η σχέση μεταξύ μεταβλητών. Στη μηχανική μάθηση, χρησιμοποιείται για την πρόβλεψη του αποτελέσματος ενός συμβάντος, που βασίζεται στη σχέση μεταβλητών που λαμβάνονται από το σύνολο δεδομένων. Η γραμμική παλινδρόμηση, είναι ένας τύπος παλινδρόμησης όπου χρησιμοποιείται στη Μηχανική Μάθηση. Κάθε μοντέλο γραμμικής παλινδρόμησης εκπροσωπείται ως:

$$\text{Έξοδος} = \text{Συντελεστής 1} + \text{Συντελεστής 2} * \text{την Είσοδο}$$

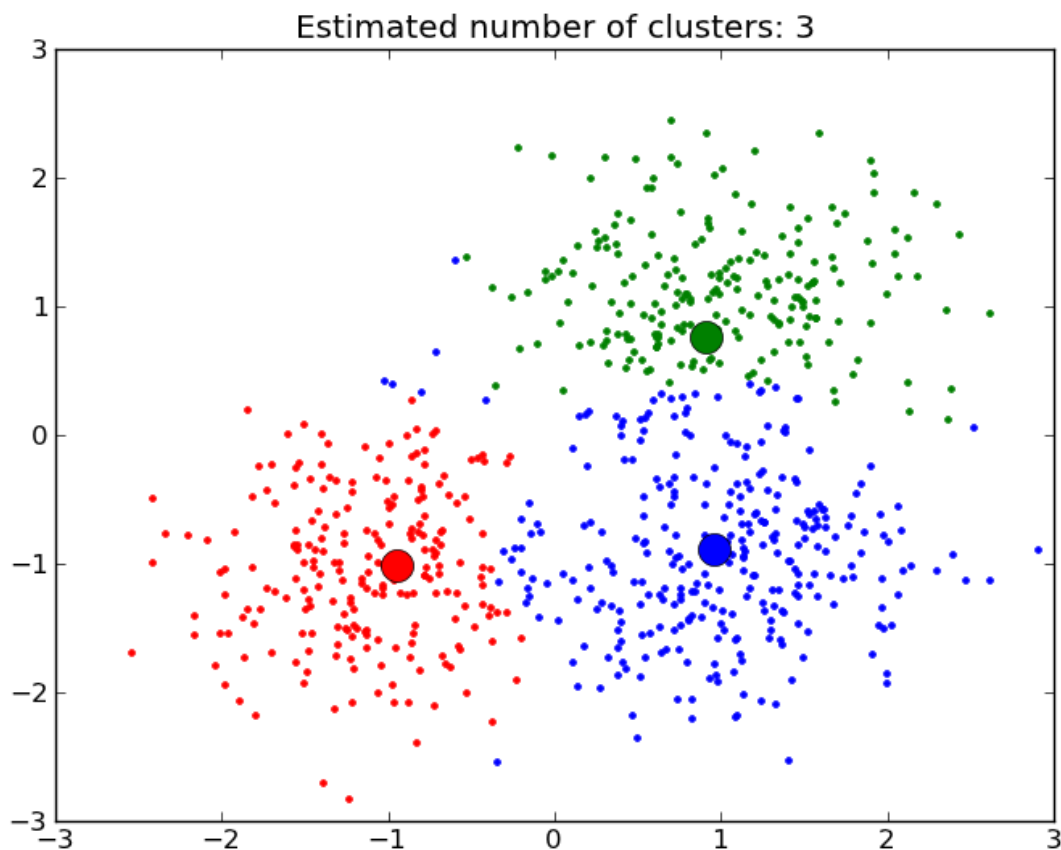


Εικόνα 7. Παράδειγμα υλοποίησης Παλινδρόμησης

1.6.2.3 Συσταδοποίηση - Clustering

Η συσταδοποίηση ή ομαδοποίηση (clustering), (Εικόνα 8), είναι μια από τις πιο δημοφιλείς μεθόδους μάθησης χωρίς επίβλεψη. Στόχος του αλγορίθμου μάθησης, είναι να καθορίσει συνεκτικά υποσύνολα, (συστάδες, clusters), των παραδειγμάτων εκπαίδευσης. Το πλήθος των συστάδων, είτε δίνεται εκ των προτέρων ως μέρος της γνώσης που προϋπάρχει, είτε καθορίζεται από τον αλγόριθμο μάθησης. Έτσι, ο αλγόριθμος πρέπει να καθορίζει ένα σχετικά μικρό πλήθος συνεκτικών συστάδων, δηλ. υποσυνόλων παρόμοιων παραδειγμάτων. Η επιλογή του μέτρου ομοιότητας, (απόστασης), είναι το πιο σημαντικό κομμάτι της προ-υπάρχουσας γνώσης και υψίστης σημασίας, για μια επιτυχή και ουσιαστική ομαδοποίηση.

Κάποιες εφαρμογές της συσταδοποίησης, είναι η κατάτμηση του αγοραστικού κοινού μιας εταιρίας, ή η ομαδοποίηση μαθητών οι φοιτητών σε ευρύτερες ομάδες ατόμων με παρόμοιες διανοητικές και μαθησιακές δυνατότητες ή δυσκολίες.



Εικόνα 8. Συσταδοποίηση



1.6.2.4 Δένδρα Απόφασης

Τα δέντρα ταξινόμησης, αποτελούν μια ευρέως χρησιμοποιούμενη μέθοδος μηχανικής μάθησης. Η μέθοδος αυτή παίρνει ως είσοδο ένα διάνυσμα τιμών σε κάποιες ιδιότητες και επιστρέφει μια έξοδο. Αυτή η έξοδος μπορεί να είναι διακριτή, οπότε ορίζεται ένα πρόβλημα ταξινόμησης, ενώ αν η έξοδος αυτή είναι συνεχής έχουμε ένα πρόβλημα παλινδρόμησης.

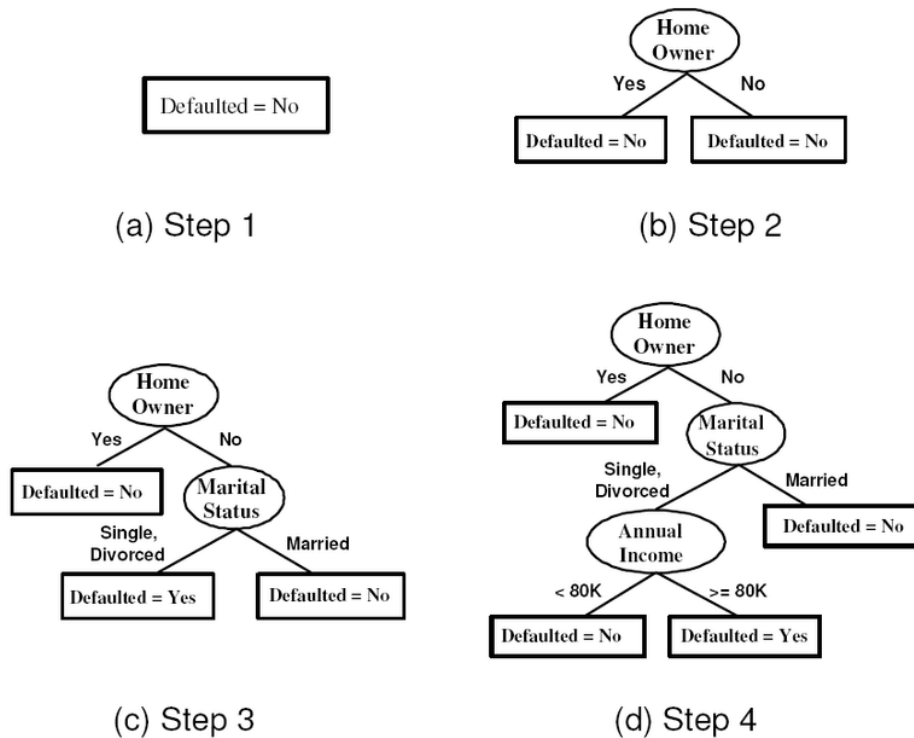
Τα δέντρα ταξινόμησης, χρησιμοποιούνται για να προβλέψουν την τιμή της μεταβλητής που μοντελοποιούν, με βάση τις τιμές των θεωρούμενων ανεξάρτητων χαρακτηριστικών. Ένα δέντρο απόφασης, αναπαριστά μια διαδικασία λήψης απόφασης, όπου για κάθε πιθανό σημείο ή κατάσταση, έχουμε ένα κόμβο, ενώ για κάθε επιλογή που μπορεί να γίνει σε ένα σημείο απόφασης αναπαρίσταται ένας "κόμβος - παιδί". Κάθε κόμβος τώρα, ορίζει μια συνθήκη ελέγχου της τιμής κάποιου χαρακτηριστικού των περιπτώσεων, ενώ κάθε κλαδί που φεύγει από ένα κόμβο, αντιστοιχεί σε μία διαφορετική διακριτή τιμή του χαρακτηριστικού που σχετίζεται με τον κόμβο. Στα κλαδιά τέλος, καταλήγουν οι τελικοί κόμβοι που ανήκουν σε ένα μόνο σύνολο, όπου και είναι οι τελικές αποφάσεις ή ενέργειες που θα γίνουν.

Για να κατασκευάσουμε ένα δέντρο απόφασης, ακολουθούμε την εξής διαδικασία:

- 1) Δημιουργούμε ένα κόμβο που περιέχει όλες τις εγγραφές.
- 2) Διασπάμε τον κόμβο με βάση μια συνθήκη, έτσι ώστε να διαχωριστούν οι εγγραφές σε κάποιο από τα γνωρίσματα.
- 3) Γίνεται αναδρομική κλίση του (2) βήματος σε κάθε κόμβο μέχρι να φτάσουμε στο τελικό κόμβο, έτσι ώστε να είναι η τελική απόφαση. Όταν κάποιος κόμβος δεν έχει παραδείγματα, τότε αντιστοιχίζεται σε μία κατηγορία. Όταν σε κάποιο κόμβο, υπάρχουν θετικά και αρνητικά παραδείγματα, τότε ο κόμβος χαρακτηρίζεται διφορούμενος.
- 4) Αφού κατασκευαστεί το δένδρο αποφάσεων, μπορούν να γίνουν κάποιες βελτιστοποιήσεις με την μέθοδο κλαδέματος.

Ο αριθμός των πιθανών δένδρων απόφασης είναι εκθετικός. Πολλοί αλγόριθμοι για να κατασκευάσουν ένα δένδρο απόφασης, προσπαθούν να το υλοποιήσουν, λαμβάνοντας

μια σειρά από τοπικά βέλτιστες αποφάσεις. Μερικοί από τους γνωστούς αλγόριθμους υλοποίησης δένδρων αποφάσεων είναι ο Hunt's Algorithm [7], (Εικόνα 6), όπου ήταν και η βάση αρκετών υπάρχοντων αλγορίθμων όπως οι CART, ID3, C4.5. Εναλλακτικά, τα δένδρα μπορούν να αναπαρασταθούν και ως σύνολα κανόνων if και then, όπου ονομάζονται κανόνες ταξινόμησης.



Εικόνα 6. Τρόπος λειτουργίας Αλγορίθμου Hunt

Εκτός από την κατασκευή του δένδρου, πρέπει να καθοριστούν οι συνθήκες ελέγχου για τα γνωρίσματα. Οι συνθήκες ελέγχου, εξαρτώνται από τον τύπο των γνωρισμάτων και διακρίνονται σε συνεχείς, διακριτές, διατεταγμένες, καθώς και από το είδος διαχωρισμού, στο οποίο θα γίνει και είναι ο δυαδικός ή ο πολλαπλός διαχωρισμός. Με βάση τώρα το δυαδικό διαχωρισμό, το σύνολο τιμών διαχωρίζεται σε δυο υποσύνολα, έτσι ώστε να βρεθεί ο βέλτιστος διαχωρισμός, ενώ με τον πολλαπλό, χρησιμοποιούνται τόσες διασπάσεις, όσες είναι και οι διαφορετικές τιμές που δίνονται.

Μία εναλλακτική ταξινόμηση στα δένδρα αποφάσεων, αποτελούν οι κανόνες ταξινόμησης, όπου θεωρούνται από τις πιο κατανοητές αναπαραστάσεις για τον άνθρωπο, με τις κυριότερες κατηγορίες να είναι α) η προϋπόθεση και β) το συμπέρασμα κανόνα.



Στον κανόνα προϋπόθεσης, υπάρχει ένα σύνολο ελέγχων, το οποίο είναι όμοιο με τους ελέγχους του κόμβου ενός δένδρου απόφασης, ενώ στον κανόνα με συμπέρασμα, γίνεται η εκχώρηση α) ταξινόμησης, β) συνόλου ταξινομήσεως ή γ) κατανομής πιθανότητας. Όταν συναντούμε ανεξάρτητους κανόνες χρησιμοποιούνται οι λογικές διαζεύξεις, κατά κύριο λόγο η OR, ώστε να ατυχούμε τη δημιουργία κανόνα. Το πρόβλημα που συναντάμε είναι ότι κάποιες φορές οι υποδείξεις των κανόνων είναι διαφορετικές για το ίδιο παράδειγμα. Για να μετατρέψουμε τέλος ένα δένδρο σε ένα σύνολο κανόνων, θα πρέπει αρχικά να θέσουμε έναν κανόνα για κάθε φύλλο. Η προϋπόθεση περιέχει μια συνθήκη για κάθε κόμβο που συναντάται από τη ρίζα ως το φύλλο, ενώ ως συμπέρασμα ορίζεται η τάξη εκχώρησης. Οι παραγόμενοι κανόνες είναι σαφείς και ορίζονται μονοσήμαντα, ενώ αλλάζοντας τη σειρά που θα εκτελεστούν, δεν θα υπάρξει αλλοίωση του αποτελέσματος.

1.6.2.5 Μάθηση βασισμένη σε Στιγμιότυπα

Η μάθηση βασισμένη σε στιγμιότυπα, αποτελεί μια απλή προσέγγιση του προβλήματος της μηχανικής μάθησης. Παρόλη την απλότητα όμως, παρουσιάζει μια θεμελιώδη διαφορά με όλες τις υπόλοιπες οικογένειες αλγορίθμων, όπου κωδικοποιούν τα παραδείγματα εκπαίδευσης, σε μια συμπαγή περιγραφή. Έτσι, η μάθηση βασισμένη σε στιγμιότυπα, διατηρεί αυτούσια τα δεδομένα εκπαίδευσης. Επίσης, ένας αλγόριθμος μάθησης βασισμένος σε στιγμιότυπα, στερείται του σταδίου της εκπαίδευσης, καθώς ο αλγόριθμος αρκείται στην απλή απομνημόνευση όλων των στιγμιότυπων εκπαίδευσης που του παρέχονται, τα οποία χρησιμοποιεί μόνο όταν κληθεί να αποφανθεί για ένα άγνωστο στιγμιότυπο. Η απόφαση αυτή λαμβάνεται με βάση την ομοιότητα του αγνώστου στιγμιότυπου με τα αποθηκευμένα. Ο έλεγχος ομοιότητας δύο στιγμιότυπων επιτυγχάνεται με τη χρήση μιας συνάρτησης απόστασης, η οποία επιλέγεται κατά αναλογία με τη φύση του εκάστοτε προβλήματος. Γίνεται αντιληπτό ότι οι αλγόριθμοι της κατηγορίας αυτής δεν κατασκευάζουν ένα καθολικό μοντέλο που να αναπαριστά τη γνώση που απέκτησαν από τα δεδομένα της εκπαίδευσης, αλλά ο προσδιορισμός της συνάρτησης στόχου γίνεται τοπικά, με κάθε ταξινόμηση ενός αγνώστου στιγμιότυπου, αντλώντας πληροφορίες από τα χαρακτηριστικά της ομάδας στιγμιότυπων με τα οποία συγγενεύει.



2. Μηχανική όραση

2.1 Ορισμός

Με τον όρο «μηχανική όραση» ή «όραση υπολογιστών» (Computer vision) ονομάζεται η τεχνολογία που επιτρέπει την μετατροπή των δεδομένων από μια συσκευή όρασης (φωτογραφικές μηχανές, βιντεοκάμερες, σαρωτές και άλλες συσκευές), σε δυαδικά, ψηφιακά δεδομένα. Η μηχανική όραση ή όραση υπολογιστών είναι η τεχνολογία που, ουσιαστικά, επιτρέπει σε μηχανές να «βλέπουν», αφού από τη στιγμή της αναπαράστασης μιας εικόνας με ψηφιακό τρόπο, είναι δυνατή η αλγοριθμική επεξεργασία και η εξαγωγή πληροφορίας και μετρικών. Η μηχανική όραση μπορεί επίσης να περιγραφεί ως συμπλήρωμα (αλλά όχι απαραίτητως αντίθετο) της βιολογικής όρασης. Στην τελευταία, μελετώνται η οπτική αντίληψη στους ανθρώπους και τα ζώα και καθορίζονται μοντέλα για το πώς

αυτά τα συστήματα λειτουργούν υπό το πρίσμα των φυσιολογικών διαδικασιών. Η μηχανική όραση από την άλλη μελετά και περιγράφει το τεχνητά συστήματα όρασης που εφαρμόζονται σε λογισμικό ή/και σε υλικό υπολογιστών.

Η διεπιστημονική ανταλλαγή μεταξύ της βιολογικής και μηχανικής όρασης αποδεικνύεται όλο και περισσότερο καρποφόρα και για τους δύο τομείς. Οι υποκατηγορίες της μηχανικής όρασης περιλαμβάνουν την αναδημιουργία σκηνής, την ανίχνευση συμβάντων, την ανίχνευση κίνησης, την αναγνώριση αντικειμένων, τη μάθηση, την ευρετηριοποίηση και την αποκατάσταση εικόνας.

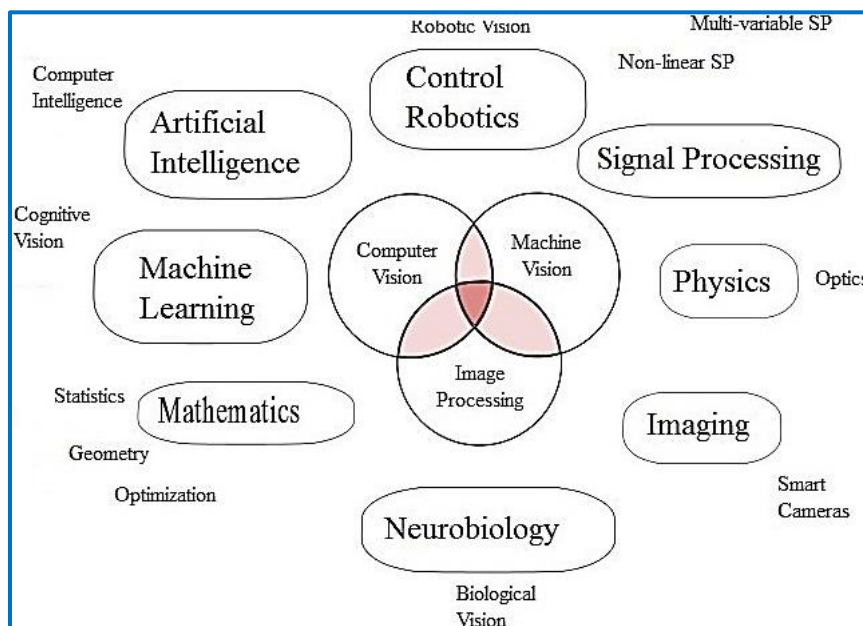
Η μηχανική όραση μπορεί επίσης να περιγραφεί ως συμπλήρωμα (αλλά όχι απαραίτητως αντίθετο) της βιολογικής όρασης, όπου τελευταία, μελετώνται η οπτική αντίληψη στους ανθρώπους και τα ζώα και καθορίζονται μοντέλα για το πώς αυτά τα συστήματα λειτουργούν υπό το πρίσμα των φυσιολογικών διαδικασιών. Η μηχανική όραση από την άλλη μελετά και περιγράφει το τεχνητά συστήματα όρασης που εφαρμόζονται σε λογισμικό ή και σε υλικό υπολογιστών.

2.2 Χρήση

Η μηχανική όραση χρησιμοποιείται σε ένα πλήθος εφαρμογών στην επιστήμη όπως είναι η ρομποτική, η ιατρική και γηριατρική για την επίβλεψη καθώς και την παρακολούθηση ηλικιωμένων - ασθενών, αλλά και στη βιομηχανία παραγωγής αγαθών και υπηρεσιών.

Η μηχανική όραση επίσης χρησιμοποιείται κατά κόρον στα στρατιωτικά συστήματα για τον εντοπισμό και στόχευση αντίπαλων στόχων. Τα πλέον προηγμένα πυραυλικά συστήματα στέλνουν το πολεμικό υλικό όχι σε έναν συγκεκριμένο στόχο αλλά σε μια περιοχή, της οποίας η επιλογή γίνεται με βάση δεδομένα εικόνας που λαμβάνονται πριν την αποστολή του πολεμικού υλικού [1].

Εκτός των ανωτέρω, η μηχανική όραση ξεκίνησε να εφαρμόζεται στην περίπτωση των αυτόματα κινούμενων οχημάτων, όπως φορτηγά, αυτοκίνητα καθώς και αυτόνομα κινούμενα robot. Ο βαθμός αυτονομίας αυτών των οχημάτων ποικίλει από ολοκληρωτικά αυτόνομα μέχρι οχήματα όπου το σύστημα όρασης λειτουργεί βοηθητικά ως προς τον οδηγό του οχήματος. Παρακάτω παρουσιάζεται μια εικόνα με τα πεδία εφαρμογής :



Εικόνα (1). Πεδία Εφαρμογής CV

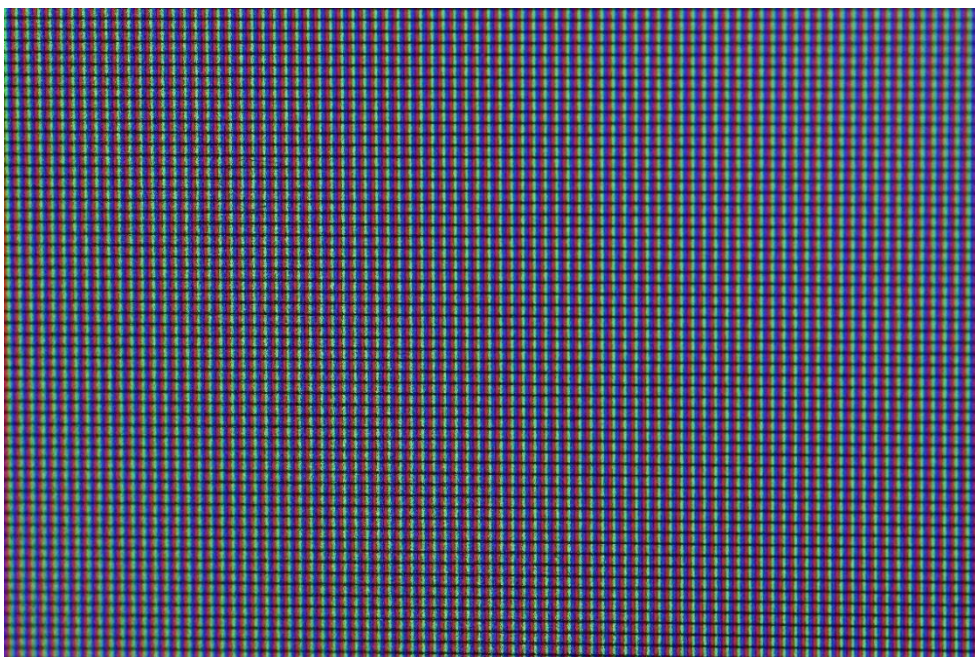
2.3 Επεξεργασία Εικόνας

Για να υπεισέλθουμε στο κομμάτι που ονομάζεται επεξεργασία εικόνας γενικότερα, θα πρέπει να γνωρίσουμε τι είναι, αλλά και από τι αποτελείται μια εικόνα.

2.3.1 Εικόνα

Μια εικόνα λοιπόν, αναπαριστά μία στιγμιαία μηχανική αποτύπωση, του τι βλέπει ο αισθητήρας που λαμβάνει την εικόνα, όπου προσομοιάζει το τι βλέπει το ανθρώπινο μάτι.

Μία εικόνα, αποτελείται από ένα σετ από εικονοστοιχεία (**pixels**), όπου είναι τα δομικά στοιχεία της Εικόνας (2). Το pixel, είναι το μικρότερο διακριτό σημείο σε μία εικόνα, όπου απεικονίζει μια συγκεκριμένη τιμή χρώματος και φωτεινότητας. Μπορούμε να φανταστούμε λοιπόν μια εικόνα ως ένα πλέγμα από σειρές και στήλες, όπου το κάθε κελί αποτελεί ένα εικονοστοιχείο της. Έτσι εάν π.χ. έχουμε μια εικόνα διαστάσεων 500x300, αυτό σημαίνει ότι αποτελείτε από 500 στήλες και 300 σειρές, άρα η εικόνα μας αποτελείτε από $500 \times 300 = 150.000$ εικονοστοιχεία.



Εικόνα (2). Αποτύπωση εικονοστοιχείων - pixels



Τα περισσότερα εικονοστοιχεία, παρουσιάζονται οπτικά σε δύο βασικές μορφές :

- a) Στην ασπρόμαυρη μορφή τους, όπου κάθε εικονοστοιχείο μπορεί να πάρει μια τιμή από 0 έως και 255. Η τιμή 0, αντιπροσωπεύει το μαύρο χρώμα και η τιμή 255 το λευκό. Οι ενδιάμεσες τιμές μεταξύ των δύο, αντιπροσωπεύουν τις διάφορες αποχρώσεις του γκρι, με τις τιμές που πλησιάζουν στο 0 να είναι σκοτεινότερες σε απόχρωση, ενώ αυτές που πλησιάζουν στο 255, να είναι ανοιχτότερες.
- b) Στην έγχρωμη μορφή τους (RGB), όπου το κάθε εικονοστοιχείο παίρνει τρεις τιμές, μία για το κόκκινο κανάλι (Red), μία για το πράσινο κανάλι (Green) και μία για το μπλε κανάλι (Blue). Η τιμή που παίρνει το κάθε ένα από τα τρία χρώματα, αποτελείται από μία ακέραιη τιμή, από το διάστημα 0 έως και 255, όπου στην ουσία η κάθε τιμή, αντιπροσωπεύει την “ποσότητα” του συγκεκριμένου χρώματος από το κάθε κανάλι ξεχωριστά και που θα περιέχει το κάθε εικονοστοιχείο. Εν συνεχεία, γίνεται συνδυασμός των τριών τιμών στο χρωματικό χώρο RGB και παίρνουμε την απόχρωση, από το συνδυασμό τους.

Εν κατακλείδι, γίνεται κατανοητό, ότι όσο πιο μικρός είναι ο αριθμός των εικονοστοιχείων που απαρτίζουν μια εικόνα, τόσο χαμηλότερη είναι και η ανάλυση – ποιότητα και αντιστρόφως ανάλογα.

2.3.2 Ψηφιακή Επεξεργασία Εικόνας

Επεξεργασία εικόνας, ονομάζεται κάθε μορφή αλγοριθμικής επεξεργασίας, ανάλυσης και χειρισμού ψηφιακών δεδομένων εικόνας ή βίντεο, όπως και το σχετικό επιστημονικό πεδίο της πληροφορικής. Στην επεξεργασία εικόνας, τόσο η είσοδος όσο και η έξοδος των υπολογισμών είναι δεδομένα εικόνας / βίντεο (έγχρωμα, ασπρόμαυρα ή σε αποχρώσεις του γκριζου). Από την επεξεργασία εικόνας εκπορεύονται επίσης και αλγόριθμοι ανάλυσης / κατανόησης εικόνας, αλλά εκεί υφίσταται επικάλυψη με το συγγενές γνωστικό πεδίο της τεχνητής



νοημοσύνης ονόματι μηχανική όραση. Μεγάλο μέρος του επιστημονικού υποβάθρου της επεξεργασίας εικόνας παρέχεται από την επεξεργασία σήματος, καθώς η ψηφιακή εικόνα μπορεί να θεωρηθεί δισδιάστατο χωρικό σήμα και το βίντεο τρισδιάστατο χωροχρονικό σήμα [2].

2.4 Εφαρμογές και Συστήματα

Ένα σύστημα μηχανικής όρασης, σχεδιάζεται λαμβάνοντας υπόψη το πεδίο εφαρμογής, για το οποίο προορίζεται. Κάποια συστήματα είναι αυτόνομες εφαρμογές λογισμικού που επιλύουν ένα συγκεκριμένο πρόβλημα μέτρησης ή εντοπισμού κάποιου χαρακτηριστικού ενώ κάποιες άλλες αποτελούν υποσυστήματα μιας ευρύτερης κατασκευής, όπως είναι τα ρομπότ.

Κάποιες λειτουργίες της μηχανικής όρασης, απαντώνται σε ένα και μόνο σύστημα - εφαρμογή, ενώ κάποιες άλλες μπορεί να είναι κοινές σε όλα ή στις περισσότερες εφαρμογές. Οι πιο συνηθισμένες λειτουργίες [3], είναι οι:

- Ανάκτηση εικόνας. Μια ψηφιακή εικόνα παράγεται με τη βοήθεια μιας ή περισσότερων συσκευών όρασης, με αυτή-ες να είναι, από μια απλή Web camera σαν αυτή που βρίσκεται σε οποιονδήποτε υπολογιστή, light-sensitive κάμερες, 3D κάμερες όπως η Kinect ή Asus, Laser Scanner radar κτλ. Ανάλογα τον τύπο της συσκευής, η τελική εικόνα μπορεί να είναι δισδιάστατη ή τρισδιάστατη, σε κλίμακα βάθους (depth cam), ή ακόμα και μια ακολουθία εικόνων.
- Προεπεξεργασία. Πριν εφαρμοστεί μια μέθοδος στην εικόνα που δημιουργήθηκε, η εικόνα πρέπει να ελεγχθεί εάν τηρεί κάποιες βασικές προδιαγραφές, που θέτει η προς εφαρμογή μέθοδος. Ο έλεγχος περιλαμβάνει λειτουργίες, όπως επαναληπτική δειγματοληψία (resampling) για να διασφαλιστεί ότι το σύστημα συντεταγμένων της εικόνας είναι το κατάλληλο. Επίσης περιλαμβάνει μείωση θορύβου (noise reduction) για να



διασφαλιστεί η απουσία λανθασμένης και άχρηστης πληροφορίας. Τέλος, ενισχύονται οι αντιθέσεις (contrast) της εικόνας ώστε η ενδιαφέρουσα πληροφορία να γίνει ευκολότερα εντοπίσιμη.

- Εξαγωγή χαρακτηριστικών. Χαρακτηριστικά που υπάρχουν μέσα στην εικόνα σε διάφορους βαθμούς λεπτομέρειας, εξάγονται από την εικόνα. Τέτοια γνωρίσματα μπορεί να είναι σχήματα όπως γραμμές, ακμές, σημεία, γωνίες και πολλά άλλα.
- Εντοπισμός και τεμαχισμός (τμηματοποίηση). Η εφαρμογή εφόσον εντοπίσει τα ενδιαφέροντα χαρακτηριστικά της εικόνας, τα αποκόβει από την υπόλοιπη εικόνα και τα προωθεί σε άλλα υποσυστήματα της εφαρμογής.
- Επεξεργασία υψηλού επιπέδου. Σε αυτό το σημείο, η είσοδος είναι ένα μικρό σύνολο δεδομένων, για παράδειγμα ένα σύνολο σημείων ή μια περιοχή της εικόνας, ή ένα συγκεκριμένο αντικείμενο. Ακολουθεί η επεξεργασία του συνόλου αυτού, όπως για παράδειγμα η εκτίμηση του μεγέθους του.
- Αναγνώριση εικόνας. Ακολουθεί η κατηγοριοποίηση των αντικειμένων της εικόνας σε κάποιες προκαθορισμένες κατηγορίες.
- Καταχώρηση εικόνας. Πρόκειται για τη σύγκριση και το συνδυασμό διαφορετικών όψεων των ίδιων αντικειμένων.
- Λήψη αποφάσεων. Εκτίμηση σύμφωνα με την μέθοδο που εφαρμόστηκε. Για παράδειγμα, αποδοχή ή απόρριψη της εικόνας ανάλογα με το αν ικανοποιεί κάποιο προκαθορισμένο πρότυπο.



3. Λογισμικό & Βιβλιοθήκες μηχανικής όρασης

2.1 Βιβλιοθήκες

Το λογισμικό μηχανικής οράσεως, είναι μια συλλογή βιβλιοθηκών υλοποιημένων σε γλώσσα προγραμματισμού C++, όπου προορίζονται για χρήση από τις εφαρμογές μηχανικής όρασης. Πρόκειται για ένα έργο λογισμικού ανοικτού κώδικα και περιλαμβάνει διάφορες υπο-βιβλιοθήκες.

Παρακάτω, παρουσιάζονται μερικές από τις κυριότερες βιβλιοθήκες [4], όπου χρησιμοποιούνται για την ανάπτυξη αλγορίθμων - εφαρμογών στο πεδίο της όρασης των υπολογιστών και που είναι :

- **OpenCV.** Η OpenCV, είναι μία εκτενής βιβλιοθήκη επεξεργασίας εικόνας ανοικτού κώδικα, χτισμένη πάνω στη βιβλιοθήκη Επεξεργασία Εικόνας της Intel - Intel Image Processing Library (IPL). Η OpenCV, ενισχύει τις εμπορικές χρήσεις της υπολογιστικής μηχανικής όρασης σε σχέση με την διεπαφή με τον άνθρωπο, με τη ρομποτική παρακολούθηση, με τη χρήση για βιομετρικές εφαρμογές ασφάλειας κτλ. Προσφέροντας ελεύθερη και ανοιχτή υποδομή, καταφέρνει μέσω της κοινότητας που την υποστηρίζει, να βελτιώνεται συνεχώς στον τομέα των επιδόσεών της [5]. Η OpenCV, είναι η κυριότερη βιβλιοθήκη από όλες και θα ασχοληθούμε μεμονωμένα στη συνέχεια.
- **V(X)L.** Το λογισμικό VXL (Vision-something-Library) [6], είναι μια συλλογή βιβλιοθηκών, υλοποιημένων σε C++ που προορίζονται για χρήση από εφαρμογές μηχανικής όρασης. Πρόκειται για ένα έργο λογισμικού ανοικτού κώδικα και περιλαμβάνει υπο-βιβλιοθήκες που, ανάλογα τη χρήση τους, έχουν αλλαγμένο το χαρακτήρα «X».

Έτσι, υπάρχει η βιβλιοθήκη VGL που είναι μια βιβλιοθήκη γεωμετρικών κλάσεων και μεθόδων (εκ του «geometry»), η VNL είναι μια βιβλιοθήκη αριθμητικών μεθόδων (εκ του «numeric»), η VIL που είναι μια βιβλιοθήκη



επεξεργασίας εικόνας (εκ του «image processing»), η VSL (εκ του «streaming I/O»), η VBL (εκ του «basic templates») και η βιβλιοθήκη εργαλείων VUL (εκ του «utilities»).

Το λογισμικό VXL προήλθε από τα παλιότερα έργα λογισμικού TargetJr και IUE και έχει σαν σκοπό την ύπαρξη του σαν ένα ελαφρύ, γρήγορο και συνεκτικό σύστημα. Όπως προαναφέρθηκε, είναι υλοποιημένο σε ANSI/ISO C++ και σχεδιάστηκε ώστε να είναι φορητό και να μπορεί να χρησιμοποιηθεί από εφαρμογές που αναπτύσσονται σε διάφορες πλατφόρμες. Η VXL αναπτύσσεται και χρησιμοποιείται από μια διεθνή κοινότητα ακαδημαϊκών και βιομηχανικών χρηστών και προγραμματιστών [6].

- **GPUCV.** Η GPUCV είναι μια ελεύθερη ανοιχτού κώδικα βιβλιοθήκη με επιτάχυνση GPU, (χρησιμοποιεί και τη cpu της κάρτας γραφικών), για την επεξεργασία της εικόνας. Προσφέρει μια διεπαφή για τον προγραμματισμό όμοια με της Intel OpenCV, ώστε να χρησιμοποιείτε σε ήδη υπάρχουσες εφαρμογές [7].
- **Torch3vision.** Αυτή η βιβλιοθήκη μηχανικής όρασης, είναι γραμμένη σε απλή γλώσσα C++ και βασίζεται στη βιβλιοθήκη μηχανικής μάθησης Torch. Αυτό το πακέτο, περιέχει Βασικούς αλγόριθμους επεξεργασίας εικόνας, καθώς και εξαγωγής χαρακτηριστικών γνωρισμάτων, όπως περιστροφή, αναστροφή, φωτομετρική εξομάλυνση, ανίχνευση ακμών, 2D DCT, 2D FFT, 2D Gabor, PCA για Eigen-Faces, LDA για Fisher-Faces [7].
- **TLib.** Η TLib, είναι μια Βιβλιοθήκη επεξεργασίας εικόνας, όπου είναι γραμμένη ειδικά για εντοπισμό αντικειμένων (object tracking), σε πραγματικό χρόνο. Απευθύνεται σε προγραμματιστές με βασικές γνώσεις



προγραμματισμού C/C++, καθώς και βασικές γνώσεις επεξεργασίας εικόνων [7].

Κάθε βιβλιοθήκη, είναι ελαφριά από άποψη κατανάλωσης υπολογιστικών πόρων και αρκετά αυτόνομη, υπό την έννοια πως δεν απαιτείται αναφορά σε άλλες βιβλιοθήκες. Ομοίως, οι κύριες βιβλιοθήκες δεν εξαρτώνται παρά από τα απολύτως απαραίτητα, ώστε ο χρήστης να μπορεί να μεταγλωττίσει και να χρησιμοποιήσει μόνο ό,τι χρειάζεται.

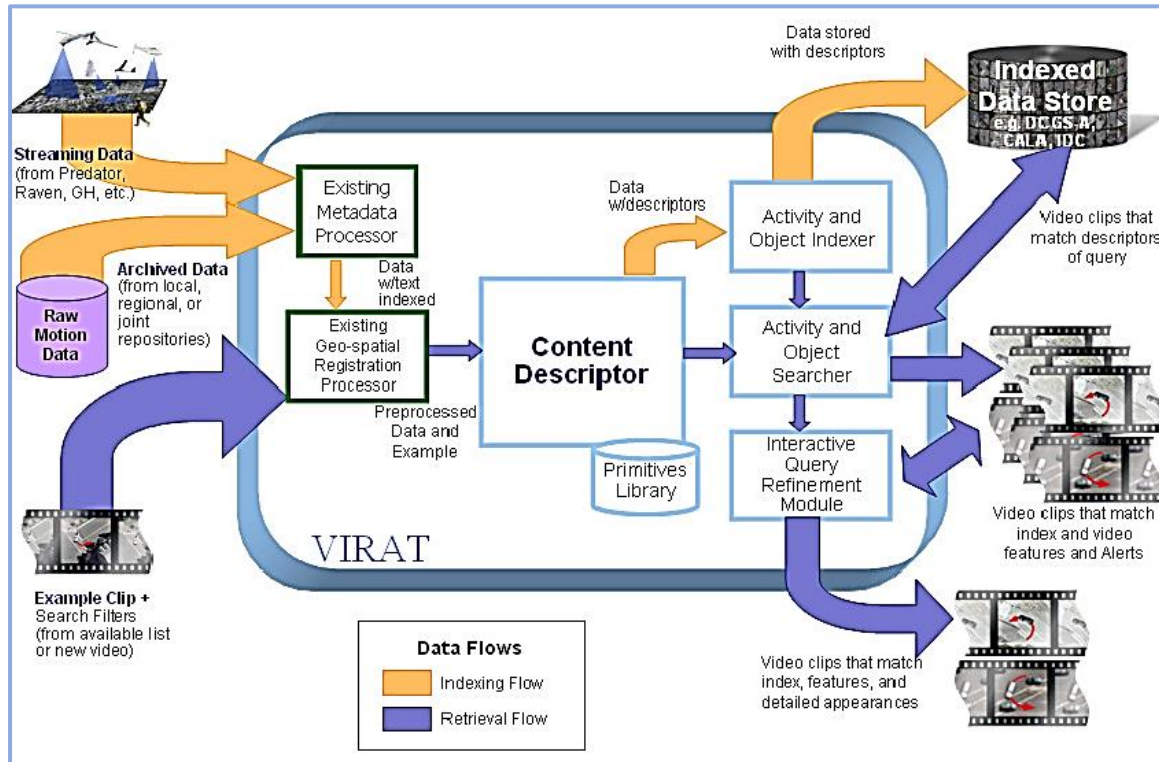
Εκτός από τις κύριες βιβλιοθήκες, (core libraries), υπάρχουν επιπλέον και άλλες βιβλιοθήκες που ασχολούνται με διαφόρων ειδών επεξεργασίες, όπως με αλγορίθμους αριθμητικών μεθόδων, επεξεργασία εικόνας, συστημάτων ελέγχου, γεωμετρίας κάμερας, σχεδιασμό γραφικών διεπαφών χρήστη (GUI), εντοπισμό γνωρισμάτων (feature tracking), τοπολογία, τρισδιάστατη απόδοση, αλλά και πολλά άλλα.

2.2 Λογισμικό

Σήμερα, υπάρχουν πλέον αρκετά λογισμικά για την επεξεργασία των εικόνων. Μερικά χαρακτηριστικά έργα - λογισμικά είναι το VIRAT, το MeVisLab, το IVT, καθώς και το Mocolo.

α) Το λογισμικό **VIRAT** (Video Image Retrieval & Analysis Tool) [8], είναι ένα project προσανατολισμένο στη βίντεο εποπτεία (video surveillance). Χρηματοδοτείται από τον οργανισμό Information Processing Technology Office (IPTO), όπου αποτελεί τμήμα του DARPA (Defense Advanced Research Projects Agency) [9]. Ο σκοπός του προγράμματος, είναι η δημιουργία μιας βάσης δεδομένων, που να είναι σε θέση να αποθηκεύσει τεράστιες ποσότητες αρχείων video και να κάνει εύκολη την αναζήτηση πάνω σε αυτό, με βάση «περιεχόμενου video που ενδιαφέρει τον χρήστη» (content-based searching). Επίσης, στοχεύει στην δημιουργία λογισμικού, που να μπορεί να παρέχει έκτακτες ειδοποιήσεις (alerts), σε ευφυή συστήματα

ελέγχου όπως για παράδειγμα ότι «ένα άτομο εισήλθε στο κτίριο». Στην εικόνα (3), παρουσιάζεται η αρχιτεκτονική του VIRAT.

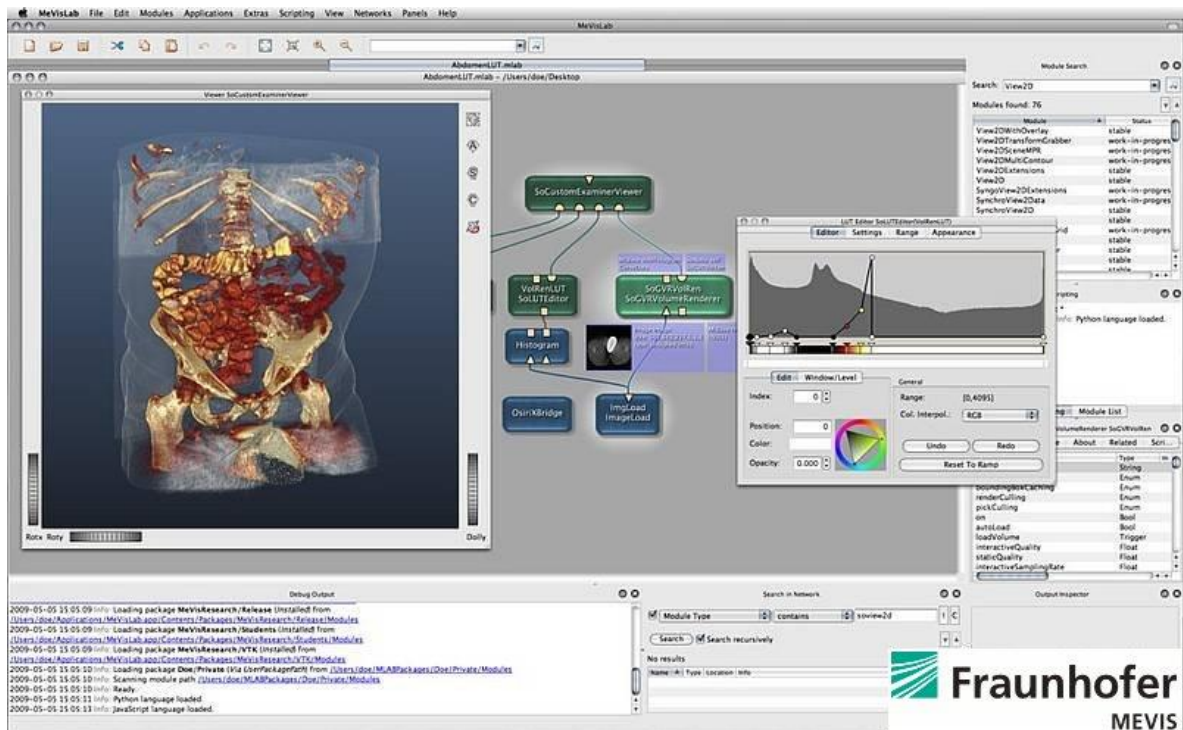


Εικόνα (3). Αρχιτεκτονική VIRAT

β) Το λογισμικό **MeVisLab** [10], είναι ένα διαλειτουργικό (cross-platform) application framework, με προσανατολισμό την επεξεργασία ιατρικών εικόνων και την απεικόνιση επιστημονικών δεδομένων (scientific visualization). Περιλαμβάνει προχωρημένους αλγορίθμους καταγραφής εικόνων, τμηματοποίηση (segmentation), μορφολογική και λειτουργική ανάλυση εικόνας.

Το MeVisLab έχει υλοποιηθεί σε C++ και χρησιμοποιεί το framework Qt για την δημιουργία γραφικών διεπαφών χρήστη. Έχει σχεδιαστεί για να μπορεί να εκτελείται σε περιβάλλον Windows, Linux και Mac OS X. Η ανάπτυξη του είναι κύρια ευθύνη των οργανισμών MeVis Medical Solutions AG και Fraunhofer MEVIS. Όσον αφορά τη διανομή του, υπάρχει μια freeware έκδοση του MeVisLab SDK, υπάρχουν modules ανοικτού κώδικα που εντάσσονται στα MeVisLab Public Sources και είναι

διαθέσιμα από το MeVisLab Community Sources Project. Η εικόνα (4) παρουσιάζει το GUI περιβάλλον του MeVisLab.



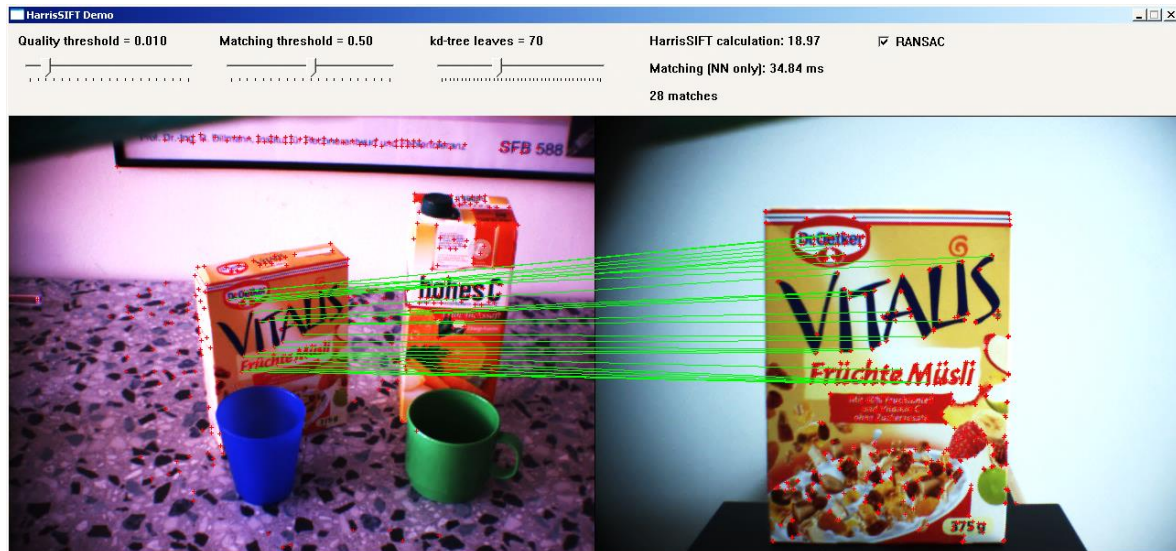
Εικόνα (4). MeVisLab

γ) Το λογισμικό **IVT** (Integrating Vision Toolkit) [11], είναι μια ισχυρή και υψηλής ταχύτητας βιβλιοθήκη μηχανικής όρασης, υλοποιημένη σε C++. Ενσωματώνει μια εύκολη στη χρήση και την κατανόηση, αντικειμενοστραφή αρχιτεκτονική και προσφέρει το δικό του, διαλειτουργικό GUI toolkit. Η βιβλιοθήκη είναι διαθέσιμη σαν ελεύθερο (free) λογισμικό, κάτω από την άδεια χρήσης BSD. Η βιβλιοθήκη έχει γραφτεί αποκλειστικά σε ANSI C++ και συνεπώς μεταγλωττίζεται επιτυχώς από οποιονδήποτε σύγχρονο C++ compiler, όπως τις διάφορες εκδόσεις του Visual Studio και του gcc. Επίσης, είναι διαλειτουργικό λογισμικό και μπορεί να εκτελεστεί σχεδόν σε οποιοδήποτε υπολογιστικό σύστημα όπως Windows, Mac OS X και Linux, προσφέροντας ταυτόχρονα, ένα περιβάλλον με μεταγλωττιστή C++.

Το ενσωματωμένο GUI toolkit προσφέρει υλοποιήσεις για Windows (Win32 API), Linux (GTK), Mac OS X (Cocoa) και Qt. Η βιβλιοθήκη IVT αναπτύσσεται από ομάδα



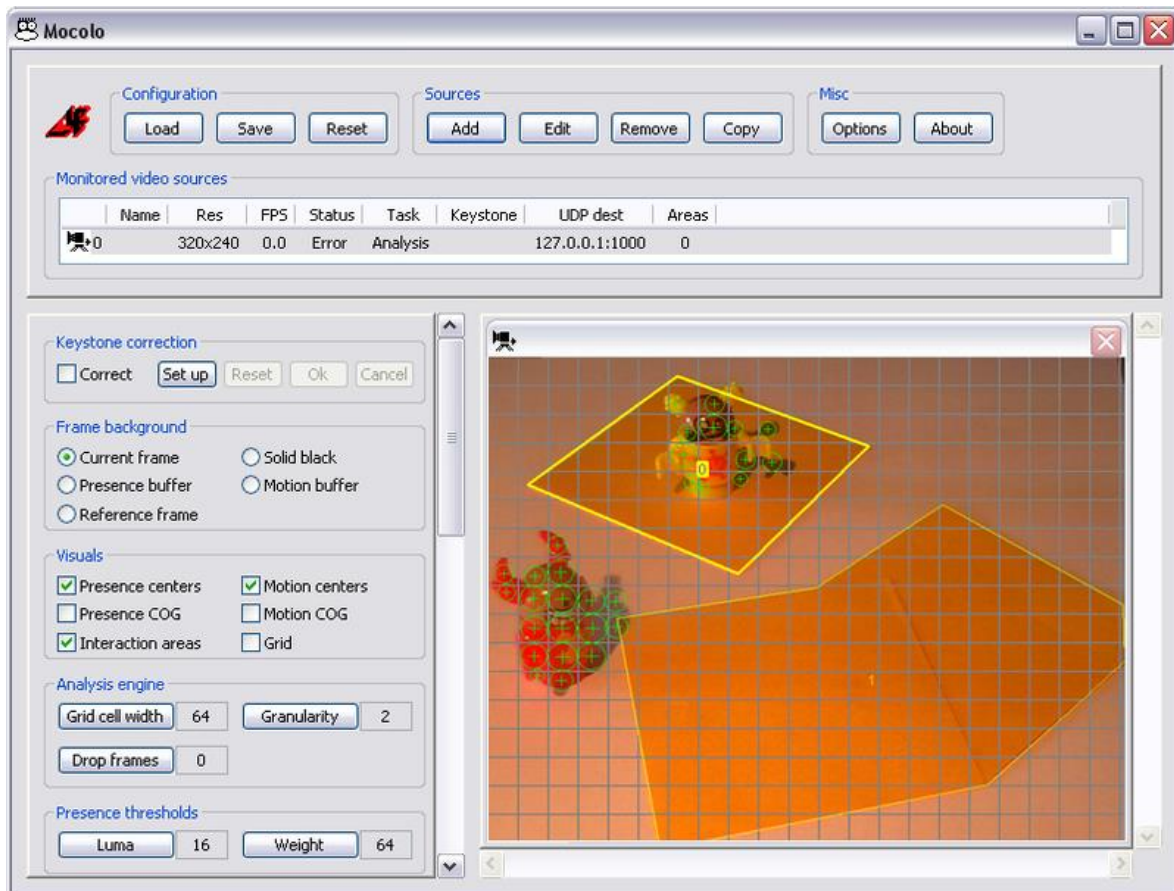
προγραμματιστών στο ίδρυμα Karlsruhe Institute of Technology (KIT). Η πρώτη έκδοση κυκλοφόρησε το 2005. Από το 2009, η βιβλιοθήκη IVT συντηρείται σε συνεργασία με την εταιρεία Keyetech. Εικόνα (5).



Εικόνα (5). IVT Library

δ) Το λογισμικό **Mocolo** [12], είναι ένα server λογισμικό ανάλυσης video, σχεδιασμένο για να εκτελείται σε περιβάλλον Microsoft Windows και το οποίο χρησιμοποιείται κυρίως σε διαδραστικές εφαρμογές. Παρακολουθεί (monitoring) ροές video, εντοπίζει αλληλεπίδραση μεταξύ αντικειμένων όπως παρουσία ή κίνηση και στέλνει τα αποτελέσματα της ανάλυσης σε client εφαρμογές, υπό την μορφή Open Sound Control (OSC) πακέτων.

Τα κύρια χαρακτηριστικά του είναι ο κλειστός κώδικας, η μικρή ανάγκη σε υπολογιστικούς πόρους, η δυνατότητα να αναλύει πολλαπλές πηγές βίντεο την ίδια στιγμή και η συνδεσιμότητα με οποιαδήποτε εφαρμογή λογισμικού χειρίζεται πακέτα OSC. Όπως προαναφέρθηκε, το λογισμικό Mocolo, χρησιμοποιεί μικρές ποσότητες υπολογιστικής ισχύος και καθόλου το υποσύστημα γραφικών του υπολογιστικού συστήματος στο οποίο εκτελείται (GPU). Αποτέλεσμα είναι να επιτρέπει έτσι, την ταυτόχρονη εκτέλεση άλλων client εφαρμογών. Εικόνα (6)



Εικόνα (6). Mocolo

4. Βιβλιοθήκη OpenCV

3.1 Γενικά

Η εργαλειοθήκη OpenCV (Open Computer Vision) είναι μια βιβλιοθήκη που υλοποιεί μεθόδους τεχνητής όρασης. Πρόκειται για ένα έργο λογισμικού ανοικτού κώδικα (Open Source), που διατίθεται δωρεάν, τόσο σε πακέτο εγκατάστασης όσο και σε μορφή πηγαίου κώδικα. Επίσης, υπάρχουν διαφορετικά πακέτα ανάλογα με την πλατφόρμα και το λειτουργικό σύστημα που ενδιαφέρει τον κάθε χρήστη.

Η βιβλιοθήκη είναι υλοποιημένη στις γλώσσες προγραμματισμού C και C++ και είναι σχεδιασμένη, έτσι ώστε να μεταγλωττίζεται και να εκτελείται σε Linux, Windows και MacOS. Ταυτόχρονα, είναι ενεργή η ανάπτυξη διεπαφών για Python, Ruby, Matlab και άλλες γλώσσες. Η OpenCV, σχεδιάστηκε ώστε να έχει υψηλή



υπολογιστική απόδοση ενώ επικεντρώνεται στις εφαρμογές πραγματικού χρόνου. Επίσης η βιβλιοθήκη μπορεί να εκμεταλλευτεί άριστα, τις δυνατότητες των νέων πολυπύρηνων επεξεργαστών.

Εάν υπάρχει επιθυμία για βελτιστοποίηση της OpenCV, υπάρχει δυνατότητα αγοράς της βιβλιοθήκης IPP (Intergraded Performance Primitives library) της Intel, η οποία αποτελείται από πιστοποιημένες υπορουτίνες χαμηλού επιπέδου, που βελτιώνουν κατά πολύ την απόδοση της. Η OpenCV, χρησιμοποιεί αυτόματα την βιβλιοθήκη IPP κατά το run-time και εφόσον έχει εγκατασταθεί.

Ένας από τους στόχους της OpenCV, είναι να κάνει διαθέσιμη στους προγραμματιστές εφαρμογών, μια απλή στη χρήση υποδομή ή διεπαφή τεχνητής όρασης, η οποία θα συνδράμει στη γρήγορη και ευκολότερη ανάπτυξη εφαρμογών τεχνητής όρασης. Η OpenCV, διαθέτει πάνω από 500 συναρτήσεις, οι οποίες αφορούν πολλά πεδία της τεχνητής όρασης περιλαμβάνοντας: εργοστασιακό έλεγχο προϊόντων, ιατρική όραση, ασφάλεια, διεπαφή χρήστη, βαθμονόμηση κάμερας, στέρεο όραση, ρομποτική και άλλα.

Επειδή η τεχνητή όραση και η μηχανική μάθηση είναι έννοιες άρρηκτα συνδεδεμένες, η OpenCV, διαθέτει επίσης μια ολοκληρωμένη και γενικού σκοπού βιβλιοθήκη μηχανικής μάθησης, την Machine Learning Library (MLL), η οποία επικεντρώνεται στην στατιστική αναγνώριση προτύπων και στο clustering δεδομένων. Η MML [13], είναι χρήσιμη για τις διαδικασίες όρασης που βρίσκονται στον πυρήνα της OpenCV αλλά μπορεί να χρησιμοποιηθεί και για οποιοδήποτε πρόβλημα αυτού του είδους.

3.2 Χρήση της OpenCV

Η OpenCV, έχει δομηθεί έτσι, ώστε να μπορεί να φτιαχτεί ένα εμπορικό προϊόν με το σύνολο ή μέρος της OpenCV ελεύθερα [14]. Εν μέρει και λόγω αυτών των ελεύθερων όρων αδειοδότησης, υπάρχει μια τεράστια κοινότητα χρηστών, όπου



περιλαμβάνει ανθρώπους από μεγάλες εταιρείες όπως οι IBM, Microsoft, Intel, SONY, Siemens, και Google, καθώς και ερευνητικά κέντρα όπως τα Stanford, MIT, CMU, Cambridge και INRIA. Το OpenCV είναι δημοφιλές σε όλο τον κόσμο, με μεγάλες κοινότητες χρηστών στην Κίνα, την Ιαπωνία, την Ρωσία, και την Ευρώπη.

Από την alpha έκδοση του Ιανουαρίου του 1999, η OpenCV έχει χρησιμοποιηθεί σε πολλές εφαρμογές, προϊόντα, και έρευνες. Οι εφαρμογές περιλαμβάνουν συρραφή εικόνων από δορυφόρους και web χάρτες, ευθυγράμμιση σαρωμένων εικόνων, μείωση του θορύβου σε ιατρικές εικόνες, ανάλυση αντικειμένων, συστήματα ασφάλειας και ανίχνευσης διείσδυσης, αυτόματα συστήματα παρακολούθησης, συστήματα ελέγχου παραγωγής, βαθμονόμηση κάμερας, στρατιωτικές εφαρμογές, καθώς και σε μη επανδρωμένα εναέρια, επίγεια και υποβρύχια οχήματα. Έχει ακόμη χρησιμοποιηθεί στην αναγνώριση ήχων και μουσικής, όπου οι τεχνικές αναγνώρισης εικόνων εφαρμόζονται στο φασματογράφημα του ήχου. Η OpenCV αποτελούσε ένα βασικό μέρος του συστήματος όρασης στο ρομπότ του Stanford «Stanley», το οποίο κέρδισε το βραβείο των δυο εκατομμυρίων δολαρίων στον αγώνα DARPA Grand Challenge Desert Robot Race.

3.3 Αρχιτεκτονική της OpenCV

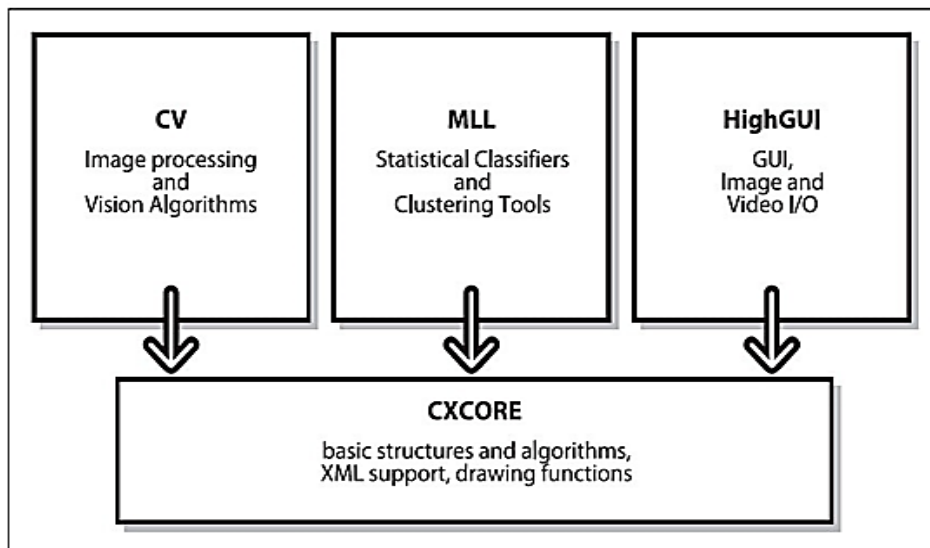
Το πακέτο λογισμικού OpenCV, σε γενικές γραμμές διαρθρώνεται σε πέντε κύρια μέρη[15], εικόνα (7). Η CV, περιέχει την βασική επεξεργασία εικόνας και τους αλγόριθμους τεχνητής όρασης υψηλότερου επιπέδου. Η ML είναι η βιβλιοθήκη μάθησης της μηχανής, όπου περιλαμβάνει πολλούς στατιστικούς ταξινομητές και εργαλεία ομαδοποίησης (clustering).

Η HighGUI περιέχει ρουτίνες εισόδου/εξόδου, τις συναρτήσεις για την αποθήκευση και φόρτωση βίντεο και εικόνων, ενώ η CXCore περιέχει τις βασικές δομές δεδομένων και το περιεχόμενο. Επίσης υπάρχει η CvAux, η οποία περιέχει

περιοχές που δεν χρησιμοποιούνται (ενσωματωμένο HMM αναγνώρισης προσώπου) αλλά και πειραματικούς αλγόριθμους (κατάτμηση φόντου, πρόσοψης).

Η OpenCV σχεδιάστηκε για να είναι φορητή, δηλαδή να μπορεί να χρησιμοποιηθεί σε εφαρμογές που αναπτύσσονται σε όσο το δυνατόν περισσότερες υπολογιστικές πλατφόρμες.

Αρχικά, σχεδιάστηκε και υλοποιήθηκε για να μεταγλωττίζεται επιτυχώς με τους μεταγλωττιστές Borland C++, Microsoft Visual C++ και για τους compilers της Intel. Εννοείται ότι ο C και C++ κώδικας έπρεπε να είναι αρκετά τυποποιημένος, ώστε να κάνει την υποστήριξη πολλών πλατφορμών ευκολότερη.



Εικόνα (7). Αρχιτεκτονική OpenCV

5. Χαρακτηριστικά Εικόνας

Τα χαρακτηριστικά μιας εικόνας, ή αλλιώς τα δεδομένα που προκύπτουν από μια εικόνα χωρίζονται σε δύο μεγάλες κατηγορίες, όπου είναι:

- 1) Τα χαρακτηριστικά που έχουν έμμεση σχέση με το περιεχόμενο της εικόνας όπως είναι ο τύπος αρχείου, το μέγεθος, η ανάλυση κτλ.



2) Τα χαρακτηριστικά που έχουν άμεση σχέση με την εικόνα και το περιεχόμενό της και όπου συναντούμε δύο υποκατηγορίες και που είναι, α) τα χαμηλού επιπέδου χαρακτηριστικά, (αυτά που είναι ορατά και άμεσα αντιληπτά από τον άνθρωπο), β) Τα υψηλού επιπέδου χαρακτηριστικά και που δεν γίνονται αντιληπτά από το ανθρώπινο μάτι.

2.5 Χρώμα Εικόνων

Το χρώμα της εικόνας ως ιδιότητα, το ανθρώπινο μάτι έχει το πλεονέκτημα να το αντιλαμβάνεται με μεγάλη ακρίβεια. Από την άλλη μεριά όμως, έχει ως μειονέκτημα ότι δεν μπορεί να εκφράσει αυτή την ιδιότητα ή να τη μετρήσει, κάτι που κάνει η μηχανική όραση.

Έτσι, για να μπορέσουμε να επιτύχουμε τη μέτρηση του χρώματος τεχνητά, χρησιμοποιούμε τα χαρακτηριστικά, αλλά και τα διαστήματα του χρώματος.

Ο απλούστερος τρόπος για την αναπαράσταση του χρώματος ενός pixel στη ψηφιακή επεξεργασία, είναι με τη χρήση ενός φυσικού αριθμού από το 0 έως το N, με το N να αντιπροσωπεύει τον αριθμό των χρωμάτων της χρωματικής παλέτας που έχει η εικόνα.

Το ανθρώπινο μάτι, βλέποντας μια εικόνα, αντιλαμβάνεται κάποια χαρακτηριστικά των χρωμάτων, όπως είναι α) η φωτεινότητα (Luminance), β) τον Κορεσμό (Saturation), γ) την απόχρωση (Hue), δ) τη λαμπρότητα (Brightness) και ε) τη χρωματικότητα (Lightness). Αυτά τα χαρακτηριστικά λοιπόν, είναι υπεύθυνα για τη δημιουργία των διάφορων χρωματικών περιοχών που δημιούργησε ο άνθρωπος.

Από τη μεριά της μηχανικής όρασης τώρα και αναλόγως της εφαρμογής που υλοποιείται, χρησιμοποιούνται μια ποικιλία από διαστήματα χρώματος [16]. Τα διαστήματα χρώματος όπου χρησιμοποιούνται συχνότερα καθώς είναι πιο κοντά στην ανθρώπινη όραση, είναι τα RGB (Red Green Blue), YUV ή (YCrCb στην ψηφιακή μορφή), LUV, LAB, HSV, HIS και HLS, CMY, όπου και θα αναλύσουμε στη συνέχεια.

2.6 Χρωματικοί χώροι

Οι χρωματικοί χώροι, χρησιμεύουν για την περιγραφή του τρόπου κωδικοποίησης της χρωματικής πληροφορίας μιας εικόνας μέσω των χρωμάτων. Τα χρώματα ουσιαστικά, είναι η κωδικοποίηση που υπάρχει στο νευρικό σύστημα του ανθρώπου, διακρίνοντας μήκη κύματος του φωτός, τα οποία δέχονται τα ανθρώπινα μάτια. Το ανθρώπινο μάτι διεγείρεται από μήκη κύματος που κυμαίνονται από 400nm έως 700nm περίπου εικόνα (8). Τα χρωματικά μοντέλα, βοηθούν στην περιγραφή και στην αναπαραγωγή των χρωμάτων. Κάθε χρωματικό μοντέλο, ορίζεται από ένα σύστημα συντεταγμένων, καθώς και από έναν υποχώρο μέσα σε αυτό το σύστημα, όπου συγκεκριμένα βασικά χρώματα, είναι η βάση αυτών των χρωματικών μοντέλων. Σπάνια συμπίπτει το ίδιο χρωματικό αποτέλεσμα από δύο εφαρμογές, ακόμα και αν εφαρμόζεται ίδιο, ακριβώς, χρωματικό μοντέλο. Έτσι το κάθε χρώμα είναι μοναδικό και αναπαρίσταται μοναδικά [17].

Χρώμα	Περιοχή μηκών κύματος (nm)	Περιοχή συχνοτήτων (Hz)
Ερυθρό	~ 630–700 nm	~ 476–429 $\times 10^{12}$ Hz
Πορτοκαλί	~ 590–630 nm	~ 510–476 $\times 10^{12}$ Hz
Κίτρινο	~ 560–590 nm	~ 535–510 $\times 10^{12}$ Hz
Πράσινο	~ 500–560 nm	~ 600–535 $\times 10^{12}$ Hz
Κυανό	~ 440–500 nm	~ 680–600 $\times 10^{12}$ Hz
Ιώδες	~ 400–440 nm	~ 750–680 $\times 10^{12}$ Hz

Εικόνα (8). Χρωματικός χώρος

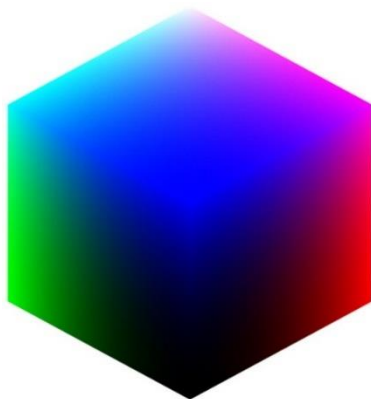
Στην επεξεργασία μίας εικόνας, η χρήση του χρώματος λαμβάνει χώρα για δύο κυρίως λόγους. Αρχικά, επειδή το χρώμα είναι ισχυρό χαρακτηριστικό για την περιγραφή και συνήθως απλοποιεί την αναγνώριση αντικειμένων. Και δεύτερον,

επειδή ο ανθρώπινος οφθαλμός είναι ικανός να διακρίνει χιλιάδες έγχρωμες αποχρώσεις, έναντι των 24 αποχρώσεων περίπου του γκρι.

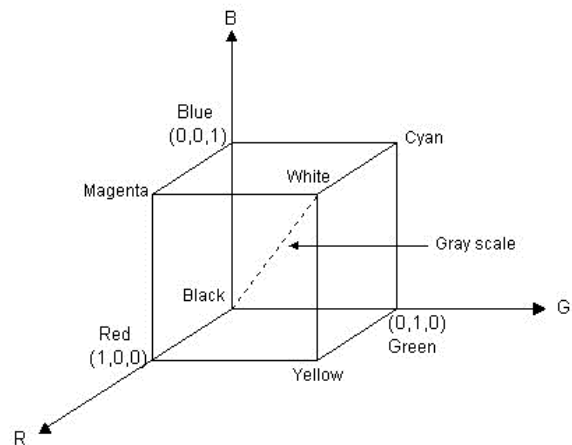
2.6.1 Ο χρωματικός χώρος RGB

Ο χώρος RGB (Red Green Blue), είναι από τους πιο διαδεδομένους και ευρέως χρησιμοποιούμενος χρωματικός χώρος. Η χρήση του έγκειται περισσότερο σε οθόνες συσκευών, καθώς παραμετροποιεί τα χρώματα μιας έγχρωμης οθόνης. Το RGB μοντέλο, βασίζεται στη φιλοσοφία που ακολουθεί ο αμφιβληστροειδής χιτώνας του ανθρώπινου ματιού. Έτσι μπορεί να περιγραφεί ένα χρώμα μέσω των τριών συντεταγμένων του (R,G, & B), όπου η κάθε συντεταγμένη δείχνει την περιεκτικότητα σε κάθε χρώμα (Red, Green & Blue).

Γεωμετρικά, το RGB μοντέλο ανήκει στις γεωμετρικές παραστάσεις, αναπαριστώμενο από τον RGB Cube (τριχρωματικό κύβο), όπου η κάθε διάσταση του κύβου, αντιστοιχεί σε ένα από τα τρία βασικά χρώματα του μοντέλου. Οι δε ακμές του κύβου, αναπαριστούν το συνδυασμό των τριών χρωμάτων μεταξύ τους, εικόνα (9α,β).



(9α) Χρωματικός Κύβος



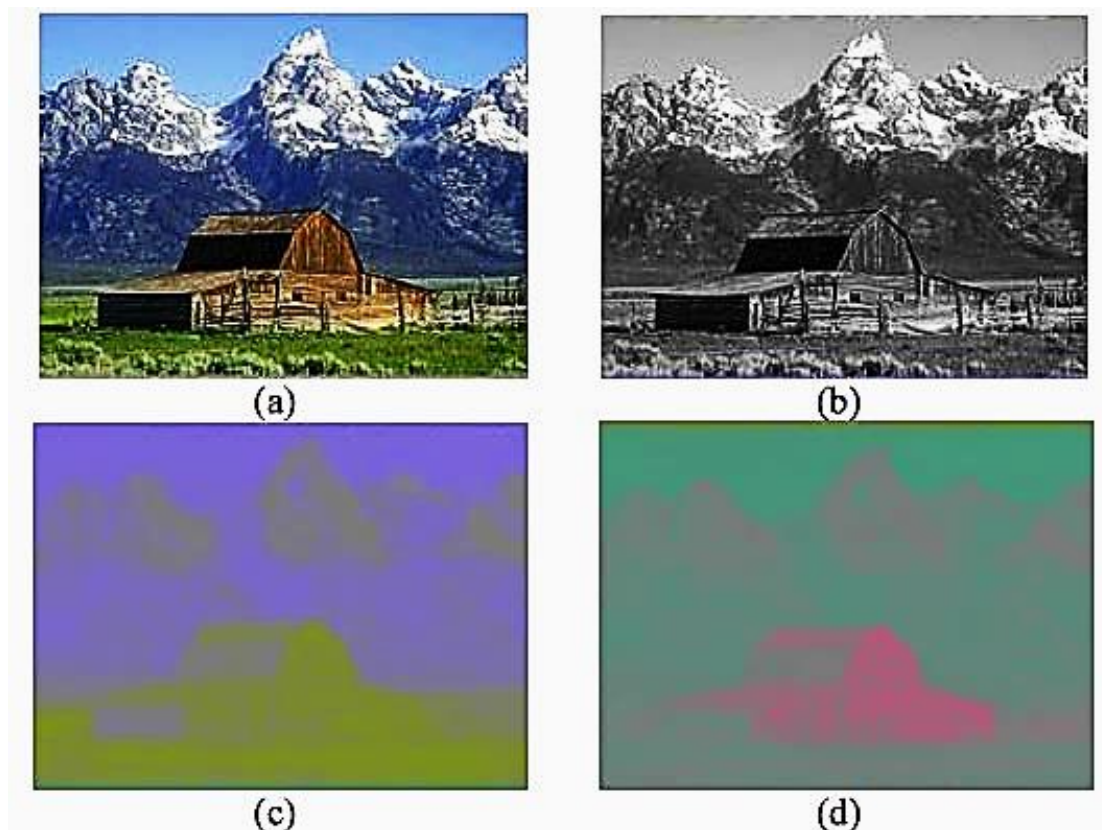
(9β) Ακμές Κύβου

Στην περίπτωση όπου έχουμε 24-bit χρώμα, τότε οι τιμές όπου παίρνει το κάθε χρώμα, είναι από 0 έως 255 και ο κάθε άξονας του κύβου, δίνει πλέον διαφορετικές τιμές στο κάθε χρώμα. Αυτό σημαίνει ότι μπορούμε να αναπαραστήσουμε 256^3 ή περίπου 16,7 εκατομμύρια διαφορετικά χρώματα. Έτσι η τιμή 0 σε ένα άξονα,

σημαίνει ότι αυτό το χρώμα εμφανίζει την ελάχιστη φωτεινότητα, ενώ η τιμή 255 τη μέγιστη φωτεινότητα που μπορεί να πάρει.

2.6.2 Ο χρωματικός χώρος YUV

Ο χρωματικός χώρος YUV αντιστοιχεί στο σύστημα για την κωδικοποίηση χρώματος που χρησιμοποιείται για την αναλογική τηλεόραση. Με το χώρο αυτό μπορούμε να ορίσουμε τη φωτεινότητα και το χρώμα με αριθμούς. Μπορούμε επίσης, να εξάγουμε από την αρχική εικόνα, μια άλλη σε αποχρώσεις του γκρι (Y) και μπορούμε να πάρουμε την πληροφορία χρώματος (U και V). Η πληροφορία χρώματος που εξάγουμε είναι η διαφορά χρώματος των σημάτων, δηλαδή στο μπλε αφαιρούμε τη φωτεινότητα ($B - Y$) και το ίδιο κάνουμε και στο κόκκινο ($R - Y$). Η μετατροπή μιας RGB εικόνας σε YUV παρουσιάζεται στην εικόνα (10).



Εικόνα (10). Η RGB εικόνα (a), Το στοιχείο Y (b), Το στοιχείο U (c), Το στοιχείο V (d)



Τα δεδομένα που λαμβάνουμε από μία βιντεοκάμερα στο χρωματικό χώρο RGB μέσα από μια διαδικασία που ονομάζεται χρωματική μετατροπή χώρου, μετατρέπονται σε σύνθετα αναλογικά σήματα (YUV) ή ψηφιακά YCbCr.

Η μετατροπή του RGB χρωματικού χώρου στον YUV γίνεται βάση της ακόλουθης εξίσωσης:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \begin{array}{l} Y \in [0, 255] \\ U \in [-111, 111] \\ V \in [-157, 157] \end{array}$$

2.6.3 Ο χρωματικός χώρος YCbCr

Ο χρωματικός χώρος YCbCr δημιουργήθηκε για τις ανάγκες του ψηφιακού βίντεο. Είναι ένας χώρος προσαρμοσμένος και αντισταθμισμένος με το χρωματικό χώρο YUV. Το Y αντιπροσωπεύει τη φωτεινότητα και τα Cb, Cr αντιπροσωπεύουν την πληροφορία χρώματος. Το Cb αναπαριστά τη χρωματική διαφορά ως προς το μπλε για παράδειγμα (B – Y) και το Cr εκφράζει τη χρωματική διαφορά ως προς το κόκκινο όπως (R – Y). Αξίζει να σημειωθεί ότι η εικόνα που εξάγουμε από το Y είναι στην ουσία ένα αντίγραφο της αρχικής αλλά σε κλίμακα αποχρώσεων του γκρι.

Ο μετασχηματισμός για τη μετατροπή από το χώρο RGB στο χώρο YCbCr προκύπτει από την εξίσωση :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

όπου στη σχέση αυτή οι τιμές που παίρνουν τα R,G,B βρίσκονται στο διάστημα [0,1] και η τιμή που παίρνει το Y βρίσκεται στο διάστημα [16, 240].

2.6.4 Οι χρωματικοί χώροι LUV, LAB

Οι χώροι LUV και LAB, αποτελούν μη γραμμικούς μετασχηματισμούς του χρωματικού χώρου XYZ. Είναι σχεδιασμένοι έτσι ώστε να υπάρχει μια περισσότερο



ομοιόμορφη σχέση ανάμεσα στις γεωμετρικές και στις αντιληπτές αποστάσεις αναμεταξύ των χρωμάτων και όταν αυτά εξετάζονται κάτω από τον ίδιο φωτισμό.

Ο χρωματικός χώρος LAB, είναι το καθαυτό χρωματικό μοντέλο που περιγράφει όλα τα ορατά χρώματα. Κάθε χρώμα έχει τρεις παραμέτρους και μπορεί να περιγραφεί με τις τιμές (L), (a) και (B). Η πρώτη παράμετρος (L), αναπαριστά τη φωτεινότητα του χρώματος, με την μικρότερη τιμή να αντιστοιχεί στο μαύρο χρώμα. Η δεύτερη παράμετρος (a), αναπαριστά τη θέση του χρώματος ανάμεσα στο κόκκινο και το πράσινο χρώμα, με την μικρότερη τιμή να αντιστοιχεί στο πράσινο χρώμα. Η Τρίτη και τελευταία παράμετρος (b), αναπαριστά τη θέση του χρώματος μεταξύ του κίτρινου και μπλε χρώματος, με την μικρότερη τιμή να αντιστοιχεί στο μπλε χρώμα. Και οι τρεις παράμετροι, αναπαριστούν τα χρώματα, πάντα με το ίδιο λευκό σημείο αναφοράς, επιτρέποντας έτσι τη διαχείριση των χρωμάτων κάτω από διαφορετικές συνθήκες φωτισμού [18].

2.6.5 Ο χρωματικός χώρος HSV

Ο χρωματικός χώρος HSV (Hue, Saturation, Value), εκμεταλλεύεται τον τρόπο που εμείς οι άνθρωποι αντιλαμβανόμαστε το χρώμα. Συγκεκριμένα, συνηθίζεται να περιγράψουμε τις διάφορες σκηνές, όχι σε συνθήκες κόκκινου, πράσινου και μπλε, αλλά ως απόχρωση, καθαρότητα και ένταση.

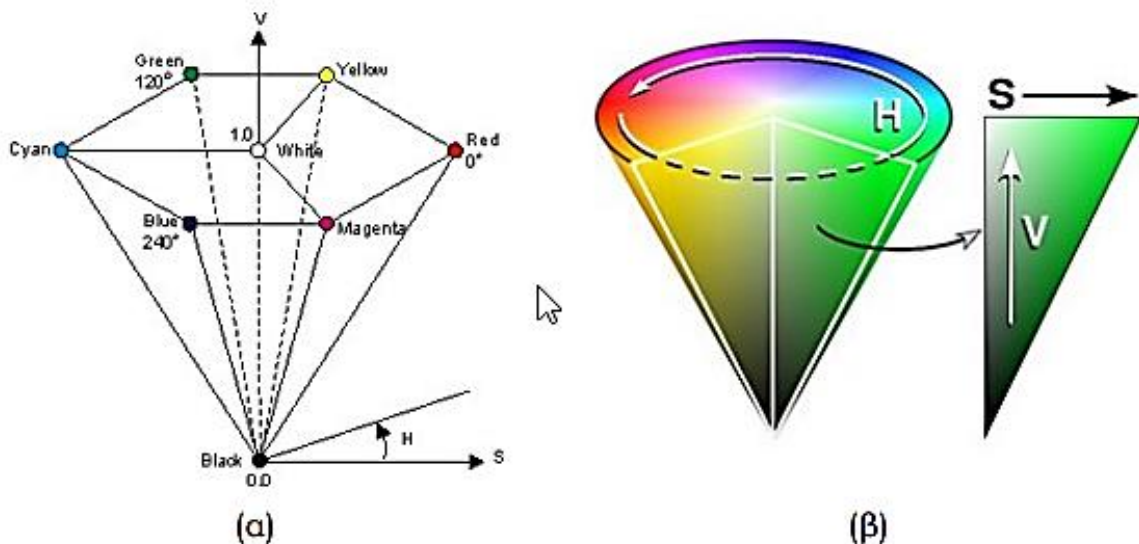
Βλέπουμε τα πράγματα ως χρώματα ή αποχρώσεις, οι οποίες είτε έχουν μια “ξεπλυμένη” όψη, είτε βαθύ και έντονο χαρακτήρα. Το Hue (απόχρωση), είναι το χρώμα που γίνεται αντιληπτό λόγω του μήκους κύματος. Το Saturation (καθαρότητα), είναι ο βαθμός καθαρότητας του χρώματος, δηλαδή το κατά πόσο το χρώμα έχει πρόσμιξη λευκού μέσα. Το Value (τιμή) αναφέρεται στο βαθμό μίξης ενός καθαρού χρώματος με το μαύρο.

Το σύνολο των τριών αυτών ιδιοτήτων μπορεί να παράγει οποιοδήποτε χρώμα βρίσκεται στη φύση. Η τρισδιάστατη αναπαράσταση του HSV, προκύπτει από τον

κύβο RGB. Αν κοιτάξουμε στον RGB κύβο κατά μήκος της διαγώνιου του γκρι, μπορούμε να δούμε ένα εξάγωνο, το οποίο είναι το HSV εξάγωνο.

Η απόχρωση δίνεται από τη γωνία με τον οριζόντιο άξονα με το κόκκινο στις 0° , το κίτρινο στις 60° , το πράσινο στις 120° , το κυανό στις 180° , το μπλέ στις 240° και το ματζέντα στις 300° . Να σημειωθεί ότι τα συμπληρωματικά χρώματα, έχουν 180° διαφορά.

Η χρωματική καθαρότητα κυμαίνεται μεταξύ $0.0 \leq S \leq 1.0$ και είναι ο λόγος της καθαρότητας μιας συγκεκριμένης απόχρωσης προς τη μέγιστη καθαρότητα ($S=1$). Όταν $S=0$, βρισκόμαστε στην κλίμακα του γκρι, δηλαδή στη διαγώνιο του RGB κύβου εικόνα (11).



Εικόνα (11). Σχηματική αναπαράσταση του χρωματικού χώρου HSV (α) HSV εξάγωνο, (β) Χρωματική άποψη του HSV εξαγώνου

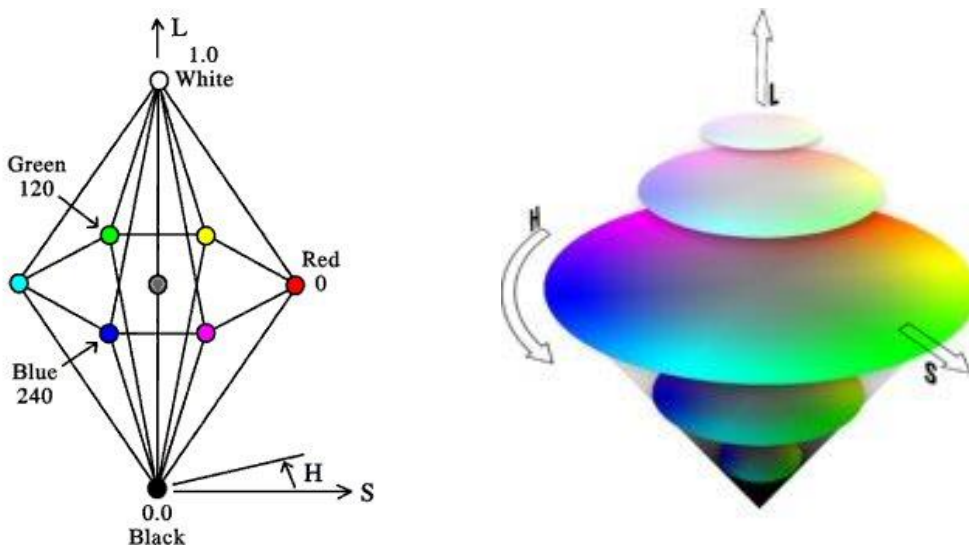
Για την επιλογή ενός χρώματος, διαλέγουμε αρχικά μια καθαρή απόχρωση (καθορίζουμε δηλαδή την τιμή του H και θέτουμε $S=V=1$). Στη συνέχεια προσθέτοντας μαύρο μειώνουμε την τιμή του V και προσθέτοντας άσπρο μειώνουμε το S .

Το HSV παρουσιάζει δυο βασικά πλεονεκτήματα. Πρώτον, η τιμή V είναι ανεξάρτητη από το χρώμα και δεύτερον η απόχρωση H και η χρωματική καθαρότητα S , είναι

στενά συσχετισμένες με τον τρόπο αντίληψης του χρώματος από το ανθρώπινο μάτι. Αυτά τα χαρακτηριστικά καθιστούν το μοντέλο HSV ιδανικό εργαλείο, για την ανάπτυξη αλγορίθμων επεξεργασίας εικόνας, βασισμένων στην αίσθηση χρώματος από το ανθρώπινο οπτικό σύστημα.

2.6.6 Οι χρωματικοί χώροι HIS & HLS

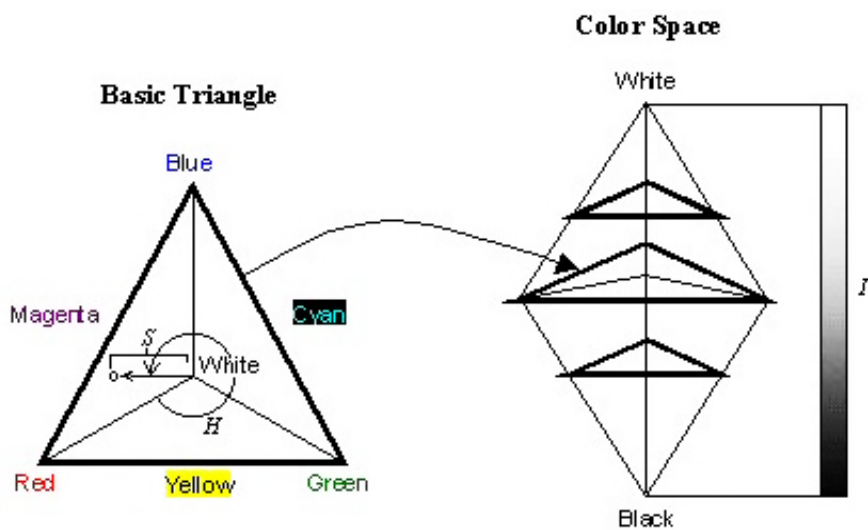
Το χρωματικό μοντέλο HLS αποτελείται από τις παραμέτρους απόχρωση (Hue), φωτεινότητα (Lightness) και καθαρότητα (Saturation). Ο ρόλος κάθε συνιστώσας αλλάζει ελαφρά σε σχέση με τον χώρο HSV. Ο τόνος έχει μεν την ίδια έννοια, αλλά εδώ μία γωνία 0° αντιστοιχεί σε μπλε χρώμα, ενώ το κόκκινο χρώμα βρίσκεται σε γωνία 120° . Όπως συμβαίνει και στο HSV μοντέλο, τα συμπληρωματικά χρώματα απέχουν γωνία 180° . Η καθαρότητα στο HLS μοντέλο υποδηλώνει την “αγνότητα” του τόνου. Όταν η καθαρότητα μειώνεται, ο τόνος γίνεται πιο γκρι. Μηδενική τιμή καθαρότητας σημαίνει απόχρωση του γκρι. Τέλος, η φωτεινότητα υποδηλώνει το ποσοστό μαύρου ή λευκού, που περιέχεται σε ένα χρώμα. Αυξανόμενη φωτεινότητα προσθέτει λευκό στον τόνο, ενώ αντίστοιχα όσο μειώνεται η φωτεινότητα προστίθεται μαύρο στον τόνο. Η μορφή του μοντέλου HLS φαίνεται στις εικόνες (12α, β).



Εικόνα (12). α) HLS δωδεκάεδρο (β) HLS διαμορφωμένος ως διπλός κώνος

Το χρωματικό μοντέλο HSI αποτελείται από τις παραμέτρους απόχρωση (Hue), καθαρότητα (Saturation) και φωτεινότητα (Intensity). Οι παράμετροι του χρώματος στο μοντέλο HSI ορίζονται στην επιφάνεια του τριγώνου που φαίνεται στην εικόνα (13α). Η απόχρωση H ενός χρώματος O είναι η γωνία που σχηματίζεται μεταξύ του άξονα του κόκκινου με το ευθύγραμμο τμήμα που ενώνει την αρχή (άσπρο) με το σημείο O . Όταν δηλαδή $H=0^\circ$, το χρώμα είναι κόκκινο, όταν $H=60^\circ$ το χρώμα είναι κίτρινο κτλ. Η χρωματική καθαρότητα S του χρώματος στο σημείο O είναι η απόσταση της ευθείας που ενώνει την αρχή με το σημείο O . Δείχνει δηλαδή το βαθμό στον οποίο το χρώμα αυτό είναι ανανεμιγμένο με το άσπρο. Όσο μεγαλύτερη είναι η απόσταση από την αρχή τόσο μεγαλύτερη είναι και η χρωματική καθαρότητα του χρώματος.

Η ένταση στο μοντέλο HSI ορίζεται σε σχέση με τη θέση που κατέχει πάνω σε μια γραμμή που διαπερνά το κέντρο (άσπρο). Εντάσεις που τοποθετούνται κάτω από την επιφάνεια του τριγώνου τείνουν στο μαύρο όσο απομακρύνονται, ενώ τείνουν στο άσπρο εντάσεις που είναι πάνω από την επιφάνεια. Συνδυάζοντας την απόχρωση, την χρωματική καθαρότητα και την ένταση σε έναν τρισδιάστατο χώρο, καταλήγουμε σε μια δομή με μορφή πυραμίδας, όπως φαίνεται στην εικόνα (13β).

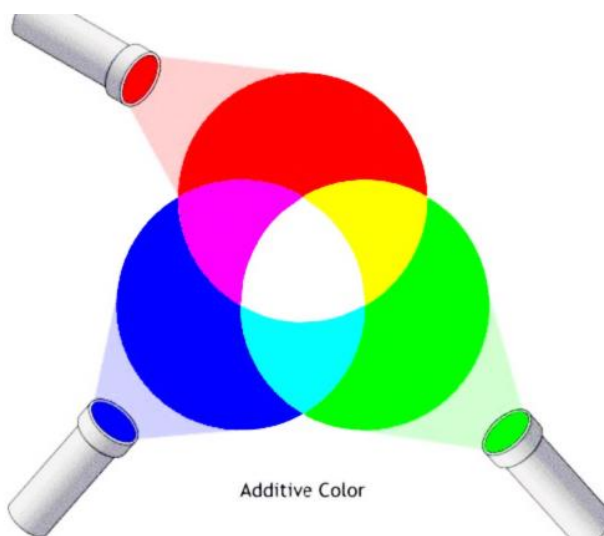


Εικόνα (13). α) Το τρίγωνο χρωμάτων HIS β) Η συνολική δομή του μοντέλου HSI

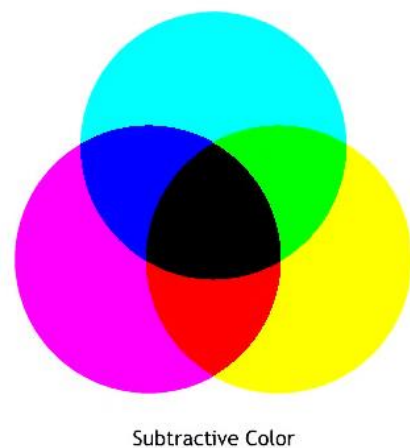
Οποιοδήποτε σημείο βρίσκεται πάνω στην εξωτερική επιφάνεια της δομής του μοντέλου, αντιστοιχεί σε χρώμα με πλήρη χρωματική καθαρότητα (εξαιρούνται το άσπρο όπου $S=0$ και το μαύρο όπου η παράμετρος S δεν ορίζεται). Η ένταση I εξαρτάται από τη θέση της κάθετης προβολής του σημείου πάνω στον κάθετο άξονα.

2.6.7 Χρωματικός χώρος CMY

Ο χρωματικός χώρος CMY (Cyan, Magenta, Yellow), σχετίζεται με το RGB, γιατί πρόκειται στην ουσία για συμπληρωματικά μοντέλα με τις ίδιες βασικές αρχές και ιδιότητες, εικόνα (14α, β). Όπως γνωρίζουμε, το RGB χρησιμοποιείται στις οθόνες των υπολογιστών, των οποίων η επιφάνεια στην οποία αναπαράγονται τα χρώματα, είναι μαύρου χρώματος. Γι' αυτό και η αρχή των αξόνων $(0, 0, 0)$ στο RGB είναι το μαύρο. δηλαδή ξεκινώντας από το μαύρο και προσθέτοντας κατάλληλες ποσότητες κόκκινου, πράσινου και μπλε, λαμβάνουμε όλα τα χρώματα. Το CMY επινοήθηκε για τις ανάγκες των εκτυπωτών στους οποίους και χρησιμοποιείται. Η διαφορά με το RGB έγκειται, ότι στο CMY η επιφάνεια που αναπαράγονται τα χρώματα είναι το χαρτί, το οποίο είναι άσπρο, σε αντίθεση με τη μαύρη οθόνη των υπολογιστών. Έτσι, οποιαδήποτε αναπαραγωγή χρωμάτων, ξεκινάει έχοντας ως βάση το λευκό.



Εικόνα (14). α) Αθροιστική Πρόσμιξη χρωμάτων RGB



β) Αφαιρετική Πρόσμιξη CMY



Στη θεωρία είναι δυνατό να δημιουργήσουμε κάθε χρωματική απόχρωση χρησιμοποιώντας τα RGB και CMY μοντέλα χρώματος. Στην πράξη όμως, τα τρία βασικά χρώματα, γαλάζιο, μωβ και κίτρινο δεν υπάρχουν διαθέσιμα ως αμιγή χρώματα, αλλά προκύπτουν πάντοτε μέσω πρόσμιξης με μια συγκεκριμένη αναλογία χρώματος από τα άλλα βασικά χρώματα. Το αποτέλεσμα είναι ότι δεν είναι δυνατό να τυπωθεί καθαρό μαύρο και περιορίζεται ο αριθμός των ικανών για εκτύπωση χρωμάτων. Με σκοπό να υπερπηδηθεί αυτό το πρόβλημα, το χρωματικό μοντέλο CMY έχει επεκταθεί για να σχηματίσει το μοντέλο CMYK. Η διαφορά μεταξύ των δύο αυτών μοντέλων είναι ελάχιστη. Στην πράξη όμως το αποτέλεσμα είναι σημαντικό, καθώς επιπρόσθετος, στα τρία βασικά χρώματα γαλάζιο, μωβ και κίτρινο, το μοντέλο χρωμάτων CMYK εμπεριέχεται και το μαύρο.

2.6.8 Περιληπτική σύγκριση των χρωματικών χώρων

- Ο χώρος RGB, είναι ο πιο διαδεδομένος χώρος ο οποίος είναι συμβατός με όλες τις οθόνες και τους αισθητήρες. Μειονέκτημα του είναι η κακή επεξεργασία εικόνας λόγω της υψηλής συσχέτισης που τον χαρακτηρίζει.
- Ο χώρος RGB παρέχει ανεξαρτησία στο χρώμα από φωτεινότητα και λάμψη αλλά μπορεί να παρουσιάσει προβλήματα θορύβου σε χαμηλή ένταση.
- Οι χώροι YUV και YCbCr μπορούν να υπολογιστούν εύκολα με μεγάλο πλεονέκτημα το Y, το οποίο βοηθά στην εύρεση ακμών. Μειονέκτημα τους είναι ότι χρησιμοποιούν γραμμικό μετασχηματισμό και ακατάλληλες αποστάσεις.
- Οι χώροι CMY & CMYK, καθώς και ο RGB, εξαρτώνται από τις συσκευές για τις οποίες προορίζονται να χρησιμοποιηθούν.

2.7 Σχήμα εικόνας (Shape)

Ένα ακόμα χαρακτηριστικό όπου το ανθρώπινο μάτι μπορεί να ξεχωρίσει και να αντιληφθεί, είναι το σχήμα (Shape) μέσα σε μια εικόνα [19], εξάγοντας έτσι αποτέλεσμα για το πιο αντικείμενο αντιπροσωπεύει. Όσο εύκολη λοιπόν είναι η



αναγνώριση του αντικειμένου από τον άνθρωπο, τόσο δύσκολη είναι για τη μηχανική υπολογιστική όραση, όπου και για το λόγω αυτό δεν χρησιμοποιείται αρκετά.

Υπάρχουν βέβαια εφαρμογές όπου χρησιμοποιούν το χαρακτηριστικό του σχήματος. Για να επιτευχθεί αναγνώριση μέσω του σχήματος, χρησιμοποιούνται τα χαρακτηριστικά όπου είναι οι ακμές, και μέσω των μεθόδων: Sobel – Fourier – Prewitt – Robert – Laplacian of Gaussian – Zero cross – Canny, οι σταθερές moment [20] κ.α., καθώς και με τμηματοποίηση της εικόνας, αλλά και με προσδιορισμό των ακμών, που αναλύουμε σε επόμενο κεφάλαιο.

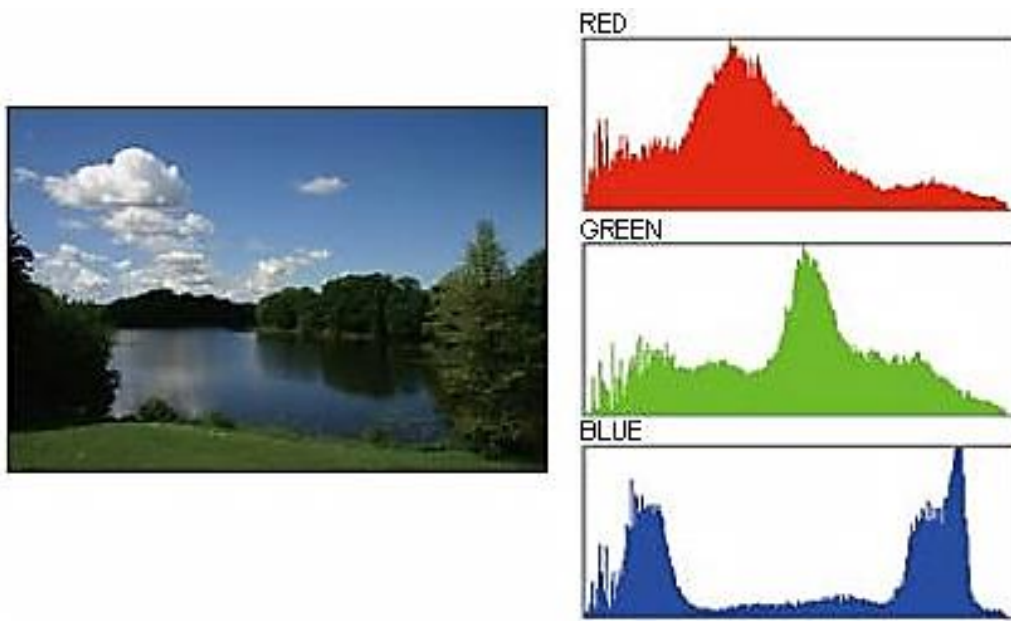
Τα σχήματα, χωρίζονται ανάλογα με τον τρόπο που εξάγονται, σε δύο κατηγορίες [20]. Αυτές είναι :

- Αυτά που βασίζονται στα όρια του σχήματος (Boundary-based), όπου χρειάζονται μόνο τα εξωτερικά όρια του αντικειμένου (περιγράμματα).
- Αυτά που βασίζονται στην περιοχή του σχήματος (Region-based), όπου λαμβάνεται υπόψη η συνολική περιοχή του σχήματος.

3. Χαρακτηριστικά χρώματος

3.1 Ιστόγραμμα Χρώματος Εικόνας

Η αναπαράσταση της κατανομής των τριών βασικών καναλιών χρώματος (Red, Green, Blue), καλείτε ιστόγραμμα χρώματος εικόνας. εικόνα (15). Η αναπαράσταση, μας δίνει σημαντικές πληροφορίες για τα χρώματα μιας εικόνας, χωρίς όμως να αποτελεί καλό τρόπο σύγκρισης δύο εικόνων. Το ιστόγραμμα χρώματος, αγνοεί τελείως το σχήμα και είναι ευαίσθητο στη μεταβολή φωτεινότητας.



εικόνα (15). Ιστόγραμμα Χρώματος εικόνας

3.2 Φωτεινότητα (Brightness)

Η φωτεινότητα μιας εικόνας, ορίζεται ως η μέση φωτεινότητα των εικονοστοιχείων της.

3.3 Κορεσμός (Saturation)

Ως κορεσμό, ορίζουμε τη χρωματική πληρότητα ενός χρώματος, σε σχέση με την φωτεινότητά της Εικόνα (16). Ποσοστό 100% του κορεσμού, αντιστοιχεί στο καθαρό χρώμα, ενώ ποσοστό 0%, αντιστοιχεί σε άσπρο, μαύρο, ή γκρι χρώμα, ανάλογα με την απόχρωση.



Εικόνα (16). Κορεσμός Μπλε χρώματος



4. Τεχνικές Ανάλυσης Εικόνας

4.1 Τεχνικές Τμηματοποίησης Εικόνων

Η τμηματοποίηση εικόνων [13], είναι ένα από τα πρωταρχικά και κύρια βήματα στην ανάλυση εικόνων και κυρίως για την αναγνώριση αντικειμένων. Βασικός σκοπός της, είναι η αναγνώριση ομοιογενών περιοχών της εικόνας, ως διακεκριμένων τμημάτων ή αντικειμένων.

Το πρόβλημα της τμηματοποίησης ή κατάτμησης εικόνων, μπορεί να οριστεί σε ένα γενικό επίπεδο, ως η αναζήτηση ενός διαχωρισμού της εικόνας σε ομοιογενείς περιοχές, κατά κάποιο τρόπο που να αντιστοιχεί στην ανθρώπινη αντίληψη. Η δυσκολία του προβλήματος βρίσκεται σε υψηλό επίπεδο, καθώς διαφορετικοί άνθρωποι μπορούν να κατατμήσουν την ίδια σκηνή με διαφορετικούς τρόπους, καθώς και τα κριτήρια με τα οποία ορίζεται η ομοιογένεια των περιοχών ποικίλουν από εικόνα σε εικόνα, καθώς είναι συχνά δύσκολη, η διατύπωση και ποσοτικοποίησή τους.

Η κλασική προσέγγιση του προβλήματος, κάνει την υπόθεση ότι τα τμήματα της εικόνας, μπορούν να ομαδοποιηθούν βάσει κάποιου χαρακτηριστικού χαμηλού επιπέδου. Στα χαρακτηριστικά χαμηλού επιπέδου, εντάσσονται η φωτεινότητα της εικόνας, το χρώμα, η υφή, ή η έξοδος μίας συστοιχίας φίλτρων. Τα κριτήρια που χρησιμοποιούνται στη διατύπωση των αλγορίθμων κατάτμησης βασίζονται στην ομοιογένειά των χαρακτηριστικών στο εσωτερικό των κατατμημένων περιοχών και εκατέρωθεν των ορίων τους, ενώ οι επιθυμητές λύσεις, μπορούν να περιοριστούν εισάγοντας επιπλέον όρους που αφορούν το μήκος των διαχωριστικών γραμμών και την έκταση αυτών των περιοχών.

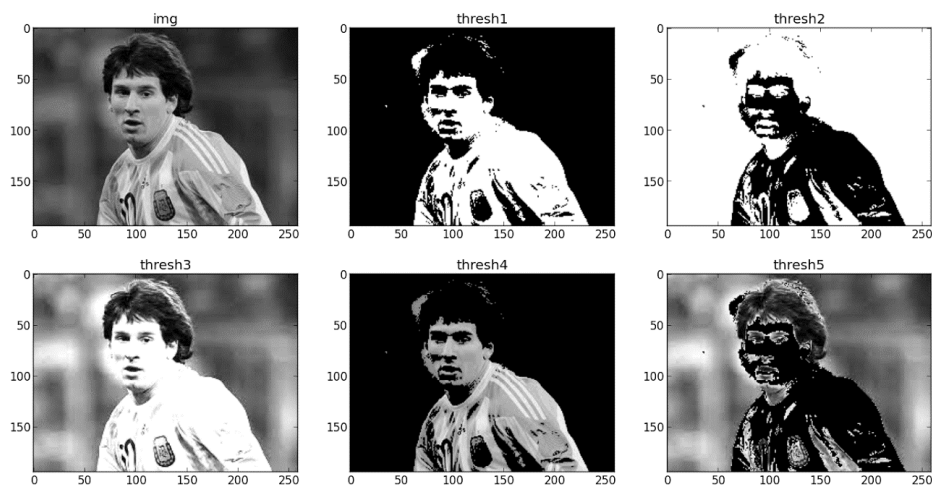
Οι παράμετροι που συσχετίζονται με την τμηματοποίηση εικόνων, είναι πολλές. Βασικά θέματα στη τμηματοποίηση είναι η επιλογή της πιο κατάλληλης μεθόδου διαχωρισμού των αντικειμένων από το φόντο, η μέτρηση της απόδοσής της, καθώς και η κατανόηση της εφαρμογής της στην τελική ανάλυση της εικόνας. Στις

παρακάτω παραγράφους αναλύονται οι βασικότερες μέθοδοι τμηματοποίησης της εικόνας.

4.1.1 Τεχνική Κατωφλίωσης Ιστογράμματος

Η κατωφλίωση (thresholding) [13], είναι η απλούστερη μέθοδος για τμηματοποίηση της εικόνας. Από μια grayscale εικόνα, είναι δυνατή η παραγωγή μιας δυαδικής (binary) εικόνας. Οι μέθοδοι κατάτμησης ιστογράμματος (histogram-based methods), εικόνα (17), είναι πολύ αποδοτικές, ιδιαίτερα όταν συγκρίνονται με άλλες μεθόδους κατάτμησης εικόνας και αυτό επειδή απαιτούν, μόνο μια σάρωση όλων των pixel της εικόνας.

Σε αυτήν την τεχνική, κατασκευάζεται ένα ιστόγραμμα από όλα τα pixel στην εικόνα. Οι λεγόμενες «κοιλιάδες» και οι «κορυφές» στο ιστόγραμμα χρησιμοποιούνται για να εντοπίσουν τις συστάδες (clusters) στην εικόνα. Ως μέτρο, μπορεί να χρησιμοποιηθεί το χρώμα ή η ένταση του κάθε pixel. Η κατάτμηση γίνεται με δύο τρόπους, όπου είναι μέσω του χρώματος. Έτσι έχουμε μονοδιάστατο ιστόγραμμα της εικόνας για μονόχρωμες και πολυδιάστατο ιστόγραμμα για έγχρωμες εικόνες. Η λογική που ακολουθείτε, είναι ότι η ανίχνευση κορυφών ή κοιλιάδων στο ιστόγραμμα, οδηγούν στην εύρεση αντικειμένων και του φόντου αντίστοιχα.



Εικόνα (17). Κατωφλίωση Ιστογράμματος



Στις πολύχρωμες εικόνες, η διαδικασία είναι πιο περίπλοκη, λόγω των τριών ιστογραμμάτων του χρωματικού χώρου, όπου είναι το R.G.B. (Red Green Blue – Κόκκινο Πράσινο Μπλε). Το αποτέλεσμα είναι ότι η επεξεργασία γίνεται χωριστά σε κάθε χρωματικό κανάλι και στο τέλος μέσω του συνδυασμού των αποτελεσμάτων, επικρατεί η ισχυρότερη υπόθεση κατάτμησης, δηλαδή αυτή που μας δίνει το αντικείμενο που θέλουμε καλύτερα και μέσω του συνδυασμού απόχρωσής του.

Επίσης, οι αναπαραστάσεις H.S.V. (Hue Saturation Value - Απόχρωση Κορεσμός Μέγεθος) και H.S.I. (Hue Saturation Intensity - Απόχρωση Κορεσμός Ένταση), δίνουν το πλεονέκτημα της εύρεσης αντικειμένων με βάση την απόχρωση των χρωμάτων, αφού συνήθως τα αντικείμενα έχουν ομοιογενή χρώματα, ενώ με αυτό τον τρόπο, οι εναλλαγές σε φωτεινότητα λόγω προβλημάτων φωτισμού όπως σκιές, κυρτές και κοίλες επιφάνειες και άλλα, δεν επηρεάζουν το αποτέλεσμα. Κοινό πρόβλημα σε τέτοιες τεχνικές είναι η παρουσία θορύβου, όπου θα εμφανίσει κορυφές στο ιστόγραμμα, έτσι η ομαλοποίηση (smoothing) του ιστογράμματος ως προ επεξεργασία του αλγορίθμου κατάτμησης ιστογράμματος, μερικές φορές κρίνεται απαραίτητη.

Όταν η κατάτμηση γίνεται τώρα μέσω της φωτεινότητας του κάθε pixel, τα αντικείμενα γίνονται ευκολότερα αντιληπτά και λιγότερο σύνθετα. Μιλώντας για μια ασπρόμαυρη (grayscale) εικόνα, όπου φαίνεται πιο καθαρά η έννοια της φωτεινότητας, κάθε αντικείμενο με σχετικά μεγάλο μέγεθος δημιουργεί μια κατανομή pixel στο ιστόγραμμα της εικόνας γύρω από τη μέση τιμή φωτεινότητάς του.

4.1.2 Τεχνική με βάση την ομαδοποίηση

Οι τεχνικές κατάτμησης με τη χρήση ιστογράμματος κατωφλίου απαιτούν υπολογισμό του ιστογράμματος, πράγμα που χρειάζεται χρόνο και υπολογιστική



μνήμη. Για το λόγο αυτό προτάθηκε η τεχνική ομαδοποίησης σημείων στο χρωματικό χώρο.

Η τεχνική αυτή προϋποθέτει ότι ο αναλυτής προσαρμόζει τον επιθυμητό αριθμό των κλάσεων των εικονοστοιχείων με ένα σημείο βαρύτητας το οποίο είναι και το κέντρο της ομάδας και το οποίο σχετίζεται άμεσα με την κλάση στην οποία υπάγεται. Η τεχνική αυτή ελαχιστοποιεί τον ολικό διασκορπισμό μέσα στην κάθε κλάση.

Αρχικά οι τοποθεσίες των σημείων βαρύτητας καθορίζονται τυχαία από το χρωματικό χώρο. Στη συνέχεια μετά από μια επανάληψη τα σημεία αυτά ενημερώνονται σύμφωνα με τα παρακάτω:

- Το κάθε εικονοστοιχείο ανατίθεται στην κλάση της οποίας το κέντρο βάρους είναι το πλησιέστερο σε χρώμα.
- Το κέντρο βάρους κάθε κλάσης ενημερώνεται με το να λαμβάνει υπόψη τα εικονοστοιχεία που του ανατίθενται.
- Η διακύμανση των κέντρων βάρους προσδιορίζεται ανάμεσα στο προηγούμενο ($t - 1$) και το τρέχον βήμα t από τον τύπο στην παρακάτω εξίσωση :

$$\varepsilon = \sum_{j=1 \dots N\omega} \| C_j(t) - C_j(t - 1) \|^2$$

- Αν η επανάληψη γίνει μεγαλύτερη από κάποιο κατώφλι τότε η διαδικασία επιτυγχάνει μια νέα επανάληψη.

4.1.3 Τεχνικές Τεχνητών νευρωνικών δικτύων – Neural Networks

Τα τεχνητά νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν στην κατάτμηση εικόνας με βάση το χρώμα. Διαφέρουν από τις άλλες τεχνικές στο ότι χρησιμοποιούν μια μη γραμμική εκτέλεση λειτουργιών κατάτμησης για να καταλήξουν στην τελική

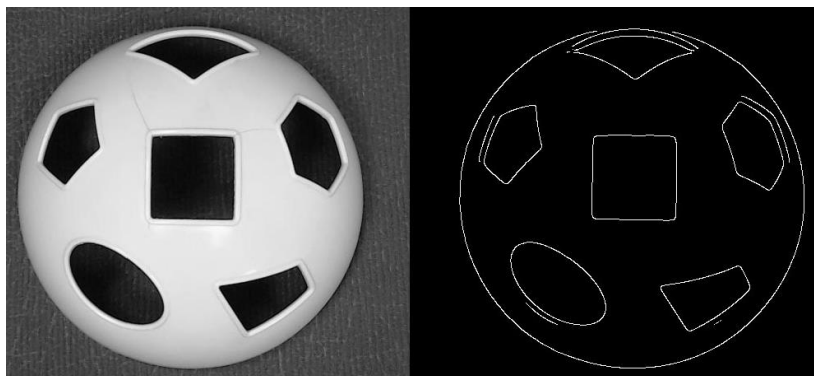
υπόθεση. Οι περιοχές στην εικόνα αναπαριστούνται από τους πολλαπλούς νευρώνες μέσω των διασυνδέσεων τους.

Την τεχνική αναγνώρισης μέσω Μηχανικής Μάθησης και νευρωνικών δικτύων, θα την αναλύσουμε σε επόμενο κεφάλαιο, καθώς θα χρησιμοποιηθεί ως κύρια τεχνική για την αναγνώριση αντικειμένων.

4.1.4 Αναγνώριση Ακμών

Η ανίχνευση και η αναγνώριση ακμών [14], αποτελούν ένα από τα βασικότερα εργαλεία της επεξεργασίας εικόνας. Μέσω αυτών των εργαλείων, επιτυγχάνεται η ανίχνευση και η εξαγωγή χαρακτηριστικών από την εικόνα, για τον εντοπισμό σημείων ή αντικειμένων, μέσα σε αυτή.

Οι περισσότερες μέθοδοι ανίχνευσης ακμών, στηρίζονται στον εντοπισμό σημείων με αξιόλογες μεταβολές στην φωτεινότητα, ασυνέχειες στο βάθος (κλίμακα του γκρι) ή στον προσανατολισμό (γωνίας), καθώς και αλλαγές στις ιδιότητες των υλικών, εικόνα (18).



Εικόνα (18). Ανίχνευση ακμών

Η εφαρμογή της ανίχνευσης ακμών σε μια εικόνα, (σε ιδανική περίπτωση), οδηγεί στη δημιουργία ενός συνόλου συνδεδεμένων καμπυλών όπου δείχνουν τα όρια των αντικειμένων και των επιφανειών, καθώς επίσης και τις καμπύλες που αντιστοιχούν σε ασυνέχειες προσανατολισμού.

Ένα συνηθισμένο πρόβλημα που συναντάται συχνά, είναι ότι σε λαμβανόμενες σε πραγματικό πεδίο εικόνες δύο διαστάσεων, (π.χ. φωτογραφίες ενός τοπίου), τα



αντικείμενα της εικόνας καλύπτονται από άλλα αντικείμενα μερικώς, με αποτέλεσμα να μην έχουμε ολοκληρωμένο το σχήμα τους και να δυσκολεύεται η ανίχνευσή τους.

Σε μια έγχρωμη εικόνα RGB, ο συνολικός χάρτης ακμών δίνεται από τον τύπο :

$$G(x, y) = \sqrt{G_R^2 + G_G^2 + G_B^2}$$

Με, τα G_R , G_G , G_B , να είναι οι χάρτες των τριών χρωματικών καναλιών

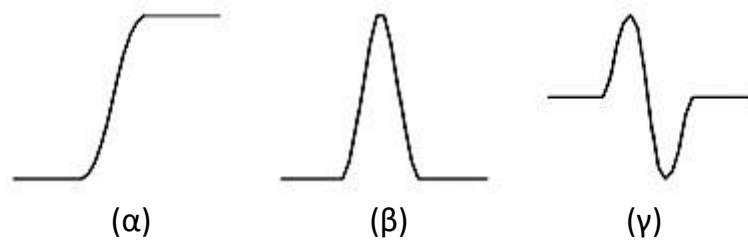
Για τη σωστή ανίχνευση των ακμών τώρα, πρέπει να ισχύει ένα κριτήριο, όπου το R θα πρέπει να είναι όσο μεγαλύτερο γίνεται και υπολογίζεται από τον τύπο :

$$R = \frac{1}{\max [I_i, I_A]} \sum_{i=1}^{I_A} \frac{1}{1 + ad_i^2}$$

Όπου:

- I_I να είναι ο ιδανικός αριθμός σημείων
- I_A να είναι ο αριθμός σημείων που ανίχνευσε η μέθοδος
- α να είναι ο παράγοντας κλιμάκωσης της εικόνας
- d_i να είναι η μετατόπιση του σημείου ακμής από την πραγματική του θέση

Η πληθώρα κριτηρίων για την ορθή ανίχνευση των ακμών, οδήγησε στη δημιουργία αρκετών μεθόδων για το σκοπό αυτό. Οι πιο ευρέως διαδεδομένοι μέθοδοι επεξεργασίας εικόνας με προσδιορισμό ακμών που χρησιμοποιούνται σήμερα, είναι οι μέθοδοι με χρήση πρώτων παραγώγων των Sobel, Prewitt και Roberts, η μέθοδος με χρήση δεύτερων παραγώγων και χρήση τελεστή Laplace (Laplacian of Gaussian, Zero-cross), καθώς και η μέθοδος με χρήση του αλγορίθμου εξαγωγής ακμών Canny. Παρακάτω δίνεται η αναπαράσταση μιας ακμής(α), της πρώτης παραγώγου της (β), καθώς και της δεύτερης παραγώγου της (γ). Εικόνα (19).



Εικόνα (19). Παράγωγοι ακμής

4.1.4.1 Μέθοδος Sobel

Σε αυτή τη μέθοδο, γίνεται η χρήση των πρώτων παραγώγων, όπως και στις μεθόδους Prewitt και Roberts. Εικόνα (20). Η αναπαράσταση της εικόνας είναι ως μια συνάρτηση $f(i,j)$, με μερική παράγωγο $G(i,j) \leftrightarrow [f(i,j) - f(i-1,j)] / T$. Για την υλοποίησή τους, χρησιμοποιούν μια μάσκα διαστάσεων 3×3 , με τιμές $x_i, i = 0, \dots, 8$ όπου αντιπροσωπεύουν τους συντελεστές με τους οποίους πολλαπλασιάζονται οι τιμές φωτεινότητας των εικονοστοιχείων, με το κεντρικό x_0 εικονοστοιχείο. Η μάσκα της μεθόδου είναι η :

$$L_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ και } L_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



Εικόνα (20). Μέθοδος Sobel

4.1.4.2 Μέθοδος Laplacian of Gaussian (LoG)

Ως αρχική μέθοδος, υπήρξε η Laplacian, (μετασχηματισμός Laplace), όπου όμως ήταν ευαίσθητη στο θόρυβο, δημιουργώντας έτσι περισσότερες ακμές από όσες

χρειάζονταν πραγματικά για την αναγνώριση. Έτσι, προστέθηκε στη μέθοδο αυτή ένα φίλτρο Gaussian, ώστε να επιφέρει στην εικόνα μια “Θολότητα” (blur) και έτσι να μειώνεται ο “θόρυβος” στην εικόνα. Εικόνα (21).

Η σειρά βημάτων που ακολουθείται είναι :

- Συνέλιξη τα αρχικής εικόνας I , με μια δισδιάστατη συνάρτηση Gaussian G
- Εφαρμογή της Laplacian συνάρτησης, όπου προκύπτει η εικόνα L .

$$\text{Φίλτρο LoG : } h(x, y) = C \left(\frac{2\sigma^2 - r^2}{\sigma^4} \right) e^{-\frac{r^2}{2\sigma^2}}$$

$$\text{Όπου, } r^2 = x^2 + y^2$$

C ο συντελεστής γενίκευσης της συνάρτησης,

σ η τυπική απόκλιση

- Στις ακμές ανήκουν εκείνα τα εικονοστοιχεία της L για τα οποία υπάρχει διασκελισμός του μηδενός.



Εικόνα (21). Μέθοδος Laplacian of Gaussian (LoG)

4.1.4.3 Μέθοδος Zero-Cross

Η μέθοδος Zero-Cross¹, προσδιορίζει τα σημεία μέσω του διασκελισμού του μηδενός. Δηλαδή, ανιχνεύει τα εικονοστοιχεία όπου είναι πλησιέστερα προς τις

¹ <http://homepages.inf.ed.ac.uk/rbf/HIPR2/zeros.htm>

θέσεις που έχουμε διασκελισμό του μηδενός, (απότομη και μεγάλη αλλαγή στη φωτεινότητα των εικονοστοιχείων). Εικόνα (22).



Εικόνα (22). Μέθοδος Zero - Cross

4.1.4.4 Μέθοδος Canny

Η μέθοδος Canny για ανίχνευση ακμών σε εικόνες, θεωρείται ο βέλτιστος που μπορούμε να ακολουθήσουμε για ανίχνευση ακμών, παρουσία του λευκού θορύβου. Πρόθεση της μεθόδου Canny, ήταν να βελτιώσει τους ήδη υπάρχοντες αλγόριθμους στην περιοχή της ανίχνευσης ακμών. Για να το πετύχει αυτό, όρισε κάποια κριτήρια για να αξιολογήσει την αποτελεσματικότητα των αλγόριθμων αυτών.

Πρώτο και πιο προφανές κριτήριο, ήταν η ελαχιστοποίηση του σφάλματος. Είναι πολύ σημαντικό να ανιχνεύονται όλες οι πραγματικές ακμές (πραγματική είναι μια ακμή που υφίσταται και στον τρισδιάστατο πραγματικό κόσμο), και ταυτόχρονα, να μην ανιχνεύονται ακμές που δεν υπάρχουν, ή να έχουμε "διπλές" ανιχνεύσεις σε μια ακμή.

Δεύτερο κριτήριο, ήταν οι ακμές να είναι σωστά τοποθετημένες τοπικά. Η απόσταση μεταξύ της πραγματικής ακμής και της ακμής που εντοπίζει ο αλγόριθμος, πρέπει να ελαχιστοποιηθεί. Επίσης η ακμή πρέπει να ορίζεται σαφώς



και όχι να παίρνει εκτεταμένες διαστάσεις.

Βασιζόμενη λοιπόν σε αυτά τα κριτήρια, η μέθοδος του Canny κατέληξε σε έναν αλγόριθμο, όπου αρχικά στην εικόνα εφαρμόζεται ένα γκαουσιανό ψηφιακό φίλτρο (gaussian). Αυτό στοχεύει στην ελαχιστοποίηση της επίδρασης του θορύβου, με τη διαδικασία αυτή να ονομάζεται ομαλοποίηση της εικόνας (smoothing).

Η ψηφιακή μορφή του φίλτρου, είναι ένα τετραγωνικό μητρώο συνέλιξης. Όσο μεγαλώνει η διάσταση του φίλτρου και η τυπική απόκλιση (σ) της γκαουσιανής δυσδιάστατης κατανομής, τόσο περισσότερο εξομαλύνεται η εικόνα και μειώνεται η επίδραση του λευκού θορύβου.

Πιο αναλυτικά, η εκτέλεση της μεθόδου ανίχνευσης ακμών Canny, απαιτεί τη σειριακή εκτέλεση μιας σειράς ενεργειών όπου είναι :

- a. Εφαρμογή Gaussian φίλτρου στην εικόνα, με $\mu=0$ και καθορισμένο εξ αρχής το σ (τυπική απόκλιση). Το δισδιάστατο φίλτρο Gaussian, δίνεται από τη σχέση :

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Το φιλτράρισμα, πραγματοποιείτε σε δύο βήματα, πρώτα ως προς τον άξονα x και έπειτα ως προς τον άξονα y

- b. Προσδιορισμός κλίσεως της φιλτραρισμένης πλέον εικόνας, όπου επιτυγχάνεται μέσω της εφαρμογής масκών, (π.χ. Sobel). Ακολούθως, προσδιορίζονται τα πλάτη και οι κλίσεις κάθε εικονοστοιχείου.
- c. Σε αυτό το βήμα, γίνεται καταστολή των μη μέγιστων τιμών. Από τον προηγούμενο υπολογισμό πλατών, εξαλείφονται οι ακμές που έχουν τοπικά μικρότερη τιμή, κάτι που οδηγεί σε λέπτυνση αλλά και φιλτράρισμα ακμών.

4.1.5 Μεγέθυνση Περιοχής

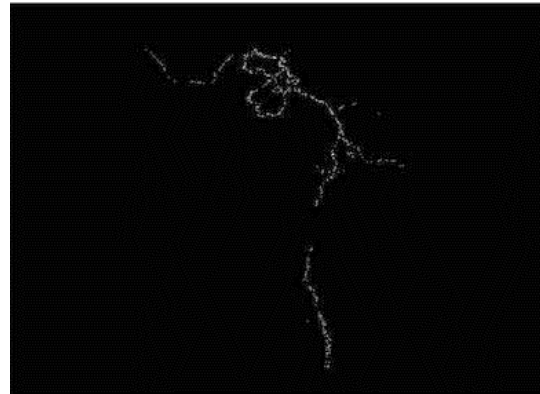
Η πρώτη μέθοδος κατάτμησης αυξανόμενης περιοχής (region growing method) ήταν η αυξανόμενη περιοχή με σπόρο (seeded region growing). Στις τεχνικές αυτής

της κατηγορίας, η επεξεργασία αρχίζει με τον καθορισμό κάποιων αρχικών σημείων στην εικόνα που ονομάζονται σπόροι (seeds). Κάθε ένα από αυτά καθορίζει μια περιοχή (region) και την επεκτείνει προσθέτοντας γειτονικά σημεία, εφόσον ικανοποιούν ένα προκαθορισμένο κριτήριο ομοιογένειας, λογική του οποίου είναι να ξεχωρίζει στα γειτονικά σημεία, ποια είναι όμοια ή όχι.

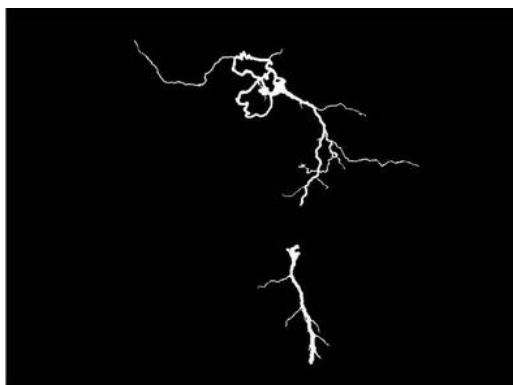
Οι περιοχές, λοιπόν, αυξάνονται συνεχώς με τη σύγκριση όλων των γειτονικών pixels των σπόρων. Η επέκταση αυτή σταματάει όταν κανένα άλλο σημείο δεν μπορεί να προστεθεί. Η διαφορά μεταξύ του μεγέθους της έντασης ενός pixel και του κέντρου της περιοχής, που συνήθως ονομάζεται δ , μπορεί να χρησιμοποιηθεί ως μέτρο συσταδοποίησης. Το pixel με τη μικρότερη διαφορά συνδέεται με την περιοχή. Αυτή η διαδικασία συνεχίζεται έως ότου συνδεθούν όλα τα pixels στην περιοχή αυτή. Εικόνες (23α, β, γ, δ).



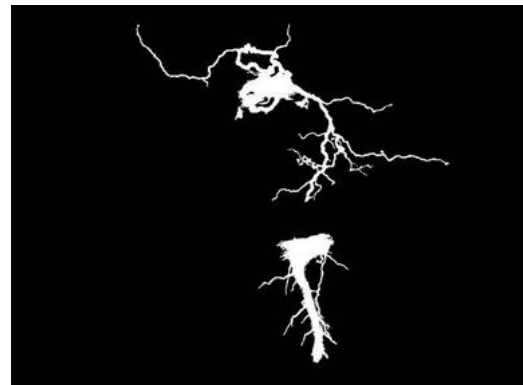
Εικόνα (23α)



Εικόνα (23β)



Εικόνα (23γ)



Εικόνα (23δ)



Μια παραλλαγή αυτής της τεχνικής, που προτείνεται από τους Haralick και Shapiro (1985), είναι βασισμένη στις εντάσεις των pixel. Το κέντρο με τη διασπορά της περιοχής και η ένταση του υποψήφιου pixel χρησιμοποιούνται για να συνυπολογίσουν μια δοκιμαστική στατιστική. Εάν αυτή είναι αρκετά μικρή, το pixel προστίθεται στην περιοχή, και το κέντρο με τη διασπορά της περιοχής υπολογίζονται εκ νέου. Διαφορετικά, το pixel απορρίπτεται, και χρησιμοποιείται για να διαμορφώσει μια νέα περιοχή.

4.1.6 Τεχνικές ασαφούς λογικής - Fuzzy Techniques

Οι τεχνικές κατάτμησης που προαναφέραμε, λαμβάνουν καθαρές αποφάσεις σε ότι έχει να κάνει με τις περιφέρειες (regions). Παρόλα αυτά, οι περιφέρειες σε μια εικόνα δεν είναι πάντα διακριτές. Αν γίνει μια λανθασμένη υπόθεση στα αρχικά στάδια της επεξεργασίας τότε θα επηρεαστούν και όλα τα επόμενα στάδια που θα ακολουθήσουν αφού οι αποφάσεις σε κάθε επίπεδο βασίζονται στα αποτελέσματα του προηγούμενου επιπέδου.

Ένα σύστημα όρασης μηχανής πρέπει να έχει επαρκή ευελιξία για την επεξεργασία της αβεβαιότητας σε όλα τα επίπεδα έτσι ώστε σε περίπτωση λαθών να μην καταδικάζεται το αποτέλεσμα.

Οι τεχνικές ασαφούς λογικής παρέχουν το μηχανισμό για να αναπαρασταθεί και να διαχειριστεί η αβεβαιότητα και η ασάφεια. Ο μηχανισμός αυτός προσφέρει τη δυνατότητα να μεταφέρονται οι αβέβαιες υποθέσεις και η εκτίμηση τους σε ανώτερα επίπεδα. Σε κάθε επίπεδο συγκρίνονται μεταξύ τους για να αναδείξουν την απόφαση που θα επικρατήσει στο τελευταίο στάδιο επεξεργασίας.

Μια από τις πιο δημοφιλείς τεχνικές ασαφούς λογικής είναι η F.C.M. Η τεχνική αυτή, επιτρέπει ασαφή όρια ανάμεσα στις περιοχές. Τα στάδια λειτουργίας του αλγορίθμου ξεκινούν με τον καθορισμό της αβεβαιότητας ενός εικονοστοιχείου σε σχέση με το σύμπλεγμα (cluster) στο οποίο ανήκει, μέσω της απόστασής του από το κέντρο του συμπλέγματος. Το επόμενο βήμα είναι ο επαναπροσδιορισμός των



συμπλεγμάτων και ο υπολογισμός του κέντρου τους. Τέλος, γίνεται επανάληψη των προηγούμενων βημάτων μέχρι η απόσταση που έχουν τα κέντρα των συμπλεγμάτων, να καταφέρει να φτάσει στο μέγιστο σημείο της.

4.1.7 Τεχνικές βασισμένες στη φυσική – Physics based

Οι τεχνικές κατάτμησης χρώματος με βάση τη φυσική, εκμεταλλεύονται τις ιδιότητες των υλικών με τη χρήση των μοντέλων της φυσικής και στοχεύουν στην ανίχνευση των ορίων των αντικειμένων, εξαλείφοντας παράλληλα τις ψευδείς ακμές που προέρχονται από σκιά ή φωτισμό.

Οι πιο διαδεδομένες τεχνικές κατάτμησης χρώματος με βάση τη φυσική, είναι το «διχρωματικό μοντέλο ανάκλασης – dichromatic reflection model» και το «ενδεικτικό μοντέλο ανάκλασης χρώματος - Approximate Color Reflectance Model (ACRM)». Το A.C.R.M. χωρίζει τα αντικείμενα σε δυο κύριες κλάσεις, α) τα οπτικά ομογενή υλικά όπως πλαστικά, υφάσματα και μπογιές, και β) τα οπτικά ανομοιογενή υλικά όπως μέταλλα και γυαλιά.

Τα μοντέλα φυσικής και δίνοντας κυρίως σημασία σε αντανάκλασεις, έχουν επιτυχή εφαρμογή σε συγκεκριμένους χώρους, στους οποίους είναι γνωστά από πριν τα φυσικά αντικείμενα της εικόνας και οι ιδιότητές τους.

4.2 Τμηματοποίηση για την ανάπτυξη λογισμικού αναγνώρισης αντικειμένων

Ένα από τα βασικότερα χαρακτηριστικά του περιεχόμενου μίας εικόνας είναι οι ακμές της. Οι ακμές της εικόνας συγκεντρώνουν, ουσιαστικά, το μεγαλύτερο μέρος της πληροφορίας εντός της εικόνας. Στο σύνολο της βιβλιογραφίας που πραγματεύεται το πρόβλημα της αναγνώρισης προτύπων, και της τμηματοποίησης γενικότερα είναι δύσκολο να εντοπιστεί προσέγγιση που να μη λαμβάνει υπόψη, είτε με άμεσο είτε με έμμεσο τρόπο, τις ακμές.

Πάρα πολλές μελέτες, έχουν αναδείξει την αξία των ακμών στην αναγνώριση αντικειμένων και την τμηματοποίηση. Η αναπαράσταση μίας εικόνας με τις ακμές της παρουσιάζει το επιπρόσθετο πλεονέκτημα ότι το πλήθος δεδομένων, μειώνεται

σημαντικά χωρίς να χάνεται το κύριο περιεχόμενό της. Η ανίχνευση ακμών, απαιτεί μικρότερο υπολογιστικό κόστος και οδηγεί πολλές φορές, σε καλύτερο αποτέλεσμα ειδικά όταν το ζητούμενο, είναι αποκλειστικά η ορθή τμηματοποίηση.

- 1 Στα πλαίσια της παρούσας εργασίας, θα αναπτυχθεί λογισμικό για την αναγνώριση αντικειμένων μέσα σε μία σειρά συνεχόμενων ψηφιακών εικόνων (Video). Η τεχνική που θα χρησιμοποιηθεί σε αυτό το λογισμικό θα είναι η Haar-like features, των Viola & Jones. Η τεχνική αυτή θεωρείται η πιο διαδεδομένη και θα αναλυθεί σε επόμενο κεφάλαιο.





ROS



Open Source Robotics Foundation





5. Robotic Operating System (ROS)

Το ROS [21] είναι ένα μεταλειτουργικό σύστημα, το οποίο χρησιμοποιείται σήμερα από εκατοντάδες ερευνητικές ομάδες και εταιρείες του κλάδου της ρομποτικής. Επίσης λόγω της ευκολίας που παρέχει, το καθιστά ως το καλύτερο σημείο εισόδου στην ρομποτική για ερασιτέχνες αλλά και για εκπαιδευτικούς σκοπούς.

Το ROS, αναπτύχθηκε για πρώτη φορά το 2007 από το Stanford Artificial Intelligence Laboratory, με ονομασία *Switchyard*, ώστε να υποστηρίξει το έργο STAIR (STanford AI Robot), του ομώνυμου εργαστηρίου. Από το 2008 και μέχρι το 2013, η ανάπτυξη συνεχίστηκε αρχικά από το ερευνητικό ινστιτούτο ρομποτικής Willow Garage και εν συνεχεία με περισσότερα από είκοσι ερευνητικά εργαστήρια και ερευνητές, να συνεργάζονται με τους μηχανικούς της Willow Garage για τη περαιτέρω ανάπτυξή του. Το Φεβρουάριο του 2013, η ανάπτυξη του ROS πέρασε υπό την εποπτεία του Open Source Robotics Foundation (OSRF).

Το ίδρυμα ανοιχτού λογισμικού Ρομποτικής (OSRF), είναι μια ανεξάρτητη μη κερδοσκοπική εταιρεία Έρευνας & Ανάπτυξης. Η αποστολή της OSRF είναι να στηρίξει την ανάπτυξη, τη διανομή, και την υιοθέτηση του λογισμικού ανοικτού κώδικα, για χρήση σε ρομποτική έρευνα, την εκπαίδευση και την ανάπτυξη του τελικού προϊόντος.

Το ROS, προς το παρόν λειτουργεί μόνο σε πλατφόρμες που βασίζονται σε UNIX και κατά κύριο λόγο, σε συστήματα Ubuntu και Mac OS X. Άλλες πλατφόρμες που υποστηρίζουν το ROS και που έχουν αναπτυχθεί από την κοινότητα, είναι το Fedora, το Gentoo, το Arch κ.α.

Ο πυρήνας του συστήματος ROS, μαζί με τα εργαλεία και τις βιβλιοθήκες, αναβαθμίζονται τακτικά και κυκλοφορούν ως διανομές ROS [22] με διαφορετικά ονόματα. Οι διανομές αυτές με χρονολογική σειρά φαίνονται στην εικόνα (24).

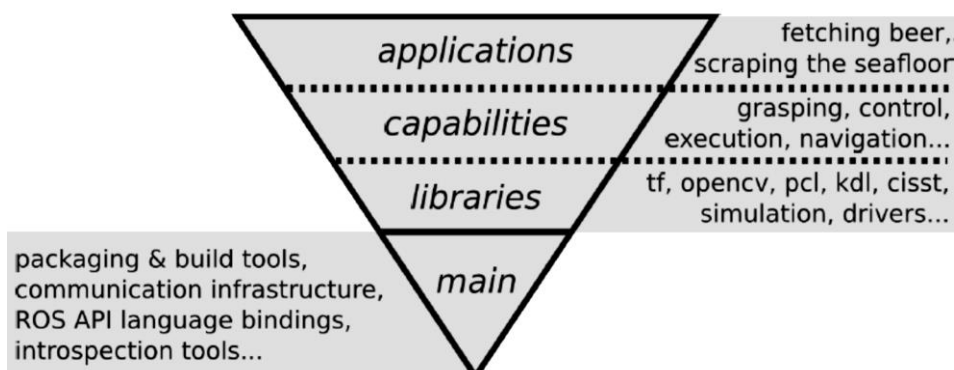
Distro	Release date	Turtle	EOL date
ROS Kinetic Kame (Recommended)	May 23rd, 2016		May, 2021
ROS Jade Turtle	May 23rd, 2015		May, 2017
ROS Indigo Igloo	July 22nd, 2014		April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013		May, 2015
ROS Groovy Galapagos	December 31, 2012		July, 2014
ROS Fuerte Turtle	April 23, 2012		---
ROS Electric Emys	August 30, 2011		---
ROS Diamondback	March 2, 2011		---
ROS C Turtle	August 2, 2010		---
ROS Box Turtle	March 2, 2010		---

Εικόνα (24). Διανομές ROS

Το ROS όπως προαναφέραμε, είναι ένα μεταλειτουργικό σύστημα ανοιχτού κώδικα για ρομπότ. Προσφέρει υπηρεσίες ενός κανονικού λειτουργικού συστήματος, συμπεριλαμβάνοντας επίσης διάφορα επίπεδα, όπως:

- Αφαίρεσης υλικού (hardware abstraction)
- Έλεγχο συσκευής χαμηλού επιπέδου (low-level device control)
- Εφαρμογή των λειτουργιών που χρησιμοποιούνται για τη λειτουργία του ρομπότ
- Ανταλλαγή μηνυμάτων μεταξύ των διεργασιών
- Διαχείριση των πακέτων

Μια συνοπτική απεικόνιση των επιπέδων δίνεται στο παρακάτω εικόνα (25) :



Εικόνα (25). Επίπεδα ROS



Εκτός των ανωτέρω λειτουργιών, το ROS έχει ως πρωταρχικό του στόχο, την επαναχρησιμοποίηση του κώδικα στη ρομποτική έρευνα και ανάπτυξη. Αυτό το πετυχαίνει παρέχοντας μια σειρά από εργαλεία και βιβλιοθήκες, που χρησιμεύουν για τη λήψη, την κατασκευή, τη συγγραφή αλλά και την εκτέλεση κώδικα σε πολλούς διαφορετικούς υπολογιστές, λόγω της εφαρμογής τους κάτω από το πλαίσιο του ROS και που είναι κοινό για όλους τους υπολογιστές και τα λειτουργικά τους συστήματα.

Σχετικά με την επαναχρησιμοποίηση του κώδικα, το ROS υποστηρίζεται από ένα ομόσπονδο σύστημα αποθετηρίου κώδικα, επιτρέποντας έτσι την συνεργασία καθώς και τη διανομή του. Έτσι με αυτό το σχεδιασμό, από το επίπεδο συστήματος αρχείων, μέχρι το επίπεδο κοινότητας, επιτυγχάνεται η ανεξαρτησία αποφάσεων για τη συγγραφή κώδικα σε σχέση με την έρευνα και ανάπτυξη ρομποτικών εφαρμογών, όπου στο τέλος όλα μπορούν να συνενωθούν κάτω από την υποδομή του ROS.

5.1 Συστατικά στοιχεία του ROS

Τα συστατικά στοιχεία του ROS, αποτελούνται από ένα καταμεμημένο πλαίσιο διαδικασιών όπου είναι γνωστοί ως nodes - “κόμβοι”. Οι κόμβοι, στην ουσία είναι ένα μοναδικά σχεδιασμένο εκτελέσιμο κομμάτι μιας διαδικασίας, όπου μπορεί να είναι ένας, ή περισσότεροι. Έτσι όλες αυτές οι διαδικασίες, μπορούν να ομαδοποιηθούν σε packages - “πακέτα”, όπου είναι ο φάκελος που περιέχει των κώδικα, τα εκτελέσιμα αρχεία, έναν ή περισσότερους κόμβους και γενικά ότι αρχείο χρειάζεται για να εκτελεστεί το πρόγραμμα. Με τη σειρά τους, τα πακέτα οργανώνονται σε stacks - “στοίβες”, όπου είναι η ονομαστική συλλογή πακέτων με σκοπό τον εύκολο διαμοιρασμό τους.

5.2 Μια γενική ιδέα των επιπέδων του ROS

Η λειτουργία του ROS, είναι χωρισμένη σε επίπεδα. Έτσι υπάρχουν τρία εννοιολογικά επίπεδα, όπου είναι τα κάτωθι:

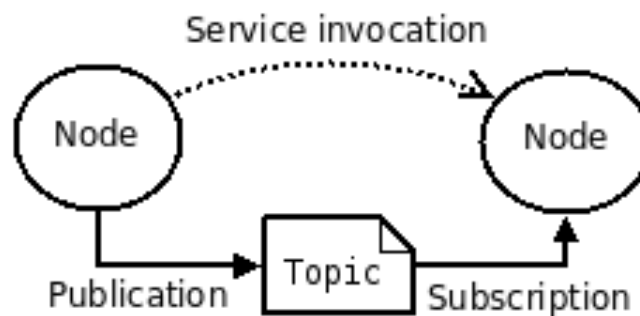
- Επίπεδο συστήματος Αρχείων
- Επίπεδο γραφικής παράστασης υπολογισμού
- Επίπεδο κοινότητας

5.2.1 Επίπεδο συστήματος Αρχείων

Στο επίπεδο συστήματος αρχείων, υπάρχουν οι πόροι του συστήματος που είναι αποθηκευμένοι στο δίσκο, όπως τα “πακέτα” (packages), τα “μέταπακέτα” (metapackages), τα “δηλωτικά αρχεία πακέτων” (Package Manifests), τα “μηνύματα” (message types, msg) και τέλος οι “υπηρεσίες” (service types, srv).

5.3 Επίπεδο γραφικής παράστασης υπολογισμού

Σε αυτό το επίπεδο, υπάρχουν οι “κόμβοι” (nodes), ο ROS Master, ο Parameter Sever (κομμάτι του Master), τα “μηνύματα” (messages), τα Topics, τα Services και τέλος τα Bags που είναι σημαντικά για την δημιουργία αλλά και το τεστ των αλγορίθμων, καθώς καταγράφουν, αποθηκεύουν και αναπαραγάγουν τα δεδομένα. Ένα παράδειγμα του πως επικοινωνούν οι κόμβοι γραφικά, φαίνεται στην Εικόνα (27).



Εικόνα (27). ROS Nodes Concept

Το επίπεδο αυτό προσφέρει τις πηγές για τις εφαρμογές του ROS, πετυχαίνοντας έτσι την εμπλοκή διαφόρων κοινοτήτων. με αποτέλεσμα το διαμοιρασμό εφαρμογών και τεχνογνωσίας. Σε αυτές ανήκουν οι “εκδόσεις” του ROS (Distributions), τα “αποθετήρια” εφαρμογών (Repositories), το ROS Wiki, οι ROS Mailing Lists, τα Q&A καθώς και το Blog (ros.org Blog).

Μέσω της συνεργασίας και των τριών επιπέδων, το ROS προσφέρει τη λειτουργικότητα που χρειάζεται μια ρομποτική συσκευή ή εφαρμογή.

5.4 Επίπεδο Κοινότητας ROS

Το επίπεδο κοινότητας του ROS, αποσκοπεί στην ένωση κάτω από μια πλατφόρμα, όλων των κοινοτήτων του ROS, με αποτέλεσμα το διαμοιρασμό εφαρμογών και τεχνογνωσίας. Σε αυτές ανήκουν οι “εκδόσεις” του ROS (Distributions), τα “αποθετήρια” εφαρμογών (Repositories), το ROS Wiki, οι ROS Mailing Lists, τα Q&A καθώς και το Blog (ros.org Blog).

Μέσω της συνεργασίας και των τριών επιπέδων, το ROS προσφέρει τη λειτουργικότητα που χρειάζεται μια ρομποτική συσκευή ή εφαρμογή.





6. Τεχνική Εκπαίδευσης Μοντέλου με Μηχανική Μάθηση

6.1 Εγκατάσταση TensorFlow και Anaconda Frameworks

Πρώτα εγκαθιστούμε το περιβάλλον Anaconda, και ενεργοποιούμε ένα εικονικό περιβάλλον χρήσης του, ώστε να γίνει χρήση της βιβλιοθήκης TensorFlow με GPU.

Τα βήματα όπου ακολουθούμε είναι:

1. Από το αρχικό μενού των Windows, εντοπίζουμε το command line του Anaconda και το ανοίγουμε δικαιώματα διαχειριστή. Ακολούθως από το anaconda terminal, δημιουργούμε το εικονικό μας περιβάλλον (tensor ή ότι όνομα θέλουμε), χρησιμοποιώντας την εντολή

```
C:\> conda create -n tensorflow1 pip python=3.5
```

2. Αφού δημιουργηθεί, ενεργοποιούμε το περιβάλλον με την ακόλουθη εντολή:

```
C:\> activate tensor
```

3. Ακολούθως εγκαθιστούμε το framework Tensorflow-gpu με την εντολή:

```
(tensor) C:\> pip install --ignore-installed --upgrade tensorflow-gpu
```

4. Εγκαθιστούμε και τα υπόλοιπα πακέτα βιβλιοθηκών με τις ακόλουθες εντολές:

- (tensor) C:\> conda install -c anaconda protobuf

- (tensor) C:\> pip install pillow

- (tensor) C:\> pip install lxml

- (tensor) C:\> pip install Cython

- (tensor) C:\> pip install jupyter

- (tensor) C:\> pip install matplotlib

- (tensor) C:\> pip install pandas

- (tensor) C:\> pip install opencv-python

(Σημείωση: Τα πακέτα 'pandas' και 'opencv-python' δεν χρειάζονται από το TensorFlow, αλλά χρησιμοποιούνται από τα Python scripts για τη δημιουργία των TFRecords, καθώς και για τη χρήση στο πρόγραμμα ανίχνευσης των αντικειμένων.

Ρύθμιση μεταβλητής περιβάλλοντος PYTHONPATH

Ακολούθως, θα πρέπει να δημιουργηθεί μια μεταβλητή PYTHONPATH όπου να δείχνει στα `\models`, `\models\research` και `\models\research\slim directories`. Αυτό επιτυγχάνεται εκτελώντας την ακόλουθη εντολή (από οποιονδήποτε κατάλογο):



```
(tensor) C:\> set
```

```
PYTHONPATH=C:\tensor\models;C:\tensor\models\research;C:\tensor\models\research\slim
```

(Προσοχή: Η εντολή για τη μεταβλητή `PYTHONPATH` πρέπει να εκτελείται κάθε φορά που θα χρησιμοποιείται το TensorFlow.)

Μεταγλώττιση αρχείων Protocol buffers και εγκατάσταση

Ως επόμενο στάδιο, μεταγλωττίζουμε τα αρχεία Protobuf, όπου χρησιμοποιούνται από το TensorFlow για τις ρυθμίσεις του μοντέλου και την παραμετροποίηση της εκπαίδευσης. Για την μεταγλώττιση, κάθε αρχείο `.proto`, στον κατάλογο `\object_detection\protos`, πρέπει να μεταγλωττιστεί ξεχωριστά με εντολή στο anaconda terminal.

Έτσι, στο Anaconda Command Prompt, μεταβαίνουμε στον κατάλογο `\models\research` και εκτελούμε μια-μια τις ακόλουθες εντολές:

```
protoc --python_out=..\object_detection\protos\anchor_generator.proto
```

```
protoc --python_out=..\object_detection\protos\argmax_matcher.proto
```

```
protoc --python_out=..\object_detection\protos\bipartite_matcher.proto
```

```
protoc --python_out=..\object_detection\protos\box_coder.proto
```

```
protoc --python_out=..\object_detection\protos\box_predictor.proto
```

```
protoc --python_out=..\object_detection\protos\eval.proto
```

```
protoc --python_out=..\object_detection\protos\faster_rcnn.proto
```

```
protoc --python_out=..\object_detection\protos\faster_rcnn_box_coder.proto
```

```
protoc --python_out=..\object_detection\protos\grid_anchor_generator.proto
```

```
protoc --python_out=..\object_detection\protos\hyperparams.proto
```

```
protoc --python_out=..\object_detection\protos\image_resizer.proto
```

```
protoc --python_out=..\object_detection\protos\input_reader.proto
```

```
protoc --python_out=..\object_detection\protos\losses.proto
```

```
protoc --python_out=..\object_detection\protos\matcher.proto
```

```
protoc --python_out=..\object_detection\protos\mean_stddev_box_coder.proto
```

```
protoc --python_out=..\object_detection\protos\model.proto
```

```
protoc --python_out=..\object_detection\protos\optimizer.proto
```

```
protoc --python_out=..\object_detection\protos\pipeline.proto
```

```
protoc --python_out=..\object_detection\protos\post_processing.proto
```

```
protoc --python_out=..\object_detection\protos\preprocessor.proto
```



```
protoc --python_out=..\object_detection\protos\region_similarity_calculator.proto
```

```
protoc --python_out=..\object_detection\protos\square_box_coder.proto
```

```
protoc --python_out=..\object_detection\protos\ssd.proto
```

```
protoc --python_out=..\object_detection\protos\ssd_anchor_generator.proto
```

```
protoc --python_out=..\object_detection\protos|string_int_label_map.proto
```

```
protoc --python_out=..\object_detection\protos\train.proto
```

```
protoc --python_out=..\object_detection\protos\keypoint_box_coder.proto
```

```
protoc --python_out=..\object_detection\protos\multiscale_anchor_generator.proto
```

```
protoc --python_out=..\object_detection\protos\graph_rewriter.proto
```

Αυτό θα δημιουργήσει αρχεία τύπου <<name>>_pb2.py από το κάθε αρχείο <<name>>.proto στο φάκελο \object_detection\protos.

Τέλος, εκτελούμε μέσα από το C:\tensorflow\models\research κατάλογο τις ακόλουθες εντολές:

```
(tensor) C:\tensorflow1\models\research> python setup.py build
```

```
(tensor) C:\tensorflow1\models\research> python setup.py install
```

Έλεγχος καλής λειτουργίας TensorFlow

Το TensorFlow Object Detection API έχει πλέον εγκατασταθεί και είμαστε έτοιμοι να δημιουργήσουμε δεδομένα ώστε να τα εκπαιδεύσουμε. Για τον έλεγχο ορθής λειτουργίας του TensorFlow, μπορούμε να εκτελέσουμε μέσα από τον κατάλογο \object_detection, το object_detection_tutorial.ipynb script με το Jupyter notebook και με την εντολή:

```
(tensor) C:\tensorflow1\models\research\object_detection> jupyter notebook  
object_detection_tutorial.ipynb
```

Με την εντολή αυτή, θα μας ανοίξει στον τοπικό μας browser το Jupiter notebook, όπου θα μπορούσαμε να εκτελέσουμε τον demo κώδικα γραμμή-γραμμή και ανά μπλοκ.

Αν όλα έχουν γίνει σωστά, τότε θα μας εμφανιστεί μια εικόνα με τα αντίστοιχα labels στα αντικείμενα που αναγνωρίζει.

6.2 Συλλογή και επεξεργασία φωτογραφιών

Με την επιτυχή εγκατάσταση του TensorFlow, είμαστε έτοιμοι να προχωρήσουμε στη συλλογή και επεξεργασία των φωτογραφιών όπου θα τροφοδοτήσουν την εκπαίδευση.

Συλλογή Φωτογραφιών

Το TensorFlow χρειάζεται μεγάλο πλήθος από φωτογραφίες που απεικονίζουν το αντικείμενο που θέλουμε να αναγνωρίσουμε, έτσι ώστε να μπορέσει να μας δώσει ένα



αξιόπιστο και λειτουργικό μοντέλο εκπαίδευσης. Για να φτιάξουμε ένα αξιόπιστο μοντέλο, οι φωτογραφίες πρέπει να απεικονίζουν και διάφορα άλλα αντικείμενα εκτός από το αντικείμενο που θέλουμε. Επίσης μια σειρά από διαφορετικά backgrounds και συνθήκες φωτισμού, θα έδιναν ακόμα καλύτερα αποτελέσματα.

Για τον ταξινομητή αναγνώρισης θυρών καθώς και την κατάσταση που βρίσκονται, (ανοιχτές - κλειστές), χρησιμοποιήθηκαν δύο διαφορετικές καταστάσεις μιας πόρτας. Αυτές ήταν οι:

- Open_door
- Closed_door

Έτσι, συλλέξαμε από διάφορες πηγές όπως το web, καθώς και φωτογραφίες όπου τραβήξαμε με δικά μας μέσα, στις δύο διαφορετικές καταστάσεις. Το σημαντικό ήταν, ότι οι φωτογραφίες απεικόνιζαν πόρτες με διαφορετικές αρχιτεκτονικές και όχι ένα σχήμα απλό παραλληλόγραμμο όπου έχουν οι κοινές πόρτες. Σε νούμερα, η συλλογή των φωτογραφιών ανήρθε στις 1200 φωτογραφίες, όπου ένα ποσοστό 80% χρησιμοποιήθηκε για την εκπαίδευση και τοποθετήθηκε στον κατάλογο `object_detection\images\train`. Το υπόλοιπο 20%, χρησιμοποιήθηκε για τον έλεγχο ορθότητας του μοντέλου και τοποθετήθηκε στον κατάλογο `object_detection\images\test`.

Κάτι που πρέπει να προσέξουμε, είναι ότι οι φωτογραφίες δεν θα πρέπει να ξεπερνάνε σε μέγεθος τα 200 Kb, καθώς και την ανάλυση να μην ξεπερνάει τα 1280X720 px.

Επεξεργασία Φωτογραφιών

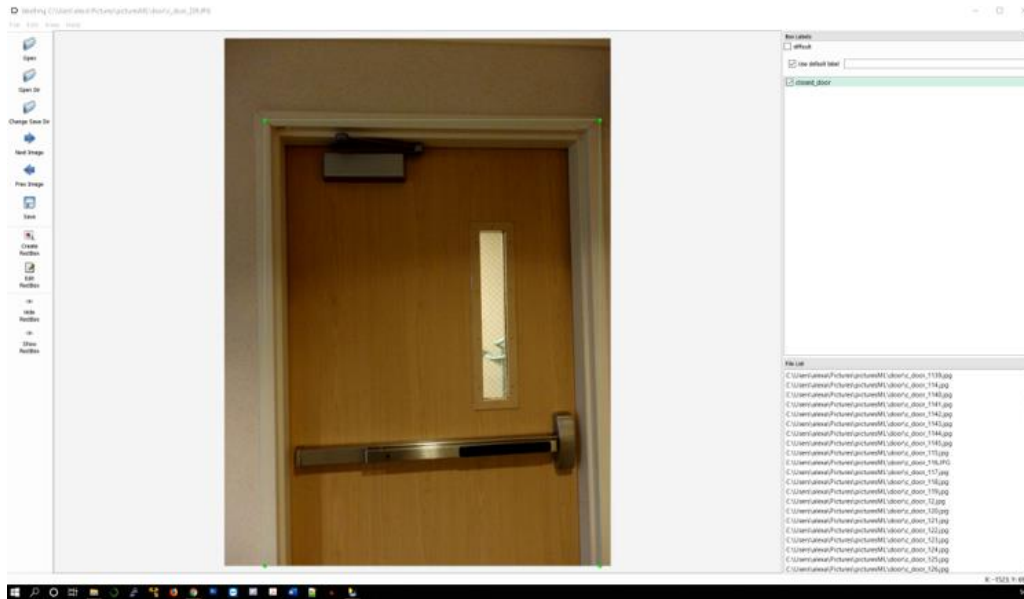
Αφού συλλέξαμε τις φωτογραφίες, ήρθε η ώρα να τις κάνουμε annotation, (να τις επεξεργαστούμε), ώστε να προσθέσουμε ετικέτες στο αντικείμενο που θέλουμε να αναγνωρίσουμε. Για το σκοπό αυτό, χρησιμοποιούμε το εργαλείο Labellmg από τον παρακάτω σύνδεσμο:

[Labellmg GitHub link \(GitHub - tzutalin/labellmg: Labellmg is a graphical image annotation tool and label object bounding boxes in images\)](#)

[Labellmg download link \(https://www.dropbox.com/s/tq7zfrcl44vxan/windows_v1.6.0.zip?dl=1\)](https://www.dropbox.com/s/tq7zfrcl44vxan/windows_v1.6.0.zip?dl=1)

Όταν εγκαταστήσουμε το εργαλείο, επιλέγουμε από το μενού του να διαβάζει τις φωτογραφίες από το φάκελο που είναι για εκπαίδευση. Όταν τελειώσουμε με αυτές, επαναλαμβάνουμε το ίδιο και για τις φωτογραφίες, όπου βρίσκονται στο φάκελο test. *Εικόνα (28)*.

Το Labellmg, θα δημιουργήσει για κάθε μία φωτογραφία, ένα αντίστοιχο αρχείο .xml. Σε αυτό το αρχείο, εγγράφονται τα δεδομένα της κάθε εικόνας μαζί με την ετικέτα τους. Επίσης, τα αρχεία .xml, θα χρησιμοποιηθούν για την παραγωγή των TFRecords, όπου με τη σειρά τους θα χρησιμοποιηθούν ως είσοδος στον εκπαιδευτή του TensorFlow.



Εικόνα (28). LabelImg

6.3 Παραγωγή Δεδομένων Εκπαίδευσης

Ως ακόλουθη διαδικασία μετά την ετικετοποίηση των φωτογραφιών μας, μπορούμε να προχωρήσουμε στην παραγωγή των TFRecords αρχείων που παρέχουν τα δεδομένα εισόδου για την εκπαίδευση του μοντέλου μας. Για το σκοπό αυτό θα χρησιμοποιήσουμε τα `rython scripts xml_to_csv.py` και `generate_tfrecord.py` με μερικές τροποποιήσεις, από τον παρακάτω σύνδεσμο: https://github.com/datitran/raccoon_dataset

Έτσι, από τον κατάλογο `\object_detection` εκτελούμε σε `anaconda terminal` την ακόλουθη εντολή: `(tensor) C:\tensor\models\research\object_detection> python xml_to_csv.py`

Η εντολή αυτή θα δημιουργήσει μέσα στον φάκελο `\object_detection\images` δύο αρχεία που είναι τα `train_labels.csv` και `test_labels.csv`

Ακολούθως, ανοίγουμε με έναν text editor το αρχείο `generate_tfrecord.py`, και το διαμορφώνουμε ως εξής:

```
# TO-DO replace this with label map
def class_text_to_int(row_label):
    if row_label == 'closed_door':
        return 1
    elif row_label == 'open_door':
        return 2
    else:
        return None
```



```
29
30 # TO-DO replace this with label map
31 def class_text_to_int(row_label):
32     if row_label == 'closed_door':
33         return 1
34     elif row_label == 'open_door':
35         return 2
36     else:
37         None
38
```

Μετά την τροποποίηση, εκτελούμε μέσα από τον κατάλογο `\object_detection` τις ακόλουθες εντολές όπου μας παράγουν τα αρχεία TFRecord:

Για τον φάκελο train: `python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --output_path=train.record`

Για τον φάκελο test: `python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --output_path=test.record`

Με την εκτέλεση των εντολών, θα δημιουργηθούν μέσα στο φάκελο `\object_detection` δύο αρχεία όπου είναι τα `train.record` και `test.record`, όπου είναι τα αρχεία που θα χρησιμοποιηθούν για την εκπαίδευση του ταξινομητή μας.

6.4 Δημιουργία ετικετών και παραμετροποίηση εκπαίδευσης

Το τελευταίο πράγμα που πρέπει να γίνει πριν ξεκινήσουμε την εκπαίδευση, είναι να δημιουργήσουμε τις ετικέτες και να παραμετροποιήσουμε τις ρυθμίσεις για την εκπαίδευση των δεδομένων μας.

6.4.1 Ετικέτες αντικειμένων

Με την ετικετοποίηση, στην ουσία καθοδηγούμε τον εκπαιδευτή στο να αποδίδει συγκεκριμένα ονόματα, (ετικέτες), στα αντικείμενα όπου εκπαιδεύεται να αναγνωρίζει.

Για να προδιαγράψουμε τις ετικέτες, δημιουργούμε ένα νέο αρχείο μέσα στο φάκελο `C:\tensor\models\research\object_detection\training`, με όνομα `labelmap.pbtxt`. Ανοίγουμε το αρχείο με έναν text editor και γράφουμε τα ακόλουθα:

```
item {
  id: 1
  name: 'closed_door'
}
```

```
item {
  id: 2
  name: 'open_door'
}
```

```
1  item {
2    id: 1
3    name: 'closed_door'
4  }
5
6  item {
7    id: 2
8    name: 'open_door'
9  }
```

Θα πρέπει να προσεχθεί τα ονόματα των ετικετών, να είναι ίδια με τα ονόματα που έχουμε δώσει στο αρχείο *generate_tfrecord.py*.

6.4.2 Παραμετροποίηση εκπαίδευσης

Σε αυτό το αρχείο, ορίζουμε το μοντέλο, καθώς και τις παραμέτρους για τη διαδικασία εκπαίδευσης. Αρχικώς, μεταβαίνουμε στο φάκελο *C:\tensor\models\research\object_detection\samples\configs* και αντιγράφουμε, (με copy-paste), το αρχείο του μοντέλου *faster_rcnn_inception_v2_pets.config* στο φάκελο *..\object_detection\training*. Ακολούθως, το ανοίγουμε με έναν text editor και κάνουμε τις ακόλουθες αλλαγές:

- Στη γραμμή 9, αλλάζουμε την παράμετρο *num_classes* βάζοντας τον αριθμό των διαφορετικών αντικειμένων που θα αναγνωρίζει ο ταξινομητής μας και που στη δική μας περίπτωση είναι 2.
- Στη γραμμή 110, αλλάζουμε το *fine_tune_checkpoint* σε:
fine_tune_checkpoint:
"C:/tensor/models/research/object_detection/faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt"
- Στις γραμμές 126 και 128, στην παράμετρο *train_input_reader* αλλάζουμε τα *input_path* και *label_map_path* σε:
input_path: *"C:/tensor/models/research/object_detection/train.record"* και
label_map_path:
"C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"
- Στη γραμμή 132, αλλάζουμε την παράμετρο *num_examples* βάζοντας τον αριθμό των φωτογραφιών που έχουμε στο φάκελο *..\images\test*
- Τέλος, στις γραμμές 140 και 142, αλλάζουμε τα *input_path* και *label_map_path* σε:
input_path: *"C:/tensor/models/research/object_detection/test.record"*
label_map_path:
"C:/tensor/models/research/object_detection/training/labelmap.pbtxt"

Αποθηκεύουμε το αρχείο και είμαστε έτοιμοι για τη διαδικασία εκπαίδευσης.

6.5 Εκτέλεση εκπαίδευσης

Μεταβαίνουμε στο terminal στον κατάλογο `object_detection` και εκτελούμε την ακόλουθη εντολή:

```
python train.py --logtostderr --train_dir=training/ --  
pipeline_config_path=training/faster_rcnn_inception_v2_pets.config
```

Σε κάθε στάδιο της διαδικασίας, παράγεται μια αναφορά σχετική με το *loss*, όπου στην αρχή της διαδικασίας, ξεκινάει με υψηλές τιμές, όπου κατά τη διάρκεια της εκπαίδευσης, μειώνονται συνεχώς. Μια τιμή αντιπροσωπευτική όπου μας δείχνει ότι η εκπαίδευση του μοντέλου μας είναι αρκετά καλή, είναι κάτω από 2. Τη διαδικασία εκπαίδευσης, το ρυθμό μείωσης του *loss*, καθώς και άλλες μετρικές, μπορούμε να τις δούμε μέσα από το περιβάλλον του *tensorboard*. Για να ενεργοποιήσουμε το περιβάλλον, ανοίγουμε ένα νέο terminal και μεταβαίνουμε στον κατάλογο:

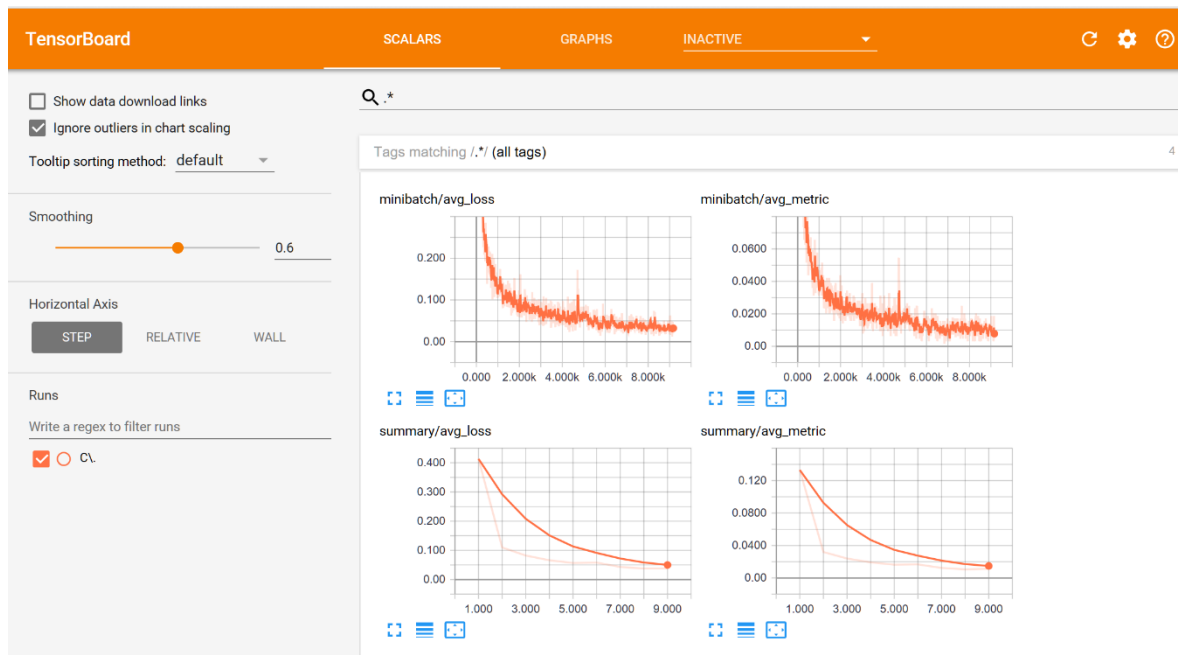
```
C:\tensorflow1\models\research\object_detection
```

Ακολουθώντας, εκτελούμε την εντολή:

```
(tensor) C:\tensor\models\research\object_detection>tensorboard --logdir=training
```

Με την εκτέλεση της εντολής ενεργοποιείται ένας localhost browser στην πόρτα 6006.

(<http://localhost:6006>).



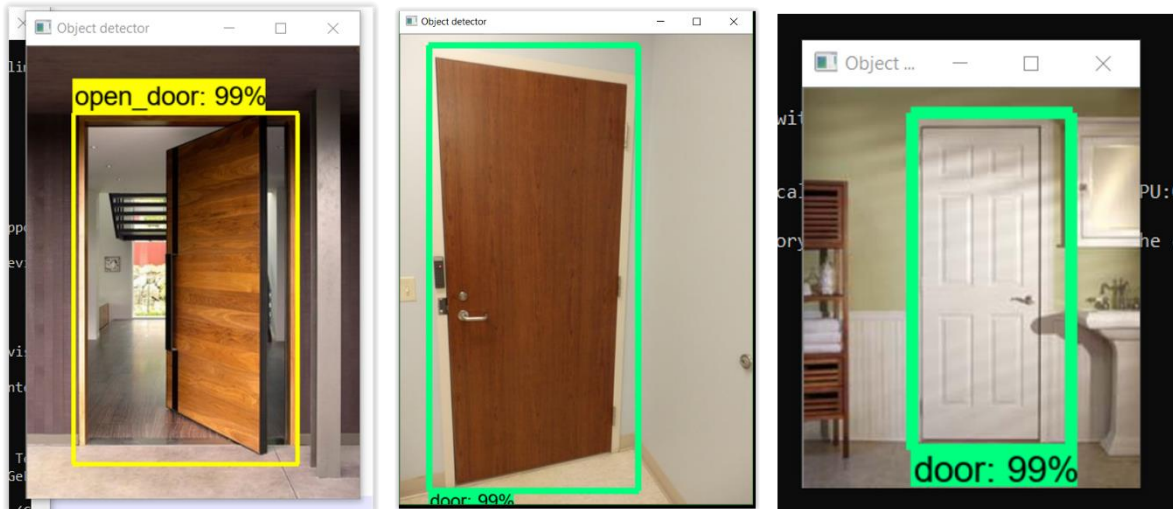
6.6 Εξαγωγή Inference graph

Από τη στιγμή που έχει τελειώσει η εκπαίδευση, δημιουργούμε το *frozen inference graph* (*.pb file*) μέσα από τον κατάλογο `object_detection`. Η εντολή που χρησιμοποιούμε είναι η ακόλουθη, (όπου "xxxx" στο "model.ckpt-xxxx" βάζουμε το .ckpt αρχείο με το μεγαλύτερο νούμερο που βρίσκεται στο φάκελο training):



```
python export_inference_graph.py --input_type image_tensor --pipeline_config_path  
training/faster_rcnn_inception_v2_pets.config --trained_checkpoint_prefix  
training/model.ckpt-XXXX --output_directory inference_graph
```

Αυτή η εντολή, θα μας δημιουργήσει το αρχείο *frozen_inference_graph.pb* στο φάκελο *\object_detection\inference_graph*. Το αρχείο .pb είναι αυτό που περιέχει τον ταξινομητή μας.



Βιβλιογραφία – Πηγές

- [1] McCulloch, W. and Pitts, W., “A logical calculus and the ideas immanent in the nervous activity”, Bulletin of Mathematical Biophysics, vol.5, 1943, <https://www.cs.cmu.edu/~epxing/Class/10715/reading/McCulloch.and.Pitts.pdf>
- [2] BAYES Theorem, Wikipedia, https://en.wikipedia.org/wiki/Bayes%27_theorem
- [3] Machine Learning in Action, http://library02.embl.de/InmagicGenie/DocumentFolder/TableOfContents_H897.pdf
- [4] K Nearest Neighbors (KNN), <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
- [5] Decision Trees, Wikipedia, https://en.wikipedia.org/wiki/Decision_tree_learning
- [6] D. Kragic, L. Petersson, and H. I. Christensen. Visually guided manipulation tasks. Robotics and Autonomous Systems, 40(2-3):193–203, August 2002.
- [7] People Detection and Tracking Through Stereo Vision for Human-Robot Interaction, Munoz – Salinas et al, 2005, https://link.springer.com/chapter/10.1007/11579427_34