

ΑΕΙ ΠΕΙΡΑΙΑ ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ



Α.Ε.Ι. Πειραιά Τ.Τ.

ΜΗΧΑΝΙΚΗ ΟΡΑΣΗ : ΒΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΗΝ ΕΞΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΑΣ

ΠΥΡΓΑ ΕΥΤΥΧΙΑ 44546

ΓΙΑΝΝΗΣ ΠΑΝΑΓΟΠΟΥΛΟΣ 44636

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ ΓΡΗΓΟΡΗΣ ΝΙΚΟΛΑΟΥ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΜΑΡΤΙΟΣ 2017

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

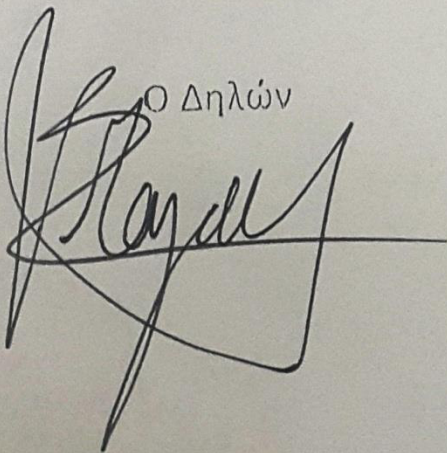
Ο / Η κάτωθι υπογεγραμμένος / η Μαναφίλουδης Ιωάννης.....
του Αιτωλίου....., με αριθμό μητρώου 44636..... φοιτητής / τριά του
Τμήματος **Μηχανικών Αυτοματισμού Τ.Ε.** του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

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

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

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

Ο Δηλών



Ημερομηνία

6/4 / 2017

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

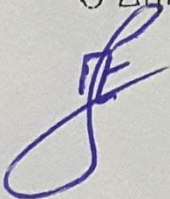
Ο / Η κάτωθι υπογεγραμμένος / η Πύρρα Ευτυχία,
του Φεωρηίου, με αριθμό μητρώου 44546 φοιτητής / τρια του
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

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

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

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

Ο Δηλών



Ημερομηνία

6/4/2017

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τους γονείς μας και τους φίλους μας για τη στήριξη τους στα χρόνια των σπουδών μας. Καθώς επίσης τον καθηγητή μας κύριο Γρηγόρη Νικολάου για τη καθοδήγηση του όπως επίσης και για την υπομονή που έδειξε κατά το διάστημα της εκπόνησης της παρούσας εργασίας.

Πίνακας περιεχομένων

Κατάλογος Εικόνων	5
Κατάλογος Πινάκων	6
Περίληψη	7
Εισαγωγή	
1.1 Μηχανική όραση - Ιστορική αναδρομή	9
1.2 Τύποι εικόνας	10
1.3 Χρωματικοί χώροι	11
1.4 Ιστόγραμμα εικόνων	13
1.5 Στάδια επεξεργασίας εικόνας	14
Κεφάλαιο 1^ο : Κατάτμηση Εικόνας (Image Segmentation)	
1.1 Τι είναι η κατάτμηση εικόνας	16
1.2 Μέθοδοι κατάτμησης βασισμένοι στις ομοιότητες	
1.2.1 Κατάτμηση εικόνας βασισμένη στις περιοχές	17
1.2.2 Κατάτμηση εικόνας με τον αλγόριθμο του υδροκρίτη	19
1.2.3 Κατάτμηση εικόνας με κατωφλίωση	20
1.3 Πρακτικά παραδείγματα με τη χρήση του matlab	
1.3.1 Κατάτμηση εικόνας με μεθόδους κατωφλίωσης	
1.3.1.1 Auto Thresholding	21
1.3.1.2 Adaptive Thresholding	22
1.3.1.3 Otsu's Thresholding Method	23
1.3.2 Κατάτμηση εικόνας βάση των τριών καναλιών του RGB φάσματος	26
Κεφάλαιο 2: Αναγνώριση Ακμών (Edge Detecting)	
2.1 Τι είναι η αναγνώριση ακμών	33
2.2 Μέθοδοι αναγνώρισης ακμών	
2.2.1 Αναγνώριση ακμών με τη μέθοδο Roberts	34
2.2.2 Αναγνώριση ακμών με τη μέθοδο Prewit	35
2.2.3 Αναγνώριση ακμών με τη μέθοδο Sobel	36
2.2.4 Αναγνώριση ακμών με τη μέθοδο Canny	37
2.3 Πρακτικά παραδείγματα με τη χρήση του matlab	
2.2.1 Αναγνώριση ακμών με τη μέθοδο Roberts	38
2.2.2 Αναγνώριση ακμών με τη μέθοδο Prewit	40
2.2.3 Αναγνώριση ακμών με τη μέθοδο Sobel	42
2.2.4 Αναγνώριση ακμών με τη μέθοδο Canny	44
2.2.5 Αναγνώριση ακμών χρησιμοποιώντας την εντολή edge	46
Κεφάλαιο 3: Βασικοί Αλγόριθμοι για την εξαγωγή πληροφορίας (Matlab)	
3.1 Εύρεση βασικών γεωμετρικών μεγεθών	
3.1.1 Μεθοδολογία ανάπτυξης αλγόριθμου εύρεσης βασικών γεωμετρικών μεγεθών	55
3.1.2 Πρακτική υλοποίηση	56
3.2 Μέτρηση γωνιών μεταξύ αντικειμένων	
3.2.1 Μεθοδολογία ανάπτυξης αλγόριθμου μέτρησης γωνιών μεταξύ αντικειμένων	59
3.2.2 Πρακτική υλοποίηση	60
3.3 Αναγνώριση βασικών γεωμετρικών σχημάτων	
3.3.1 Μεθοδολογία ανάπτυξης αλγόριθμου αναγνώρισης βασικών γεω/κών σχημάτων	63
3.3.2 Πρακτική υλοποίηση	64
3.4 Αναγνώριση χρωμάτων	
3.4.1 Μεθοδολογία ανάπτυξης αλγόριθμου αναγνώρισης χρωμάτων	68
3.4.2 Πρακτική υλοποίηση	69
3.5 Καταμέτρηση αντικειμένων	
3.5.1 Μεθοδολογία ανάπτυξης αλγόριθμου καταμέτρησης αντικειμένων	74
3.5.2 Πρακτική υλοποίηση	75

3.6	Εύρεσης αθροίσματος κερμάτων	
3.6.1	Μεθοδολογία ανάπτυξης αλγόριθμου εύρεσης αθροίσματος κερμάτων	78
3.6.2	Πρακτική υλοποίηση	79
Κεφάλαιο 4: Ανάπτυξη ευφών αλγορίθμων για την αναγνώριση προτύπων		
4.1	Ευφή συστήματα αναγνώρισης προτύπων	
4.1.1	Συνελικτικά Νευρωνικά Δίκτυα	83
4.1.2	Πρόγραμμα εκπαίδευσης οπτικής αναγνώρισης χαρακτήρων (OCR Trainer)	87
4.2	Αναγνώριση χειρόγραφων αριθμών στο matlab	
4.2.1	Μεθοδολογία ανάπτυξης αλγόριθμου με τη χρήση ocr trainer	88
4.2.2	Πρακτική υλοποίηση	89
4.2.3	Μεθοδολογία ανάπτυξης αλγόριθμου με τη χρήση ΣΝΔ	92
4.2.4	Πρακτική υλοποίηση	93
Συμπεράσματα – Παρατηρήσεις		95
Βιβλιογραφία		96

Κατάλογος Εικόνων

Εικόνα 1 Binary	10
Εικόνα 2 Gray Scale	10
Εικόνα 3 Colour	10
Εικόνα 4 Multispectral	10
Εικόνα 5 Στάδια Επεξεργασίας Εικόνας	14
Εικόνα 6 Αρχική εικόνα	24
Εικόνα 7 Ιστόγραμμα εικόνας	24
Εικόνα 8 Κατάτμηση εικόνας με προσαρμοσμένη κατωφλίωση (Adaptive Thresholding)	25
Εικόνα 9 Κατάτμηση εικόνας με προσαρμοσμένη κατωφλίωση (Auto Thresholding)	25
Εικόνα 10 Κατάτμηση εικόνας με τη μέθοδο του Ότσου (Otsu Thresholding)	25
Εικόνα 11 Αρχική εικόνα Test 1 ^ο	28
Εικόνα 12 Αρχική εικόνα Test 2 ^ο	28
Εικόνα 13 Ιστόγραμμα εικόνας Test 1 ^ο	29
Εικόνα 14 Ιστόγραμμα εικόνας Test 2 ^ο	29
Εικόνα 15 Καναλιών του RGB φάσματος Τεστ 1 ^ο	30
Εικόνα 16 Καναλιών του RGB φάσματος Τεστ 2 ^ο	30
Εικόνα 17 Κατάτμηση εικόνας βάση των τριών καναλιών του RGB φάσματος Τεστ 1 ^ο	31
Εικόνα 18 Κατάτμηση εικόνας βάση των τριών καναλιών του RGB φάσματος Τεστ 2 ^ο	31
Εικόνα 19 Αρχική εικόνα	47
Εικόνα 20 Ιστόγραμμα εικόνας	47
Εικόνα 21 Υπολογισμός x, y κλίσης με τη μέθοδο του Roberts	48
Εικόνα 22 Υπολογισμός x, y κλίσης με τη μέθοδο του Prewit	48
Εικόνα 23 Υπολογισμός x, y κλίσης με τη μέθοδο του Sobel	48
Εικόνα 24 Υπολογισμός x, y κλίσης με τη μέθοδο του Canny	49
Εικόνα 25 Υπολογισμός μέτρου κλίσης με τη μέθοδο του Roberts	49
Εικόνα 26 Υπολογισμός μέτρου κλίσης με τη μέθοδο του Prewit	49
Εικόνα 27 Υπολογισμός μέτρου κλίσης με τη μέθοδο του Sobel	50
Εικόνα 28 Υπολογισμός μέτρου κλίσης με τη μέθοδο του Canny	50
Εικόνα 29 Εικόνα μετά την εφαρμογή κατωφλίωσης με τη μέθοδο του Roberts	50
Εικόνα 30 Εικόνα μετά την εφαρμογή κατωφλίωσης με τη μέθοδο του Prewit	51
Εικόνα 31 Εικόνα μετά την εφαρμογή κατωφλίωσης με τη μέθοδο του Sobel	51
Εικόνα 32 Εικόνα μετά την εφαρμογή κατωφλίωσης με τη μέθοδο του Canny	51
Εικόνα 33 Ανίχνευση ακμών με τη μέθοδο του Roberts	52
Εικόνα 34 Ανίχνευση ακμών με τη μέθοδο του Prewit	52
Εικόνα 35 Ανίχνευση ακμών με τη μέθοδο του Sobel	52
Εικόνα 36 Ανίχνευση ακμών με τη μέθοδο του Canny	53
Εικόνα 37 Ανίχνευση ακμών με τη χρήση της εντολής edge	53
Εικόνα 38 Αρχική εικόνα	58
Εικόνα 39: Ιστόγραμμα εικόνας	58
Εικόνα 40: Αρχική εικόνα	61
Εικόνα 41: Ιστόγραμμα εικόνας	61
Εικόνα 42: περικομμένη εικόνα με εστίαση στη γωνία που μας ενδιαφέρει να υπολογίσουμε	62
Εικόνα 43: Δυαδική περικομμένη εικόνα	62
Εικόνα 44: Τελική εικόνα στην οποία αναγράφεται το αποτέλεσμα μετά τον υπολογισμό	62
Εικόνα 45: Αρχική εικόνα Test 1 ^ο	65
Εικόνα 46: Αρχική εικόνα Test 2 ^ο	65
Εικόνα 47: Ιστόγραμμα εικόνας Test 1 ^ο	66
Εικόνα 48: Ιστόγραμμα εικόνας Test 2 ^ο	66
Εικόνα 49: Αναγνώριση βασικών γεωμετρικών σχημάτων Test 1 ^ο	67
Εικόνα 50: Αναγνώριση βασικών γεωμετρικών σχημάτων Test 2 ^ο	67

Εικόνα 51: Αναγνώριση χρωμάτων Μαύρο	72
Εικόνα 52: Αναγνώριση χρωμάτων Μπλε	72
Εικόνα 53: Αναγνώριση χρωμάτων Κυανό	72
Εικόνα 54: Αναγνώριση χρωμάτων Πράσινο	72
Εικόνα 55: Αναγνώριση χρωμάτων Ματζέντα	73
Εικόνα 56: Αναγνώριση χρωμάτων Κόκκινο	73
Εικόνα 57: Αναγνώριση χρωμάτων Άσπρο	73
Εικόνα 58: Αναγνώριση χρωμάτων Κίτρινο	73
Εικόνα 59: Ιστόγραμμα εικόνας Test 1 ^ο	76
Εικόνα 60: Ιστόγραμμα εικόνας Test 2 ^ο	76
Εικόνα 61: Καταμέτρηση αντικειμένων Τεστ 1 ^ο	77
Εικόνα 62: Καταμέτρηση αντικειμένων Τεστ 2 ^ο	77
Εικόνα 63: Αρχική εικόνα.....	80
Εικόνα 64: Ιστόγραμμα εικόνας	80
Εικόνα 65: Άθροισμα κερμάτων.....	81
Εικόνα 66: Αρχική εικόνα Test 1 ^ο	90
Εικόνα 67: Αρχική εικόνα Test 2 ^ο	90
Εικόνα 68: Αρχική εικόνα Test 3 ^ο	91
Εικόνα 69: Αποτελέσματα Τεστ 1 ^ο	91
Εικόνα 70: Αποτελέσματα Τεστ 2 ^ο	91
Εικόνα 71: Αποτελέσματα Τεστ 3 ^ο	91
Εικόνα 72: Αποτελέσματα Τεστ 1 ^ο	94
Εικόνα 73: Αποτελέσματα Τεστ 2 ^ο	94
Εικόνα 74: Αποτελέσματα Τεστ 3 ^ο	94
Εικόνα 75: Αποτελέσματα Τεστ 4 ^ο	94
Εικόνα 76: Αποτελέσματα Τεστ 5 ^ο	94

Κατάλογος Πινάκων

Πίνακας 1: Μοντέλα Ακμών	33
Πίνακας 2: Δυαδική αναπαράσταση χρωμάτων	68
Πίνακας 3: Δυαδική αναπαράσταση χρωμάτων	71

Περίληψη

Στη παρούσα εργασία περιγράφονται βασικές έννοιες που αφορούν την ψηφιακή επεξεργασία εικόνας με σκοπό την εξαγωγή πληροφορίας για την εικόνα αυτή. Πρόκειται για ένα κομμάτι της μηχανικής όρασης.

Πιο συγκεκριμένα ασχοληθήκαμε με τη κατάτμηση εικόνας και επιλέξαμε να παρουσιάσουμε κάποιες βασικές μεθόδους κατάτμησης εικόνας οι οποίες παρουσιάζονται σε δυο επίπεδα αρχικά σε θεωρητικό επίπεδο καταγράφοντας τις βασικές αρχές που τις διέπουν και μετά σε πρακτικό επίπεδο με υλοποίηση, κάποιων μεθόδων από αυτών που παρουσιάζονται θεωρητικά, στο Matlab.

Επίσης ασχοληθήκαμε με ολοκληρωμένους αλγορίθμους με σκοπό στοχευόμενης πληροφορίας όπως η καταμέτρηση αντικειμένων, η αναγνώριση χρωμάτων και η αναγνώριση γεωμετρικών σχημάτων

Abstract

In the current paper are described basic concepts that are related with digital image processing having as a main purpose the export of information consisted in an image. As a matter of fact, it is considered as a part of computer vision.

More specifically we will be discussing, concepts like image segmentation and we also chose to present a few basic methods of image segmentation that are parted and presented in two different levels, the theoretical and the practical one. To start from in the theoretical level, we'll be writing down the fundamental theory that describes them and at a second basis in the practical level we'll be constructing some of the methods that are presented theoretically on Matlab.

Also, we have worked with different algorithms having as a main purpose to extract specific information such as, counting down different objects, color recognition and object recognition.

Εισαγωγή

1.1 Μηχανική όραση – Ιστορική Αναδρομή

Η μηχανική όραση, υπολογιστική όραση ή τεχνητή όραση είναι ένα επιστημονικό πεδίο της τεχνητής νοημοσύνης το οποίο επιχειρεί να αναπαράγει αλγοριθμικά την αίσθηση της όρασης, συνήθως σε ηλεκτρονικό υπολογιστή ή ρομπότ. Η μηχανική όραση σχετίζεται με τη θεωρία και την τεχνολογία που εμπλέκονται στη σχεδίαση και κατασκευή συστημάτων που λαμβάνουν και αναλύουν δεδομένα από ψηφιακές εικόνες. Τα εν λόγω δεδομένα μπορούν να είναι φωτογραφίες, βίντεο, όψεις από πολλαπλές κάμερες, πολυδιάστατες εικόνες από ιατρικό σαρωτή κλπ.

Η μηχανική όραση επιδιώκει να εφαρμόσει θεωρίες και μοντέλα στην κατασκευή μηχανικών συστημάτων με δυνατότητα όρασης.

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 με αποτέλεσμα 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 υπολογιστικής όρασης αποδεικνύεται όλο και περισσότερο καρποφόρα και για τους δύο τομείς.

Ιστορικά, η υπολογιστική όραση αναδύθηκε μετά το 1980 ως αποτέλεσμα επέκτασης του πεδίου της πληροφορικής το οποίο καλείται ψηφιακή επεξεργασία εικόνας σε αλγορίθμους ανάλυσης και κατανόησης εικόνων. Είχαν προηγηθεί η μαθηματική μοντελοποίηση της φυσικής όρασης, έστω σε ένα βασικό επίπεδο, και οι πρώτες προσπάθειες για αναπαραγωγή της αίσθησης της όρασης σε αυτόνομα ρομπότ. Ως τότε ο όρος μηχανική όραση σχετιζόταν με την ηλεκτρολογία και τη ρομποτική, συνήθως σε βιομηχανικό πλαίσιο. Κατά τη δεκαετία του 1980, μετά την εμφάνιση της υπολογιστικής όρασης, οι δύο όροι σταδιακά συνέκλιναν και συγχωνεύθηκαν ως επιστημονικά πεδία, σαν διακριτός τομέας της τεχνητής νοημοσύνης με εφαρμογές όχι μόνο στη ρομποτική αλλά και σε δεκάδες ακόμα κλάδους.

Από τη δεκαετία του 1990 κι έπειτα η μηχανική όραση έχει γνωρίσει αλματώδη ανάπτυξη, έχει συνδεθεί με το γνωστικό πεδίο της μηχανικής μάθησης και έχει δώσει σημαντικά απτά αποτελέσματα, με αλγορίθμους όρασης πραγματικού χρόνου να υλοποιούνται ακόμα και σε φτηνά κινητά τηλέφωνα εξοπλισμένα με κάμερα. Στο εν λόγω πλαίσιο, η μηχανική όραση έχει διαδραματίσει θεμελιώδη ρόλο στην εξέλιξη της ενισχυμένης πραγματικότητας. Μετά την ευρύτατη διάδοση του Kinect, ενός καινοτόμου περιφερειακού διασύνδεσης μεταξύ χρηστών και υπολογιστικών συστημάτων, και τη σχετική άνθιση του τρισδιάστατου (στερεοσκοπικού) οπτικού περιεχομένου ύστερα από τη μεγάλη επιτυχία της κινηματογραφικής ταινίας Άβαταρ το 2009, η μηχανική όραση έχει αρχίσει να εξετάζει πιο ενδελεχώς και την αξιοποίηση δεδομένων βάθους (π.χ. από στερεοσκοπική κάμερα ή ξεχωριστούς αισθητήρες βάθους) για την επίτευξη των στόχων της.

1.2 Τύποι εικόνας

Δυαδικές εικόνες (Binary Images): Οι δυαδικές εικόνες είναι ο απλούστερος τύπος εικόνων και μπορούν να πάρουν δύο τιμές, συνήθως άσπρο η μαύρο, η 0 και 1. Μια δυαδική εικόνα αναφέρεται ως εικόνα 1 μπιτ επειδή παίρνει μόνο 1 δυαδικό ψηφίο για να αντιπροσωπεύσει κάθε εικονοστοιχείο. Αυτοί οι τύποι εικόνων χρησιμοποιούνται συχνά στις εφαρμογές όπου οι μόνες πληροφορίες που απαιτούνται είναι γενική μορφή ή το περίγραμμα του αντικείμενου που πραγματεύεται η εικόνα, παραδείγματος χάριν οπτική αναγνώριση χαρακτήρα (OCR). Οι δυαδικές εικόνες δημιουργούνται συχνά από τις εικόνες στη κλίμακα του γκρι μέσω της διαδικασίας καταφλίωσης εικόνας, όπου όποια τιμή εικονοστοιχείου είναι μεγαλύτερη του κατωφλίου αντικαθιστάτε με την τιμή 1 (λευκό) και οι υπόλοιπες με την τιμή 0 (μαύρο).

Εικόνες στη κλίμακα του γκρι (Gray scale Images): Οι εικόνες στη κλίμακα του γκρι αναφέρονται ως τις μονοχρωματικές εικόνες (ενός-χρώματος). Ο αριθμός των bit που χρησιμοποιούνται για κάθε εικονοστοιχείο καθορίζει τον αριθμό διαθέσιμων διαφορετικών επιπέδων γκρι. Μια τυπική εικόνα στη κλίμακα του γκρι περιέχει 8bits/εικονοστοιχείο, το οποίο μας επιτρέπει να έχουμε 256 διαφορετικά επίπεδα του γκρι.

Έγχρωμες εικόνες (Color Images): Οι εικόνες χρώματος μπορούν να διαμορφωθούν ως μονοχρωματικά στοιχεία εικόνας τριών-καναλιών, όπου κάθε κανάλι αντιστοιχεί σε ένα διαφορετικό χρώμα. Οι πραγματικές πληροφορίες που αποθηκεύονται στα ψηφιακά στοιχεία της εικόνας είναι οι πληροφορίες των επιπέδων του γκρι σε κάθε φασματική ζώνη. Μια τυπική έγχρωμη εικόνα αναπαριστάτε ως RGB εικόνα. Χρησιμοποιώντας τα οκτάμπιτα μονοχρωματικά πρότυπα, η αντίστοιχη έγχρωμη εικόνα θα είχε 24 bits/εικονοστοιχείο (8-bits για κάθε χρωματικό κανάλι κόκκινο, πράσινο και μπλε).

Πολυφασματικές εικόνες (Multispectral Images): Οι πολυφασματικές εικόνες περιέχουν πληροφορίες εκτός του ανθρώπινου αντιληπτικού εύρους. Αυτό μπορεί να περιλάβει τις υπέρυθρες ακτίνες, την υπεριώδη ακτίνα, την ακτίνα X. Αυτές οι εικόνες δεν είναι εικόνες υπό τη συνηθισμένη έννοια επειδή οι πληροφορίες που περιέχουν δεν είναι άμεσα ορατές από το ανθρώπινο σύστημα. Εντούτοις, οι πληροφορίες αντιπροσωπεύονται συχνά με οπτική μορφή με τη χαρτογράφηση των διαφορετικών φασματικών ζωνών των συνιστωσών του RGB.

Εικόνα 1 Binary



Εικόνα 2 Gray Scale



Εικόνα 3 Color



Εικόνα 4 Multispectral



1.3 Χρωματικοί χώροι

Με τον όρο χρωματικός χώρος εννοούμε ένα μοντέλο που χρησιμοποιείται για την αναπαράσταση του χρώματος με τη μορφή τιμών έντασης. Ο χρωματικός χώρος ορίζει τον τρόπο με τον οποίο αναπαρίσταται η χρωματική πληροφορία. Σκοπός του είναι να διευκολύνεται ο ορισμός των χρωμάτων στα πλαίσια της τυποποίησης. Στην ουσία, ένα χρωματικό μοντέλο είναι ένα τρισδιάστατο σύστημα συντεταγμένων και ένα υποσύστημα μέσα σε αυτό όπου κάθε χρώμα αναπαρίσταται από ένα σημείο. Σήμερα, τα περισσότερα μοντέλα είναι προσαρμοσμένα στα φυσικά εξαρτήματα υπολογιστικών συστημάτων ή σε εφαρμογές όπου είναι επιθυμητή η διαχείριση των χρωμάτων.

Στην πρώτη κατηγορία ανήκουν το μοντέλο RGB (Red, Green, Blue) για έγχρωμες οθόνες και κάμερες, το μοντέλο CMYK (Cyan, Magenta, Yellow) για έγχρωμους εκτυπωτές. Στη δεύτερη κατηγορία έχουμε το μοντέλο HSV (Hue, Saturation, Value). Στη συνέχεια παρουσιάζουμε τους ευρύτερα χρησιμοποιούμενους χρωματικούς χώρους και τα χαρακτηριστικά τους

Χρωματικά μοντέλα εικόνων

- **RGB:** Στο συγκεκριμένο χρωματικό μοντέλο υπάρχουν 3 βασικά χρώματα το κόκκινο (RED-R), το πράσινο (GREEN-G) και το μπλε (BLUE-B) και βασίζεται σε ένα καρτεσιανό σύστημα συντεταγμένων. Ο χρωματικός χώρος RGB καλύπτει ένα αρκετά μεγάλο μέρος του ορατού φάσματος. Όλα τα χρώματα του χώρου αυτού καθορίζονται από τρεις μεταβλητές με τιμές 0 έως 255, οι οποίες συμβολίζουν την ένταση του κάθε βασικού χρώματος με πρώτη την τιμή για το κόκκινο μετά την τιμή για το πράσινο και τέλος για το μπλε. Στην εικόνα το κόκκινο έχει τιμή 255,0,0 το πράσινο 0,255,0 το μπλε 0,0,255. Κάθε χρώμα μπορεί να αναπαρασταθεί από μια μίξη συγκεκριμένων τιμών έντασης σε καθεμία από τις παραπάνω χρωματικές συνιστώσες. Οι διάφορες χρωματικές αποχρώσεις μπορούν να αναπαρασταθούν σε σύστημα τριών διαστάσεων αν φανταστούμε ότι κάθε άξονας αντιστοιχεί σε μια χρωματική συνιστώσα. Το RGB είναι ένα προσθετικό μοντέλο. Πάντα δημιουργούμε χρώμα ξεκινώντας από το μαύρο (0,0,0). Όταν τα βασικά του χρώματα προστεθούν σε ίσες αναλογίες στην μέγιστη τιμή τους τότε δημιουργούν το λευκό (255,255,255). Στο matlab η αναπαράσταση μιας RGB εικόνας διαστάσεων $M \times N$ γίνεται με έναν πίνακα τριών διαστάσεων $M \times N \times 3$ που περιέχει εικονοστοιχεία χρώματος. Κάθε εικονοστοιχείο είναι μια τριπλέτα χρωμάτων που αντιστοιχεί στις συνιστώσες του κόκκινου, του πράσινου και του μπλε για το συγκεκριμένο σημείο. Μια RGB εικόνα, δηλαδή, μπορεί να αναπαρασταθεί ως μια “στοίβα” από τρεις εικόνες στην κλίμακα του γκρι οι οποίες όταν τροφοδοτηθούν στις εισόδους για κόκκινο, πράσινο και μπλε μιας έγχρωμης οθόνης παράγουν μια έγχρωμη εικόνα.
- **CMYK:** Το χρωματικό μοντέλο CMY είναι άρρηκτα συνδεδεμένο με τις εκτυπώσεις. Υπάρχουν 3 βασικά χρώματα το κυανό, το ματζέντα και το κίτρινο οι συνδυασμοί των οποίων δημιουργούν όλα τα άλλα χρώματα. ο συνδυασμός των τριών δημιουργεί το μαύρο χρώμα. Αυτό όμως στην πράξη δεν επαληθεύεται και στην πραγματικότητα το χρώμα που προέκυπτε στις εκτυπώσεις ήταν μεταξύ σκούρου καφέ και μαύρου. Έτσι για να επιτύχουμε έντονα γεμάτα μαύρα προστέθηκε και το μαύρο χρώμα στο μοντέλο που τελικά έγινε CMYK όπου C = Cyan, M=Magenta, Y = Yellow, K=Key (μαύρο). Οι τιμές που μπορούν να δοθούν είναι 0%-100%. Στο CMYK τα διάφορα χρώματα προκύπτουν αναμειγνύοντας

τα βασικά χρώματα στο λευκό χαρτί. Το λευκό χρώμα όμως περιέχει όλο το χρωματικό φάσμα. Όταν εκτυπώνεται ένα συγκεκριμένο χρώμα στην πραγματικότητα αφαιρούνται κάποια μήκη κύματος από το λευκό για αυτό ονομάζεται αφαιρετικό μοντέλο. Ο χρωματικός χώρος CMYK περιέχει λιγότερες αποχρώσεις από τον RGB. Οι περισσότερες αποχρώσεις CMYK είναι μέσα στον RGB με κάποιες εξαιρέσεις στις περιοχές των κυανών και κίτρινων.

- **CIE L*a*b*:** Στηρίζεται στην αρχή ότι ένα χρώμα δεν μπορεί να είναι συγχρόνως πράσινο και κόκκινο ή κίτρινο και μπλε. Έχει τρεις μεταβλητές μία για την φωτεινότητα (luminance) και δύο για χρωματικές τιμές. Έτσι το L* συμβολίζει την φωτεινότητα το a* την τιμή από πράσινο έως κόκκινο και το b* από κίτρινο έως μπλε. Όπως βλέπουμε ο χρωματικός χώρος RGB περιλαμβάνεται μέσα στον CIE L*a*b*.
- **HSV:** Το χρωματικό μοντέλο HSV (Hue, Saturation, Value) εκμεταλλεύεται τον τρόπο που οι άνθρωποι αντιλαμβάνονται το χρώμα. Συγκεκριμένα συνηθίζεται να περιγράψουμε τις διαφορές σκηνές, όχι σε συνθήκες κόκκινου, πράσινου και μπλε, αλλά ως απόχρωση, καθαρότητα και ένταση. Βλέπουμε τα πράγματα ως χρώματα ή αποχρώσεις, οι οποίες έχουν είτε 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 διαγωνίου 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 μπλε 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. 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.

1.4 Ιστόγραμμα εικόνων

Ιστόγραμμα ονομάζουμε ένα γράφημα που δείχνει την κατανομή των δεδομένων. Τα ιστογράμματα στην επεξεργασία εικόνας χρησιμοποιούνται για να δείξουν την κατανομή των τιμών των εικονοστοιχείων σε μια εικόνα.

Το ιστόγραμμα μίας ψηφιακής εικόνας με επίπεδα του γκρι στο διάστημα $[0, L-1]$ είναι μία διακριτή συνάρτηση, όπου είναι το k επίπεδο γκρι και είναι το πλήθος των εικονοστοιχείων της εικόνας, που έχουν τιμή $h(r_k) = n_k$ επιπέδου γκρι. Συνήθως, κανονικοποιούμε το ιστόγραμμα, διαιρώντας κάθε τιμή με τον συνολικό αριθμό των εικονοστοιχείων της εικόνας, έστω n . Τότε, το κανονικοποιημένο ιστόγραμμα δίνεται από την συνάρτηση $p(r_k) = \frac{n_k}{n}$, για $k=0,1,\dots,L-1$. Θα μπορούσαμε να πούμε, ότι η δίνει μία προσέγγιση της πιθανότητας της εμφάνισης ενός γκρι επιπέδου r_k .

Τα ιστογράμματα μπορεί να φανούν πολύ χρήσιμα γιατί μας βοηθούν να εξάγουμε συμπεράσματα για τη μορφή μιας εικόνας. Για παράδειγμα

- Μία σκούρα εικόνα (dark image) οι τιμές του γκριζου θα είναι συγκεντρωμένες στα χαμηλότερα επίπεδα.
- Σε μία φωτεινή εικόνα αντίθετα οι τιμές του γκριζου θα είναι συγκεντρωμένες σε υψηλότερα επίπεδα.
- Μία εικόνα με χαμηλό contrast θα έχει για παράδειγμα τις τιμές συγκεντρωμένες στο κέντρο.

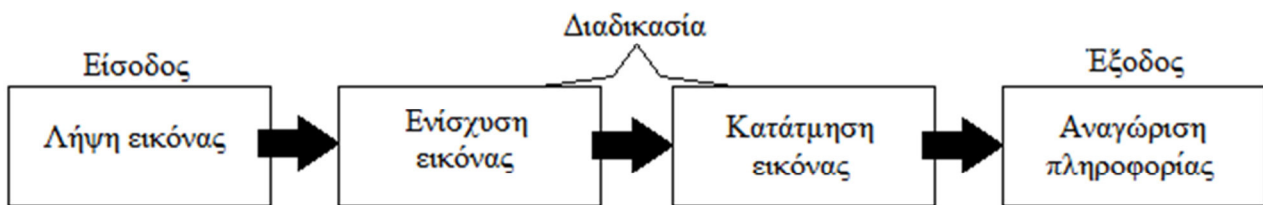
Επομένως, σε μια ιδανική περίπτωση, το ιστόγραμμα «απλώνεται» σε όλο το εύρος των διαθέσιμων γκρι επιπέδων και η εικόνα είναι κατανοητή από το ανθρώπινο μάτι. Την τεχνική αυτή υλοποιεί η εξισορρόπηση ιστογράμματος (histogram equalization), στοχεύει δηλαδή στο να «μετατρέψει» την κατανομή που ακολουθούν τα επίπεδα του γκρι μιας εικόνας σε ομοιόμορφη κατανομή.

1.5 Στάδια επεξεργασίας εικόνας

Απόκτηση εικόνας (Image Acquisition): Αυτό το βήμα σκοπεύει να συλλάβει την εικόνα μέσω της φωτογραφικής μηχανής. Η ποιότητα της εικόνας εξαρτάται από τις παραμέτρους της φωτογραφικής μηχανής, τις συνθήκες φωτισμού, το μέγεθος των αντικειμένων και την απόσταση – οπτική γωνία που θα ληφθεί η εικόνα. Για τα καλύτερα αποτελέσματα, οι φωτογραφικές μηχανές με υψηλότερη ανάλυση προτιμώνται.

Ενίσχυση εικόνας (Image Enhancement): Ο στόχος της ενίσχυσης εικόνας είναι να υποβληθεί σε επεξεργασία η εικόνα έτσι ώστε η προκύπτουσα εικόνα είναι καταλληλότερη από την αρχική εικόνα για τη συγκεκριμένη εφαρμογή. Κατά τη διάρκεια αυτής της διαδικασίας, μια ή περισσότερες ιδιότητες της εικόνας τροποποιούνται. Η επιλογή των ιδιοτήτων και ο τρόπος με τον οποίο η εικόνα πρέπει να τροποποιηθεί εξαρτώνται από τη συγκεκριμένη εφαρμογή. Τέτοιου είδους τροποποιήσεις περιλαμβάνουν μετασχηματισμούς των επίπεδων του γκρι, τροποποιήσεις του ιστογράμματος της εικόνας, φιλτράρισμα της εικόνας κτλ.

Κατάτμηση εικόνας (Image Segmentation): Βασικός στόχος της είναι να χωρίσει την εικόνα σε τμήματα τα οποία έχουν υψηλή συσχέτιση με τα αντικείμενα ή τις περιοχές του πραγματικού κόσμου που περιλαμβάνονται στην εικόνα. Μπορεί να χωριστεί σε περιοχές που να είναι ομοιογενείς σε σχέση με ένα χαρακτηριστικό που επιλέγεται η βάση των απότομων μεταβολών της έντασης.



Εικόνα 5: Στάδια Επεξεργασίας Εικόνας

Κεφάλαιο 1^ο

Κατάτμηση Εικόνας (Image Segmentation)

1.1 Τι είναι η κατάτμηση εικόνας

Η κατάτμηση εικόνας αποτελεί ένα από τα σημαντικότερα βήματα που οδηγούν στην ανάλυση των ψηφιακών εικόνων. Ο βασικός στόχος της είναι να χωρίσει την εικόνα σε τμήματα τα οποία έχουν υψηλή συσχέτιση με τα αντικείμενα ή τις περιοχές του πραγματικού κόσμου που περιλαμβάνονται στην εικόνα. Ο διαχωρισμός της εικόνας σε περιοχές μπορεί να πραγματοποιηθεί με πολλούς τρόπους, όμως δεν έχει βρεθεί ένας αλγόριθμος που να μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εικόνα και το συγκεκριμένο γεγονός παραμένει μια πρόκληση για τους επιστήμονες του συγκεκριμένου κλάδου.

Μια εικόνα μπορεί να χωριστεί σε περιοχές που να είναι ομοιογενείς σε σχέση με ένα χαρακτηριστικό που επιλέγεται, όπως είναι η φωτεινότητα, το χρώμα, η ανακλαστικότητα, η υφή κ.α.

Οι διαδικασίες κατάτμησης διαχωρίζονται σε δυο βασικές κατηγορίες με βάση τα επικρατέστερα χαρακτηριστικά που χρησιμοποιούν: Την κατηγορία της κατάτμησης στηριζόμενη στις ομοιογένειες και την κατηγορία της κατάτμησης που στηρίζεται στις ασυνέχειες. Στην 1^η κατηγορία η προσέγγιση που ακολουθούμε για τη κατάτμηση της εικόνας, στηρίζεται στην απότομη μεταβολή της έντασης όπως για παράδειγμα οι ακμές. Ενώ στη 2^η κατηγορία η κατάτμηση προσεγγίζεται βάση των περιοχών που χαρακτηρίζονται ως παρόμοιες σύμφωνα με ένα σύνολο προκαθορισμένων κριτηρίων.

Ορισμός: Έστω πως R αναπαριστά ολόκληρη την περιοχή του χώρου που καταλαμβάνεται από μια εικόνα. Στην περίπτωση αυτή, η κατάτμηση εικόνας μπορεί να θεωρηθεί ως μια διαδικασία η οποία διαχωρίζει το χώρο R σε n υποπεριοχές R_1, R_2, \dots, R_n και με τέτοιο τρόπο ώστε να ικανοποιούνται οι ακόλουθες ιδιότητες.^[1]

- I. $\bigcup_{i=1}^n R_i = R$
- II. Το σύνολο R_i ($i=1, 2, 3, \dots, n$) είναι ένα συνδεδεμένο σύνολο
- III. $R_i \cap R_j = \emptyset$ για όλες τις τιμές των i, j με $i \neq j$
- IV. $Q(R_i) = \text{TRUE}$ για $i=1, 2, 3, \dots, n$
- V. $Q(R_i \cup R_j) = \text{FALSE}$ για δυο οποιεσδήποτε συνεχόμενες περιοχές R_i και R_j .

Η συνθήκη (I) υποδηλώνει πως η διαδικασία κατάτμησης θα πρέπει να είναι πλήρης, δηλαδή πως το κάθε εικονοστοιχείο θα πρέπει να ανήκει σε κάποια περιοχή. Η συνθήκη (II) επιβάλλει την απαίτηση να είναι όλα μιας περιοχής να είναι να είναι συνδεδεμένα υπό κάποια προκαθορισμένη έννοια. Η συνθήκη (III) υποδηλώνει πως οι περιοχές στις οποίες χωρίζεται μια εικόνα θα πρέπει να είναι ξένες μεταξύ τους, ενώ η συνθήκη (IV) πραγματεύεται τις ιδιότητες που θα πρέπει να ικανοποιούνται από τα εικονοστοιχεία μιας περιοχής που έχει υποστεί κατάτμηση. Τέλος η συνθήκη (V) υποδηλώνει πως δυο παρακείμενες περιοχές R_i και R_j , θα πρέπει να είναι διαφορετικές από την οπτική γωνία του κατηγορήματος Q .^[2]

[1]Rafael C. Gonzales, Richard E. Woods, Ψηφιακή Επεξεργασία Εικόνας, Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2014, Σελ 985.

[2]Rafael C. Gonzales, Richard E. Woods, Ψηφιακή Επεξεργασία Εικόνας, Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2014, Σελ 986.

1.2 Μέθοδοι κατάτμησης εικόνας βασισμένοι στις ομοιότητες

1.2.1 Κατάτμηση εικόνας βασισμένη στις περιοχές

Αυτή η μέθοδος εκμεταλλεύεται την αρχή της ομοιογένειας παίρνοντας ως δεδομένο ότι τα εικονοστοιχεία μέσα σε μια περιοχή κατέχουν παρόμοια χαρακτηριστικά και είναι ανόμοια με τα εικονοστοιχεία σε άλλες περιοχές.

Ο στόχος της κατάτμησης αυτής είναι να παραχθεί μια ομοιογενής περιοχή που είναι μεγαλύτερη στο μέγεθος. Η απλούστερη προσέγγιση κατάτμησης εικόνας βασισμένη στην υπόθεση της ομοιότητας είναι ότι κάθε εικονοστοιχείο συγκρίνεται με το γείτονά του για έλεγχο ομοιότητας βάση κάποιου κριτηρίου (επίπεδα του γκρι, τη σύσταση, το χρώμα, τη μορφή). Εάν το αποτέλεσμα είναι θετικό, εκείνο το ιδιαίτερο εικονοστοιχείο «προστίθεται» στο εικονοστοιχείο με το οποίο συγκρίθηκε με αυτό το τρόπο μια περιοχή «αυξάνεται» επιπλέον. Η ανάπτυξη σταματά όταν αποτυγχάνει η δοκιμή ομοιότητας.

Η κατάτμηση με βάση τις περιοχές χωρίζεται σε δυο βασικούς αλγορίθμους.

➤ Αλγόριθμος αύξησης περιοχών (Region Growing)

Η μέθοδος της κατωφλίωσης, ενώ έχει σαν αποτέλεσμα την κατάτμηση της εικόνας σε περιοχές, δεν λαμβάνει υπόψη καθόλου τις χωρικές σχέσεις των εικονοστοιχείων κατά την διαδικασία της επιλογής. Ο αλγόριθμος αύξησης περιοχών αντίθετα, για να συμπεριλάβει ένα αντικείμενο σε μια περιοχή, ελέγχει αρχικά τις χωρικές σχέσεις των εικονοστοιχείων της εικόνας και στη συνέχεια κάποιο κριτήριο ένωσης των εικονοστοιχείων (πχ. μικρή διαφορά φασματικών τιμών).

Η λογική των αλγορίθμων κατάτμησης με αύξηση περιοχών είναι να ξεκινάει η διαδικασία από συγκεκριμένα σημεία πάνω στην εικόνα ή από τυχαίες θέσεις πάνω σε αυτή. Τα συγκεκριμένα αυτά σημεία μπορεί να προέρχονται είτε από άλλα στάδια επεξεργασίας, είτε από στατιστική ανάλυση των εικόνων, είτε από κάποιον μετασχηματισμό. Μετά το πρώτο βήμα της επιλογής αρχικών σημείων, ο αλγόριθμος προχωράει στην σειριακή επεξεργασία του κάθε σημείου σπόρου (seed point) στην αρχική εικόνα. Ορίζεται ένα κριτήριο συνένωσης (πχ. οι διαφορές τόνων του γκρι) το οποίο καθορίζει εφόσον μπορεί να πραγματοποιηθεί μια συνένωση ή όχι. Ακολουθεί μια επαναληπτική διαδικασία κατά την οποία ελέγχονται τα εικονοστοιχεία που γειτνιάζουν με τα αρχικά σημεία και με βάση το κριτήριο ένωσης, αυξάνεται το εμβαδόν του αντικειμένου, συγχωνεύοντας νέα εικονοστοιχεία.

Συνολικά, η όλη διαδικασία επαναλαμβάνεται μέχρι που να μην μπορούν να πραγματοποιηθούν πλέον συγχωνεύσεις σε όλη την επιφάνεια της εικόνας. Σε αυτό το στάδιο θα έχουν προκύψει το πολύ ισάριθμα αντικείμενα με τα αρχικά σημεία, ενώ τα εικονοστοιχεία που δεν συγχωνεύτηκαν, αποτελούν το υπόβαθρο της κατάτμησης. Αυτό σημαίνει ότι με την αύξηση περιοχών δεν επιτυγχάνεται πλήρης κατάτμηση της εικόνας σε αντικείμενα.

➤ Αλγόριθμος συνένωσης περιοχών (Region Merging)

Οι αλγόριθμοι κατάτμησης εικόνας με συνένωση περιοχών είναι από τους πιο διαδεδομένους στην βιβλιογραφία και χρησιμοποιούνται εκτεταμένα στην Τηλεπισκόπηση και ειδικότερα στην Αντικειμενοστραφή Ανάλυση Εικόνας. Σε αυτή την οικογένεια αλγορίθμων ουσιαστικά ορίζεται ένα χαρακτηριστικό μέγεθος που εκφράζει την ομοιογένεια περιοχών.

Ο βασικός στόχος αυτών των αλγορίθμων είναι ο διαχωρισμός της εικόνας σε αντικείμενα για τα οποία όμως η ομοιογένεια να είναι μέγιστη. Το κριτήριο της ομοιογένειας μπορεί να βασίζεται στο επίπεδο του τόνου του γκριζού, στο χρώμα, στην υφή, στο σχήμα ή ακόμα και σε κάποιο μοντέλο σημασιολογικής πληροφορίας (semantic information). Με το ορισμό που ορίστηκε για τη κατάτμηση εικόνας στην ενότητα 1.1 ορίσαμε και τις βασικές συνθήκες για την κατάτμηση εικόνας σε περιοχές. Για τους αλγορίθμους συνένωσης περιοχών ισχύουν επιπλέον οι παρακάτω συνθήκες:

I. $H(R_i) = \text{TRUE}$ όπου $i=1,2,\dots,S$.

II. $H(R_i \cup R_j) = \text{FALSE}$ $i \neq j$, R_i γειτονικό με το R_j

όπου S είναι ο συνολικός αριθμός αντικειμένων σε μια εικόνα και $H(R_i)$ είναι μια δυαδική συνάρτηση ελέγχου ομοιογένειας για την περιοχή R_i .

Οι περιοχές που θα προκύψουν από την κατάτμηση της εικόνας θα πρέπει να είναι ταυτόχρονα ομοιογενής και μέγιστες, και με τον όρο μέγιστες εννοείται ότι το κριτήριο ομοιογένειας θα πρέπει να μην είναι αληθές, για κανένα γειτονικό αντικείμενο.

Η πιο φυσική από τις μεθόδους κατάτμησης εικόνας με βάση περιοχές είναι η διαδικασία κατά την οποία η εικόνα θεωρείται σαν ένα σύνολο από περιοχές, οι οποίες αρχικά αποτελούνται από μεμονωμένα εικονοστοιχεία. Αυτές οι αρχικές περιοχές σχεδόν σίγουρα δεν ικανοποιούν το κριτήριο της (I) και επομένως οι περιοχές θα αρχίσουν να ενώνονται για όσο η συνάρτηση (II) παραμένει αληθής. Όταν τα δυο αυτά κριτήρια ικανοποιηθούν, η διαδικασία συνένωσης θα σταματήσει.

1.2.2 Κατάτμηση εικόνας με τον αλγόριθμο του υδροκρίτη

Ο αλγόριθμος υδροκρίτη (watershed) είναι ένας αλγόριθμος κατάτμησης εικόνας που βασίζεται σε περιοχές και χρησιμοποιεί έναν μορφολογικό μετασχηματισμό για τον προσδιορισμό περιοχών στην εικόνα. Οι έννοιες του υδροκρίτη και των λεκανών απόθεσης, είναι πολύ γνωστές στον τομέα της τοπογραφίας.

Οι γραμμές των ορίων του υδροκρίτη χωρίζουν μεμονωμένες λεκάνες απόθεσης. Εάν υπολογιστεί η κλίση με βάση τους τόνους του γκρι μιας εικόνας, παράγεται μια εικόνα κλίσεων. Τότε η εικόνα μπορεί να ερμηνευτεί σαν μια τοπογραφική τρισδιάστατη επιφάνεια, στην οποία οι τιμές της κλίσης του γκρι, αντιστοιχούν στις τιμές του υψομέτρου. Επομένως, οι ακμές των περιοχών αντιστοιχούν σε ψηλούς υδροκρίτες και οι περιοχές χαμηλών κλίσεων αντιστοιχούν σε λεκάνες απόθεσης.

Ο στόχος της κατάτμησης που βασίζεται σε περιοχές είναι να δημιουργήσει ομοιογενείς περιοχές. Στο μετασχηματισμό του υδροκρίτη, οι λεκάνες απόθεσης της τοπογραφικής επιφάνειας είναι ομοιογενείς με την έννοια ότι όλα τα εικονοστοιχεία που ανήκουν στην ίδια λεκάνη απόθεσης, συνδέονται με το ελάχιστο υψόμετρο της περιοχής με μια σειρά εικονοστοιχείων που παρουσιάζουν σταδιακή μείωση υψομέτρου κατά μήκος της διαδρομής.

Προσεγγίσεις του αλγόριθμου του υδροκρίτη

- Τα τοπικά ελάχιστα της κλίσης της εικόνας μπορούν να επιλεγούν σαν δείκτες, σε αυτήν την περίπτωση το αποτέλεσμα που παράγεται είναι υπερ-τμηματοποίηση της εικόνας και το δεύτερο βήμα περιλαμβάνει τη συγχώνευση περιοχών.
- Ο βασισμένος στους δείκτες μετασχηματισμός του υδροκρίτη χρησιμοποιεί τις συγκεκριμένες θέσεις δεικτών που είτε ρητά έχουν καθοριστεί από το χρήστη είτε έχουν καθοριστεί αυτόματα με τους μορφολογικούς χειριστές είτε άλλους τρόπους.

1.2.3 Κατάτμηση Εικόνας με κατωφλίωση

Η κατωφλίωση μιας εικόνας κλίμακας του γκριζου, είναι η απλούστερη μέθοδος κατάτμησης. Η κατωφλίωση είναι ένας μετασχηματισμός μιας αρχικής εικόνας R σε μια δυαδική εικόνα g με τον ακόλουθο τρόπο:

$$\begin{aligned} g(i, j) &= 1 \text{ για } R(i, j) \geq T \\ g(i, j) &= 0 \text{ για } R(i, j) < T \end{aligned} \quad (1)$$

όπου T είναι το κατώφλι, η τιμή 1 δίνεται στα αντικείμενα της εικόνας και η τιμή 0 στο υπόβαθρο (background) ή αντίστροφα.

Για να είναι επιτυχής η κατάτμηση με τη μέθοδο της κατωφλίωσης, είναι σημαντικό να γίνει σωστή επιλογή της τιμής του κατωφλιού. Η επιλογή αυτή μπορεί να γίνει με δυο τρόπους, είτε με τη βοήθεια του χρήστη, ή μπορεί να είναι το αποτέλεσμα μιας μεθόδου αυτόματης ανίχνευσης κατωφλιού (auto thresholding).

Δεδομένου ότι ακόμα και στις απλούστερες εικόνες τείνουν να υπάρχουν διαφοροποιήσεις του τόνου του γκρι μέσα στα αντικείμενα και στο υπόβαθρο, οι οποίες μπορεί να προέρχονται από σφάλματα του δέκτη, σε διαφοροποιήσεις του φωτισμού κατά τη διάρκεια της λήψης ή σε ένα μεγάλο αριθμό άλλων παραγόντων. Μόνο σε ένα μικρό αριθμό περιπτώσεων μπορεί η κατωφλίωση να είναι επιτυχής με την εφαρμογή του ίδιου κατωφλιού σε όλη την έκταση της εικόνας (global thresholding).

Αντιθέτως η κατάτμηση με τη χρήση διαφόρων κατωφλιών (που ονομάζεται και προσαρμοσμένη κατωφλίωση – adaptive thresholding), μπορεί να αποδώσει καλύτερα αποτελέσματα σε τέτοιες περιπτώσεις. Σε αυτού του είδους την κατωφλίωση, η τιμή για το κατώφλι ορίζεται σαν συνάρτηση ενός ή περισσότερων τοπικών χαρακτηριστικών της εικόνας.

1.3 Πρακτικά παραδείγματα με τη χρήση του matlab

1.3.1 Κατάτμηση εικόνας με μεθόδους κατωφλίωσης

1.3.1.1 Auto Thresholding

```
%Εισαγωγή εικόνας
I = imread(' jump.jpg');

%Μετατροπή εικόνας από έγχρωμη κλίμακα σε κλίμακα του γκρι
I=rgb2gray(I);

subplot(1,2,1)
imshow(I)
title('Original Image')

Id = im2double(I); % I is a uint8 grayscale image
T = 0.5*(min(Id(:)) + max(Id(:)));
deltaT = 0.01; % convergence criterion
done = false;
while ~done
g = Id >= T;
Tnext = 0.5*(mean(Id(g)) + mean(Id(~g)));
done = abs(T - Tnext) < deltaT;
T = Tnext;
end

%Μετατροπή εικόνας σε δυαδική μορφή όπου όποια τιμή εικονοστοιχείου είναι μεγαλύτερη του T
αντικαθιστάτε με την τιμή 1 (λευκό) και οι υπόλοιπες με την τιμή 0 (μαύρο).
BW = im2bw(I,T);

subplot(1,2,2)
imshow(BW,[])
title('Thresholded Image');
```

Λειτουργία κώδικα

Αρχικά διαβάζουμε μια εικόνα, αφού, την μετατρέψουμε από έγχρωμη κλίμακα, σε κλίμακα γκρι, βρίσκει μια τιμή για το T βασισμένη στο στα μέγιστα και ελάχιστα της εικόνας καθώς και στις μέσες τιμές. Τέλος, χρησιμοποιώντας την τιμή που βρήκαμε, σαν κριτήριο επιλογής μετατροπής των εικονοστοιχείων, μετατρέπουμε την εικόνα σε δυαδική μορφή, πρακτικά όποια τιμή εικονοστοιχείου είναι μεγαλύτερη του T αντικαθιστάτε με την τιμή 1 (λευκό) και οι υπόλοιπες με την τιμή 0 (μαύρο).

1.3.1.2 Adaptive Thresholding

```
%Εισαγωγή εικόνας
I = imread('jump.jpg');

%Μετατροπή εικόνας από έγχρωμη κλίμακα σε κλίμακα του γκρι
I=rgb2gray(I);

subplot(1,2,1)
imshow(I)
title('Original Image')

%Εφαρμόζει στην εικόνα την συνάρτηση adaptt
BW = blkproc(I,[310 310],@adaptt);

subplot(1,2,2)
imshow(BW,[])
title('Thresholded Image');
```

```
function [ y ] = adaptt( x )
if std2(x)<1
    y=ones(size(x,1),size(x,2));
else
    y=im2bw(x,graythresh(x));
end
end
```

Λειτουργία κώδικα

Αρχικά διαβάζουμε μια εικόνα και αφού την μετατρέψουμε από έγχρωμη κλίμακα σε κλίμακα γκρι, εφαρμόζουμε την συνάρτηση `adaptt` χρησιμοποιώντας την εντολή `blkproc`. Η συνάρτηση λειτουργεί βάση τυπικής απόκλισης η οποία αν είναι μικρότερη του 1 τότε το συγκεκριμένο εικονοστοιχείο καταχωρείται ως φόντο ενώ αν είναι μεγαλύτερη ως μέρος του αντικειμένου.

1.3.1.3 Otsu's Thresholding Method

```
%Εισαγωγή εικόνας
I = imread('jump.jpg');

%Μετατροπή εικόνας από έγχρωμη κλίμακα σε κλίμακα του γκρι
I1=rgb2gray(I);

%Βρίσκουμε το ιστόγραμμα της εικόνας
imhist(I1)
figure
subplot(1,2,1)
imshow(I)
title('Original Image')

%Βρίσκουμε μια global τιμή κατωφλίωσης μέσω της μεθόδου του Otsu
T = graythresh(I1);

%Μετατροπή εικόνας σε δυαδική μορφή όπου όποια τιμή εικονοστοιχείου είναι μεγαλύτερη του T
αντικαθιστάτε με την τιμή 1 (λευκό) και οι υπόλοιπες με την τιμή 0 (μαύρο).
BW = im2bw(I1,T);

subplot(1,2,2)
imshow(BW,[])
title('Thresholded Image');
```

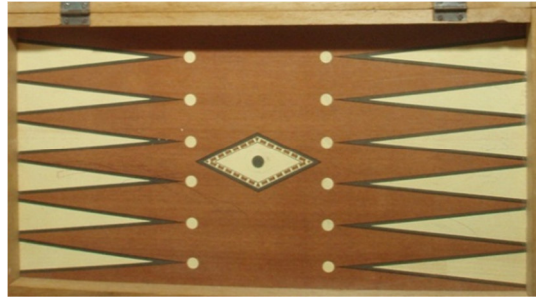
Λειτουργία κώδικα

Αρχικά διαβάζουμε μια εικόνα, αφού, την μετατρέψουμε από έγχρωμη κλίμακα, σε κλίμακα γκρι, βρίσκουμε το ιστόγραμμα της. Μετά, χρησιμοποιούμε την εντολή `graythresh`, η οποία, πρακτικά εφαρμόζει τη μέθοδο του Otsu, ώστε, να βρει μια global τιμή κατωφλίωσης. Τέλος, χρησιμοποιώντας την τιμή που βρήκαμε, σαν κριτήριο επιλογής μετατροπής των εικονοστοιχείων, μετατρέπουμε την εικόνα σε δυαδική μορφή, πρακτικά όποια τιμή εικονοστοιχείου είναι μεγαλύτερη του T αντικαθιστάτε με την τιμή 1 (λευκό) και οι υπόλοιπες με την τιμή 0 (μαύρο).

Συμπεράσματα-Αποτελέσματα

Οι παραπάνω αλγόριθμοι είναι εξίσου αποτελεσματικοί, όμως βέλτιστα αποτελέσματα περνούμε από τον πρώτο αλγόριθμο.

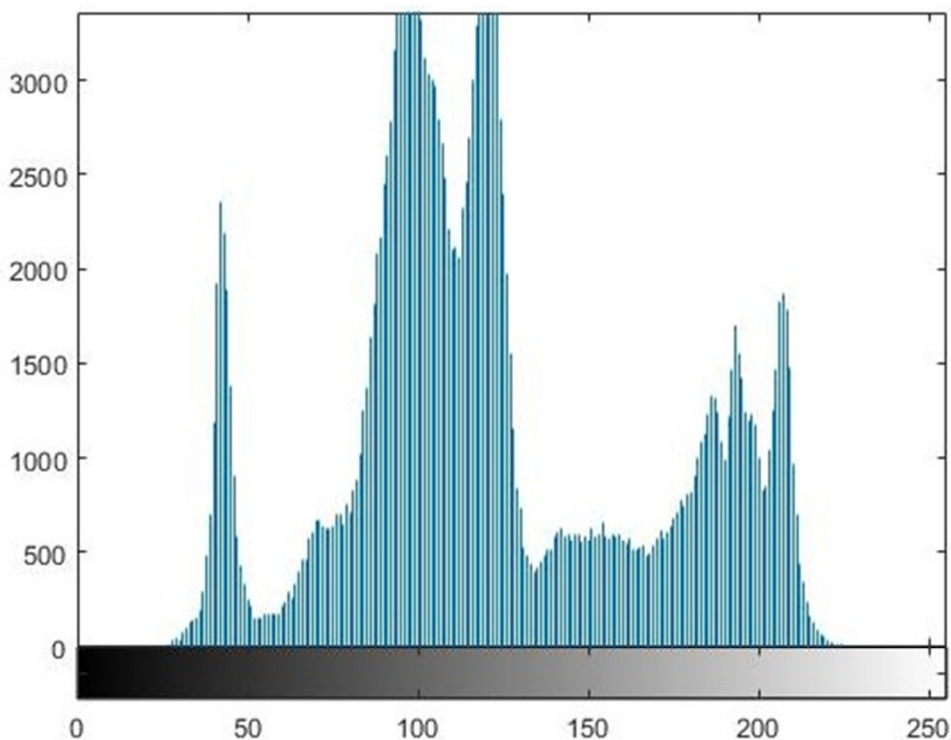
Αρχική εικόνα:



Εικόνα 6: Αρχική εικόνα

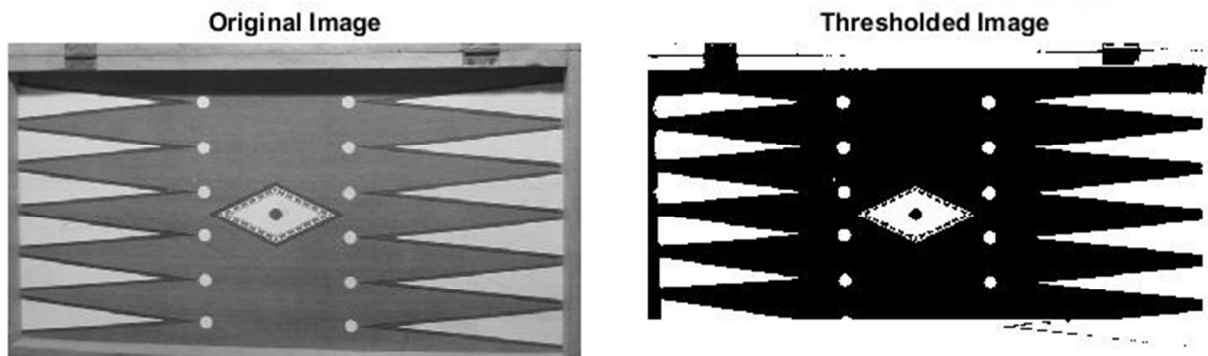
Ιστόγραμμα εικόνας:

Το ιστόγραμμα της εικόνας, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι, στα οποία, και στηρίζονται οι αλγόριθμοι καταψήφωσης ώστε να βρουν ένα ικανοποιητικό κατώφλι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι σε φυσιολογικά πλαίσια. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων όπου στη συγκεκριμένη εικόνα είναι χαμηλά.

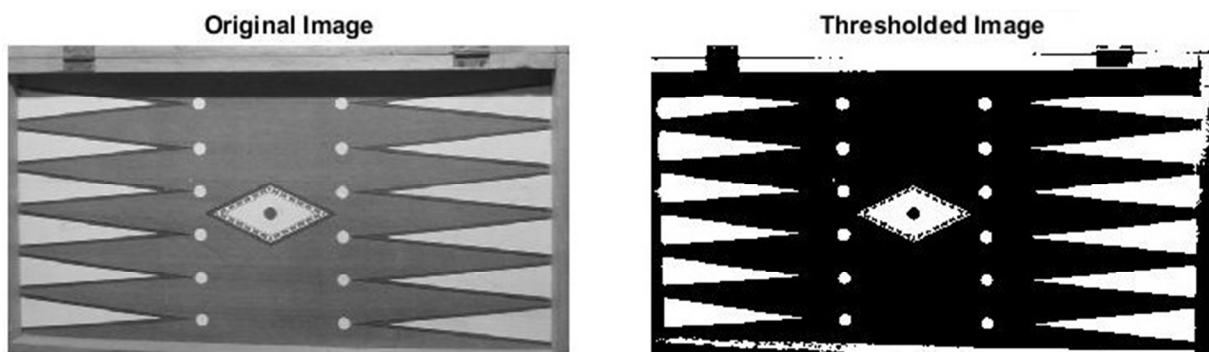


Εικόνα 7: Ιστόγραμμα εικόνας

Αποτελέσματα:



Εικόνα 8: Κατάτμηση εικόνας με προσαρμοσμένη κατωφλίωση (Adaptive Thresholding)



Εικόνα 9: Κατάτμηση εικόνας με μέθοδο αυτόματης αντίγνωσης κατωφλιού (Auto Thresholding)



Εικόνα 10: Κατάτμηση εικόνας με τη μέθοδο του Ότσου (Otsu Thresholding)

1.3.2 Κατάτμηση εικόνας βάση των τριών καναλιών του RGB φάσματος

```
%Εισαγωγή εικόνας
image = imread('jump.jpg');
[height, width, planes] = size(image);
rgb = reshape(image, height, width * planes);
figure('Name','RGB Planes','units','normalized','outerposition',[0 0 1 1])

% Απεικόνιση RGB επιπέδων
imagesc(rgb);

% Εμφάνιση μπάρας χρωμάτων
c = colorbar;
w = c.LineWidth;
c.LineWidth = 1.5;
title('RGB Planes')

figure('Name','GraySpace','units','normalized','outerposition',[0 0 1 1])
imshow(rgb)
r = double(image(:, :, 1));      % Κόκκινο κανάλι
g = double(image(:, :, 2));      % Πράσινο κανάλι
b = double(image(:, :, 3));      % Μπλε κανάλι
a = zeros(size(image, 1), size(image, 2));
just_red = cat(3, r, a, a);
just_green = cat(3, a, g, a);
just_blue = cat(3, a, a, b);

figure('Name','RGB Channels','units','normalized','outerposition',[0 0 1 1])
subplot(1,3,1)
imshow(just_red/255)
title('Red channel')

subplot(1,3,2)
imshow(just_green/255)
title('Green channel')

subplot(1,3,3)
imshow(just_blue/255)
title('Blue channel')

% apply the blueness, redness and greenness calculation
redness = (double(g)- double(b));
greeness = 2.0+(double(b)- double(r));
blueness = 4.0+(double(r)- double(g));

% Απεικόνιση RGB επιπέδων
figure('Name','Color Calculations','units','normalized','outerposition',[0 0 1 1])
subplot(1,3,1)
imagesc(redness);
title('Redness')
```

```

subplot(1,3,2)
imagesc(greeness);
title('Greeness')

subplot(1,3,3)
imagesc(blueness);
title('Blueness')

%εφαρμογή κατωφλίωσης ώστε να γίνει κατάτμηση του αντικειμένου από το φόντο
MR = redness < -25;
MG = greeness < 20;
MB = blueness < -1;

figure('Name','ForegroundThresholding','units','normalized','outerposition',[0 0 1 1])
subplot(1,3,1)
imagesc(MR);
title({'ForeGround Thresholding';'Based on Red Channel'})

subplot(1,3,2)
imagesc(MG);
title({'ForeGround Thresholding';'Based on Green Channel'})

subplot(1,3,3)
imagesc(MB);
title({'ForeGround Thresholding';'Based on Blue Channel'})

LR = bwlabel(MR);
LG = bwlabel(MG);
LB = bwlabel(MB);

id1 = LR(200, 200);
id2 = LG(200, 200);
id3 = LB(200, 200);

% Εφαρμογή μάσκας ώστε να εμφανίζεται το επιθυμητό αντικείμενο
obj1 = (LR == id1);
obj2 = (LG == id2);
obj3 = (LB == id3);

figure('Name','Final Results','units','normalized','outerposition',[0 0 1 1])
subplot(1,3,1)
imshow(obj1)
title({'Segmented Image';'Based on Red Channel'})
subplot(1,3,2)
imshow(obj2)
title({'Segmented Image';'Based on Green Channel'})
subplot(1,3,3)
imshow(obj3)
title({'Segmented Image';'Based on blue Channel'})

```

Λειτουργία κώδικα

Διαχωρίζουμε την εικόνα βάση των τριών καναλιών του RGB φάσματος και στη συνέχεια υπολογίζουμε το ποσοστό των κυρίαρχων χρωμάτων σε κάθε εικονοστοιχείο. Κατά αυτόν το τρόπο γίνεται η ομαδοποίηση των εικονοστοιχείων βάση του χρώματος και κατά συνέπεια η κατάτμηση της εικόνας.

Συμπεράσματα-Αποτελέσματα

Με δεδομένο τα χρώματα που εμπεριέχονται σε κάθε φωτογραφία καθώς και στο κύριο αντικείμενο της, η κατάτμηση, διαφέρει από κανάλι σε κανάλι. Ανάλογα με την τιμή κατοφλίσωσης που θα δώσουμε επιτυγχάνουμε αντίστοιχα και τη σωστή κατάτμηση.

Αρχικές εικόνες:



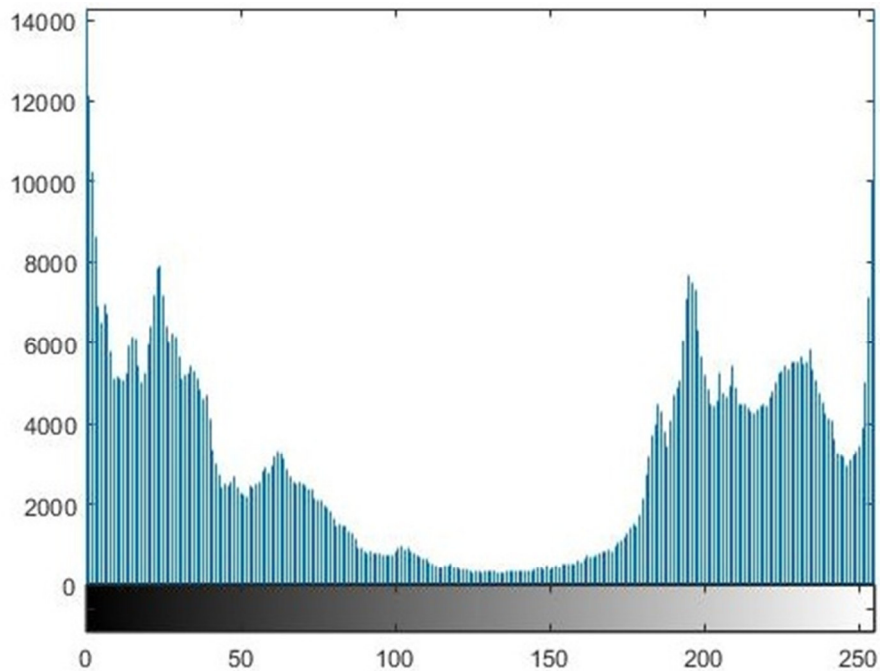
Εικόνα 11: Αρχική εικόνα Test 1°



Εικόνα 12: Αρχική εικόνα Test 2°

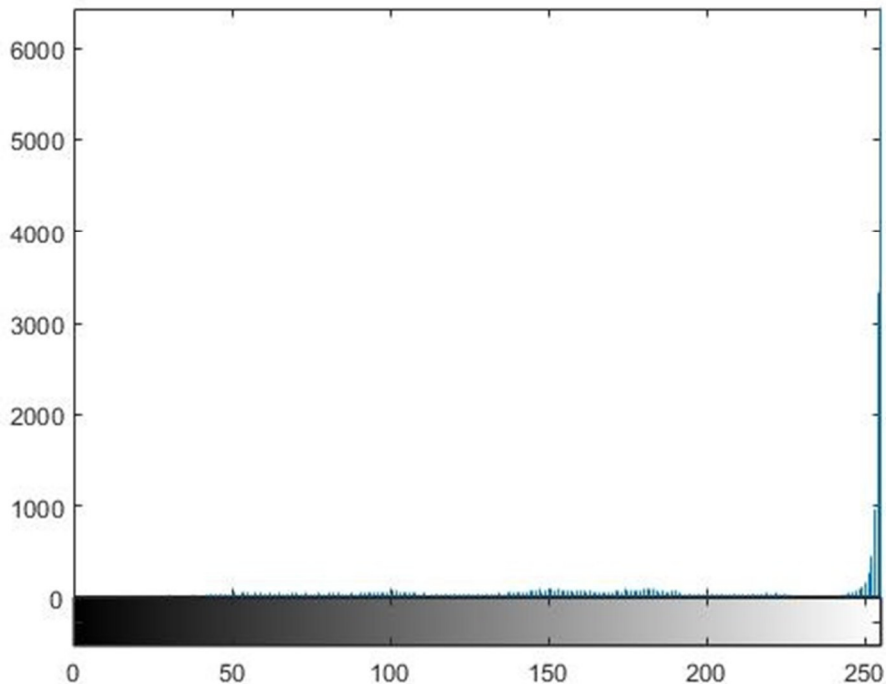
Ιστογράμματα εικόνων:

Το ιστόγραμμα της εικόνας για 1^ο τεστ, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι χαμηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων οπού στη συγκεκριμένη εικόνα είναι χαμηλά.



Εικόνα 13: Ιστόγραμμα εικόνας Test 1^ο

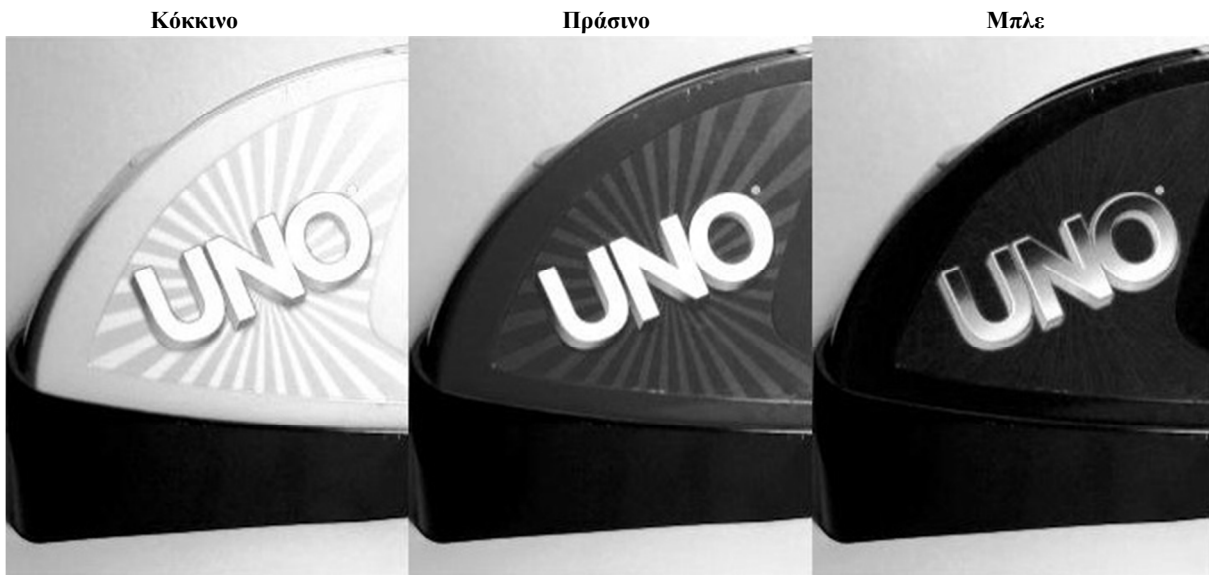
Το ιστόγραμμα της εικόνας για το 2^ο τεστ, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι υψηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων οπού στη συγκεκριμένη εικόνα είναι πάρα πολύ χαμηλά.



Εικόνα 14: Ιστόγραμμα εικόνας Test 2^ο

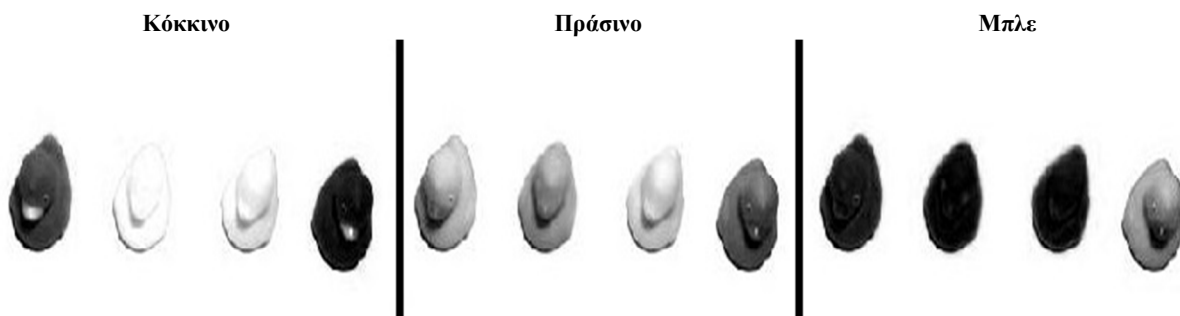
Κανάλια του RGB φάσματος της εικόνας:

Χωρίζοντας την εικόνα στα τρία κανάλια του RGB φάσματος, γίνεται πιο εύκολη η αναγνώριση των χρωμάτων. Πιο συγκεκριμένα, από την αρχική φωτογραφία ξέρουμε ότι το κόκκινο χρώμα, είναι το κυρίαρχο χρώμα στη φωτογραφία και επιβεβαιώνεται από το διαχωρισμό της εικόνας στα τρία κανάλια του RGB φάσματος. Μιας και στο κανάλι του κόκκινου, βλέπουμε το αντικείμενο ανοιχτόχρωμο, ενώ, στο πράσινο κανάλι βλέπουμε σαν ανοιχτόχρωμο μόνο τη λέξη που υπάρχει γραμμένη πάνω στο αντικείμενο, γιατί, στην αρχική εικόνα είναι κίτρινη που το κίτρινο είναι παράγωγο χρώμα του πράσινου.



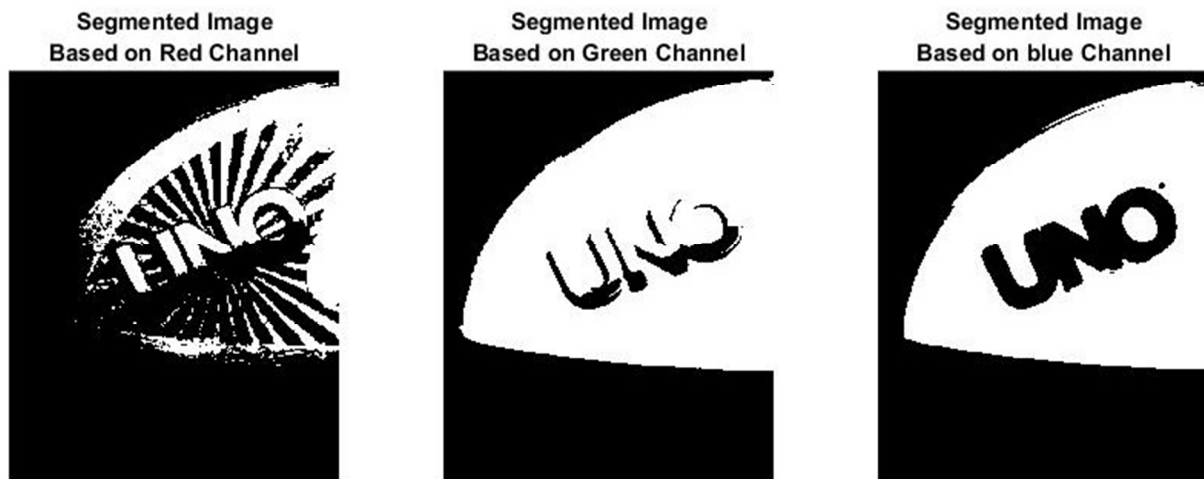
Εικόνα 15: Κανάλια του RGB φάσματος Τεστ 1°

Χωρίζοντας την εικόνα στα τρία κανάλια του RGB φάσματος, γίνεται πιο εύκολη η αναγνώριση των χρωμάτων. Πιο συγκεκριμένα, στο κανάλι του κόκκινου, βλέπουμε το κίτρινο και το πορτοκαλί παπάκι ανοιχτόχρωμο, αυτό συμβαίνει γιατί είναι παράγωγα του κόκκινου και συναφή μεταξύ τους μιας και με τη χρήση του ενός μπορούμε να δημιουργήσουμε το άλλο. Στο πράσινο κανάλι αρκετά ανοιχτόχρωμο είναι το κίτρινο και το πράσινο παπάκι γιατί το κίτρινο είναι παράγωγο του πράσινου. Ενώ στο μπλε μόνο το μπλε.

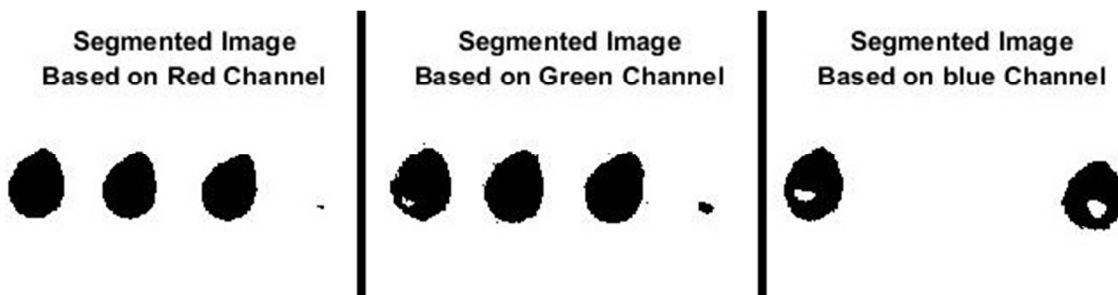


Εικόνα 16: Κανάλια του RGB φάσματος Τεστ 2°

Αποτελέσματα:



Εικόνα 17: Κατάτμηση εικόνας βάση των τριών καναλιών του RGB φάσματος Τεστ 1^ο



Εικόνα 18: Κατάτμηση εικόνας βάση των τριών καναλιών του RGB φάσματος Τεστ 2^ο

Κεφάλαιο 2

Αναγνώριση Ακμών (Edge Detecting)

2.1 Τι είναι η αναγνώριση ακμών

Ανίχνευση ακμών είναι μια μέθοδος κατάτμησης η οποία στηρίζεται στην ανίχνευση απότομων τοπικών μεταβολών έντασης. Όσο πιο απότομη είναι η μεταβολή τόσο πιο εύκολη είναι η ανίχνευση των ακμών. Αντιθέτως όταν στην εικόνα υπάρχει θόρυβος η ανίχνευση των ακμών είναι πιο δύσκολη. Στην ιδανική περίπτωση, το αποτέλεσμα της εφαρμογής ενός ανιχνευτή ακμών σε μια εικόνα οδηγεί σε ένα σύνολο συνδεδεμένων καμπυλών που δείχνουν τα όρια (περιγράμματα) των διαφόρων αντικειμένων της εικόνας.

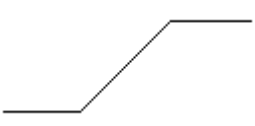




Τα εικονοστοιχεία των ακμών είναι εικονοστοιχεία στα οποία η ένταση μιας συνάρτησης εικόνας μεταβάλετε απότομα ενώ ακμές είναι ένα σύνολο από σημεία ασυνέχειας τα οποία καθορίζουν το όριο ανάμεσα σε δύο ομοιόμορφες περιοχές.

Υπολογιστικά οι ακμές για συνεχείς συναρτήσεις μπορούν να υπολογιστούν με τον υπολογισμό των μερικών παραγώγων της πρώτης παραγώγου και εντοπισμό των τοπικών μέγιστων. Μια δεύτερη μέθοδος με πλεονεκτήματα σε αξιοπιστία στηρίζεται στις διελεύσεις των μερικών παραγώγων της δεύτερης παραγώγου από το μηδέν (zero crossing). Μια μεταβολή της συνάρτησης της εικόνας μπορεί να περιγραφεί με την βάθμωση (gradient) προς την κατεύθυνση της μέγιστης μεταβολής. Πρόκειται για διανυσματική μεταβλητή με μέτρο και κατεύθυνση.

$$\text{Μέτρο: } |\text{grad}(I(x, y))| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

$$\text{Κατεύθυνση: } \varphi = \tan^{-1}\left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}}\right)$$

Πίνακας 1: Μοντέλα Ακμών

	Ράμπα
	Βηματική
	Γραμμική
	Οροφής
	Θορυβώδης

2.2 Μέθοδοι αναγνώρισης ακμών

2.2.1 Αναγνώριση ακμών με τη μέθοδο Roberts

Ο ανιχνευτής ακμών Roberts τον οποίο χρησιμοποιήσαμε στην παρούσα εργασία είναι ένας τοπικός διαφορικός τελεστής που δέχεται μια εικόνα εισόδου $A(m,n)$, ανήκει στις τεχνικές ανίχνευσης ακμών με χρήση πρώτης παραγώγου.

Η τεχνική εφαρμόζεται με χρήση ενός τελεστή (τελεστής Roberts) ο οποίος αποτελείται από δύο φίλτρα (μάσκες ακμών), ένα για να ανιχνεύσει τις αλλαγές στην κάθετη κατεύθυνση (G_y) και το άλλο για να ανιχνεύσει τις αλλαγές στην οριζόντια (G_x).

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ και } G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Κατά τη διαδικασία ανίχνευσης ακμών πραγματοποιείται συνέλιξη μεταξύ της εικόνας και των δύο αυτών μασκών με τη κάθε μια ξεχωριστά. Μετά την εφαρμογή αυτής της διαδικασίας ακολουθεί καταφλίωση που συνήθως έγκειται στη διατήρηση ενός ποσοστού των σημείων ακμών που έχουν το υψηλότερο μέτρο κλίσης.

Υπολογίζεται το μετρό κλίσης για κάθε εικονοστοιχείο, στην περίπτωση που αυτό υπερβεί το κατώφλι που υπολογίσαμε κατά τη διαδικασία της καταφλίωσης τότε ορίζεται πως το εικονοστοιχείο αποτελεί σημείο ακμής και λαμβάνει τιμή 255 (αλλιώς λαμβάνει τιμή 0). Με τη διαδικασία αυτή σαρώνεται όλη η εικόνα και όσα σημεία ανιχνευθούν ότι αποτελούν σημεία ακμών αναδεικνύονται στο προσκήνιο. Τέλος υπάρχει το βήμα της λέπτυνσης των πιθανών γραμμών ώστε να έχουν πάχος ενός εικονοστοιχείου.

2.2.2 Αναγνώριση ακμών με τη μέθοδο Prewit

Ο ανιχνευτής ακμών Prewit τον οποίο χρησιμοποιήσαμε στην παρούσα εργασία. Στον ανιχνευτή ακμών του Prewit η διαδικασία είναι ακριβώς η ίδια όπως και με τον ανιχνευτή ακμών του Sobel, δηλαδή δίνει έμφαση σε περιοχές υψηλής χωρικής συχνότητας, οι οποίες αντιστοιχούν σε ακμές. Ανήκει στις τεχνικές ανίχνευσης ακμών με χρήση πρώτης παραγώγου.

Η τεχνική εφαρμόζεται με χρήση ενός τελεστή (τελεστής Prewit) ο οποίος αποτελείται από δύο φίλτρα (μάσκες ακμών), ένα για να ανιχνεύσει τις αλλαγές στην κάθετη κατεύθυνση (G_y) και το άλλο για να ανιχνεύσει τις αλλαγές στην οριζόντια (G_x).

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ και } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Κατά τη διαδικασία ανίχνευσης ακμών πραγματοποιείται συνέλιξη μεταξύ της εικόνας και των δύο αυτών μασκών με τη κάθε μια ξεχωριστά. Μετά την εφαρμογή αυτής της διαδικασίας ακολουθεί καταφλίωση που συνήθως έγκειται στη διατήρηση ενός ποσοστού των σημείων ακμών που έχουν το υψηλότερο μέτρο κλίσης.

Υπολογίζεται το μετρό κλίσης για κάθε εικονοστοιχείο, στην περίπτωση που αυτό υπερβεί το κατώφλι που υπολογίσαμε κατά τη διαδικασία της καταφλίωσης τότε ορίζεται πως το εικονοστοιχείο αποτελεί σημείο ακμής και λαμβάνει τιμή 255 (αλλιώς λαμβάνει τιμή 0). Με τη διαδικασία αυτή σαρώνεται όλη η εικόνα και όσα σημεία ανιχνευθούν ότι αποτελούν σημεία ακμών αναδεικνύονται στο προσκήνιο. Τέλος υπάρχει το βήμα της λέπτυνσης των πιθανών γραμμών ώστε να έχουν πάχος ενός εικονοστοιχείου.

2.2.3 Αναγνώριση ακμών με τη μέθοδο Sobel

Ο ανιχνευτής ακμών Sobel (Sobel edge detector), τον οποίο χρησιμοποιήσαμε στην παρούσα εργασία, είναι μια από τις πιο γνωστές μεθόδους και δίνει έμφαση σε περιοχές υψηλής χωρικής συχνότητας, οι οποίες αντιστοιχούν σε ακμές.

Η τεχνική εφαρμόζεται με χρήση ενός τελεστή (τελεστής Sobel) ο οποίος αποτελείται από δύο φίλτρα (μάσκες ακμών), ένα για να ανιχνεύσει τις αλλαγές στην κάθετη κατεύθυνση (G_y) και το άλλο για να ανιχνεύσει τις αλλαγές στην οριζόντια (G_x).

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ και } G_y = \begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Κατά τη διαδικασία ανίχνευσης ακμών πραγματοποιείται συνέλιξη μεταξύ της εικόνας και των δύο αυτών μασκών με τη κάθε μια ξεχωριστά. Μετά την εφαρμογή αυτής της διαδικασίας ακολουθεί καταφλίωση που συνήθως έγκειται στη διατήρηση ενός ποσοστού των σημείων ακμών που έχουν το υψηλότερο μέτρο κλίσης.

Υπολογίζεται το μετρό κλίσης για κάθε εικονοστοιχείο, στην περίπτωση που αυτό υπερβεί το κατώφλι που υπολογίσαμε κατά τη διαδικασία της καταφλίωσης τότε ορίζεται πως το εικονοστοιχείο αποτελεί σημείο ακμής και λαμβάνει τιμή 255 (αλλιώς λαμβάνει τιμή 0). Με τη διαδικασία αυτή σαρώνεται όλη η εικόνα και όσα σημεία ανιχνευθούν ότι αποτελούν σημεία ακμών αναδεικνύονται στο προσκήνιο. Τέλος υπάρχει το βήμα της λέπτυνσης των πιθανών γραμμών ώστε να έχουν πάχος ενός εικονοστοιχείου.

Ο τελεστής Sobel υπολογίζει την απόκλιση της έντασης κάθε εικονοστοιχείο σε σχέση με τα γειτονικά του και δίνει ως αποτέλεσμα την πιο πιθανή μέγιστη αύξηση από τα σκούρα, στα πιο ανοικτά, καθώς και τη μεταβολή αυτής. Ουσιαστικά δείχνει πώς μεταβάλλεται η τιμή φωτεινότητας στις οριζόντιες και κάθετες κατευθύνσεις, και γι' αυτό τα σημεία μεγάλης μεταβολής είναι πολύ πιθανό να είναι ακμές. Ανήκει στις τεχνικές ανίχνευσης ακμών με χρήση πρώτης παραγώγου.

2.2.4 Αναγνώριση ακμών με τη μέθοδο Canny

Ο αλγόριθμος Canny αποτελεί μια από τις καλύτερες τεχνικές ανίχνευσης ακμών. Αποτελεί τον πιο διαδεδομένο αλγόριθμο κατάτμησης εικόνας βασισμένο σε ακμές. Είναι ένας σύνθετος πολυσταδιακός αλγόριθμος που αναπτύχθηκε το 1986 από τον John F. Canny με στόχο τη βέλτιστη και ευρεία ανίχνευση ακμών. Τα στάδια τα οποία εκτελείτε παρατίθενται παρακάτω:

Αρχικά η εικόνα φιλτράρεται με γκαουσιανό φίλτρο με στόχο την αφαίρεση θορύβου. Το αποτέλεσμα είναι μια πιο θαμπή εικόνα από την αρχική. Τα γκαουσιανά φίλτρα είναι γραμμικά φίλτρα με συντελεστές που επιλέγονται από το σχήμα της γκαουσιανής συνάρτησης μηδενικής μέσης τιμής και σ τυπικής απόκλισης που (σε μία διάσταση) έχει την μορφή :

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

Για την επεξεργασία εικόνας και για εύρεση των συντελεστών του παραθύρου χρησιμοποιούμε την αντίστοιχη (διακριτή) σχέση:

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

Η μάσκα που χρησιμοποιήσαμε εμείς για να φιλτράρουμε την εικόνα :

$$\frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Στη συνέχεια υπολογίζεται η πρώτη παράγωγος της εικόνας η οποία περιλαμβάνει την πληροφορία των ακμών. Ο υπολογισμός αυτός επιτυγχάνεται με την εφαρμογή μασκών όπως Sobel, Prewit ή Roberts. Με αυτό τον τρόπο υπολογίζονται οι παράγωγοι της εικόνας στην κατακόρυφη και την οριζόντια διεύθυνση. Στη συνέχεια οι κατευθύνσεις των ακμών υπολογίζονται σε οριζόντιες, κατακόρυφες και διαγώνιες.

Ακολουθεί το βήμα της συμπίεσης των μη τοπικών μεγίστων. Πραγματοποιείται μια αναζήτηση στις τιμές της παραγώγου της εικόνας για να βρεθούν αν οι υψηλές τιμές συμπίπτουν με την κατεύθυνση της κλίσης.

Στη συνέχεια πραγματοποιείται έλεγχος για το αν τα πιθανά σημεία των ακμών παρουσιάζουν συνέχεια ή αν είναι μεμονωμένα. Οι πιθανότητες είναι ότι τα περισσότερα σημεία των ακμών θα βρίσκονται πάνω σε συνεχόμενες γραμμές.

Τέλος υπάρχει το βήμα της λέπτυνσης των πιθανών γραμμών ώστε να έχουν πάχος ενός εικονοστοιχείου.

2.3 Πρακτικά παραδείγματα με τη χρήση του matlab

2.3.1 Αναγνώριση ακμών με τη μέθοδο Roberts

```
%Εισαγωγή εικόνας
I = imread ('tourta.jpg');
I=rgb2gray(I);
alfa=0.068;

%Υπολογισμός x, y κλίσης
Rx=[1 0;0 -1];
Ix=filter2(Rx,I);
Ry=[0 1;-1 0];
Iy=filter2(Ry,I);

% Υπολογισμός μέτρου κλίσης
edge_r=sqrt(Ix.^2+Iy.^2);

%Thresholding
I_max=max(max(edge_r));
I_min=min(min(edge_r));
level=alfa*(I_max-I_min)+I_min;
edge_t=max(edge_r,level);
edge_t(edge_t==round(level))=0;

%Thinning
[n,m]=size(edge_t);
for i=2:n-1,
for j=2:m-1,
    if edge_t(i,j) > level,
        X=[-1,0,+1;-1,0,+1;-1,0,+1];
        Y=[-1,-1,-1;0,0,0;+1,+1,+1];
        Z=[edge_t(i-1,j-1),edge_t(i-1,j),edge_t(i-1,j+1);
            edge_t(i,j-1),edge_t(i,j),edge_t(i,j+1);
            edge_t(i+1,j-1),edge_t(i+1,j),edge_t(i+1,j+1)];
        XI=[Ix(i,j)/edge_r(i,j), -Ix(i,j)/edge_r(i,j)];
        YI=[Iy(i,j)/edge_r(i,j), -Iy(i,j)/edge_r(i,j)];
        ZI=interp2(X,Y,Z,XI,YI);
        if edge_t(i,j) >= ZI(1) & edge_t(i,j) >= ZI(2)
            I_temp(i,j)=I_max;
        else
            I_temp(i,j)=I_min;
        end
    else
        I_temp(i,j)=I_min;
    end
end
end
end
```



```

%Αποτελέσματα
figure
subplot(2,2,1)
imshow(Ix/255)
title('Ix');
subplot(2,2,2)
imshow(Iy/255)
title('Iy');
subplot(2,2,3)
imshow(edge_r/255)
title('Gradient Image');
subplot(2,2,4)
imshow(edge_t/255,[])
title('Thresholded Image');
figure
imshowpair(I,I_temp,'montage')
title('Original Image           Edge Image');

```

Λειτουργία κώδικα

Αφότου εισάγουμε την εικόνα τη μετατρέπουμε από τρισδιάστατη έγχρωμη σε δισδιάστατη στη κλίμακα του γκρι. Υπολογίζουμε την κλίση της εικόνας ως προς x και y καθώς επίσης και το μέτρο της κλίσης το οποίο το κάνουμε χρησιμοποιώντας τις μάσκες:

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ και } G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

εφαρμόσουμε στο μέτρο της κλίσης έναν αλγόριθμο κατωφλίωσης (thresholding) ώστε να δημιουργήσουμε τις ακμές όμως επειδή είναι πολύ χοντρές και παραπάνω από τις πραγματικές ολοκληρώνουμε τη μέθοδο εύρεσης ακμών εφαρμόζοντας έναν αλγόριθμο λέπτυνσης (thinning) και έτσι έχουμε τη τελική εικόνα που δείχνει τις ακμές.

2.3.2 Αναγνώριση ακμών με τη μέθοδο Pretwitt

```
%Εισαγωγή εικόνας
I = imread ('tourta.jpg');
I=rgb2gray(I);
alfa=0.068;

%Υπολογισμός x, y κλίσης
Px=[-1 0 1;-1 0 1;-1 0 1];
Ix=filter2(Px,I);
Py=[-1 -1 -1;0 0 0;1 1 1];
Iy=filter2(Py,I);

% Υπολογισμός μέτρου κλίσης
edge_p=sqrt(Ix.^2+Iy.^2);

%Threshholding
I_max=max(max(edge_p));
I_min=min(min(edge_p));
level=alfa*(I_max-I_min)+I_min;
edge_t=max(edge_p,level);
edge_t(edge_t==round(level))=0;

%Thining Edges
[n,m]=size(edge_t);
for i=2:n-1,
for j=2:m-1,
    if edge_t(i,j) > level,
        X=[-1,0,+1;-1,0,+1;-1,0,+1];
        Y=[-1,-1,-1;0,0,0;+1,+1,+1];
        Z=[edge_t(i-1,j-1),edge_t(i-1,j),edge_t(i-1,j+1);
            edge_t(i,j-1),edge_t(i,j),edge_t(i,j+1);
            edge_t(i+1,j-1),edge_t(i+1,j),edge_t(i+1,j+1)];
        XI=[Ix(i,j)/edge_p(i,j), -Ix(i,j)/edge_p(i,j)];
        YI=[Iy(i,j)/edge_p(i,j), -Iy(i,j)/edge_p(i,j)];
        ZI=interp2(X,Y,Z,XI,YI);
        if edge_t(i,j) >= ZI(1) & edge_t(i,j) >= ZI(2)
            I_temp(i,j)=I_max;
        else
            I_temp(i,j)=I_min;
        end
    else
        I_temp(i,j)=I_min;
    end
end
end
end
```

```

%Αποτελέσματα
figure
subplot(2,2,1)
imshow(Ix/255)
title('Ix');
subplot(2,2,2)
imshow(Iy/255)
title('Iy');
subplot(2,2,3)
imshow(edge_p/255)
title('Gradient Image');
subplot(2,2,4)
imshow(edge_t/255,[])
title('Thresholded Image');
figure
imshowpair(I,I_temp,'montage')
title('Original Image           Edge Image');

```

Λειτουργία κώδικα

Αφότου εισάγουμε την εικόνα τη μετατρέπουμε από τρισδιάστατη έγχρωμη σε δισδιάστατη στη κλίμακα του γκρι. Υπολογίζουμε την κλίση της εικόνας ως προς x και y καθώς επίσης και το μέτρο της κλίσης το οποίο το κάνουμε χρησιμοποιώντας τις μάσκες:

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ και } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

εφαρμόσουμε στο μέτρο της κλίσης έναν αλγόριθμο κατωφλίωσης (thresholding) ώστε να δημιουργήσουμε τις ακμές όμως επειδή είναι πολύ χοντρές και παραπάνω από τις πραγματικές ολοκληρώνουμε τη μέθοδο εύρεσης ακμών εφαρμόζοντας έναν αλγόριθμο λέπτυνσης (thining) και έτσι έχουμε τη τελική εικόνα που δείχνει τις ακμές.

2.3.3 Αναγνώριση ακμών με τη μέθοδο Sobel

```
%Εισαγωγή εικόνας
I = imread ('tourta.jpg');
I=rgb2gray(I);
alfa=0.068;

%Υπολογισμός x, y κλίσης
Sx=[-1 0 1;-2 0 2;-1 0 1];
Ix=filter2(Sx,I);
Sy=[-1 -2 -1;0 0 0;1 2 1];
Iy=filter2(Sy,I);

% Υπολογισμός μέτρου κλίσης
edge_s=sqrt(Ix.^2+Iy.^2);

%Threshholding
I_max=max(max(edge_s));
I_min=min(min(edge_s));
level=alfa*(I_max-I_min)+I_min;
edge_t=max(edge_s,level);
edge_t(edge_t==round(level))=0;
%Thining Edges
[n,m]=size(edge_t);
for i=2:n-1,
for j=2:m-1,
    if edge_t(i,j) > level,
        X=[-1,0,+1;-1,0,+1;-1,0,+1];
        Y=[-1,-1,-1;0,0,0;+1,+1,+1];
        Z=[edge_t(i-1,j-1),edge_t(i-1,j),edge_t(i-1,j+1);
            edge_t(i,j-1),edge_t(i,j),edge_t(i,j+1);
            edge_t(i+1,j-1),edge_t(i+1,j),edge_t(i+1,j+1)];
        XI=[Ix(i,j)/edge_s(i,j), -Ix(i,j)/edge_s(i,j)];
        YI=[Iy(i,j)/edge_s(i,j), -Iy(i,j)/edge_s(i,j)];
        ZI=interp2(X,Y,Z,XI,YI);
        if edge_t(i,j) >= ZI(1) & edge_t(i,j) >= ZI(2)
            I_temp(i,j)=I_max;
        else
            I_temp(i,j)=I_min;
        end
    else
        I_temp(i,j)=I_min;
    end
end
end
end
```

```

%Αποτελέσματα
figure
subplot(2,2,1)
imshow(Ix/255)
title('Ix');
subplot(2,2,2)
imshow(Iy/255)
title('Iy');
subplot(2,2,3)
imshow(edge_s/255)
title('Gradient Image');
subplot(2,2,4)
imshow(edge_t/255,[])
title('Thresholded Image');
figure
imshowpair(I,I_temp,'montage')
title('Original Image           Edge Image');

```

Λειτουργία κώδικα

Αφότου εισάγουμε την εικόνα τη μετατρέπουμε από τρισδιάστατη έγχρωμη σε δισδιάστατη στη κλίμακα του γκρι. Υπολογίζουμε την κλίση της εικόνας ως προς x και y καθώς επίσης και το μέτρο της κλίσης το οποίο το κάνουμε χρησιμοποιώντας τις μάσκες:

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ και } G_y = \begin{bmatrix} -1 & 0 & -1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

εφαρμόσουμε στο μέτρο της κλίσης έναν αλγόριθμο κατωφλίωσης (thresholding) ώστε να δημιουργήσουμε τις ακμές όμως επειδή είναι πολύ χοντρές και παραπάνω από τις πραγματικές ολοκληρώνουμε τη μέθοδο εύρεσης ακμών εφαρμόζοντας έναν αλγόριθμο λέπτυνσης (thinning) και έτσι έχουμε τη τελική εικόνα που δείχνει τις ακμές.

2.3.4 Αναγνώριση ακμών με τη μέθοδο Canny

```
% Εισαγωγή εικόνας
I = imread ('tourta.jpg');
I=rgb2gray(I);
I=double(I);
alfa=0.068;
%Εξομάλυνση εικόνας με γκαουτσιανό φίλτρο
GaussianFilter=[2 4 5 4 2;4 9 12 9 4;5 12 15 12 5;4 9 12 9 4;2 4 5 4 2]*1/159;
I=imfilter(I,GaussianFilter);

%Υπολογισμός x, y κλίσης
Cx = [-1, 0, 1; -2, 0, 2; -1, 0, 1];
Ix= imfilter(I, Cx);
Cy = [-1, -2, -1; 0, 0, 0; 1, 2, 1];
Iy= imfilter(I, Cy);

%Υπολογισμός μέτρου κλίσης
edge_c = sqrt(Ix.^2+Iy.^2);

%Threshholding
I_max=max(max(edge_c));
I_min=min(min(edge_c));
level=alfa*(I_max-I_min)+I_min;
edge_t=max(edge_c,level);
edge_t(edge_t==round(level))=0;

%Thining Edges
[n,m]=size(edge_t);
for i=2:n-1,
for j=2:m-1,
    if edge_t(i,j) > level,
        X=[-1,0,+1;-1,0,+1;-1,0,+1];
        Y=[-1,-1,-1;0,0,0;+1,+1,+1];
        Z=[edge_t(i-1,j-1),edge_t(i-1,j),edge_t(i-1,j+1);
            edge_t(i,j-1),edge_t(i,j),edge_t(i,j+1);
            edge_t(i+1,j-1),edge_t(i+1,j),edge_t(i+1,j+1)];
        XI=[Ix(i,j)/edge_c(i,j), -Ix(i,j)/edge_c(i,j)];
        YI=[Iy(i,j)/edge_c(i,j), -Iy(i,j)/edge_c(i,j)];
        ZI=interp2(X,Y,Z,XI,YI);
        if edge_t(i,j) >= ZI(1) & edge_t(i,j) >= ZI(2)
            I_temp(i,j)=I_max;
        else
            I_temp(i,j)=I_min;
        end
    else
        I_temp(i,j)=I_min;
    end
end
end
end
```

```

%Αποτελέσματα
figure
subplot(2,2,1)
imshow(Ix/255)
title('Ix');
subplot(2,2,2)
imshow(Iy/255)
title('Iy');
subplot(2,2,3)
imshow(edge_c/255)
title('Gradient Image');
subplot(2,2,4)
imshow(edge_t/255,[])
title('Thresholded Image');
figure
imshowpair(I,I_temp,'montage')
title('Original Image           Edge Image');

```

Λειτουργία κώδικα

Αφότου εισάγουμε την εικόνα τη μετατρέπουμε από τρισδιάστατη έγχρωμη σε δισδιάστατη στη κλίμακα του γκρι. Την φιλτράρουμε με ένα γκαουτσιανό φίλτρο 5x5 το οποίο πρακτικά κόβει της υψηλές συχνότητες της εικόνας και θολώνει την εικόνα αυτό το κάνουμε γιατί στην συνέχεια που θα κάνουμε την αναγνώριση ακμών να μειώσουμε την ευαισθησία του αλγορίθμου και θα αναγνωρίσει μόνο τις ισχυρές ακμές. Υπολογίζουμε τρία βασικά στοιχεία την κλίση της εικόνας ως προς x και y καθώς επίσης και το μετρό της κλίσης το οποίο το κάνουμε χρησιμοποιώντας τις μάσκες που χρησιμοποιούμε στον αλγόριθμο Sobel. Το επόμενο μας βήμα είναι εφαρμόσουμε στο μετρό της κλίσης έναν αλγόριθμο κατωφλίωσης (thresholding) ώστε να δημιουργήσουμε τις ακμές όμως επειδή είναι πολύ χοντρές και παραπάνω από τις πραγματικές ολοκληρώνουμε τη μέθοδο εύρεσης ακμών εφαρμόζοντας έναν αλγόριθμο λέπτυνσης (thinning) και έτσι έχουμε τη τελική εικόνα που δείχνει τις ακμές.

2.3.5 Αναγνώριση ακμών χρησιμοποιώντας την εντολή edge

```
alfa=0.099;  
I = imread ('tourta.jpg');  
I=rgb2gray(I);  
  
%Sobel, Prewitt, Roberts and Canny  
BW1 = edge(I,'Sobel',alfa,'both');  
BW2 = edge(I,'Prewitt',alfa,'both');  
BW3 = edge(I,'Roberts',alfa,'both');  
BW4 = edge(I,'Canny',alfa,'both');  
imshow(I,[]);  
title('original Image');  
  
figure('units','normalized','outerposition',[0 0 1 1])  
  
subplot(2,2,1)  
imshow(BW1,[]);  
title('Sobel');  
  
subplot(2,2,2)  
imshow(BW2,[]);  
title('Prewitt');  
  
subplot(2,2,3)  
imshow(BW3,[]);  
title('Roberts');  
  
subplot(2,2,4)  
imshow(BW4,[]);  
title('Canny');
```

Λειτουργία κώδικα

Πρακτικά αυτό που κάνουμε είναι να χρησιμοποιήσουμε μια έτοιμη συνάρτηση που έχει το Matlab ώστε να βρίσκει τις ακμές δίνοντας τη μέθοδο που θα εφαρμόζει και την ευαισθησία που θα πρέπει να έχει ως προς τις ακμές που πρέπει να βρει.

Συμπεράσματα-Αποτελέσματα

Συγκριτικά τα καλύτερα και πιο ακριβή αποτελέσματα τα δίνει ο αλγόριθμος του Canny. Για να πετύχουμε βέλτιστα αποτελέσματα παίζουμε με τιμή του α το οποίο καθορίζει την ευαισθησία του αλγορίθμου στις ακμές. Για να συγκρίνουμε την αποτελεσματικότητα των αλγορίθμων μεταξύ τους δίνουμε την ίδια τιμή για το α . Επίσης οι παραπάνω αλγόριθμοι δεν συναντάνε ιδιαίτερα προβλήματα σε σύνθετες εικόνες. Τέλος η συνάρτηση edge υστερεί σε αποτελεσματικότητα έναντι των αλγορίθμων που φτιάξαμε.

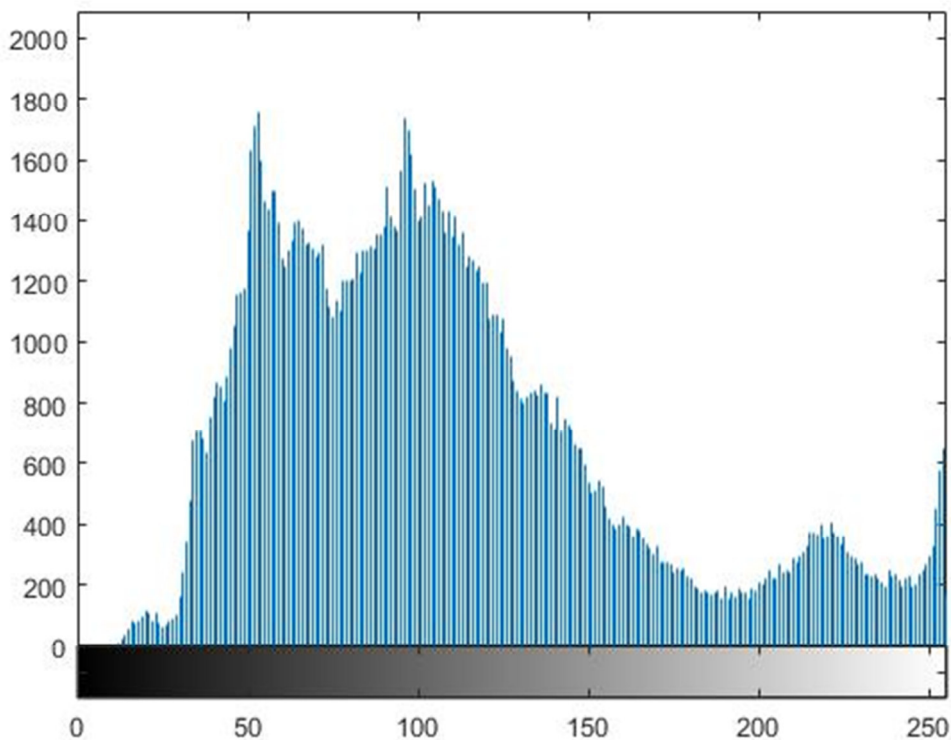
Αρχική εικόνα:



Εικόνα 19: Αρχική εικόνα

Ιστόγραμμα εικόνας:

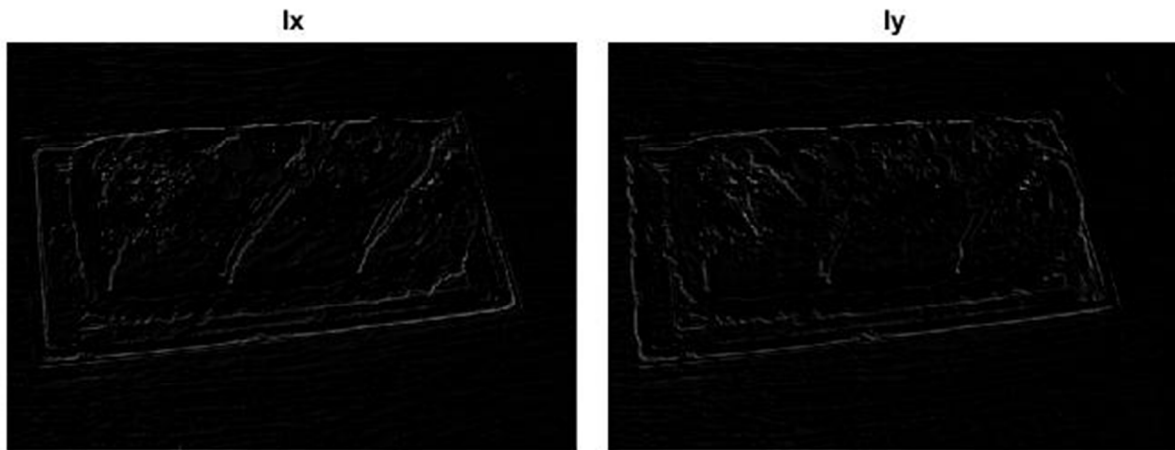
Το ιστόγραμμα της εικόνας, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι σε φυσιολογικά επίπεδα. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων όπου στη συγκεκριμένη εικόνα είναι χαμηλά.



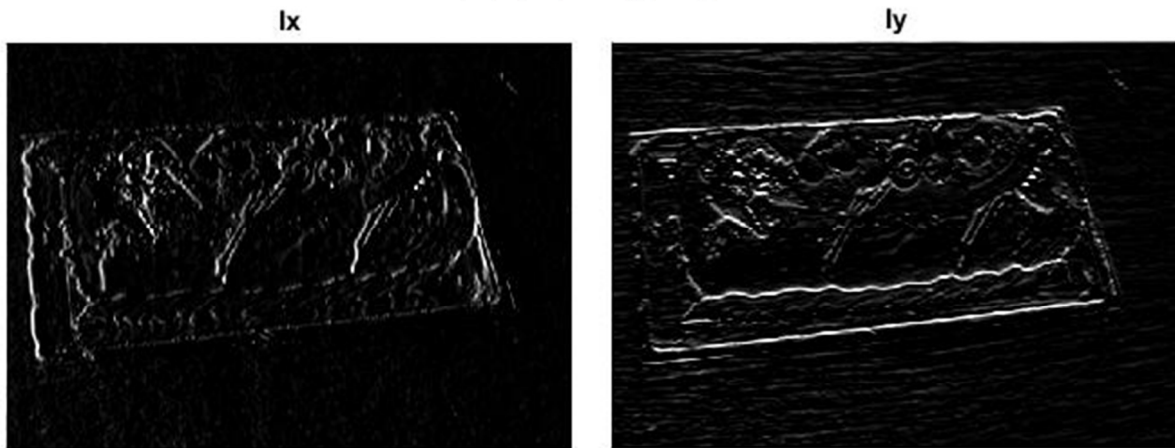
Εικόνα 20: Ιστόγραμμα εικόνας

Υπολογισμός x, y κλίσης:

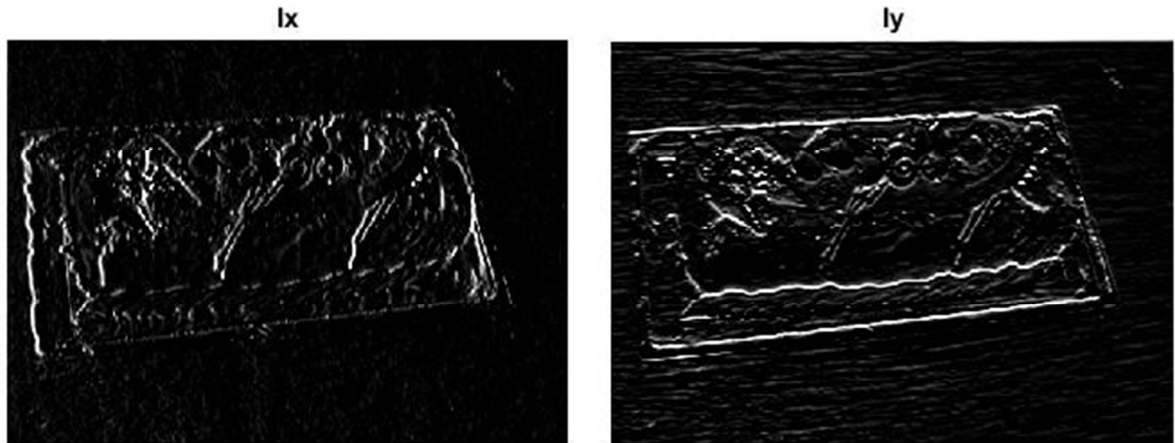
Ο υπολογισμός της κλίση μιας εικόνας μας δείχνει τη διεύθυνση του μεγαλύτερου ρυθμού μεταβολής μιας εικόνας.



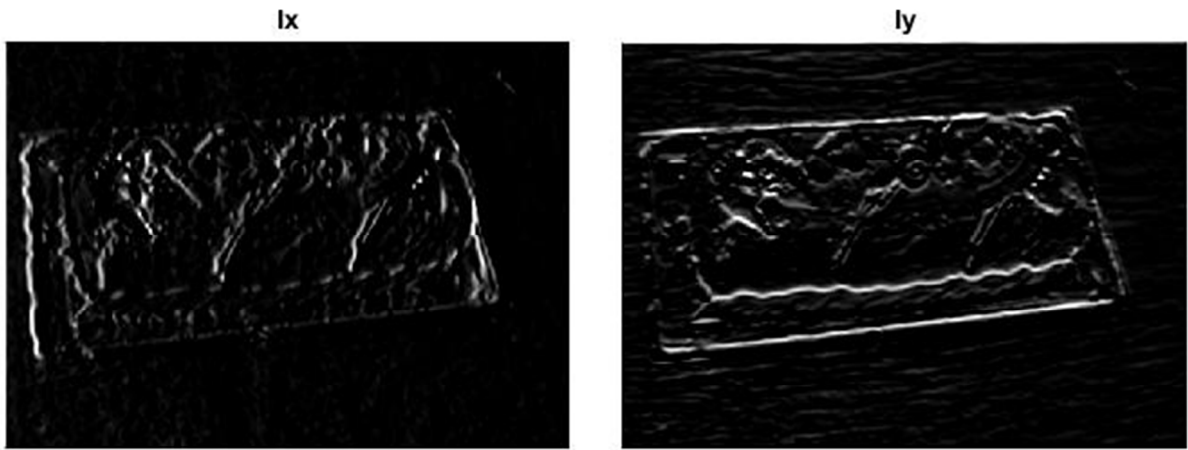
Εικόνα 21: Υπολογισμός x, y κλίσης με τη μέθοδο του Roberts



Εικόνα 22: Υπολογισμός x, y κλίσης με τη μέθοδο του Prewitt



Εικόνα 23: Υπολογισμός x, y κλίσης με τη μέθοδο του Sobel

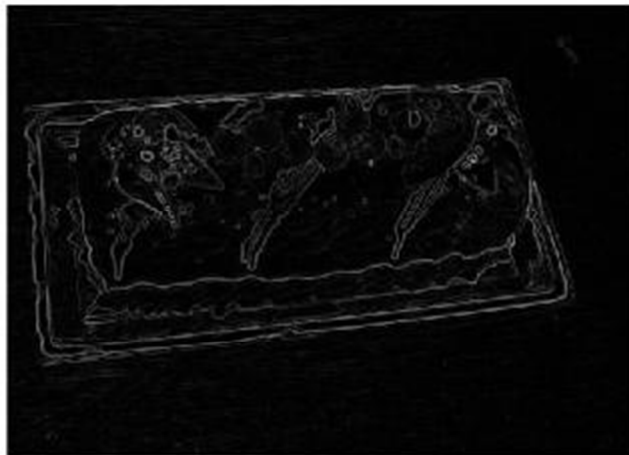


Εικόνα 24: Υπολογισμός x, y κλίσης με τη μέθοδο του Canny

Υπολογισμός μέτρου κλίσης:

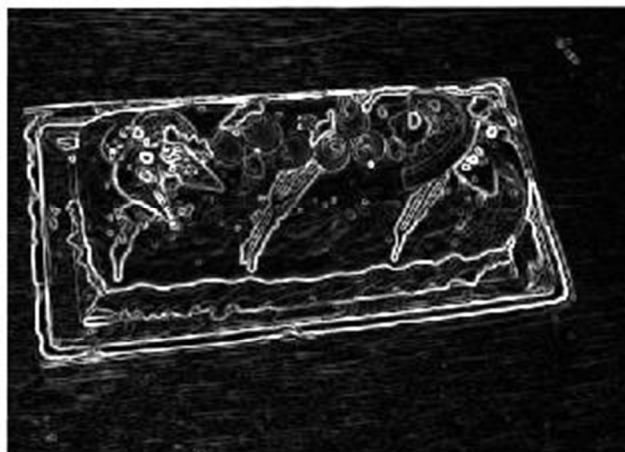
Ο υπολογισμός του μέτρου κλίσης μιας εικόνας είναι η τιμή του ρυθμού μεταβολής κατά τη διεύθυνση της κλίσης.

Gradient Image



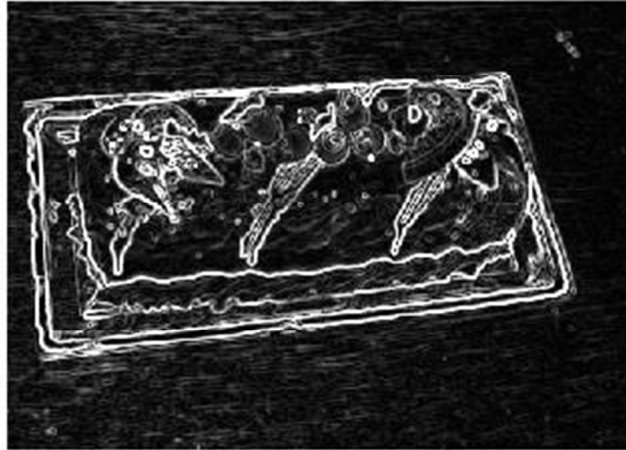
Εικόνα 25: Υπολογισμός μέτρου κλίσης με τη μέθοδο του Roberts

Gradient Image



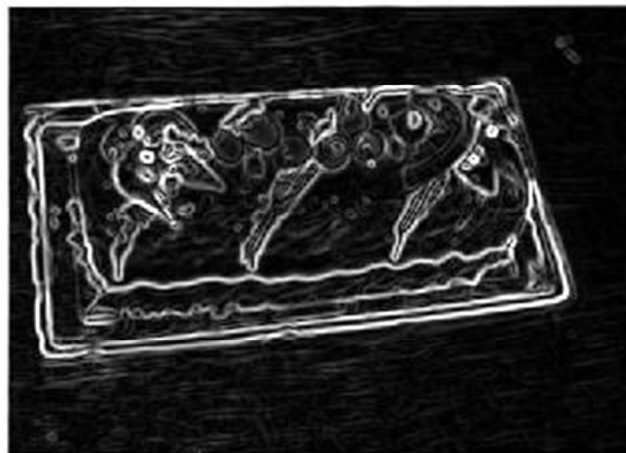
Εικόνα 26: Υπολογισμός μέτρου κλίσης με τη μέθοδο του Prewitt

Gradient Image



Εικόνα 27: Υπολογισμός μέτρου κλίσης με τη μέθοδο του Sobel

Gradient Image

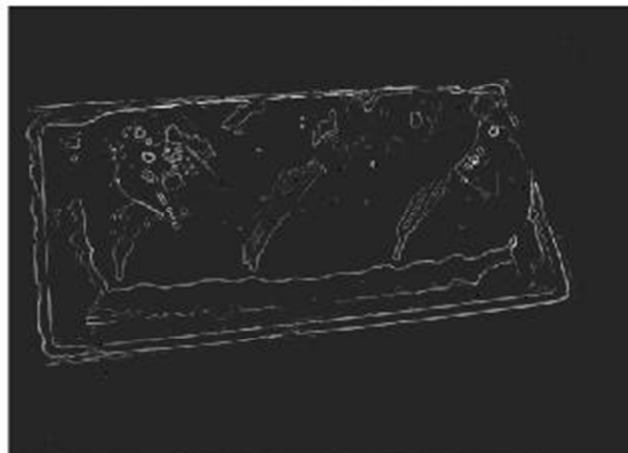


Εικόνα 28: Υπολογισμός μέτρου κλίσης με τη μέθοδο του Canny

Κατωφλίωση εικόνας:

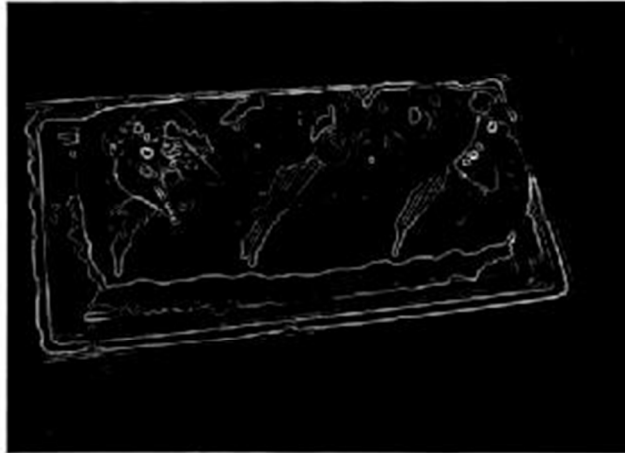
Εφαρμόζουμε στο μετρό της κλίσης έναν αλγόριθμο κατωφλίωσης (thresholding) ώστε να δημιουργήσουμε τις ακμές.

Thresholded Image

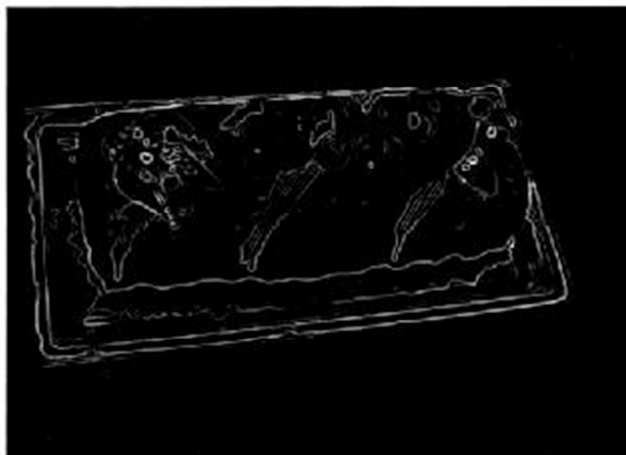


Εικόνα 29: Εικόνα μετά την εφαρμογή κατωφλίωσης με τη μέθοδο του Roberts

Thresholded Image



Εικόνα 30: Εικόνα μετά την εφαρμογή κατοφλίωσης με τη μέθοδο του Prewit
Thresholded Image



Εικόνα 31: Εικόνα μετά την εφαρμογή κατοφλίωσης με τη μέθοδο του Sobel
Thresholded Image



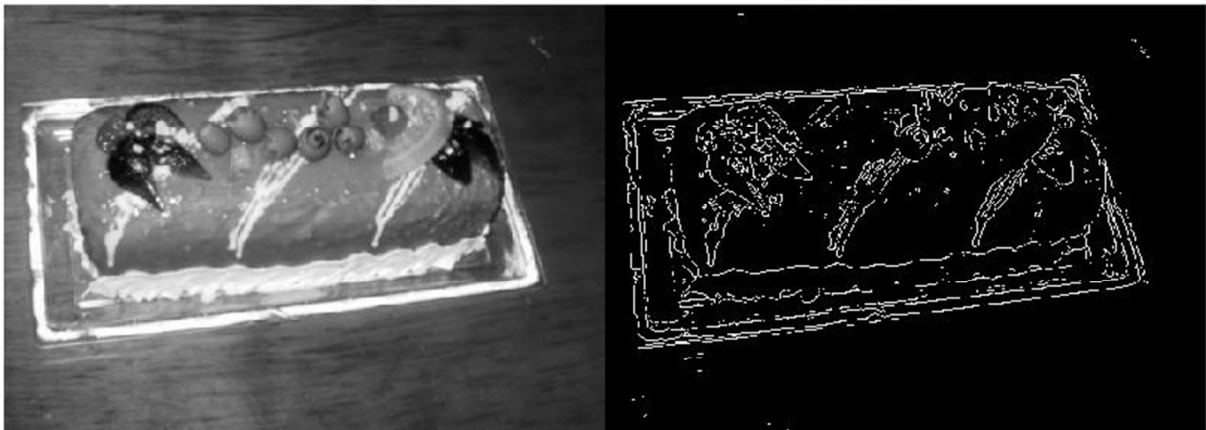
Εικόνα 32: Εικόνα μετά την εφαρμογή κατοφλίωσης με τη μέθοδο του Canny

Αποτελέσματα:

Αφού εφαρμόσαμε ένα αλγόριθμο λέπτυνσης στη κατωφλιωμένη εικόνα παίρνουμε τα τελικά αποτελέσματα.

Original Image

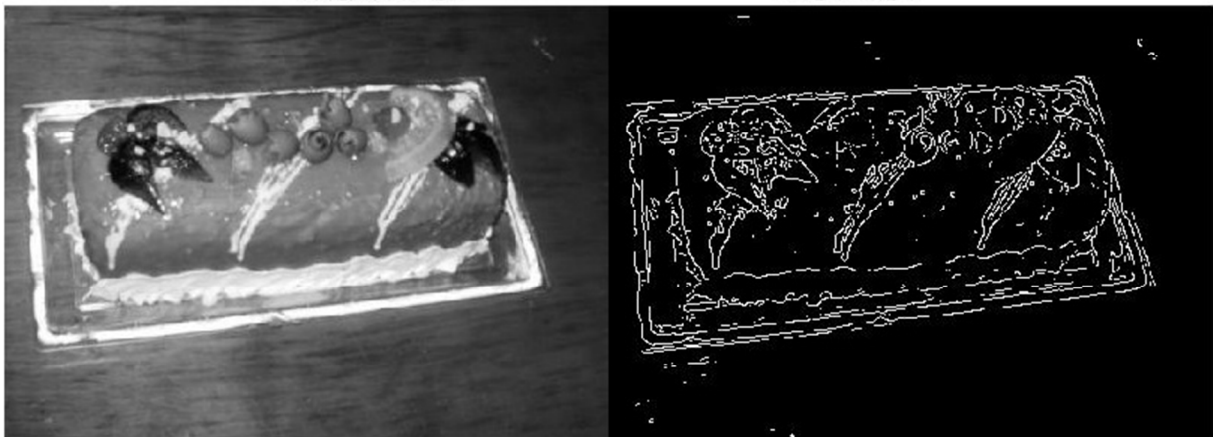
Edge Image



Εικόνα 33: Ανίχνευση ακμών με τη μέθοδο του Roberts

Original Image

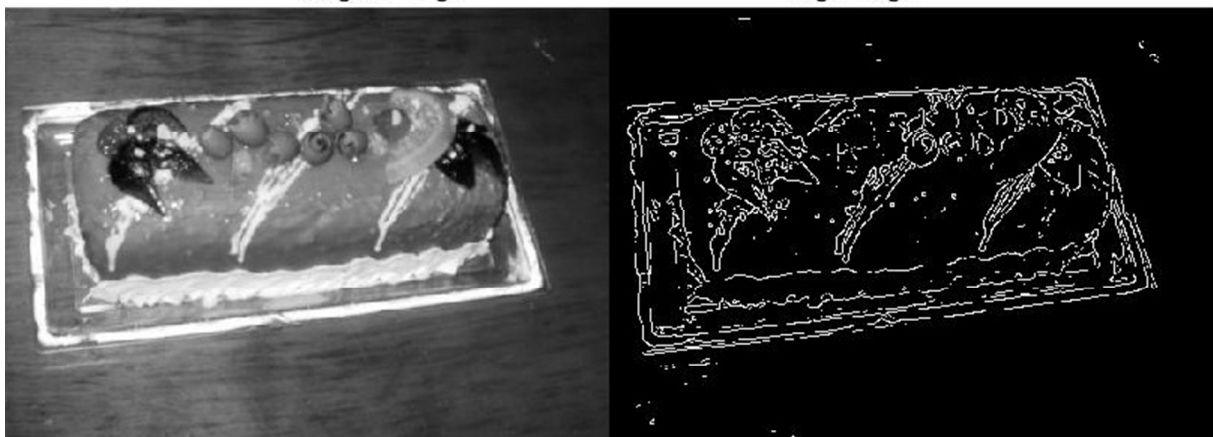
Edge Image



Εικόνα 34: Ανίχνευση ακμών με τη μέθοδο του Prewitt

Original Image

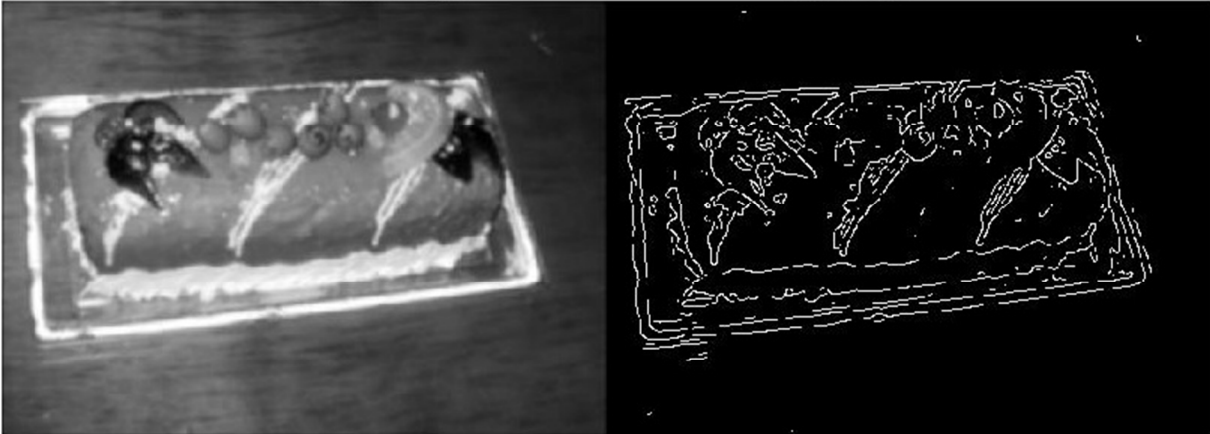
Edge Image



Εικόνα 35: Ανίχνευση ακμών με τη μέθοδο του Sobel

Original Image

Edge Image



Εικόνα 36: Ανίχνευση ακμών με τη μέθοδο του Canny

Sobel

Prewitt



Roberts

Canny



Εικόνα 37: Ανίχνευση ακμών με τη χρήση της εντολής edge

Κεφάλαιο 3

Βασικοί Αλγόριθμοι για την εξαγωγή πληροφορίας

3.1 Εύρεση βασικών γεωμετρικών μεγεθών

3.1.1 Μεθοδολογία ανάπτυξης αλγόριθμου εύρεσης βασικών μεγεθών

Για να βρούμε βασικά γεωμετρικά μεγέθη, όπως η περίμετρος και το εμβαδό στο matlab, υπάρχουν δυο βασικοί τρόποι. Είτε με τη βοήθεια της εντολής `regionprops`, η οποία τα υπολογίζει κατευθείαν μαζί με διάφορα αλλά μεγέθη. Είτε υπολογίζοντας τα ξεχωριστά.

Για να τα υπολογίσουμε, θα πρέπει αρχικά να αναγνωρίσουμε το πλήθος των αντικειμένων στην εικόνα, βρίσκοντας τα στοιχεία των αντικειμένων, που συνδέονται μεταξύ τους χρησιμοποιώντας την εντολή `bwlabel`. Μετά σαρώνουμε την εικόνα, ώστε για κάθε αντικείμενο να υπολογίσει τα εικονοστοιχεία που ανήκουν στο αντικείμενο τα οποία αποθηκεύονται σε ένα πίνακα. Με αυτό το τρόπο χρησιμοποιώντας την εντολή `numel` η οποία υπολογίζει το πλήθος στοιχείων του πίνακα υπολογίζουμε το εμβαδό του εκάστοτε αντικειμένου.

Για να υπολογίσουμε την περίμετρο βρίσκουμε το περίγραμμα του αντικειμένου χρησιμοποιώντας την εντολή `bwboundaries` τα οποία αποθηκεύουμε σε ένα πίνακα. Έτσι για να υπολογίσουμε τη περίμετρο χρησιμοποιούμε το παρακάτω τύπο υπολογισμού: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ όπου τα αντικαθιστούνται με τα στοιχεία του πίνακα και αφού εφαρμόσουμε το τύπο για όλα τα στοιχεία και τα προσθέσουμε μεταξύ τους έχουμε τη περίμετρο.

Στάδια αλγόριθμου εύρεσης βασικών μεγεθών:

1. Εισαγωγή εικόνας
2. Μετατροπή εικόνας από έγχρωμη στην κλίμακα γκρι
3. Μετατροπή εικόνας σε δυαδική μορφή
4. Εύρεση πλήθους αντικειμένων
5. Υπολογισμός μεγεθών

3.1.2 Πρακτική υλοποίηση

1^{ος} τρόπος

```
%Read Original Image
I=imread('eurocoins.jpg');
I=rgb2gray(I);
%Convert to Binary
I1=im2bw(I,graythresh(I));
B=imcomplement(I1);
%Fill the holes
C=imfill(B,'holes');
%Label the image
[Label,Total]=bwlabel(C,8);
%Object Number
for k = 1:Total
num=k;
[row, col] = find(Label==num);
%To find Bounding Box
sx=min(col)-0.5;
sy=min(row)-0.5;
breadth=max(col)-min(col)+1;
len=max(row)-min(row)+1;
BBox=[sx sy breadth len];
figure,imshow(I); hold on;
x=zeros([1 5]);
y=zeros([1 5]);
x(:)=BBox(1);
y(:)=BBox(2);
x(2:3)=BBox(1)+BBox(3);
y(3:4)=BBox(2)+BBox(4);
plot(x,y);
title([ num2str(k) '^ο αντικείμενο'])
%Find Area
Obj_area=numel(row);
Embado(k)=Obj_area;
%Find Centroid
X=mean(col);
Y=mean(row);
Centroid=[X Y];
plot(X,Y,'ro','color','r'); hold off;
%Find Perimeter
BW=bwboundaries(Label==num);
c=cell2mat(BW(1));
Perimeter=0;
for i=1:size(c,1)-1
Perimeter=Perimeter+sqrt((c(i,1)-c(i+1,1)).^2+(c(i,2)-c(i+1,2)).^2);
end
Perimetros(k)=Perimeter;
end
display(Embado);
display(Perimetros);
```

Λειτουργία κώδικα

Αφού διαβάσουμε την εικόνα, την μετατρέπουμε από έγχρωμη στην κλίμακα του γκρι και μετά σε δυαδική μορφή, με την χρήση κατωφλίωσης ώστε να έχουμε καλύτερα δυνατά αποτελέσματα και φτιάχνουμε ένα πίνακα με τα συγκοινωνούντα εικονοστοιχεία, χρησιμοποιώντας την εντολή `bwlabel`, το οποίο εξυπηρετεί δυο σκοπούς, να βρούμε το πλήθος των αντικειμένων, καθώς και τον καθορισμό των αντικειμένων πάνω στην εικόνα. Στη συνέχεια εκμεταλλεύεται τα δυο αυτά γεγονότα ώστε να υπολογίσει την περίμετρο και το εμβαδό.

2^{ος} τρόπος

```
%Read Original Image
I=imread('eurocoins.jpg');
I=rgb2gray(I);
%Convert to Binary
I1=im2bw(I,graythresh(I));
B=imcomplement(I1);
%Fill the holes
C=imfill(B,'holes');
%Label the image
[Label,Total]=bwlabel(C,8);
for k = 1:Total
num=k;
%Calculation with 'regionprops'
Sdata=regionprops(Label,'all');
Embado(k)=Sdata(num).Area;
Perimetros(k)=Sdata(num).Perimeter;
end
display(Embado);
display(Perimetros);
```

Λειτουργία κώδικα

Αφού διαβάσουμε την εικόνα, την μετατρέπουμε από έγχρωμη στην κλίμακα του γκρι και μετά σε δυαδική μορφή, με την χρήση κατωφλίωσης ώστε να έχουμε καλύτερα δυνατά αποτελέσματα και φτιάχνουμε ένα πίνακα με τα συγκοινωνούντα εικονοστοιχεία, χρησιμοποιώντας την εντολή `bwlabel`, το οποίο εξυπηρετεί δυο σκοπούς, να βρούμε το πλήθος των αντικειμένων, καθώς και τον καθορισμό των αντικειμένων πάνω στην εικόνα. Στην συνέχεια χρησιμοποιώντας την εντολή `regionprops` υπολογίζουμε την περίμετρο και το εμβαδό.

Συμπεράσματα –Αποτελέσματα

Οι δυο παραπάνω αλγόριθμοι λειτουργούν πολύ καλά δίνοντας σωστά αποτελέσματα. Επίσης ο δεύτερος αλγόριθμος είναι ουσιαστικά μια επαλήθευση του πρώτου. Τα προβλήματα που μπορεί να συναντήσουμε κατά την εκτέλεση των αλγορίθμων έχουν να κάνουν κυρίως με το αν αντικείμενα δεν έχουν καθοριστεί σωστά ώστε να είναι ξεκάθαρο το πλήθος τους και θέση τους πάνω στην εικόνα.

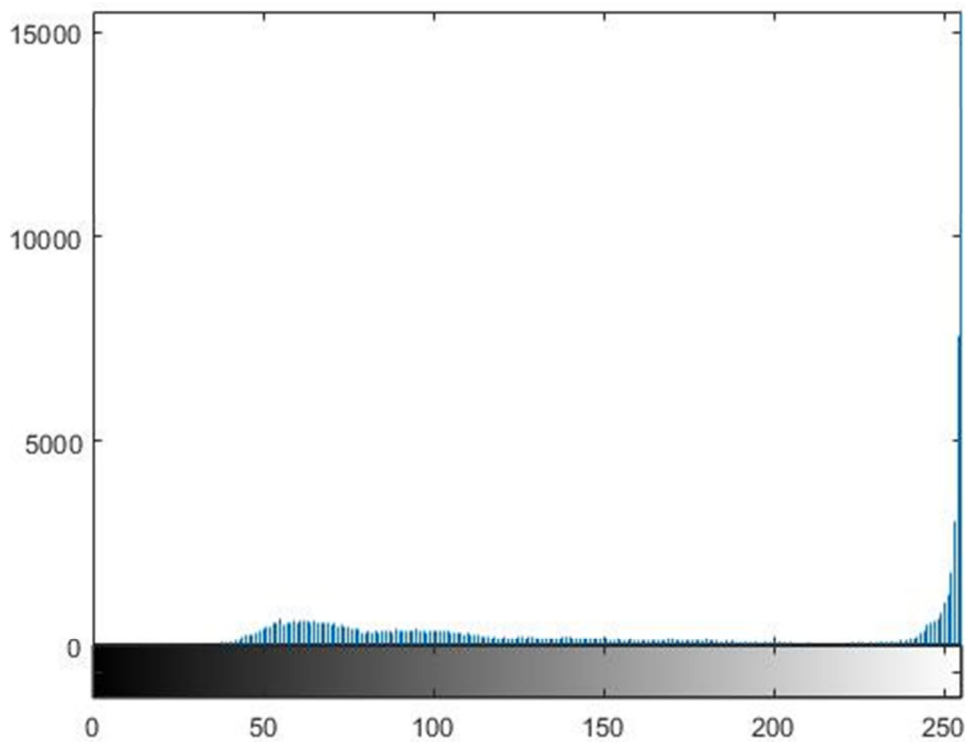
Αρχική εικόνα:



Εικόνα 38: Αρχική εικόνα

Ιστόγραμμα εικόνας:

Το ιστόγραμμα της εικόνας, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι υψηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων οπύ στη συγκεκριμένη εικόνα είναι πάρα πολύ χαμηλά.



Εικόνα 39: Ιστόγραμμα εικόνας

3.2 Μέτρηση γωνιών μεταξύ αντικειμένων

3.2.1 Μεθοδολογία ανάπτυξης αλγόριθμου μέτρησης γωνιών

Για να υπολογίσουμε την γωνία μεταξύ δυο αντικειμένων, θα πρέπει αρχικά, να κάνουμε ξεκάθαρο στο σύστημα μας, πια είναι αυτή η γωνία που θέλουμε να υπολογίσουμε, αυτό το κάνουμε περικόπτοντας την εικόνα και εστιάζοντας στο σημείο που βρίσκεται η γωνία.

Από τη στιγμή που θα κατωφλιώσουμε την εικόνα και θα βρίσκεται σε δυαδική μορφή, μπορούμε να σκεφτούμε, ότι αν τη σαρώσουμε, θα μπορέσουμε να εντοπίσουμε τους άξονες που περικλείουν την γωνία. Γιατί το πρώτο άσπρο σημείο που θα συναντήσουμε κατά τη σάρωση, θα ανήκει στους άξονες. Το οποίο και εκμεταλλευόμαστε, για να δημιουργήσουμε οπτικά τα όρια των αξόνων πάνω στην εικόνα.

Επομένως, τα στάδια υλοποίησης αλγόριθμου μέτρησης γωνιών, που προκύπτουν από την αρχική μας σκέψη, είναι τα παρακάτω:

1. Εισαγωγή εικόνας
2. Εξάγουμε την περιοχή που μας ενδιαφέρει και περιέχει τη γωνιά
3. Μετατροπή από έγχρωμη στην κλίμακα γκρι
4. Κατωφλίωση της εικόνας
5. Εύρεση των αξόνων που περικλείουν την γωνία
6. Οπτικός σχηματισμός των συγκεκριμένων αξόνων
7. Υπολογισμός της γωνιάς

3.2.2 Πρακτική υλοποίηση

```
% Εισαγωγή εικόνας
RGB = imread('GoniaTest1.jpg');
figure
imshow(RGB);
% Αρχικές τιμές ώστε να κρατήσουμε το κομμάτι της εικόνας που μας ενδιαφέρει
start_row = 220;
start_col = 150;
I = RGB(start_row:400, start_col:307, :);
figure
imshow(I)
offsetX = start_col-1;
offsetY = start_row-1;
% Μετατροπή της εικόνας από έγχρωμη σε κλίμακα του γκρι
I1 = rgb2gray(I);
% Μετατροπή σε δυαδική μορφή
BW = im2bw(I1);
BW = ~BW;%Προσοχή οι περιοχές που μας ενδιαφέρουν θα πρέπει να είναι άσπρες
figure
imshow(BW)
dim = size(BW);
% Οριζόντιος άξονας
col1 = 4;
row1 = find(BW(:,col1), 1);
% Άξονας υπο γωνία
row2 = 12;
col2 = find(BW(row2,:), 1);
% Εύρεση ορίου πρώτου άξονα
boundary1 = bwtraceboundary(BW, [row1, col1], 'N', 8, 70);
% Εύρεση ορίου δεύτερου άξονα
boundary2 = bwtraceboundary(BW, [row2, col2], 'E', 8, 90, 'counter');
figure
imshow(RGB); hold on;
% Εμφάνιση ορίων
plot(offsetX+boundary1(:,2),offsetY+boundary1(:,1),'r','LineWidth',2);
plot(offsetX+boundary2(:,2),offsetY+boundary2(:,1),'r','LineWidth',2);
ab1 = polyfit(boundary1(:,2), boundary1(:,1), 1);
ab2 = polyfit(boundary2(:,2), boundary2(:,1), 1);
vect1 = [1 ab1(1)];
vect2 = [1 ab2(1)];
dp = dot(vect1, vect2);
length1 = sqrt(sum(vect1.^2));
length2 = sqrt(sum(vect2.^2));
% εύρεση της γωνίας μεταξύ των δυο αξόνων
angle = 180-acos(dp/(length1*length2))*180/pi
intersection = [1, -ab1(1); 1, -ab2(1)] \ [ab1(2); ab2(2)];
% Σχηματισμός της ευρεθείσας γωνίας πάνω στην αρχική εικόνα
intersection = intersection + [offsetY; offsetX]
inter_x = intersection(2);
inter_y = intersection(1);
plot(inter_x,inter_y,'kx','LineWidth',2);
```

```

text(inter_x-60, inter_y-
30,[sprintf('%1.3f',angle), '{\circ}'], 'Color','k','FontSize',14,'FontWeight','bold');
interString = sprintf('%2.1f,%2.1f', inter_x, inter_y);
text(inter_x-10, inter_y+20, interString, 'Color','k','FontSize',14,'FontWeight','bold');

```

Λειτουργία κώδικα

Αφού διαβάσουμε και περικόψουμε την εικόνα, ώστε, να εστιάσουμε στο σημείο στο οποίο, γίνεται ξεκάθαρη η γωνία την οποία θέλουμε να μετρήσουμε, μετατρέπουμε την εικόνα από έγχρωμη στην κλίμακα του γκρι και μετά σε δυαδική μορφή. Στη συνέχεια, εκμεταλλευόμαστε το γεγονός ότι αν σαρώσουμε την εικόνα, ότι στο πρώτο άσπρο εικονοστοιχείο που θα διαβάσουμε, είναι, το εξωτερικό όριο για κάθε άξονα του αντικείμενου που θέλουμε να μετρήσουμε τη γωνία, οπότε, έχοντας αυτό σαν γνώμονα, σχεδιάζουμε τα αυτά τα όρια αυτών των αξόνων, τα οποία, τα χρησιμοποιούμε ώστε να υπολογίσουμε τη γωνία.

Συμπεράσματα –Αποτελέσματα

Για να μπορέσει ο παραπάνω αλγόριθμος να δώσει σωστά αποτελέσματα θα πρέπει να εστιάσουμε σωστά στη γωνία που θέλουμε να υπολογίσουμε.

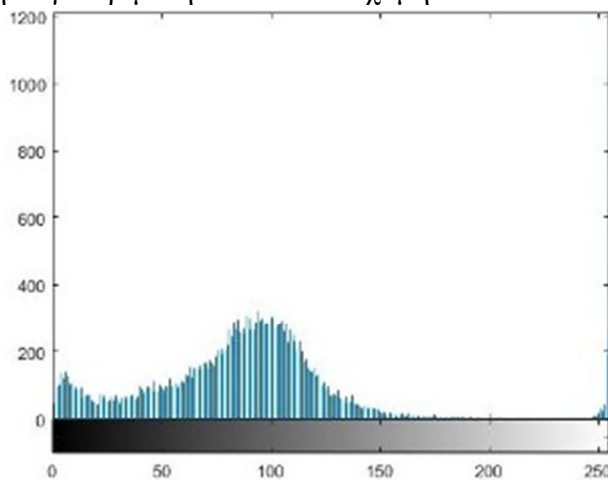
Αρχική εικόνα:



Εικόνα 40: Αρχική εικόνα

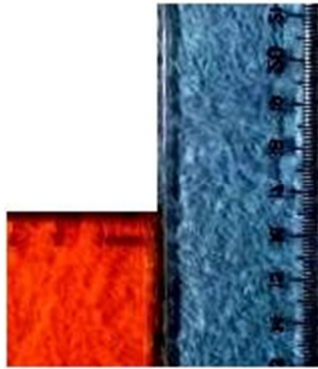
Ιστόγραμμα εικόνας:

Το ιστόγραμμα της εικόνας, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι σε φυσιολογικά επίπεδα. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων όπου στη συγκεκριμένη εικόνα είναι χαμηλά.



Εικόνα 41: Ιστόγραμμα εικόνας

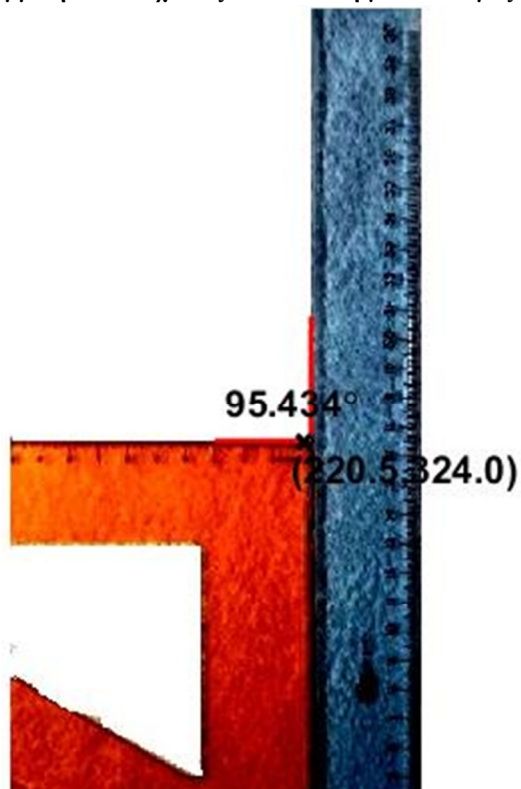
Αποτελέσματα:



Εικόνα 42: Έγχρωμη περικομμένη εικόνα έχοντας εστιάσει στη γωνία που μας ενδιαφέρει να υπολογίσουμε



Εικόνα 43: Δυαδική περικομμένη εικόνα έχοντας εστιάσει στη γωνία που μας ενδιαφέρει να υπολογίσουμε



Εικόνα 44: Τελική εικόνα στην οποία αναγράφεται το αποτέλεσμα μετά τον υπολογισμό

3.3 Αναγνώριση βασικών γεωμετρικών σχημάτων

3.3.1 Μεθοδολογία ανάπτυξης αλγόριθμου αναγνώρισης σχημάτων

Για να αναγνωρίσουμε γεωμετρικά σχήματα εκμεταλλευόμαστε ιδιότητες όπως το εμβαδό, η περίμετρος και οι σχέσεις μεταξύ αυτών. Στο Matlab αξιοποιώντας την ιδιότητα `extend` της εντολής `regionprops` μπορούμε να καταλάβουμε τι σχήμα είναι υπολογίζοντας τη τιμή του εμβαδού.

Στάδια αλγόριθμου αναγνώρισης σχημάτων :

1. Εισαγωγή εικόνας
2. Μετατροπή εικόνας σε δυαδική μορφή
3. Υπολογισμός παραμέτρων σχημάτων εικόνας
4. Αναγνώριση σχήματος

3.3.2 Πρακτική υλοποίηση

```
%Εισαγωγή εικόνας
I=imread('Basic Shapes.jpg');

%Μετατροπή εικόνας σε δυαδική μορφή
I=~im2bw(I);

%Φτιάχνουμε ένα πίνακα με τα συγκοινωνούντα εικονοστοιχεία
I1 = bwlabel(I);

%Υπολογισμός παραμέτρων σχημάτων εικόνας
STATS = regionprops(I1, 'Extent', 'Centroid');

%Δημιουργία πίνακα ώστε να αποθηκευτούν τα δεδομένα της κατηγοριοποίησης των σχημάτων
shape=cell(length(STATS),1);

%Κατηγοριοποίηση σχημάτων και αποθήκευση των αποτελεσμάτων στο πίνακα shape
for cnt = 1:length(STATS)
    if(STATS(cnt).Extent>0.25 && STATS(cnt).Extent<0.6)
        shape{cnt}=1;%triangle
    elseif(STATS(cnt).Extent>=0.78 && STATS(cnt).Extent<=0.79)
        shape{cnt}=2;%circle
    elseif(STATS(cnt).Extent==1)
        shape{cnt}=3;%square
    end
end

%Εμφάνιση αποτελεσμάτων
imshow(I)
title('RESULTS: 1=TRIAGNLE -- 2=CIRCLE -- 3=SQUARE ')

%Εισαγωγή κειμένου στη εικόνα με σκοπό την ανάδειξη των αποτελεσμάτων
for cnt = 1:length(STATS)
    text(STATS(cnt).Centroid(1),STATS(cnt).Centroid(2),num2str(shape{cnt}),'color','red');
end
```

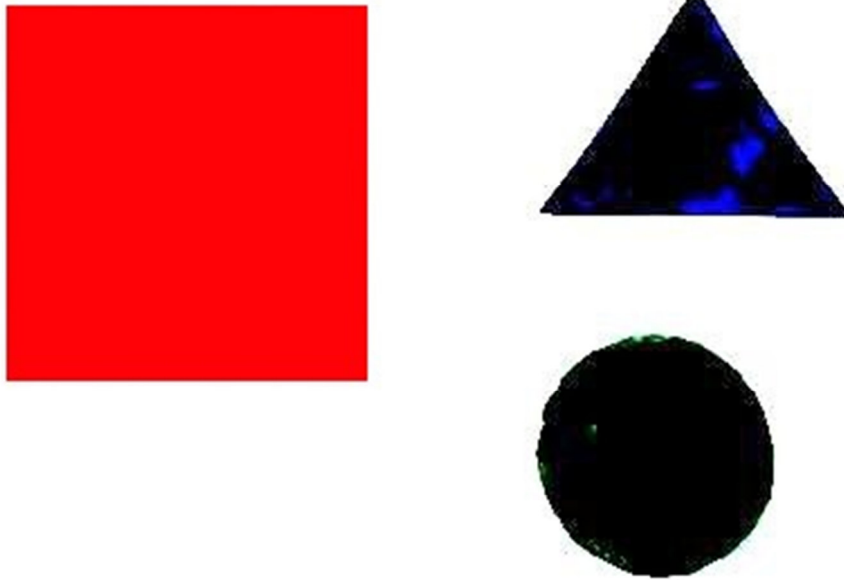
Λειτουργία κώδικα

Αφού διαβάσουμε την εικόνα την μετατρέπουμε από έγχρωμη σε δυαδική μορφή και φτιάχνουμε ένα πίνακα με τα συγκοινωνούντα εικονοστοιχεία ουσιαστικά τα περιγράμματα των αντικειμένων. Υπολογίζουμε το εμβαδό των αντικειμένων αξιοποιώντας την ιδιότητα extend της εντολής regionprops καθώς επίσης και το κέντρο του κάθε αντικείμενου χρησιμοποιώντας την ιδιότητα centroid της ίδιας εντολής. Και βάση της τιμής του extend μπορούμε να προσδιορίσουμε αν το σχήμα μας είναι τρίγωνο ($0.25 < \text{extend} < 0.6$), τετράγωνο ($\text{extend} = 1$) η κύκλος ($0.78 \leq \text{extend} \leq 0.79$). Για την απεικόνιση των αποτελεσμάτων χρησιμοποιούμε αριθμούς που έχουμε ορίσει στον έλεγχο των σχημάτων οι οποίοι εμφανίζονται στο κέντρο κάθε αντικείμενου.

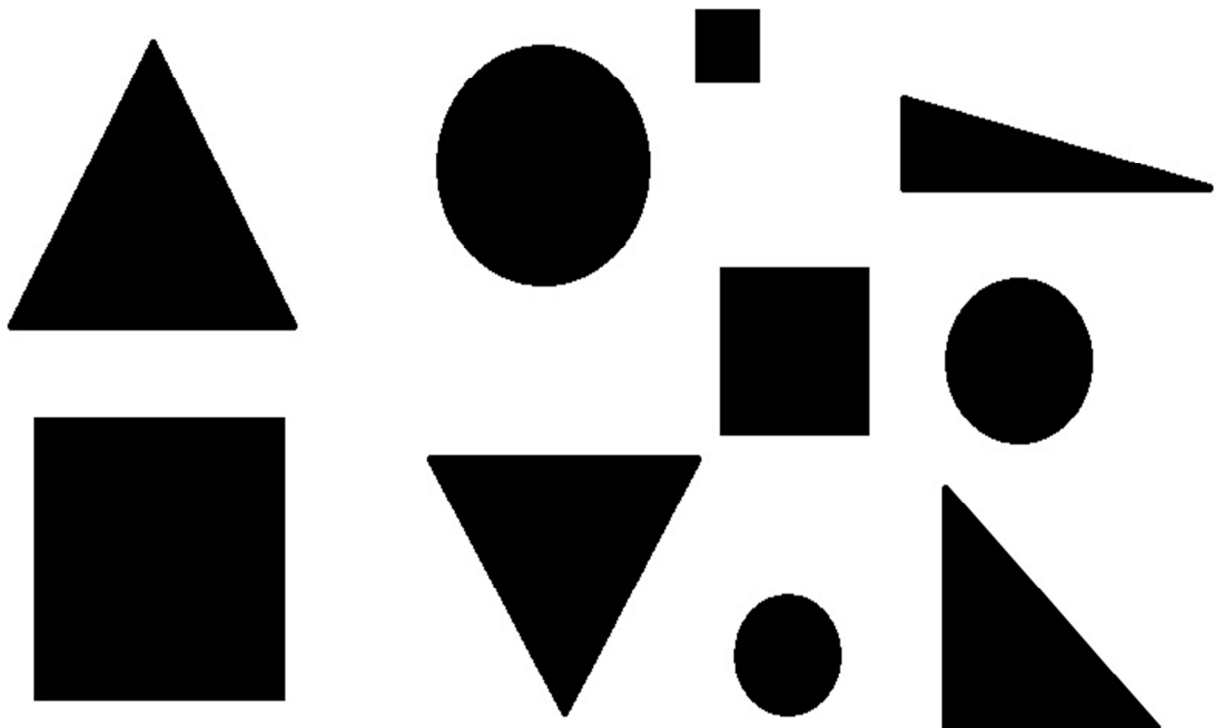
Συμπεράσματα –Αποτελέσματα

Ο συγκεκριμένος αλγόριθμος δουλεύει πολύ καλά, με τα τρία βασικά γεωμετρικά σχήματα, όταν, αυτά είναι σε ανοιχτόχρωμο φόντο, ενώ, τα ίδια είναι σχεδιασμένα χωρίς περιγράμματα, με ένα σκουρόχρωμο χρώμα. Αυτό είναι απαραίτητο γιατί όταν μετατρέπουμε την εικόνα από έγχρωμη σε δυαδική μορφή αν δεν υπάρχει η πλήρης αντίθεση στο φόντο και τα σχήματα μπορεί κάποιο σχήμα να υπολογιστεί σαν μέρος του φόντου. Επίσης τα περιγράμματα επηρεάζουν τη τιμή του extend και δεν αναγνωρίζετε καλά το σχήμα.

Αρχικές εικόνες:



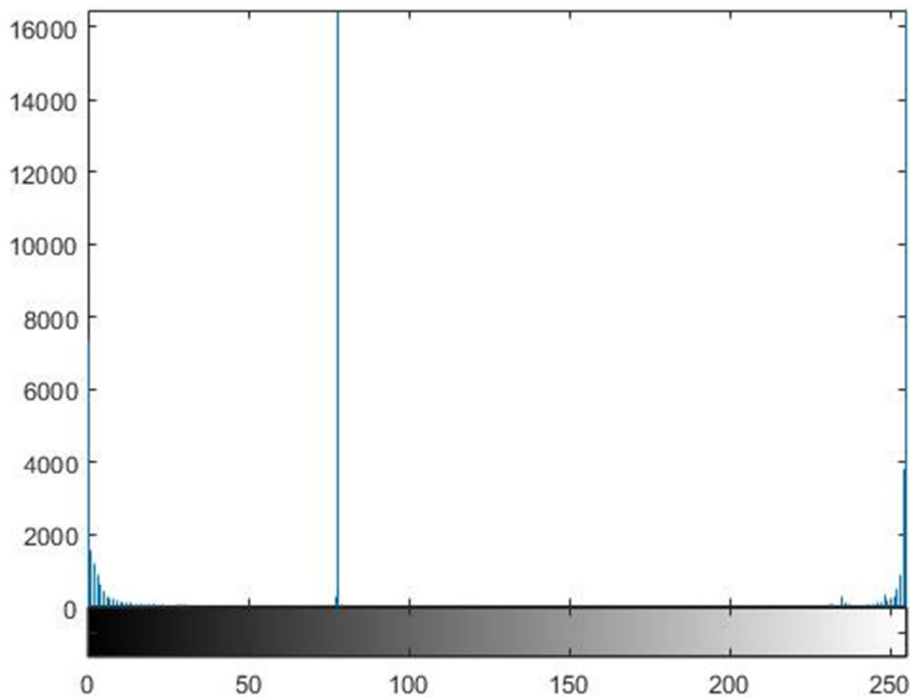
Εικόνα 45: Αρχική εικόνα Test 1^ο



Εικόνα 46 Αρχική εικόνα Test 2^ο

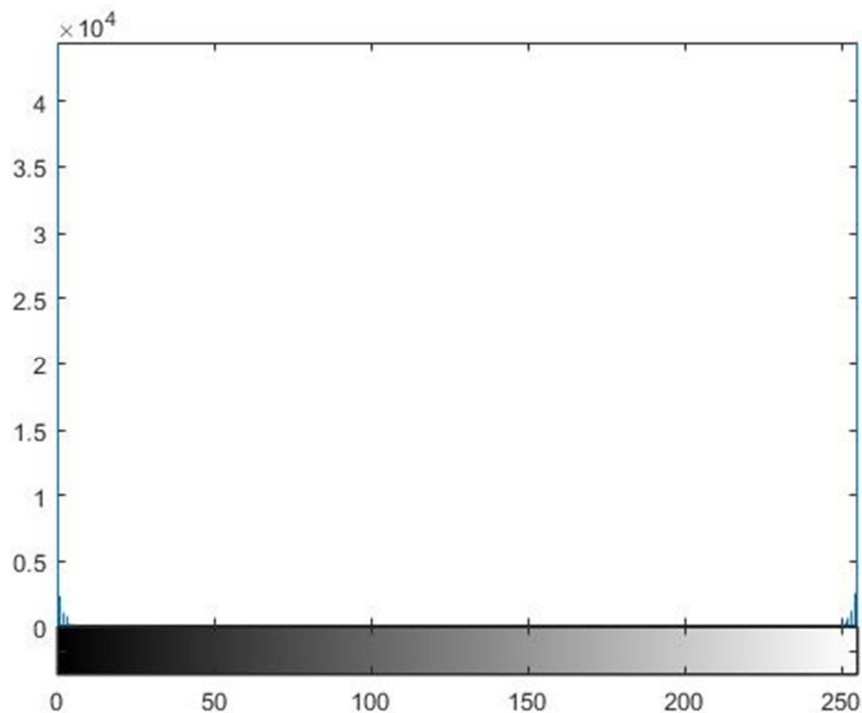
Ιστογράμματα εικόνων:

Το ιστόγραμμα της εικόνας για 1^ο τεστ, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι χαμηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων οπού στη συγκεκριμένη εικόνα είναι χαμηλά.



Εικόνα 47: Ιστόγραμμα εικόνας Test 1^ο

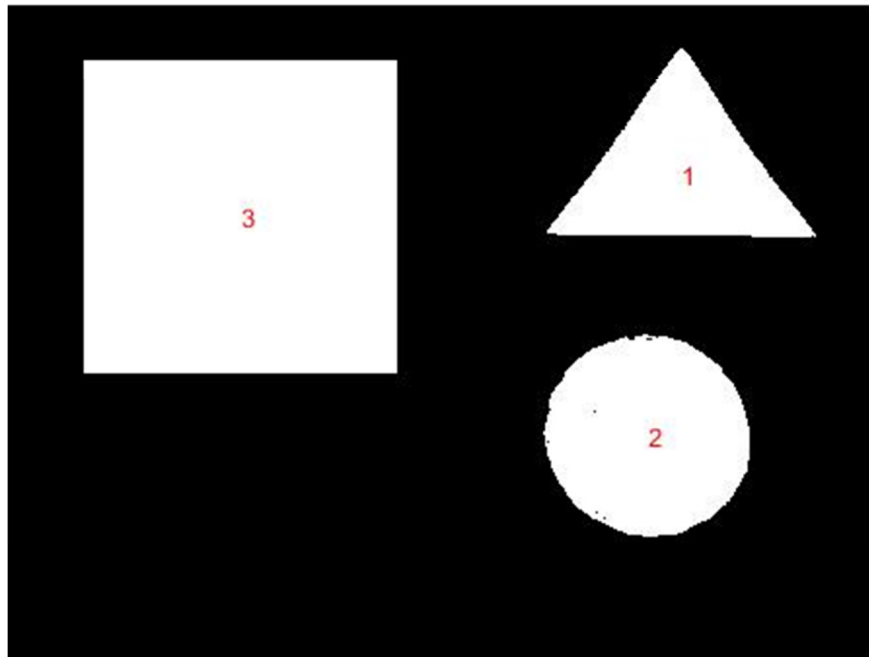
Το ιστόγραμμα της εικόνας για το 2^ο τεστ, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι υψηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων οπού στη συγκεκριμένη εικόνα είναι πάρα πολύ χαμηλά.



Εικόνα 48: Ιστόγραμμα εικόνας Test 2^ο

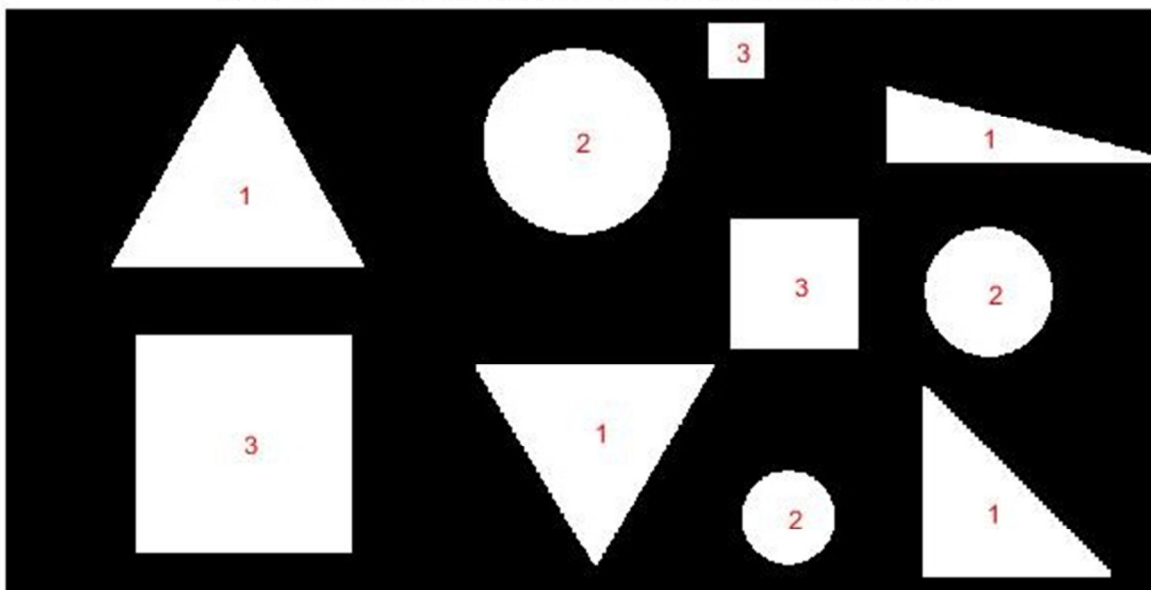
Αποτελέσματα:

RESULTS: 1=TRIAGNLE -- 2=CIRCLE -- 3=SQUARE



Εικόνα 49: Αναγνώριση βασικών γεωμετρικών σχημάτων Test 1^ο

RESULTS: 1=TRIAGNLE -- 2=CIRCLE -- 3=SQUARE



Εικόνα 50: Αναγνώριση βασικών γεωμετρικών σχημάτων Test 2^ο

3.4 Αναγνώριση χρωμάτων

3.4.1 Μεθοδολογία ανάπτυξης αλγόριθμου αναγνώρισης χρωμάτων

Έχοντας στα υπόψη μας ότι το Matlab αναγνωρίζει τα παρακάτω χρώματα στη δυαδική μορφή η οποία φαίνεται στο παρακάτω πίνακα και γνωρίζοντας επίσης ότι πάνω η κάτω από ένα ποσοστό συμμετοχής του RGB φάσματος αντιπροσωπεύεται από 0 η 1 δυαδικά (εάν είναι κάτω από 50 τότε είναι 0 ενώ αν είναι πάνω από 200 είναι 1). Εκμεταλλευόμαστε αυτό το γεγονός για να αναγνωρίσουμε τα χρώματα που αναφέρονται στο πίνακα.

Πίνακας 2: Δυαδική αναπαράσταση Χρωμάτων

Δυαδική αναπαράσταση	Χρώμα
[1 1 0]	Κίτρινο
[1 0 1]	Φούξια
[0 1 1]	Γαλάζιο
[1 0 0]	Κόκκινο
[0 1 0]	Πράσινο
[0 0 1]	Μπλε
[1 1 1]	Άσπρο
[0 0 0]	Μαύρο

Στάδια αλγόριθμου αναγνώρισης χρωμάτων :

1. Εισαγωγή εικόνας
2. Σάρωση εικόνας με σκοπό την αναγνώριση του ποσοστού συμμετοχής του RGB φάσματος
3. Εμφάνιση αποτελεσμάτων

3.4.2 Πρακτική υλοποίηση

```
% Εισαγωγή εικόνας
im=imread('Circles.jpg');
%Διαβάζει το μέγεθος της εικόνας
sz=size(im);
%Δημιουργία μιας λούπας βάση του μεγέθους της εικόνας με σκοπό να αναγνωρίσουμε τα
χρώματα που υπάρχουν σε αυτή καθώς και τη θέση στην οποία βρίσκονται. Αυτό το πετυχαίνουμε
αναγνωρίζοντας τη συμμετοχή του φάσματος RGB σε σχέση με τη δυαδική τιμή που έχει κάθε
χρώμα σύμφωνα με το Matlab.
for i=1:sz(1)
    for j=1:sz(2)
        if im(i,j,1)>200 & im(i,j,2)>200 & im(i,j,3)>200 %[1 1 1]
            white(i,j)=1;
            green(i,j)=0;
            blue(i,j)=0;
            red(i,j)=0;
            black(i,j)=0;
            yellow(i,j)=0;
            magenta(i,j)=0;
            cyan(i,j)=0;
        elseif im(i,j,1)>200 & im(i,j,2)<50 & im(i,j,3)<50 %[1 0 0]
            white(i,j)=0;
            green(i,j)=0;
            blue(i,j)=0;
            red(i,j)=1;
            black(i,j)=0;
            yellow(i,j)=0;
            magenta(i,j)=0;
            cyan(i,j)=0;
        elseif im(i,j,1)<50 & im(i,j,2)>200 & im(i,j,3)<50 %[0 1 0]
            white(i,j)=0;
            green(i,j)=1;
            blue(i,j)=0;
            red(i,j)=0;
            black(i,j)=0;
            yellow(i,j)=0;
            magenta(i,j)=0;
            cyan(i,j)=0;
        elseif im(i,j,1)<50 & im(i,j,2)<50 & im(i,j,3)>200 %[0 0 1]
            white(i,j)=0;
            green(i,j)=0;
            blue(i,j)=1;
            red(i,j)=0;
            black(i,j)=0;
            yellow(i,j)=0;
            magenta(i,j)=0;
            cyan(i,j)=0;
        elseif im(i,j,1)<50 & im(i,j,2)<50 & im(i,j,3)<50 %[0 0 0]
            white(i,j)=0;
            green(i,j)=0;
```



```

blue(i,j)=0;
red(i,j)=0;
black(i,j)=1;
yellow(i,j)=0;
magenta(i,j)=0;
cyan(i,j)=0;
elseif im(i,j,1)>200 & im(i,j,2)>200 & im(i,j,3)<50 %[1 1 0]
white(i,j)=0;
green(i,j)=0;
blue(i,j)=0;
red(i,j)=0;
black(i,j)=0;
yellow(i,j)=1;
magenta(i,j)=0;
cyan(i,j)=0;
elseif im(i,j,1)>200 & im(i,j,2)<50 & im(i,j,3)>200 %[1 0 1]
white(i,j)=0;
green(i,j)=0;
blue(i,j)=0;
red(i,j)=0;
black(i,j)=0;
yellow(i,j)=0;
magenta(i,j)=1;
cyan(i,j)=0;
elseif im(i,j,1)<50 & im(i,j,2)>200 & im(i,j,3)>200 %[0 1 1]
white(i,j)=0;
green(i,j)=0;
blue(i,j)=0;
red(i,j)=0;
black(i,j)=0;
yellow(i,j)=0;
magenta(i,j)=0;
cyan(i,j)=1;
end
end
end
end
%Εμφάνιση αποτελεσμάτων
figure
imshowpair(im,white,'montage')
title('white')
figure
imshowpair(im,green,'montage')
title('green')
figure
imshowpair(im,blue,'montage')
title('blue')
figure
imshowpair(im,red,'montage')
title('red')
figure
imshowpair(im,black,'montage')

```

```

title('black')
figure
imshowpair(im,yellow,'montage')
title('yellow')
figure
imshowpair(im,magenta,'montage')
title('magenta')
figure
imshowpair(im,cyan,'montage')
title('cyan')

```

Λειτουργία κώδικα

Έχοντας στα υπόψη μας ότι το Matlab αναγνωρίζει τα παρακάτω χρώματα στη δυαδική μορφή η οποία φαίνεται στο παρακάτω πίνακα και γνωρίζοντας επίσης ότι πάνω η κάτω από ένα ποσοστό συμμετοχής του RGB φάσματος αντιπροσωπεύεται από 0 η 1 δυαδικά (εάν είναι κάτω από 50 τότε είναι 0 ενώ αν είναι πάνω από 200 είναι 1) τότε φτιάχνουμε μια λούπα η οποία περιέχει ελέγχους αυτού του ποσοστού και ανάλογα με το αποτέλεσμα του τριψηφίου δυαδικού αριθμού αναγνωρίζει και κάνει την αντιστοίχιση στο κάθε χρώμα αποθηκεύοντας τη θέση πάνω στην εικόνα που έχουμε. Για να το κάνουμε αυτό εισάγουμε μια εικόνα και διαβάζουμε το μέγεθος της και ουσιαστικά σύμφωνα με αυτό έχουμε το πλήθος επανάληψης της λούπας και εφαρμόζουμε του ελέγχους οι οποίοι είναι τόσοι όσοι και τα διαφορετικά χρώματα που θέλουμε να αναγνωρίσουμε. Για την εμφάνιση των αποτελεσμάτων δείχνουμε σε ζευγάρι την αρχική εικόνα με την θέση που αποθηκεύσαμε για το κάθε χρώμα.

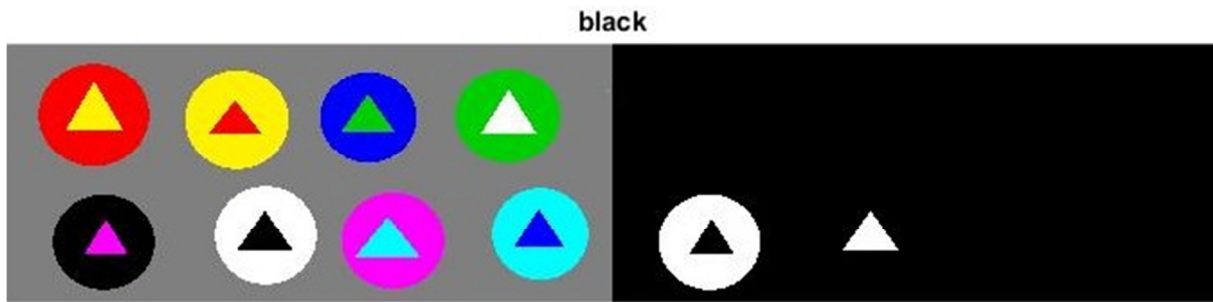
Πίνακας 3: Δυαδική αναπαράσταση Χρωμάτων

Δυαδική αναπαράσταση	Χρώμα
[1 1 0]	Κίτρινο
[1 0 1]	Φούξια
[0 1 1]	Γαλάζιο
[1 0 0]	Κόκκινο
[0 1 0]	Πράσινο
[0 0 1]	Μπλε
[1 1 1]	Άσπρο
[0 0 0]	Μαύρο

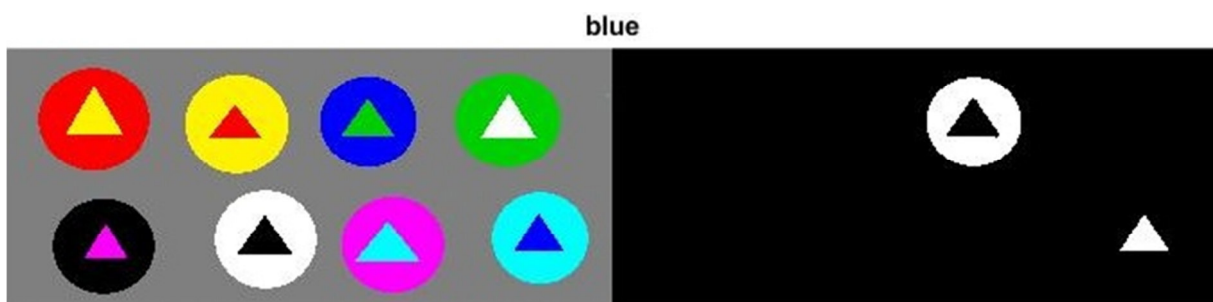
Συμπεράσματα-Αποτελέσματα

Ο συγκεκριμένος αλγόριθμος δουλεύει πολύ καλά για μονόχρωμα σχήματα σε ανοιχτόχρωμο φόντο. Αντιθέτως με το συγκεκριμένο κώδικα δεν μπορούμε να αναγνωρίσουμε αυτά τα χρώματα σε σύνθετες εικόνες.

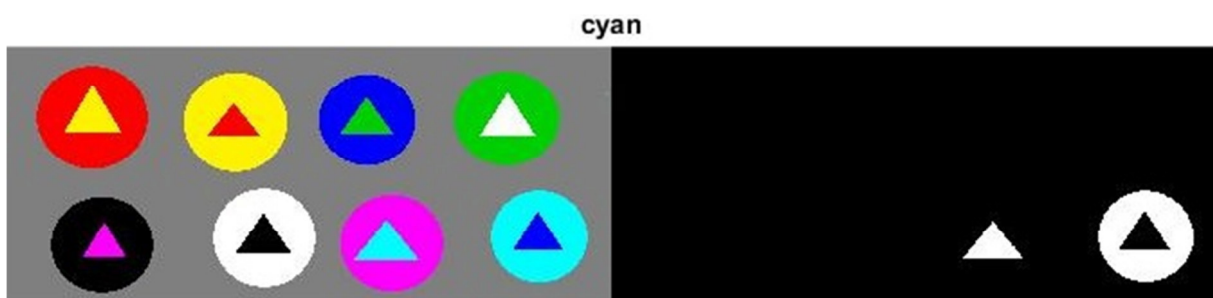
Αποτελέσματα:



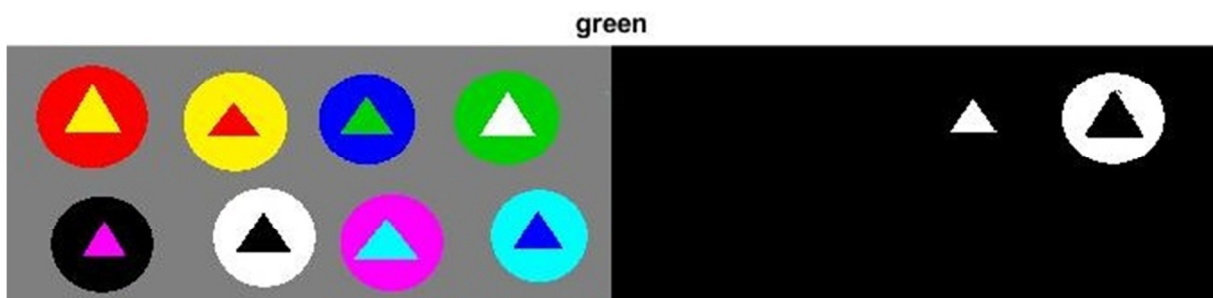
Εικόνα 51: Αναγνώριση χρωμάτων Μαύρο



Εικόνα 52: Αναγνώριση χρωμάτων Μπλε

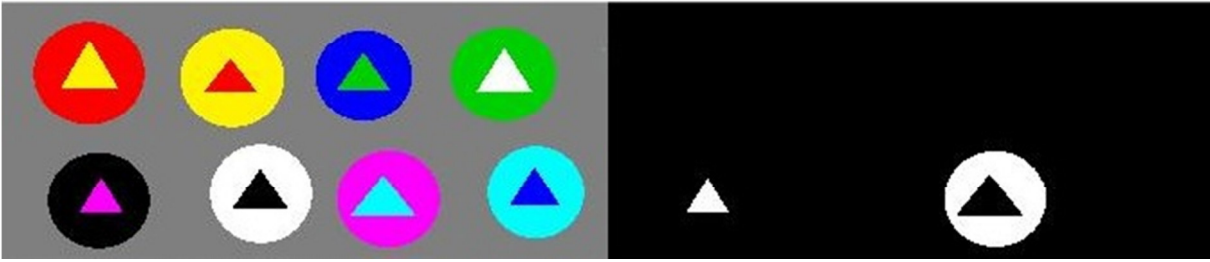


Εικόνα 53: Αναγνώριση χρωμάτων Κυανό



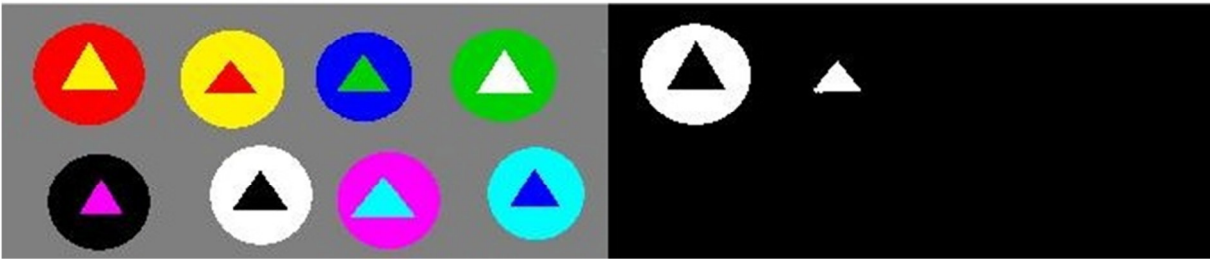
Εικόνα 54: Αναγνώριση χρωμάτων Πράσινο

magenta



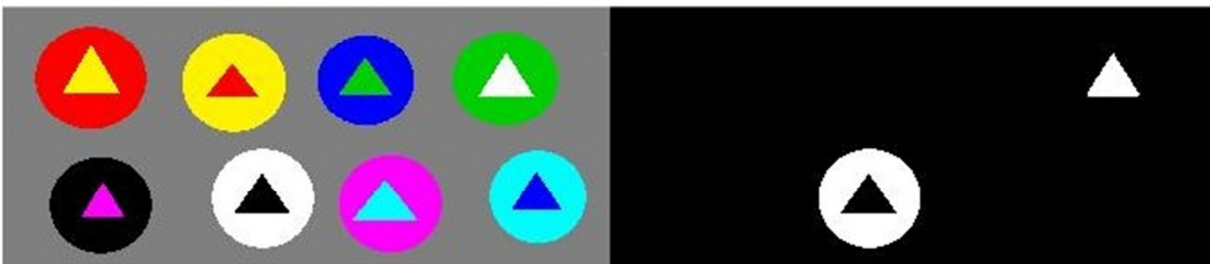
Εικόνα 55: Αναγνώριση χρωμάτων Ματζέντα

red



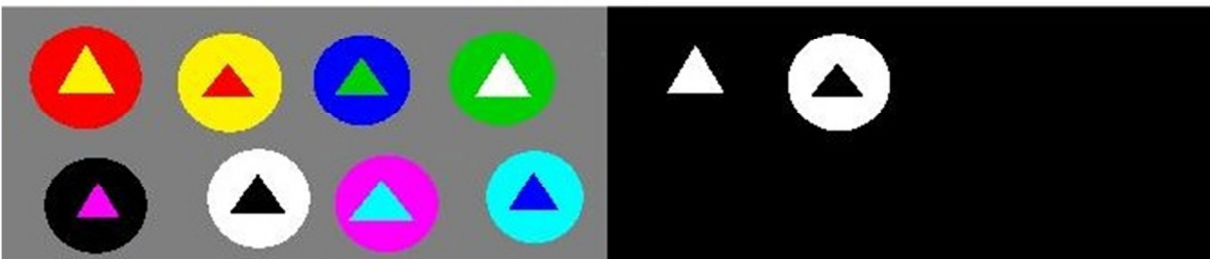
Εικόνα 56: Αναγνώριση χρωμάτων Κόκκινο

white



Εικόνα 57: Αναγνώριση χρωμάτων Άσπρο

yellow



Εικόνα 58: Αναγνώριση χρωμάτων Κίτρινο

3.5 Καταμέτρηση αντικειμένων

3.5.1 Μεθοδολογία ανάπτυξης αλγόριθμου καταμέτρησης αντικειμένων

Βασικός μας σκοπός σε πρώτο επίπεδο είναι να αναγνωρίσουμε τα περιγράμματα των αντικειμένων τις εικόνας το οποίο εάν γίνει σωστά θα έχουμε το αντικείμενο συμπαγές με το περίγραμμα του. Ουσιαστικά το πρώτο στάδιο αναφέρεται στη κατάτμηση εικόνας.

Σε δεύτερο επίπεδο διαβάζουμε τα περιγράμματα που έχουν δημιουργηθεί και τα μετράμε με αυτό το τρόπο έχουμε τη βασική λογική ενός αλγόριθμου καταμέτρησης αντικειμένων.

Όσο πιο σωστά υλοποιηθεί το πρώτο επίπεδο του αλγορίθμου το καλύτερα θα είναι τα αποτελέσματα μας.

Στάδια αλγόριθμου καταμέτρησης αντικειμένων :

1. Εισαγωγή εικόνας
2. Κατάτμηση εικόνας ώστε να βρεθούν τα περιγράμματα
3. Διαβάζουμε τα περιγράμματα με σκοπό να τα διαβάσουμε
4. Εμφάνιση αποτελεσμάτων

3.5.2 Πρακτική υλοποίηση

```
%Εισαγωγή εικόνας
I=imread('Alphabet.jpg');
figure('units','normalized','outerposition',[0 0 1 1])
subplot(2,1,1)
imshow(I)
title('Original Image')

%Μετατροπή εικόνας από έγχρωμη κλίμακα σε κλίμακα του γκρι
I=rgb2gray(I);

%Μετατροπή εικόνας σε δυαδική μορφή
I1=im2bw(I,graythresh(I));
I1=imcomplement(I1);

%Γέμισμα των κενών των αντικειμενων
I1=imfill(I1, 'holes');

%Εύρεση του περιγράμματος της εικονας
B = bwboundaries(I1);
subplot(2,1,2)
imshow(I1)
title('RESULTS')

%Εισαγωγή κειμένου στη εικόνα που εμφανίζει το αποτέλεσμα
text(20,20, strcat('\color{red} Objects Found:', num2str(length(B))), 'FontSize',18, 'FontWeight','bold')
hold on

%Σχεδιασμός του περιγράμματος
for k = 1:length(B)
boundary = B{k};
plot(boundary(:,2), boundary(:,1), 'm', 'LineWidth', 0.2)
end
```

Λειτουργία κώδικα

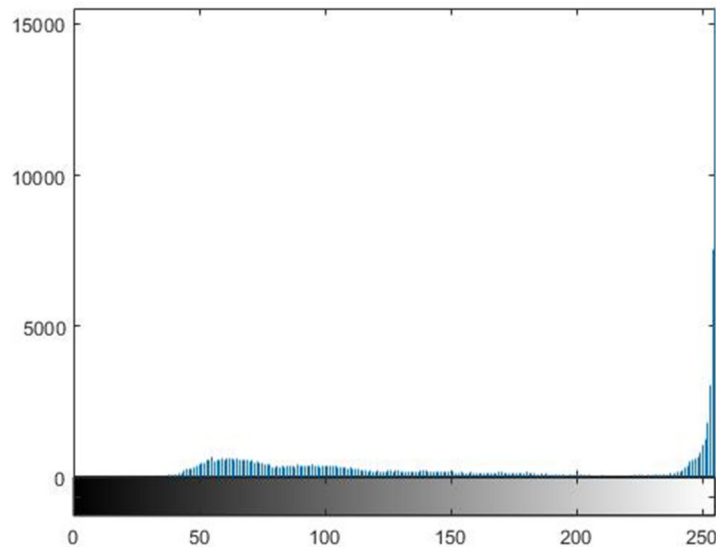
Αφού διαβάσουμε την εικόνα την μετατρέπουμε από έγχρωμη σε κλίμακα του γκρι και μετά σε δυαδική μορφή. Γεμίζουμε τα κενά που υπάρχουν στα αντικείμενα ώστε στο επόμενο βήμα που πρέπει να βρούμε τα περιγράμματα να μην μας βγάλει παραπάνω από αυτά που υπάρχουν και βρίσκουμε τα περιγράμματα τα οποία τα αποθηκεύουμε σε ένα πίνακα και πρακτικά το μέγεθος του πίνακα αυτού είναι το πλήθος των αντικειμένων που ψάχνουμε.

Συμπεράσματα –Αποτελέσματα

Ο αλγόριθμος είναι αρκετά αποτελεσματικός και εύστοχος. Συνάντα δυσκολίες σε εικόνες όπου τα αντικείμενα είναι πιο περίπλοκα με περισσότερες σκιές και λεπτομέρειες. Επίσης σε εικόνες που τα όρια των αντικειμένων δεν είναι ξεκάθαρα και μπλέκονται. Ο συγκεκριμένος αλγόριθμος θα μπορούσε να έχει καλύτερα αποτέλεσμα αν προσθέταμε ένα κομμάτι στο οποίο θα γίνονταν κατάτμηση της εικόνας ώστε τα αντικείμενα να ήταν πιο εμφανή και ξεκάθαρα.

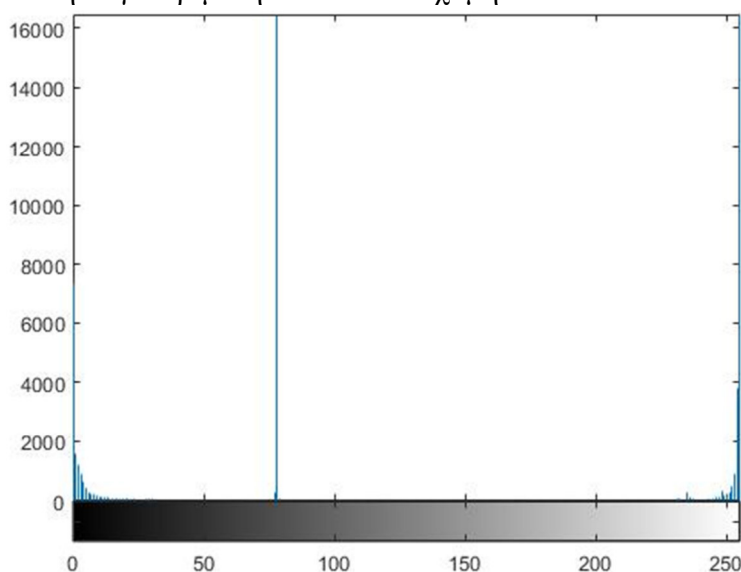
Ιστογράμματα εικόνων:

Το ιστόγραμμα της εικόνας για 1^ο τεστ, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι υψηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων όπου στη συγκεκριμένη εικόνα είναι πάρα πολύ χαμηλά.



Εικόνα 59: Ιστόγραμμα εικόνας Test 1^ο

Το ιστόγραμμα της εικόνας για 2^ο τεστ, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι χαμηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων όπου στη συγκεκριμένη εικόνα είναι χαμηλά.

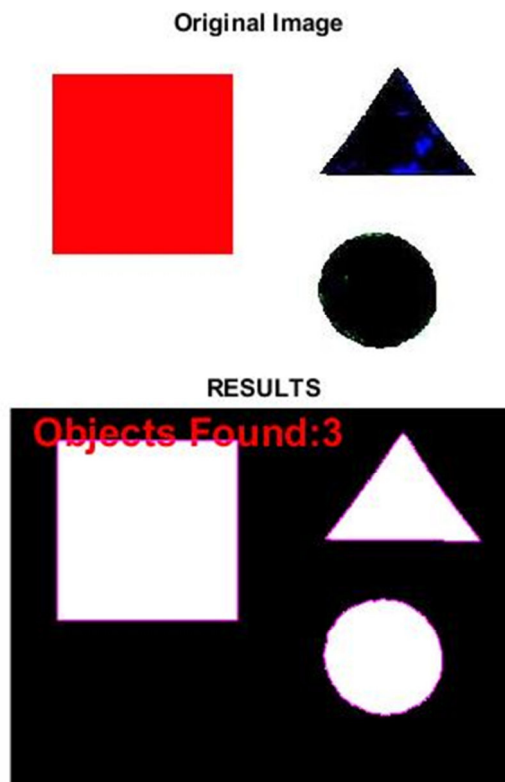


Εικόνα 60: Ιστόγραμμα εικόνας Test 2^ο

Αποτελέσματα:



Εικόνα 61: Καταμέτρηση αντικειμένων Τεστ 1^ο



Εικόνα 62: Καταμέτρηση αντικειμένων Τεστ 2^ο

3.6 Εύρεσης αθροίσματος κερμάτων

3.6.1 Μεθοδολογία ανάπτυξης αλγόριθμου εύρεσης αθροίσματος κερμάτων.

Με αφορμή τον αλγόριθμο εύρεσης της περιμέτρου και του εμβαδού μπορέσαμε να αναπτύξουμε έναν αλγόριθμο οποίος θα μπορεί να υπολογίσει την αξία των κερμάτων σε μια εικόνα έχοντας σαν γνώμονα το εμβαδό του κάθε κέρματος και βάση αυτού να τα αντιστοιχεί στην αγοραστική τους αξία.

Σε πρώτη φάση πήραμε το προηγούμενο κώδικα και υπολογίσαμε το εμβαδό για κάθε κέρμα βλέποντας ταυτόχρονα σε πιο κέρμα αντιστοιχεί. Με αυτό τον τρόπο μπορέσαμε στη συνέχεια να δημιουργήσαμε ελέγχους βασισμένους στα αποτελέσματα αυτά οποίοι θα αντιστοιχούν το εκάστοτε κέρμα στην αγοραστική του αξία.

Βασική προϋπόθεση για να λειτουργήσει ο αλγόριθμος αυτός είναι ότι η λήψη των εικόνων θα πρέπει να γίνεται από σταθερή θέση ώστε να μην επηρεάζονται τα αποτελέσματα για το εμβαδό του κάθε κέρματος.

Τέλος πρέπει να σημειωθεί ότι για να εφαρμοστεί ο συγκεκριμένος αλγόριθμος σε ένα περιβάλλον πραγματικών συνθηκών θα πρέπει να γίνει εκ νέου σετάρισμα των ελεγχων αντιστοίχισης των κερμάτων.

Στάδια αλγόριθμου εύρεσης αθροίσματος κερμάτων:

1. Εισαγωγή εικόνας
2. Μετατροπή εικόνας από έγχρωμη στην κλίμακα γκρι
3. Μετατροπή εικόνας σε δυαδική μορφή
4. Εύρεση πλήθους αντικειμένων
5. Αντιστοίχιση στην αγοραστική αξία βάση του εμβαδού
6. Υπολογισμός αθροίσματος αξίας κερμάτων

3.6.2 Πρακτική υλοποίηση

```
I=imread('eurocoins1.jpg');
I=rgb2gray(I);
%Convert to Binary
I1=im2bw(I,graythresh(I));
B=imcomplement(I1);
%Fill the holes
C=imfill(B,'holes');
%Label the image
[Label,Total]=bwlabel(C,8);
%Object Number
for k = 1:Total
num=k;
[row, col] = find(Label==num);
%Find Area
Obj_area=numel(row);
Embado(k)=Obj_area;
end
for i=1:length(Embado)
if Embado(i)<=3345
    Emabdo_to_coins(i)=0.01;
end
if Embado(i)>3345 && Embado(i)<=4000
    Emabdo_to_coins(i)=0.02;
end
if Embado(i)>4000 && Embado(i)<= 4400
    Emabdo_to_coins(i)=0.10;
end
if Embado(i)>4400 && Embado(i)<= 5000
    Emabdo_to_coins(i)=0.05;
end
if Embado(i)>5000 && Embado(i)<= 6100
    Emabdo_to_coins(i)=0.20;
end
if Embado(i)>6100 && Embado(i)<= 6360;
    Emabdo_to_coins(i)=1;
end
if Embado(i)>6360 && Embado(i)<= 6400;
    Emabdo_to_coins(i)=0.50;
end
if Embado(i)>= 8000;
    Emabdo_to_coins(i)=2;
end
end
A=sum(Emabdo_to_coins);
imshow(I)
title('RESULTS')
text(20,20, strcat('To athroisma to kermatwn isoute me:', num2str(A)), 'FontSize',18,
'FontWeight','bold')
hold on
```

Λειτουργία κώδικα

Έχοντας σαν άξονα το γεγονός ότι τα κέρματα έχουν διαφορετικό μέγεθος μεταξύ τους, ο παραπάνω κώδικας, υπολογίζει το εμβαδό του κάθε κέρματος στην εικόνα και ανάλογα με αυτό τα αποθηκεύει σε ένα πίνακα με την αγοραστική τους αξία (δηλαδή αν είναι μονόλεπτο δίλεπτο και ούτω καθεξής). Έτσι στο τέλος υπολογίζοντας το άθροισμα του πίνακα αυτού, μπορούμε να βρούμε το σύνολο της αξίας των κερμάτων.

Συμπεράσματα –Αποτελέσματα

Ο παραπάνω κώδικας λειτουργεί για κέρματα του ευρώ. Δίνει σωστά και εύστοχα αποτελέσματα με την προϋπόθεση ότι η εικόνα που θα διαβάζει θα είναι καλής ανάλυσης και θα είναι τραβηγμένες από κάμερα σε σταθερή θέση μιας το αν αλλάξει θέση η κάμερα θα πρέπει να ξανά προσδιορίσουμε τα κριτήρια αντιστοίχισης μιας και αν τραβήξουμε τη φωτογραφία από διαφορετική θέση τότε αλλάζει και το εμβαδό. Συνεπώς το κύριο μειονέκτημα του παραπάνω αλγορίθμου είναι η έλλειψη προσαρμοστικότητας σε καινούργιες συνθήκες.

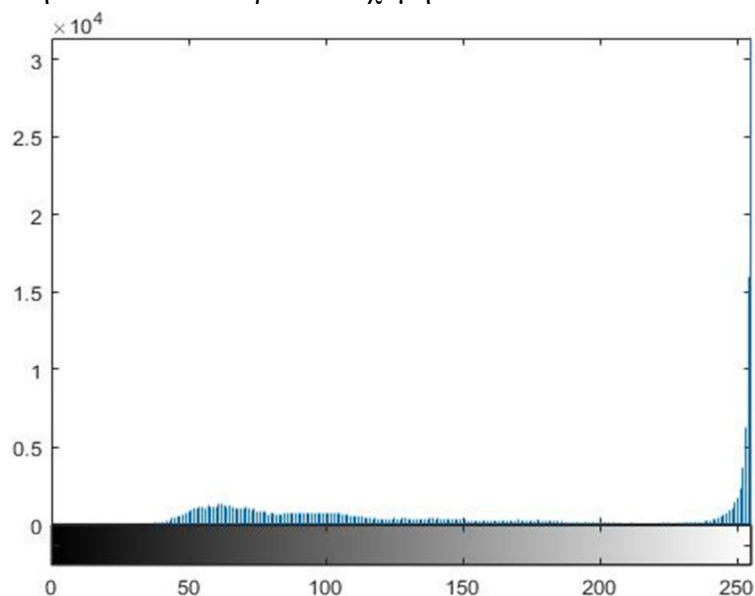
Αρχική εικόνα :



Εικόνα 63: Αρχική εικόνα

Ιστόγραμμα εικόνας:

Το ιστόγραμμα της εικόνας, δείχνει τα επίπεδα φωτεινότητας της εικόνας, στη κλίμακα του γκρι. Για τη δικιά μας εικόνα μπορούμε να παρατηρήσουμε μέσω του ιστογράμματος ότι τα επίπεδα φωτεινότητας είναι υψηλά. Επίσης βλέπουμε τα επίπεδα των αριθμών των σαφών εικονοστοιχείων όπου στη συγκεκριμένη εικόνα είναι πάρα πολύ χαμηλά.



Εικόνα 64: Ιστόγραμμα εικόνας

Αποτελέσματα:

RESULTS

To athroisma to kermatwn isoute me:7.76



Εικόνα 65: Άθροισμα κερμάτων

Κεφάλαιο 4
Ανάπτυξη ευφρών αλγορίθμων για την αναγνώριση
προτύπων

4.1 Ευφυή συστήματα αναγνώρισης προτύπων

4.1.1 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ) (Convolutional Neural Networks (CNN)) μοντελοποιούν την οπτική αντίληψη που έχουν τα ζώα και μπορούν να χρησιμοποιηθούν σαν εργαλείο για οπτική αναγνώριση, μιας και είναι κατάλληλα σχεδιασμένα να δέχονται σαν είσοδο εικόνες. Αποτελούν παρακλάδι των βαθέων νευρωνικών δικτύων (Deep Neural Networks – DNN).

Σκοπός των βαθέων νευρωνικών δικτύων είναι να μοντελοποιήσουν μια υψηλού βαθμού αφαιρετικότητα στα εισαγόμενα δείγματα χρησιμοποιώντας αρχιτεκτονικές βασισμένες σε πολλαπλά επίπεδα μη γραμμικών μετασχηματισμών. Τα βαθιά νευρωνικά δίκτυα είναι μια κατηγορία των αλγορίθμων μηχανικής μάθησης που:

- Χρησιμοποιούν μια αλληλουχία πολλών επιπέδων αποτελούμενων από μη γραμμικούς μετασχηματισμούς για εξαγωγή χαρακτηριστικών.
- Τα χαρακτηριστικά στα βαθύτερα επίπεδα προέρχονται από χαρακτηριστικά χαμηλότερων επιπέδων σχηματίζοντας μια ιεραρχική δομή.
- Μαθαίνουν πολλαπλά επίπεδα αναπαράστασης της εισόδου τα οποία ανταποκρίνονται σε διαφορετικά επίπεδα αφαιρετικότητας. Τα επίπεδα αυτά, σχηματίζουν μια εννοιολογική ιεραρχία.

Τα ΣΝΔ αποτελούνται από τεχνητούς νευρώνες με υπό-μάθηση (learnable) παραμέτρους. Οι πιο γνωστές παράμετροι είναι τα βάρη (weights) και οι πολώσεις (bias). Κάθε νευρώνας δέχεται ως είσοδο ένα διάνυσμα (X), και εξάγει το εσωτερικό γινόμενο των βαρών (W) του με την είσοδο, προστιθέμενης μιας προαιρετικής πόλωσης (b), εφαρμόζοντας προαιρετικά, μια μη-γραμμική συνάρτηση f .

Στα ΣΝΔ, ένα φίλτρο αντιπροσωπεύεται από έναν συνελικτικό πυρήνα ο οποίος είναι κατά πολύ μικρότερος χωρικά από την είσοδο. Κάθε συνελικτικό επίπεδο αποτελείται από πολλά φίλτρα, τα οποία είναι στοιχισμένα με τέτοιο τρόπο ώστε η απόκριση τους στην έξοδο να ανταποκρίνεται στην ίδια περιοχή της εισόδου (η οποία ονομάζεται και δεκτικό πεδίο). Τα φίλτρα σαρώνουν την είσοδο με τέτοιο τρόπο ώστε τα δεκτικά πεδία να αποτελούνται από αλληλεπικαλυπτόμενες περιοχές της εισόδου, προκειμένου να εξαχθεί μια πιο ομαλή αναπαράσταση της εισαγόμενης εικόνας / δείγματος. Αυτή η μέθοδος ονομάζεται διαμοιρασμός βαρών (weight sharing) και χρησιμοποιείται λόγω ότι στις φυσικές εικόνες υπάρχουν πανομοιότυπα χαρακτηριστικά διάσπαρτα στον χώρο. Με αυτή τη μέθοδο το πλήθος των ελεύθερων παραμέτρων προς μάθηση, και κατ' επέκταση η επιρρέπεια στην υπέρ-εκπαίδευση μειώνεται δραστικά, καθώς, τα επίπεδα του δικτύου δεν είναι πλήρως συνδεδεμένα μεταξύ τους αλλά τα φίλτρα συνελίσονται με τα δεκτικά πεδία που σαρώνουν όλες τις περιοχές της εισόδου. Με αυτή τη μέθοδο επίσης, σε επίπεδο υλικού, μειώνονται οι απαιτήσεις μνήμης και βελτιώνεται η συνολική απόδοση.

Αρχιτεκτονική δομή των ΣΝΔ

Ένα ΣΝΔ μπορεί να θεωρηθεί ως μια συνάρτηση μέλος $g_{w,b}$ μιας οικογένειας συναρτήσεων $G: X \rightarrow Y$, παραμετροποιημένης από τις ελεύθερες παραμέτρους της (π.χ. βάρη και πολώσεις). Ως X ορίζεται ο n -οστός διανυσματικός χώρος εισόδου (όπου $n = WxHxD$). Ως Y ορίζεται ο K -οστός διανυσματικός χώρος εξόδου (όπου K το πλήθος των κατηγοριών στην περίπτωση ενός ταξινομητή). Στην γενική περίπτωση, η οικογένεια συναρτήσεων G ορίζεται από την δομή του ΣΝΔ και αποτελείται από την σύνθεση των οικογενειών συναρτήσεων που περιγράφουν τα επίπεδα του ΣΝΔ.

$$\hat{y}_i = G(X_i) = (F_{N-1} \circ FF_{N-2} \dots F_1 \circ F_0)(X_i) \quad (1)$$

Όπου N το πλήθος των επιπέδων του ΣΝΔ. Προκειμένου να δομηθεί ένα ΣΝΔ μπορούν να χρησιμοποιηθούν επτά θεμελιώδεις τύποι επιπέδων:

1. Συνελικτικά επίπεδα / Convolutional layers.

Το συνελικτικό επίπεδο είναι το δομικό στοιχείο των ΣΝΔ. Οι υπό-μάθηση παράμετροι αποτελούνται από ένα πλήθος 3-διάστατων φίλτρων, δημιουργώντας έτσι έναν 4-διάστατο όγκο βαρών, και προαιρετικά, αλλά συστήνεται, κάθε φίλτρο μπορεί να περιέχει μια πόλωση. Η είσοδος ενός τυπικού νευρώνα του συνελικτικού επιπέδου είναι ένας τρισδιάστατο όγκος δεδομένων. Οι διαστάσεις του κάθε νευρώνα διαχωρίζονται σε: ύψος, πλάτος και βάθος. Κατά το βάθος, ο νευρώνας είναι πλήρως συνδεδεμένος με το δεκτικό του πεδίο. Ωστόσο κατά ύψος και πλάτος, όπως έχει ήδη ειπωθεί, το μέγεθος του νευρώνα είναι πολύ μικρότερο από το ύψος και το πλάτος του όγκου της εισόδου, δημιουργώντας έτσι μια χωρική τοπικότητα σύνδεσης με την είσοδο (local connectivity). Κατά την διάρκεια της διάδοσης ο νευρώνας σαρώνει την είσοδο κατά ύψος και πλάτος πραγματοποιώντας τρισδιάστατα εσωτερικά γινόμενα με τα δεδομένα που βρίσκονται κάθε φορά στο δεκτικό του πεδίο και παράγει έναν δισδιάστατο πίνακα, ο οποίος αποτελεί τον χάρτη χαρακτηριστικών της εισόδου. Στην συνέχεια αυτός ο πίνακας τροφοδοτείται στην συνάρτηση ενεργοποίησης και εξάγονται οι ενεργοποιήσεις που αποτελούν έναν μη-γραμμικό μετασχηματισμό του χάρτη χαρακτηριστικών.

2. Συναρτήσεις ενεργοποίησης / Activation functions

Κάθε νευρώνας εμπεριέχει μια συνάρτηση ενεργοποίησης η οποία πραγματοποιεί έναν μη γραμμικό μετασχηματισμό στην έξοδο του. Προκειμένου να εφαρμοστεί η οπισθοδιάδοση των σφαλμάτων, μια μη-γραμμική συνάρτηση πρέπει να είναι διαφορίσιμη και γενικώς είναι μια καλή πρακτική η συνάρτηση ενεργοποίησης να είναι φραγμένη. Μερικές ήδη γνωστές συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στα ΤΝΔ είναι οι σιγμοειδείς, οι συναρτήσεις υπερβολικής εφαπτομένης και οι συναρτήσεις τόξου εφαπτομένης. Ωστόσο, βρέθηκε πειραματικά πως οι συγκεκριμένες συναρτήσεις έχουν μειωμένη απόδοση όταν εφαρμόζονται σε ένα ΣΝΔ συγκρινόμενες με τις μονάδες γραμμικής ανόρθωσης (ReLU).

3. Επίπεδο κανονικοποίησης / Normalization layers.

Το επίπεδο κανονικοποίησης τοπικής απόκρισης είναι εμπνευσμένο από την νευροβιολογία και πραγματοποιεί ενός είδους πλευρική αναστολή. Η πλευρική αναστολή είναι μια ιδιότητα που βρέθηκε πως υφίσταται στους οπτικούς νευρώνες των θηλαστικών. Συγκεκριμένα, ένας διεγερόμενος νευρώνας μπορεί να προκαλέσει μείωση της δραστηριότητας γειτονικών νευρώνων. Ένα ακόμα υποτιμημένο χαρακτηριστικό είναι πως η πλευρική αναστολή μπορεί να συμβεί σε μη γειτονικούς νευρώνες. Στα ΣΝΔ, το επίπεδο κανονικοποίησης τοπικής απόκρισης εφαρμόζεται ως ένα ενδιάμεσο επίπεδο μεταξύ των συνελικτικών επιπέδων και είναι ικανό να πραγματοποιήσει πλευρική αναστολή με δυο διαφορετικούς τρόπους.

1. **Κατά μήκος των βαθμίδων / Across channels:** Το οποίο σημαίνει πως ενεργοποιήσεις ενός φίλτρου οι οποίες διεγείρονται από ένα χαρακτηριστικό, οδηγούν σε διέγερση και ενεργοποιήσεις άλλων φίλτρων.
 2. **Εντός μιας d-οστής βαθμίδας / Within channel** Το οποίο σημαίνει πως ενεργοποιήσεις του ίδιου φίλτρου οδηγούν σε υπέρ/υπό-διέγερση γειτονικών ενεργοποιήσεων του ίδιου φίλτρου.
- 4. Επίπεδα χωρικής υποδειγματοληψίας / Pooling layers.**
- Αυτό το είδος των επιπέδων χρησιμοποιείται για να μειώσει τους όγκους εισόδου εφαρμόζοντας μια συνάρτηση υποδειγματοληψίας κατά ύψος και πλάτος των όγκων εισόδου. Με αυτόν τον τρόπο, επιτυγχάνεται:
- Η μείωση των διαστάσεων των παραγόμενων διανυσματικών χώρων.
 - Η μείωση της πιθανότητας υπερεκπαίδευσης.
 - Επί του πρακτέου μείωση του όγκου της υπό-επεξεργασίας πληροφορίας.
- Η λειτουργία των επιπέδων χωρικής υποδειγματοληψίας βασίζεται στις ίδιες αρχές που διέπουν και τα συνελκτικά επίπεδα. Αυτό σημαίνει πως έχουν υπέρ-παραμέτρους για να δηλώσουν το μέγεθος του πυρήνα τους ($F1 \times F2$), του βήματος σάρωσης (S), και του γεμίσματος με μηδενικά (P). Τα επίπεδα αυτά λειτουργούν ανεξάρτητα σε κάθε βαθμίδα του βάθους της εισόδου και την μειώνουν χωρικά, κάνοντας χρήση μιας συγκεκριμένης συνάρτησης χωρικής υποδειγματοληψίας. Κατά την συνεχή εξέλιξη των ΣΝΔ, πολλές συναρτήσεις έχουν προταθεί για την επίτευξη τη χωρικής υπό-δειγματοληψίας. Μερικές εξ' αυτών είναι η εξαγωγή του τοπικού μεγίστου εκ των υπό- περιοχών (max-pooling), η εξαγωγή του μέσου όρου των υπό-περιοχών (average pooling) καθώς και μια πρόσφατα προτεινόμενη και υποσχόμενη μέθοδος που ονομάζεται στοχαστική χωρική υποδειγματοληψία (Stochastic pooling).
- 5. Πλήρως συνδεδεμένα επίπεδα / Fully connected layers.**
- Τα πλήρως συνδεδεμένα επίπεδα τοποθετούνται στα τελικά επίπεδα των ΣΝΔ. Στην πράξη αυτά τα δίκτυα αποτελούν μια σύνθεση επιπέδων εσωτερικού γινομένου και μιας συνάρτησης ενεργοποίησης. Το μοντέλο απόφασης του ΣΝΔ καθορίζεται στα πλήρως συνδεδεμένα επίπεδα. Κάθε νευρώνας των πλήρως συνδεδεμένων επιπέδων διατηρεί συνδέσεις με όλους τους νευρώνες του προηγούμενου επιπέδου, όπως ισχύει και στα τυπικά τεχνητά νευρωνικά δίκτυα. Λόγω αυτής της συνδεσιμότητας, οι υπό-μάθηση παράμετροι που εμπεριέχονται σε αυτά τα επίπεδα αποτελούν ένα σεβαστό ποσοστό των συνολικών ελεύθερων παραμέτρων των ΣΝΔ.
- 6. Συναρτήσεις σύνδεσης / Link functions**
- Τα επίπεδα εξόδου καθορίζονται από τον τύπο της παλινδρόμησης που θα χρησιμοποιηθεί. Όλα τα μοντέλα παλινδρόμησης είναι βασισμένα σε παραδοχές των σχέσεων μεταξύ των εισόδων και εξόδων τους. Αυτές οι παραδοχές πρέπει να αναγνωριστούν πριν οριστεί ο τύπος της παλινδρόμησης που θα χρησιμοποιηθεί.
- **Γραμμική παλινδρόμηση (linear regression):** Χρησιμοποιείται για να συνδέσει συνεχείς εισόδους X σε συνεχείς εξόδους Y . Η γραμμική παλινδρόμηση βασίζεται στην υπόθεση ότι η κατανομή της δεσμευμένης πιθανότητας $Y|X$ είναι γκαουτσιανή.
 - **Λογιστική παλινδρόμηση (logistic regression1):** Αποτελεί μια ειδική περίπτωση των γενικευμένων γραμμικών μοντέλων και γι' αυτό ανάλογη της γραμμικής παλινδρόμησης. Χρησιμοποιείται για να συνδέσει συνεχείς εισόδους X σε διακριτές εξόδους $Y=\{0,1\}$. Η λογιστική παλινδρόμηση βασίζεται στην υπόθεση πως η κατανομή της δεσμευμένης πιθανότητας $Y|X$ είναι μια κατανομή bernoulli.
 - **Η πολυωνυμική λογιστική παλινδρόμηση (multinomial logistic regression)** είναι μια παραλλαγή της λογιστικής παλινδρόμησης η οποία χρησιμοποιείται για να συνδέσει συνεχείς εισόδους X σε ένα διάνυμα Y από K αμοιβαίως αποκλειόμενες κατηγορίες.

Στα προβλήματα ταξινόμησης, τα επίπεδα εξόδου χρησιμοποιούνται για να εξάγουν την πιθανότητα κατάταξης ενός δείγματος σε μία από τις πιθανές κατηγορίες κάνοντας χρήση της λογιστικής ή της πολυωνυμικής λογιστικής παλινδρόμησης. Το επίπεδο εξόδου αποτελείται από ένα επίπεδο εσωτερικού γινομένου και μια συνάρτηση σύνδεσης (link function) των εξαγόμενων χαρακτηριστικών, που αποτελούν αυθαίρετες και συνεχείς τιμές, σε μια πιθανοτική κατανομή.

7. Συναρτήσεις σφάλματος / Loss functions (οι οποίες οδηγούν την διαδικασία της μάθησης).

Κατά την διαδικασία της εκπαίδευσης, δοθείσας της οικογένειας $G: X \rightarrow Y$, ο αλγόριθμος επιδιώκει την εύρεση μιας συνάρτησης μέλους $g_{w,b}$ ικανής να αντιστοιχίσει τα διανύσματα εισόδου $x_i \in X$ στα διανύσματα αληθείας τους $y_i \in Y$. Προκειμένου να βρεθεί αυτή η ιδανική συνάρτηση μέλος της οικογένειας $G: X \rightarrow Y$, πρέπει να γίνει χρήση μιας συνάρτησης σφάλματος η οποία να καθορίζει το πόσο λανθασμένη είναι η αντιστοίχιση που υλοποιεί η υποψήφια συνάρτηση μέλος. Το σφάλμα της εκτίμησης \hat{y}_i , η οποία αποτελεί την έξοδο του συστήματος, ορίζεται από την συνάρτηση σφάλματος $L(y_i, \hat{y}_i)$. Το σφάλμα υπολογίζεται κατά την διάρκεια της διάδοσης των δειγμάτων $x_i \in X$ δια μέσω του δικτύου, γνωρίζοντας εκ των προτέρων τα διανύσματα αληθείας τους $y_i \in Y$. Το συνολικό σφάλμα (γνωστό και ως εμπειρικό ρίσκο / empirical risk) πάνω σε ένα σετ δεδομένων από N ανεξάρτητα και ομοιόμορφα κατανομημένα δείγματα (i.i.d) υπολογίζεται ως ο αριθμητικός μέσος του σφάλματος όλων των N δειγμάτων όπως φαίνεται στην εξίσωση 2

$$E = \frac{1}{N} \cdot \sum_{n=0}^{N-1} L(y_n, g_{w,b}(X_n)) \quad (2)$$

Η επιλογή της συνάρτησης σφάλματος βασίζεται στον τύπο του προβλήματος που καλείται να επιλύσει το ΣΝΔ.

Κάθε ένα επίπεδο από τα παραπάνω είναι παραμετροποιήσιμο μέσω των υπέρ-παραμέτρων. Δεν υφίσταται συγκεκριμένος τρόπος δόμησης των ΣΝΔ ή επιλογής των υπέρ-παραμέτρων των επιπέδων προκειμένου να δημιουργηθεί ένα ΣΝΔ που να μπορεί να χρησιμοποιηθεί με απόλυτη επιτυχία. Ωστόσο, ισχύουν κάποιες γενικές παραδοχές:

- Τα συνελκτικά επίπεδα τοποθετούνται στην αρχή του ΣΝΔ προκειμένου να επενεργήσουν αφαιρετικά στην πολυπλοκότητα της εισόδου μειώνοντας τον διανυσματικό χώρο της, εκμεταλλευόμενα την χωρική επαναληπτικότητα συγκεκριμένων μοτίβων που εμφανίζονται στις φυσικές εικόνες.
- Η διαδικασία της χωρικής υποδειγματοληψίας εφαρμόζεται μεταξύ δυο διαδοχικών συνελκτικών επιπέδων με σκοπό την περαιτέρω μείωση του διανυσματικού χώρου των χαρακτηριστικών. Ωστόσο, πειραματικά αποδεικνύεται στην βιβλιογραφία πως ο τρόπος που επενεργούν στην μείωση των διαστάσεων δημιουργεί προβλήματα γενίκευσης και η τάση που επικρατεί είναι η μείωση της χρήσης τους ή/και η μείωση του θορύβου που εισάγουν με την χρήση μικρότερων υποδειγματοληπτικών περιοχών.
- Τα πλήρως συνδεδεμένα επίπεδα τοποθετούνται στο τέλος του ΣΝΔ προκειμένου να δημιουργήσουν το μοντέλο απόφασης.

4.1.2 Πρόγραμμα εκπαίδευσης οπτικής αναγνώρισης χαρακτήρων (OCR Trainer)

Τι είναι ?

Είναι μια έτοιμη εφαρμογή στο matlab την οποία μπορείς να χρησιμοποιήσεις για να εκπαιδεύσεις ένα σύστημα να ώστε να εκτελεί οπτική αναγνώριση χαρακτήρων (optical character recognition-OCR).

Πως λειτουργεί ?

Αρχικά η διαδικασία χωρίζεται σε τρία στάδια :

- **Στάδιο 1^ο** : Διαλέγεις τα δεδομένα που θες να δώσεις σαν σημείο αναφοράς ώστε να εκπαιδεύσεις το σύστημα σου. Τα δεδομένα αυτά θα πρέπει να μπορούν να περιγραφούν από ένα χαρακτήρα.
- **Στάδιο 2^ο** : Γίνεται αυτόματα κομματιασμός της εικόνας σε επιμέρους κομμάτια τα οποία ο χρήστης καλείται να ελέγξει για να διαπιστώσει αν έχει γίνει σωστά ο κομματιασμός και να διορθώσει τυχόν λάθη.
- **Στάδιο 3^ο** : Αφού πλέον έχουμε έτοιμα τα δεδομένα μας τα ονοματίζουμε με το χαρακτήρα με τον οποίο θέλουμε να αναγνωρίζονται. Τέλος εκπαιδεύουμε το σύστημα τροφοδοτώντας το με τα ονοματισμένα δεδομένα.

Όταν πλέον έχει εκπαιδευτεί το σύστημα δημιουργούμε μια συνάρτηση με την οποία μπορούμε να κάνουμε οπτική αναγνώριση των χαρακτήρων που θέλουμε.

Γενικά όσο πιο μεγαλύτερο δείγμα δώσουμε για να εκπαιδεύσουμε το σύστημα τόσο πιο σωστά αποτελέσματα θα έχουμε όταν κάνουμε την οπτική αναγνώριση των χαρακτήρων.

4.2 Αναγνώριση χειρόγραφων αριθμών

4.2.1 Μεθοδολογία ανάπτυξης αλγόριθμου αναγνώρισης χειρόγραφων αριθμών

Αρχικά το πρώτο μας μέλημα είναι να δημιουργήσουμε το δείγμα μας, το οποίο αποτελείται από μια πληθώρα αριθμών γραμμένους στο χέρι από το μηδέν έως το εννέα, το οποίο το βγάζουμε φωτογραφίες. Μετά, χρησιμοποιούμε την εφαρμογή του ocr trainer, ώστε να δημιουργήσουμε ένα 'λεξικό', στο οποίο στηρίζεται η συνάρτηση που δημιουργείτε από την εφαρμογή του ocr trainer, κατά την διάρκεια της εκπαίδευσης, την οποία θα χρησιμοποιήσουμε στο κύριο κώδικα για να αναγνωρίζουμε τους χειρογράφους αριθμούς.

Όπως να αναφέρθηκε και στην προηγούμενη ενότητα τα στάδια τα οποία περνάμε κατά τη διάρκεια χρήσης του ocr trainer ώστε να δημιουργηθεί η συνάρτηση είναι :

- **Στάδιο 1^ο** : Διαλέγεις τα δεδομένα που θες να δώσεις σαν σημείο αναφοράς ώστε να εκπαιδεύσεις το σύστημα σου. Τα δεδομένα αυτά θα πρέπει να μπορούν να περιγραφούν από ένα χαρακτήρα.
- **Στάδιο 2^ο** : Γίνεται αυτόματα κομματιασμός της εικόνας σε επιμέρους κομμάτια τα οποία ο χρήστης καλείται να ελέγξει για να διαπιστώσει αν έχει γίνει σωστά ο κομματιασμός και να διορθώσει τυχόν λάθη.
- **Στάδιο 3^ο** : Αφού πλέον έχουμε έτοιμα τα δεδομένα μας τα ονοματίζουμε με το χαρακτήρα με τον οποίο θέλουμε να αναγνωρίζονται. Τέλος εκπαιδεύουμε το σύστημα τροφοδοτώντας το με τα ονοματισμένα δεδομένα.

4.2.2 Πρακτική υλοποίηση

```
function [ocrI, results] = evaluateOCRTraining(I, roi)
% Τοποθεσία του λεξικού
trainedLanguage = 'C:\Program Files\MATLAB\R2016b\bin\myLang\tessdata\myLang.traineddata';
layout = 'Block';
if nargin == 2
results = ocr(I, roi, 'Language', trainedLanguage, 'TextLayout', layout);
else
results = ocr(I, 'Language', trainedLanguage, 'TextLayout', layout);
end
ocrI = insertOCRAnnotation(I, results);

function J = insertOCRAnnotation(I, results)
text = results.Text;
I = im2uint8(I);
if isempty(deblank(text))

text = 'Unable to recognize any text.';
[M,N,~] = size(I);
J = insertText(I, [N/2 M/2], text, 'AnchorPoint', 'Center', 'FontSize', 24, 'Font', 'Arial Unicode MS');
else
location = results.CharacterBoundingBoxes;

newlines = text == char(10);
text(newlines) = [];
location(newlines, :) = [];

spaces = isspace(text);
text(spaces) = [];
location(spaces, :) = [];

text = num2cell(text);

I = padarray(I, [50 50], uint8(255));
location(:,1:2) = location(:,1:2) + 50;

J = insertObjectAnnotation(I, 'rectangle', location, text);
end
---MAIN CODE---
I=imread('Test 1.jpg');
[ocrI, results] = evaluateOCRTraining(I);
imshow(I)

word = results.Words;

wordBBox = results.WordBoundingBoxes;
% Αποτελέσματα
figure;
Iname = insertObjectAnnotation(I, 'rectangle', wordBBox, word);
imshow(Iname);
```

Λειτουργία κώδικα

Ο κύριος κώδικας (main code) χρησιμοποιεί την συνάρτηση (evaluateOCRTraining) για να κάνει οπτική αναγνώριση στους χαρακτήρες που θα του δώσουμε και αποθηκεύει τα αποτελέσματα σε ένα πίνακα.

Συμπεράσματα –Αποτελέσματα

Ο παραπάνω κώδικας λειτουργεί άψογα σε όλα τα τεστ που έγιναν έδωσε σωστά αποτελέσματα. Αν θέλαμε να βελτιώσουμε τον κώδικα, αυτό που θα μπορούσαμε να κάνουμε είναι επεκτείνουμε της το 'λεξικό', δίνοντας μεγαλύτερο δείγμα κατά τη διάρκεια της εκπαίδευσης.



Εικόνα 66: Αρχική εικόνα Τεστ 1°

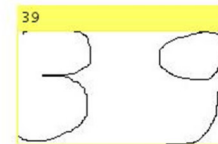
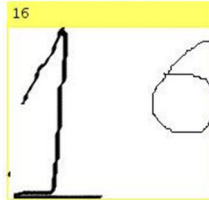


Εικόνα 67: Αρχική εικόνα Τεστ 2°

1 2 3 4 5 6 7 8 9 0

Εικόνα 68: Αρχική εικόνα Τεστ 3°

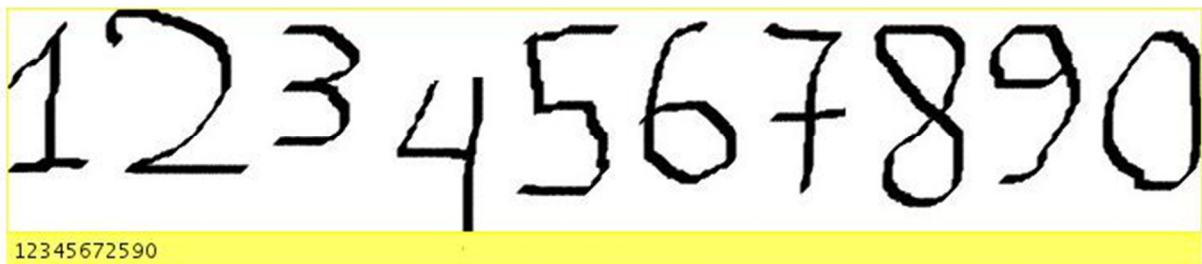
Αποτελέσματα:



Εικόνα 69: Αποτελέσματα Τεστ 1°



Εικόνα 70: Αποτελέσματα Τεστ 2°



Εικόνα 71: Αποτελέσματα Τεστ 3°

4.2.3 Μεθοδολογία ανάπτυξης αλγόριθμου αναγνώρισης χειρόγραφων αριθμών με τη χρήση ΣΝΔ

Ο συγκεκριμένος αλγόριθμος αποτελείται από τρία βασικά στάδια:

- Αρχικά το πρώτο μας μέλημα είναι να δημιουργήσουμε το δείγμα μας, το οποίο αποτελείται από μια πληθώρα αριθμών γραμμένους στο χέρι από το μηδέν έως το εννέα, το οποίο το βγάζουμε φωτογραφίες.
- Σε δεύτερη φάση οργανώνουμε σε φάκελους το δείγμα μας και δημιουργούμε μια βάση εικόνων για το δείγμα μας.
- Τέλος ορίζουμε τη δομή του ΣΝΔ και το εκπαιδεύουμε.

4.2.4 Πρακτική υλοποίηση

```
% Ορισμός μονοπατιού για την θέση που βρίσκονται τα αρχεία
digitDatasetPath = fullfile(matlabroot,'toolbox','nnet','nndemos','nndatasets','DigitDataset');
% Δημιουργία βάσης εικόνων
digitData = imageDatastore(digitDatasetPath,'IncludeSubfolders',true,'LabelSource','foldernames');
CountLabel = digitData.countEachLabel;
% Ορίζουμε με πόσα δείγματα θα εκπαιδύσουμε το δίκτυο μας
trainingNumFiles = 750;
rng(1) % For reproducibility
[trainDigitData,testDigitData] = splitEachLabel(digitData,trainingNumFiles,'randomize');
% Παραμετροποίηση των επιπέδων που θα έχει το δίκτυο
layers = [imageInputLayer([28 28 1])
          convolution2dLayer(5,20)
          reluLayer
          maxPooling2dLayer(2,'Stride',2)
          fullyConnectedLayer(10)
          softmaxLayer
          classificationLayer()];
options = trainingOptions('sgdm','MaxEpochs',15,'InitialLearnRate',0.0001);
% Εκπαίδευση και κατ' επέκταση δημιουργία του δικτύου
convnet = trainNetwork(trainDigitData,layers,options);
```

---MAIN CODE---

```
% Καλούμε το εκπαιδευμένο συνελκτικό δίκτυο
load('convnet.mat')
net = convnet;
% Εμφάνιση της δομή του δικτύου
net.Layers ;
% Εισαγωγή εικόνας
I = rgb2gray(imread('Test5.png'));
% Αναγνώριση εικόνας βάση του δικτύου
label = classify(net, I)
% Αποτελέσματα
figure
imshow(I)
text(1,4,char(label),'Color','red','FontSize',30,'FontWeight','bold')
```

Λειτουργία κώδικα

Ο κύριος κώδικας (main code) χρησιμοποιεί το εκπαιδευμένο συνελκτικό για να κάνει οπτική αναγνώριση στους χαρακτήρες που θα του δώσουμε και αποθηκεύει το αποτέλεσμα σαν ένα τον αριθμό που αναγνώρισε.

Συμπεράσματα –Αποτελέσματα

Ο παραπάνω κώδικας δίνει ακριβή αποτελέσματα με ποσοστό ακριβείας 99.64 %.

Αποτελέσματα:



Εικόνα 72: Αποτελέσματα Τεστ 1^ο



Εικόνα 73: Αποτελέσματα Τεστ 2^ο



Εικόνα 74: Αποτελέσματα Τεστ 3^ο



Εικόνα 75: Αποτελέσματα Τεστ 4^ο



Εικόνα 76: Αποτελέσματα Τεστ 5^ο

Συμπεράσματα – Παρατηρήσεις

Κατά τη διάρκεια τη εκπόνησης της παρούσας πτυχιακής βγάλαμε τα εξής συμπεράσματα :

- ✓ Δεν είναι δυνατή η γενίκευση των αλγορίθμων.
- ✓ Με δεδομένο τα χρώματα που εμπεριέχονται σε κάθε φωτογραφία καθώς και στο κύριο αντικείμενο της, η κατάτμηση, διαφέρει. Επομένως ανάλογα με την τιμή κατωφλίσωσης που θα δώσουμε επιτυγχάνουμε αντίστοιχα και τη σωστή κατάτμηση.
- ✓ Για να συγκρίνουμε την αποτελεσματικότητα μεταξύ αλγορίθμων, τους δίνουμε την ίδια παραμετροποίηση.
- ✓ Τα προβλήματα που μπορεί να συναντήσουμε κατά την εκτέλεση των αλγορίθμων έχουν να κάνουν κυρίως με το αν αντικείμενα δεν έχουν καθοριστεί σωστά ώστε να είναι ξεκάθαρο το πλήθος τους και θέση τους πάνω στην εικόνα.
- ✓ Συνάντανε δυσκολίες σε εικόνες οπού τα αντικείμενα είναι πιο περίπλοκα με περισσότερες σκιές και λεπτομέρειες. Επίσης σε εικόνες που τα όρια των αντικειμένων δεν είναι ξεκάθαρα και μπλέκονται.
- ✓ Κύριο μειονέκτημα τους είναι η έλλειψη προσαρμοστικότητας σε καινούργιες συνθήκες.
- ✓ Τα ΣΝΔ δίνουν καλύτερα και πιο εύστοχα αποτελέσματα από τους απλούς αλγορίθμους.
- ✓ Τα ΣΝΔ μπορούν να “ξεγελαστούν” εξαιρετικά εύκολα με χρήση τεχνητών εικόνων η χρήση τους αντενδείκνυται σε κρίσιμες εφαρμογές.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. J.R. Parker, Algorithms for image Processing and Computer Vision 2nd edition, Wiley Publication Inc, USA, 2011
2. Rafael C. Gonzales, Richard E. Woods, Ψηφιακή Επεξεργασία Εικόνας, Εκδόσεις Τζιόλα, Θεσσαλονίκη, 2014
3. William K. Pratt, Digital Image Processing 3rd Edition, John Wiley & Sons Inc, USA, 2011
4. Rafael C. Gonzales, Richard E. Woods, Steven L. Eddins Digital Image Processing using matlab 2nd Edition, Gatesmark Pubicing, USA, 2009
5. Chris Solomon, Toby Breckon, Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab 1st Edition, Wiley-Blackwell Publication Inc, 2011
6. Maria Petrou, Costas Petrou, Image Processing: The Fundamentals 2nd Edition, Wiley Publication Inc, 2010
7. Oge Marques, Practical Image and Video Processing Using MATLAB 1st Edition Wiley Publication Inc, 2011
8. Anil K. Jain, Fundamentals of Digital Image Processing 1st Edition, Pearson, 1988
9. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010
10. Alasdair McAndrew, An Introduction to Digital Image Processing with Matlab, Victoria University of Technology, 2004
11. Αλέξης Πρόβος, Υλοποίηση βιβλιοθήκης στο MATLAB για την ταχεία έρευνα και προτυποποίηση νέων μεθόδων εκπαίδευσης Βαθέων Συνελικτικών Νευρωνικών Δικτύων με χρήση του CAFFE, Θεσσαλονίκη, 2015
12. <http://www.ijettcs.org/>
13. <https://www.mathworks.com/>