



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
« Προηγμένα Συστήματα Πληροφορικής »

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Δημιουργία συστήματος αλληλεπίδρασης ανθρώπου υπολογιστή για την εκμάθηση του σωστού τρόπου γραφής των αριθμών με αναγνώριση χειρόγραφων χαρακτήρων και μοντελοποίηση χρηστών με την χρήση ασαφούς λογικής. Human Computer Interaction (HCI) system for learning the proper way of writing numbers using handwriting character recognition and user modeling using fuzzy logic.
Όνοματεπώνυμο Φοιτητή	Σταυρινίδης Σταύρος
Πατρώνυμο	Ηρόδοτος
Αριθμός Μητρώου	ΜΠΣΠ/13106
Επιβλέπων	Μ. Βίρβου, Καθηγήτρια

Ημερομηνία Παράδοσης : **Μάρτιος 2015**

Τριμελής εξεταστική επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

Γεώργιος Τσιχριντζής
Καθηγητής

Ευθύμιος Αλέπης
Λέκτορας

Πανεπιστήμιο Πειραιώς

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω ιδιαίτερα την καθηγήτρια και επιβλέπουσα αυτής της μεταπτυχιακής διατριβής κυρία Μαρία Βίρβου, για την καθοδήγηση της και τις συμβουλές της οι οποίες υπήρξαν πολύτιμες για την διεκπεραίωση αυτής της μεταπτυχιακής διατριβής.

Επίσης θα ήθελα να ευχαριστήσω την συμφοιτήτριά μου Γιωτοπούλου Μαρία, με την οποία είχα μια άριστη συνεργασία για την ολοκλήρωση του εκπαιδευτικού λογισμικού για την διδασκαλία των μαθηματικών για παιδιά με αυτισμό, μέρος του οποίου αποτελεί και αυτή η μεταπτυχιακή διατριβή.

Σταυρινίδης Σταύρος

Πανεπιστήμιο Πειραιώς

Πανεπιστήμιο Πειραιώς

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής για την εκμάθηση του σωστού τρόπου γραφής των αριθμών, η οποία με την χρήση κάποιας μηχανής οπτικής αναγνώρισης χαρακτήρων (OCR) θα είναι σε θέση να αναγνωρίζει χειρόγραφους χαρακτήρες (αριθμούς) από φωτογραφίες οι οποίες τραβήχτηκαν με την χρήση κάμερας. Θα χρησιμοποιηθούν τεχνικές ψηφιακή επεξεργασία εικόνας για τον καθαρισμό της εικόνας καθώς και για τον εντοπισμό και την απομόνωση της χρήσιμης πληροφορίας (αριθμός) από την αρχική φωτογραφία. Η εφαρμογή αυτή θα χρησιμοποιηθεί σαν πρόσθετη εφαρμογή ενός διαδραστικού και προσαρμοστικού συστήματος διδασκαλίας μαθηματικών για παιδιά με αυτισμό, βασισμένο στο αναλυτικό πρόγραμμα σπουδών για μαθητές με αυτισμό, του Υπουργείου Εθνικής Παιδείας και Θρησκευμάτων.

Στην πτυχιακή αυτή θα ασχοληθούμε επίσης με την μοντελοποίηση των μαθητών με την χρήση ασαφούς λογική, η οποία ανήκει στον κλάδο της υπολογιστικής νοημοσύνης.

Abstract

The purpose of this thesis is the development of an application for learning the proper way of writing numbers, which will be able to recognize handwritten characters from photographs using an optical character recognition engine (OCR). Digital image processing techniques will be used to image enhancement and to detect and extract the useful information (number) from the original picture. This application will be used as an add-on for an interactive and adaptive tutoring system for teaching math to children with autism, based on the curriculum for students with autism, by the Ministry of National Education and Religious Affairs.

In this thesis we will also deal with student modeling using fuzzy logic, which belongs to the computational intelligence field.

Πανεπιστήμιο Πειραιώς

Περιεχόμενα

1. ΚΕΦΑΛΑΙΟ 1 : ΕΙΣΑΓΩΓΗ	1
1.1. ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΑΝΘΡΩΠΟΥ ΥΠΟΛΟΓΙΣΤΗ	1
1.1.1. Έννοια της Αλληλεπίδρασης Ανθρώπου Υπολογιστή	1
1.1.2. Μονάδες εισόδου υπολογιστή	2
1.1.3. Μονάδες Εξόδου υπολογιστή	3
1.1.4. Οπτική αλληλεπίδραση ανθρώπου-υπολογιστή	3
1.1.5. Σύνοψη	3
1.2. ΥΠΟΛΟΓΙΣΤΙΚΗ ΟΡΑΣΗ	4
1.2.1. Εισαγωγή	4
1.3. ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΚΑΙ MATLAB	5
1.3.1. Στόχοι και εργαλεία	5
1.3.2. Επεξεργασία εικόνας	5
1.3.3. Ψηφιακή Επεξεργασία Εικόνας	5
1.3.4. Μορφολογική επεξεργασία εικόνας και MATLAB	6
1.3.4.1. Διάβρωση (Erosion)	6
1.3.4.2. Διαστολή (Dilation)	7
1.3.4.3. Άνοιγμα (Opening)	9
1.3.4.4. Κλείσιμο (Closing)	10
1.3.4.5. Μετασχηματισμός Top-hat	11
1.3.4.6. Μετασχηματισμός Bottom-hat	12
1.3.4.7. Γέμισμα οπών	14
1.3.4.8. Αφαίρεση αντικειμένων από τα σύνορα	15
1.3.4.9. Αφαίρεση μικρών αντικειμένων	16
1.3.5. Κατάτμηση εικόνας	17
1.3.5.1. Κατωφλίωση	18
1.3.6. Χρήσιμες συναρτήσεις επεξεργασίας εικόνας στο MATLAB	19
1.4. ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ (OCR)	20
1.4.1. Εισαγωγή	20
1.4.2. Είδη εφαρμογών	21
1.4.3. Χρήσεις OCR συστημάτων	21
1.4.4. Η μηχανή οπτικής αναγνώρισης χαρακτήρων Tesseract-OCR	21
1.4.4.1. Εισαγωγή και Ιστορία	21
1.4.4.2. Αρχιτεκτονική	22
2. ΚΕΦΑΛΑΙΟ 2 : ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ	
ΑΝΘΡΩΠΟΥ ΥΠΟΛΟΓΙΣΤΗ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΗΣ “ TESSERACT-OCR ”	23
2.1. ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΙ ΕΡΓΑΛΕΙΑ	23
2.2. ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ	24

2.3. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ (OCR ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ)	24
2.4. ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ΣΤΗΝ VB.NET	25
2.5. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΕΝΤΟΠΙΣΜΟΥ ΚΑΙ ΑΠΟΜΟΝΩΣΗΣ ΤΗΣ ΧΡΗΣΙΜΗΣ ΠΛΗΡΟΦΟΡΙΑΣ (ΑΡΙΘΜΟΥ) ΑΠΟ ΤΗΝ ΑΡΧΙΚΗ ΕΙΚΟΝΑ ΣΤΟ MATLAB	35
2.5.1. Δοκιμές	40
2.6. ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΗΣ TESSERACT-OCR	45
3. ΚΕΦΑΛΑΙΟ 3 : ΑΣΑΦΗΣ ΛΟΓΙΚΗ	47
3.1. ΤΙ ΕΙΝΑΙ ΤΑ ΑΣΑΦΗ ΣΥΣΤΗΜΑΤΑ	47
3.2. ΧΡΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΑΣΑΦΟΥΣ ΛΟΓΙΚΗΣ	48
4. ΚΕΦΑΛΑΙΟ 4 : ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΣΑΦΟΥΣ ΛΟΓΙΚΗΣ ΓΙΑ ΤΗΝ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΩΝ ΜΑΘΗΤΩΝ	49
4.1. ΣΤΟΧΟΣ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΑ	49
4.2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	49
4.3. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΑΣΑΦΟΥΣ ΛΟΓΙΚΗΣ ΣΤΟ MATLAB	50
4.4. ΧΡΗΣΗ ΣΤΗΝ VB.NET ΓΙΑ ΤΗΝ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΩΝ ΜΑΘΗΤΩΝ	53
ΕΠΙΛΟΓΟΣ	54
ΒΙΒΛΙΟΓΡΑΦΙΑ	55

1. Κεφάλαιο 1 : Εισαγωγή

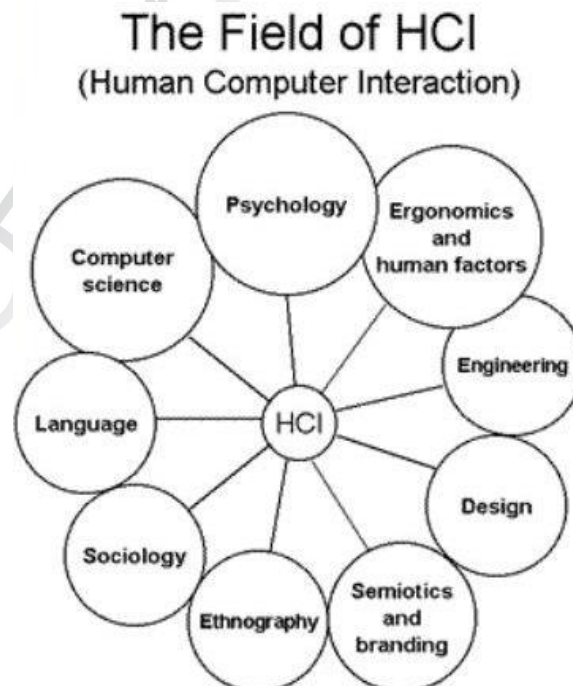
1.1. Αλληλεπίδραση ανθρώπου υπολογιστή

1.1.1. Έννοια της Αλληλεπίδρασης Ανθρώπου Υπολογιστή

Η Αλληλεπίδραση Ανθρώπου-Υπολογιστή που είναι γνωστή και ως Επικοινωνία Ανθρώπου-Υπολογιστή (HCI) συχνά την συναντάμε και ως Επικοινωνία Ανθρώπου-Μηχανής (HMI) είναι η επιστήμη ασχολείται με το σχεδιασμό, την ανάπτυξη και την αξιολόγηση διαδραστικών υπολογιστικών συστημάτων τα οποία προορίζονται για ανθρώπινη χρήση.

Η επιστήμη της αλληλεπίδρασης Ανθρώπου-Υπολογιστή αποτελείται από τρία μέρη: τον άνθρωπο, τον υπολογιστή και τον τρόπο που αλληλεπιδρούν μεταξύ τους. Με τον όρο «άνθρωπος» εννοούμε ή έναν μεμονωμένο χρήστη ή μια ομάδα χρηστών που εργάζονται μαζί και που ο καθένας ασχολείται με κάποιο κομμάτι κάποιας εργασίας. Με τον όρο «υπολογιστής» αναφερόμαστε είτε σε προσωπικούς υπολογιστές είτε σε μεγάλα υπολογιστικά συστήματα, συστήματα ελέγχου, ενσωματωμένα υπολογιστικά συστήματα ή και εφαρμογές διαδικτύου. Τέλος με τον όρο «αλληλεπίδραση» εννοούμε οποιαδήποτε επικοινωνία μεταξύ ανθρώπου και υπολογιστή. Η έννοια της διεπιφάνειας χρήστη (user interface) έχει πολύ μεγάλη σημασία για την επιστήμη της αλληλεπίδρασης ανθρώπου-υπολογιστή.

Η επιστήμη της Αλληλεπίδραση Ανθρώπου-Υπολογιστή διαφέρει από τους άλλους κλάδους της πληροφορικής, καθώς μελετά εκτός των τεχνολογιών και των μεθοδολογιών σχεδίασης και λειτουργίας υπολογιστικών συστημάτων και τις συμπεριφορές των χρηστών οι οποίοι αλληλεπιδρούν με τα υπολογιστικά αυτά συστήματα είτε ως μονάδες είτε ως ομάδες. Λόγω της ιδιαιτερότητας αυτής υπάρχουν πάρα πολλοί επιστημονικοί τομείς από τους οποίους η επιστήμη αυτή αντλεί συμπεράσματα [1].



Εικόνα 1.1

Όπως φαίνεται και στην εικόνα 1.1 Υπάρχουν πολλοί επιστημονικοί τομείς που συνεισφέρουν στην ΕΑΜ, όπως : η πληροφορική, σχετικά θέματα είναι η τεχνολογία λογισμικού για την ανάπτυξη εργαλείων, ανάλυση συστημάτων, γραφικά, γλώσσες προγραμματισμού υψηλού επιπέδου, εικονική πραγματικότητα κ.α., η γνωσιακή ψυχολογία η οποία προσπαθεί να κατανοήσει την ανθρώπινη συμπεριφορά, η κοινωνική και οργανωσιακή ψυχολογία η οποία προσπαθεί να κατανοήσει την ανθρώπινη συμπεριφορά στο κοινωνικό περιβάλλον, εργονομία η οποία σχεδιάζει εργαλεία και όργανα που ταιριάζουν σε διαφορετικές δυνατότητες χρηστών (μείωση της ακτινοβολίας της οθόνης, σωστή στάση, κτλ), η γλωσσολογία η οποία ασχολείται με την κατανόηση της φυσικής γλώσσας, κατανόηση συνομιλιών κτλ., η τεχνητή νοημοσύνη που ασχολείται με την προσομοίωση της έξυπνης ανθρώπινης συμπεριφοράς, σχετικά θέματα είναι η αναγνώριση φωνής, η τεχνητή όραση και η τεχνολογία πρακτόρων, η φιλοσοφία, κοινωνιολογία, ανθρωπολογία που ασχολούνται με θέματα όπως είναι η χρήση μεθόδων για τον σχεδιασμό και την αξιολόγηση συστημάτων, οι τέχνες σχετικά θέματα είναι η σωστή τοποθέτηση αντικειμένων στην οθόνη, χρωματική αρμονία, σωστή τοποθέτηση κειμένων κτλ. [1].

1.1.2. Μονάδες εισόδου υπολογιστή

Οι μονάδες εισόδου έχουν μεγάλη σημασία στην σχεδίαση ενός πληροφοριακού συστήματος, οι μονάδες εισόδου μπορεί να είναι :

1. Τα πληκτρολόγια, όπως τα πληκτρολόγια QWERTY τα οποία είναι τα κλασσικά πληκτρολόγια, όπου η τοποθέτηση των πλήκτρων έγινε με βάση την αντίστοιχη της γραφομηχανής, τα αλφαβητικά που είναι πληκτρολόγια όπου τα γράμματα είναι διατεταγμένα αλφαβητικά, τα πληκτρολόγια DVORAK τα οποία είναι πληκτρολόγια με διαφορετική τοποθέτηση των πλήκτρων για αύξηση ταχύτητας, τα πληκτρολόγια χορδής (Chord) που είναι πληκτρολόγια με διαφορετικό σχεδιασμό απ'τα προηγούμενα. Τα πληκτρολόγια αυτά είναι χρήσιμα κυρίως όπου απαιτείται μονόχειρη χρήση πληκτρολογίου.
2. Οι δείκτες (Pointing Devices) όπως είναι τα ποντίκια (ενός, δύο, ή τριών πλήκτρων), οι οθόνες αφής, τα Joysticks, 3-διάστατοι δείκτες (όπως γάντια δεδομένων, 3-διάστατα ποντίκια, κράνη, ρούχα), κίνηση κάποιου μέλους του χρήστη κτλ.

Εκτός από τα πληκτρολόγια, υπάρχουν και άλλοι τρόποι εισαγωγής κειμένου στον υπολογιστή όπως :

1. Αναγνώριση φωνής, τα υπάρχοντα συστήματα έχουν αρκετά μεγάλη επιτυχία στην αναγνώριση φωνημάτων. Χρησιμοποιείται μόνο σε περιορισμένα περιβάλλοντα. Το κείμενο εισάγεται με χρήση μικροφώνου, κάρτας ήχου και του αντίστοιχου λογισμικού [2][3].
2. Αναγνώριση κειμένου και χειρογραφής, όπως η σάρωση και αναγνώριση των τυπωμένων χαρακτήρων ή περιπτώσεις όπου απαιτείται η αναγνώριση και άλλων χαρακτηριστικών, π.χ., η αναγνώριση υπογραφών απαιτεί και αναγνώριση της κίνησης του υπογράφοντα, την πίεση που εξασκεί στο χαρτί κτλ. [3].

1.1.3. Μονάδες Εξόδου υπολογιστή

Οι μονάδες εξόδου μπορεί να είναι διαφόρων μορφών οθόνες όπως οθόνες καθοδικού σωλήνα (CRT), οθόνες Υγρών Κρυστάλλων (LCD), Οθόνες Πλάσματος (Plasma) καθώς και 3-διάστατες οθόνες, διαφόρων ειδών εκτυπωτές όπως εκτυπωτές ακίδων (dot-matrix), ψεκασμού (inkjet), Θερμικοί (thermal), εκτυπωτές laser κτλ, ή για πιο εξειδικευμένες ανάγκες, οι μονάδες σύνθεσης ομιλίας ή οι ακουστικές οθόνες για τυφλούς χρήστες [3].

1.1.4. Οπτική αλληλεπίδραση ανθρώπου-υπολογιστή

Η οπτική αλληλεπίδραση ανθρώπου-υπολογιστή είναι πιθανώς η πιο διαδεδομένη περιοχή έρευνας στην τεχνολογία της αλληλεπίδρασης ανθρώπου-υπολογιστή (HCI).

Λαμβάνοντας υπόψη την έκταση των εφαρμογών και την ποικιλία των ανοικτών προβλημάτων και προσεγγίσεων, οι ερευνητές προσπάθησαν να 'πιάσουν' τις διαφορετικές πτυχές των ανθρώπινων αντιδράσεων που μπορούν να αναγνωριστούν ως οπτικά σήματα. Κάποιοι από τους κυριότερους ερευνητικούς τομείς αυτής της περιοχής είναι :

- Ανάλυση εκφράσεων προσώπου
- Παρακολούθηση της κίνησης του σώματος
- Αναγνώριση χειρονομιών
- Ανίχνευση βλέμματος
- Οπτική αναγνώριση χαρακτήρων

Παρακάτω παρουσιάζονται συνοπτικά οι βασικοί στόχοι κάθε περιοχής :

Η ανάλυση εκφράσεων προσώπου ασχολείται με την οπτική αναγνώριση συναισθημάτων. Η παρακολούθηση της κίνησης ενός σώματος καθώς και η αναγνώριση χειρονομιών μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, αλλά κυρίως χρησιμοποιούνται για στην άμεση αλληλεπίδραση ανθρώπου-υπολογιστή. Η ανίχνευση βλέμματος είναι συνήθως μια έμμεση μορφή αλληλεπίδρασης ανάμεσα στον χρήστη και τον υπολογιστή η οποία χρησιμοποιείτε συνήθως στην κατανόηση του κατά πόσο ο χρήστης είναι συγκεντρωμένος ή όχι [4]. Η οπτική αναγνώριση χαρακτήρων (OCR) χρησιμοποιείτε κυρίως στην μετατροπή ενός τυπωμένου ή χειρόγραφου κειμένου σε ηλεκτρονική μορφή η οποία μπορεί να χρησιμοποιηθεί σε ένα υπολογιστικό σύστημα [5].

1.1.5. Σύνοψη

Η αλληλεπίδραση ανθρώπου-υπολογιστή αποτελεί ένα σημαντικό κομμάτι στον σχεδιασμό ενός συστήματος. Η ποιότητα ενός συστήματος εξαρτάται από τον τρόπο με τον οποίο παρουσιάζεται και χρησιμοποιείται από τους χρήστες. Ως εκ τούτου, έχει δοθεί μεγάλη προσοχή στην σωστή σχεδίαση της αλληλεπίδρασης ανθρώπου-υπολογιστή. Η νέα κατεύθυνση της έρευνας έχει στραφεί στη αντικατάσταση των παλαιών μεθόδων αλληλεπίδρασης με νέες έξυπνες και προσαρμοστικές μεθόδους [4].

1.2. Υπολογιστική όραση

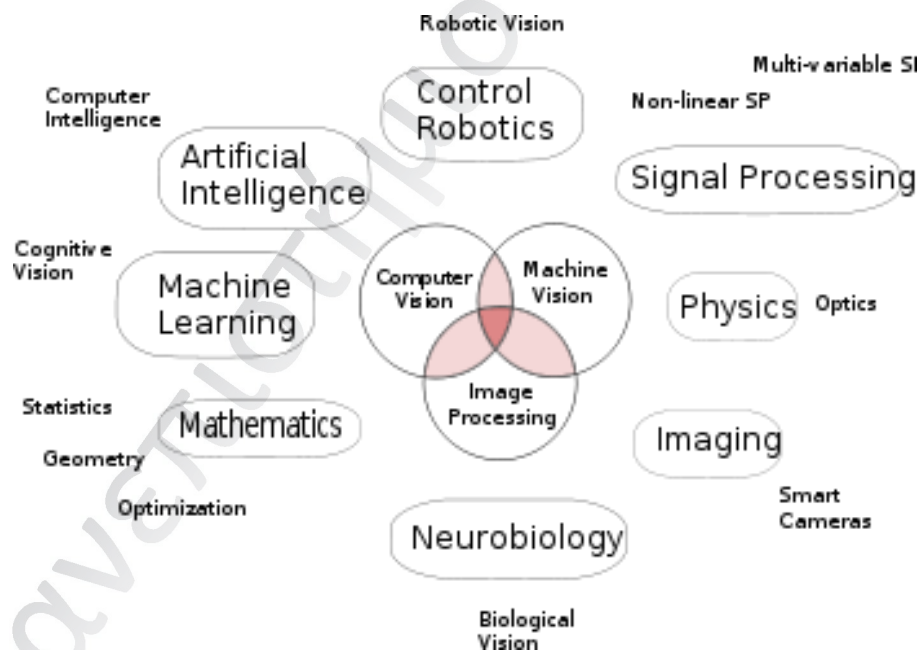
1.2.1. Εισαγωγή

Η υπολογιστική όραση γνωστή και ως τεχνητή όραση, είναι το επιστημονικό πεδίο που περιλαμβάνει μεθόδους για την απόκτηση, την επεξεργασία, την ανάλυση και την κατανόηση των εικόνων του πραγματικού κόσμου. Με άλλα λόγια, υπολογιστική όραση είναι η επιστήμη δημιουργίας συστημάτων ανάλυσης και επεξεργασίας πληροφορίας από οπτικά σήματα.

Η υπολογιστική όραση, ξεκίνησε τη δράση της ως επιστημονικό πεδίο, στη δεκαετία του 70 και από τότε συνεχώς εξελίσσεται. Στην τελευταία δεκαετία, η τεχνητή όραση, μπορούμε να πούμε ότι συνέχισε να εμβαθύνει στην αλληλεπίδραση μεταξύ της όρασης και των πεδίων της γραφικής.

Άλλη μια σημαντική τάση της τελευταίας δεκαετίας είναι η αναγνώριση αντικειμένων η οποία βασίζεται στα χαρακτηριστικά των αντικειμένων (αναγνώριση σκηνής, αναγνώριση τοποθεσίας κτλ.).

Η χρήση της τεχνητής όρασης είναι συχνό φαινόμενο ιδίως σε εφαρμογές υψηλής τεχνολογίας όπως στην βιομηχανία, την ιατρική ακόμα και στην εξερεύνηση του διαστήματος. Έτσι αναπόφευκτα έχει άμεση σχέση με άλλους τομείς των επιστημών όπως η επεξεργασία σημάτων, φυσική, τα μαθηματικά, η τεχνητή νοημοσύνη, η μηχανική μάθηση, η νευρο-βιολογία κτλ.



Εικόνα 1.2

Παρατηρώντας τη σχέση της υπολογιστικής όρασης με άλλους τομείς μπορεί να βρει κανείς, εύκολα, σημαντικές αλληλεξαρτήσεις. Διάφορες περιοχές της τεχνητής νοημοσύνης οι οποίες ασχολούνται με τον αυτόνομο σχεδιασμό και τα ρομποτικά συστήματα, συνδέονται άμεσα με την υπολογιστική όραση η οποία μπορεί να παρέχει σε αυτές ένα συστήματα που ενεργούν ως αισθητήρες, οι οποίοι θα είναι σε θέση να παρέχουν διάφορες πληροφορίες, τόσο για το περιβάλλον όσο και για το ίδιο το ρομποτικό σύστημα. Η υπολογιστική όραση και η

τεχνητή νοημοσύνη, έχουν και άλλα κοινά θέματα όπως η αναγνώριση προτύπων και οι τεχνικές εκμάθησης. Πεδία που συνδέονται πολύ στενά με την υπολογιστική όραση θεωρούνται η επεξεργασία εικόνας, καθώς και η μηχανική όραση [6].

1.3. Ψηφιακή επεξεργασία εικόνας και MATLAB

1.3.1. Στόχοι και εργαλεία

Σε αυτό το κομμάτι της πτυχιακής, θα παρουσιαστούν και θα αναλυθούν οι τεχνικές της ψηφιακή επεξεργασία εικόνας που χρησιμοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας και σκοπό έχουν να καθαρίσουν την εικόνα ώστε να διακρίνετε όσο το δυνατό περισσότερο η χρήσιμη πληροφορία.

Για την επεξεργασία των εικόνων χρησιμοποιήθηκε η γλώσσα προγραμματισμού MATLAB, η εργαλειοθήκη επεξεργασίας εικόνων του οποίου, μας παρέχει μια συλλογή από έτυμες συναρτήσεις επεξεργασίας εικόνων, δημιουργώντας ένα ιδανικό περιβάλλον για την επίλυση προβλημάτων επεξεργασίας εικόνων.

1.3.2. Επεξεργασία εικόνας

Οποιαδήποτε μορφή επεξεργασίας σήματος η οποία δέχεται σαν είσοδο μια εικόνα, ονομάζεται επεξεργασία εικόνας. Η έξοδος της επεξεργασίας αυτής μπορεί να είναι ή μια εικόνα ή ένα σύνολο χαρακτηριστικών τα οποία έχουν κάποια σχέση με αυτή την εικόνα. Οι περισσότερες τεχνικές επεξεργασίας εικόνας χειρίζονται την εικόνα σαν ένα δισδιάστατο ή τρισδιάστατο σήμα.

Ψηφιακή εικόνα είναι η αναπαράσταση μιας δισδιάστατης εικόνας σαν ένα σύνολο πεπερασμένων ψηφιακών τιμών που ονομάζονται pixels [7].

1.3.3. Ψηφιακή Επεξεργασία Εικόνας

Η Ψηφιακή επεξεργασία εικόνας είναι η χρήση αλγορίθμων για την επεξεργασία εικόνας στις ψηφιακές εικόνες. Η ψηφιακή επεξεργασία εικόνας έχει πολλά πλεονεκτήματα σε σχέση με την αναλογική επεξεργασία εικόνας.

Η ψηφιακή επεξεργασία εικόνα επικεντρώνετε στην βελτίωση της ποιότητας μιας εικόνας ώστε να είναι όσο το δυνατόν πιο καθαρή και ευκρινείς, καθώς και στην επεξεργασία των δεδομένων μιας εικόνας, αντιμετωπίζοντας προβλήματα όπως, ψηφιοποίηση και κωδικοποίηση των εικόνων, βελτιστοποίηση και αποκατάσταση, τμηματοποίηση (κατάτμηση) και περιγραφή των εικόνων και ανάλυση, κατανόηση των εικόνων με απώτερο στόχο την προσέγγιση της ανθρώπινης όρασης.

Οι εφαρμογές της Ψηφιακής Επεξεργασίας Εικόνας αυξάνονται όλο και περισσότερο με την πάροδο του χρόνου κάποιες από αυτές είναι : η αποθρομβοποίηση των εικόνων, η κατάτμηση (τεμαχισμός μιας ψηφιακής εικόνας σε πολλά τμήματα) των εικόνων, η ανίχνευση προσώπου, η αναγνώριση χαρακτήρων, η ταυτοποίηση δακτυλικών αποτυπωμάτων κ.τ.λ.

Στα πλαίσια αυτής της πτυχιακής εργασίας θα ασχοληθούμε κυρίως με την μορφολογική επεξεργασία εικόνας και με την κατωφλίωση.

1.3.4. Μορφολογική επεξεργασία εικόνας και MATLAB

Οι μορφολογικοί τελεστές είναι μέθοδοι που χρησιμοποιούνται για την επεξεργασία δυαδικών εικόνων ή εικόνων σε κλίμακα του γκρι και είναι βασισμένοι στη γεωμετρία, οι οποίοι δέχονται σαν είσοδο μία εικόνα, εφαρμόζουν πάνω της κάποιων τελεστή και επιστρέφουν μια εικόνα ίδιων διαστάσεων με την αρχική. Ένα σημαντικής σημασίας στοιχείο των μορφολογικών τελεστών, είναι το δομικό στοιχείο (structure element) το οποίο εφαρμόζεται στην αρχική εικόνα και οι διαστάσεις του είναι συνήθως πολύ μικρότερες από την αρχική εικόνα (πχ 3x3). Οι μορφολογική επεξεργασία επηρεάζει τη μορφή, τη δομή ή το σχήμα ενός αντικειμένου της εικόνας. Βασικοί μορφολογικοί τελεστές είναι η διάβρωση (Erosion), η διαστολή (Dilation), το άνοιγμα (Opening) και το κλείσιμο (Closing).

1.3.4.1. Διάβρωση (Erosion)

Η βασική λειτουργία του τελεστή διάβρωσης (Erosion) είναι η διάβρωση των συνόρων των λευκών pixel (pixels του αντικειμένου της εικόνας), με αποτέλεσμα την σμίκρυνση του αντικειμένου λόγω της αφαίρεσης των pixel από τα σύνορα του [7].

Η εντολή στο MATLAB για τον τελεστή της διάβρωσης είναι η ακόλουθη :

$$IM2 = \text{imerode}(IM, SE);$$

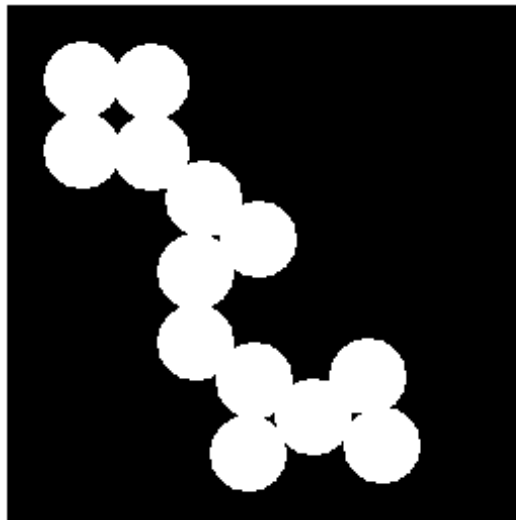
Η παραπάνω εντολή εφαρμόζει τον τελεστή διάβρωσης σε μία εικόνα (IM) η οποία είναι σε κλίμακα του γκρι ή σε μία δυαδική εικόνα και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

Η μεταβλητή SE είναι ένα δομικό στοιχείο (structure element) το οποίο επιστρέφεται από τη συνάρτηση strel (στην οποία θα αναφερθούμε αργότερα).

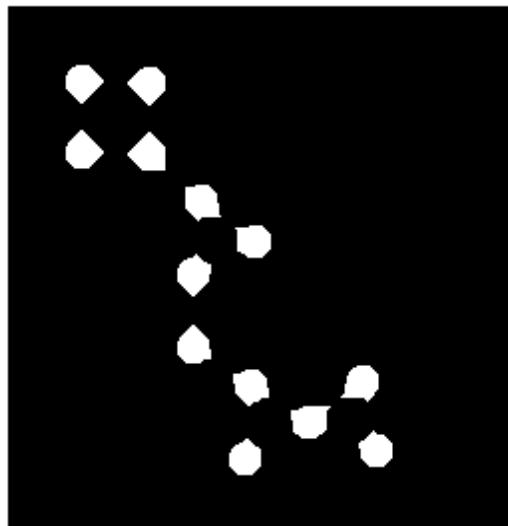
Παράδειγμα :

Διαβρώσει μιας δυαδικής εικόνας με ένα δομικό στοιχείο (structure element) τύπου δίσκο.

```
originalImage = imread('circles.png');
SE = strel('disk',11);
erodedImage = imerode(originalImage,SE);
imshow(originalImage)
figure
imshow(erodedImage) [8]
```



Εικόνα 1.3 : Αρχική



Εικόνα 1.4 : Διάβρωση

1.3.4.2. Διαστολή (Dilation)

Η βασική λειτουργία του τελεστή διαστολής (Dilation) είναι η διαστολή των συνόρων των λευκών ρίξελ (pixels του αντικειμένου της εικόνας), με αποτέλεσμα την μεγέθυνση του αντικειμένου λόγω της πρόσθεσης των pixels στα σύνορα του [7].

Η εντολή στο MATLAB για τον τελεστή της διαστολής είναι η ακόλουθη :

$$IM2 = imdilate(IM, SE);$$

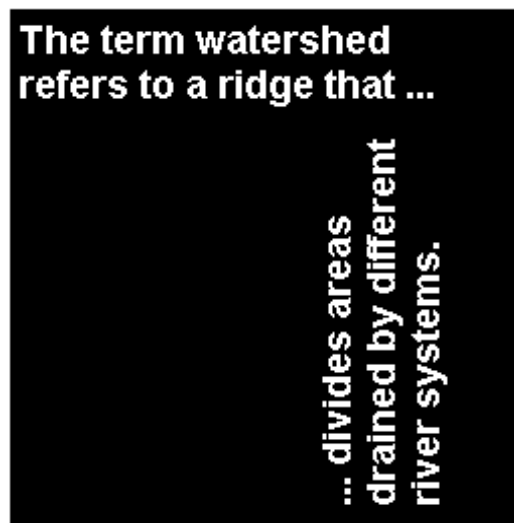
Η παραπάνω εντολή εφαρμόζει τον τελεστή διαστολής σε μία εικόνα (IM) η οποία είναι σε κλίμακα του γκρι ή σε μία δυαδική εικόνα και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

Η μεταβλητή SE είναι ένα δομικό στοιχείο (structure element) το οποίο επιστρέφεται από τη συνάρτηση strel (στην οποία θα αναφερθούμε αργότερα).

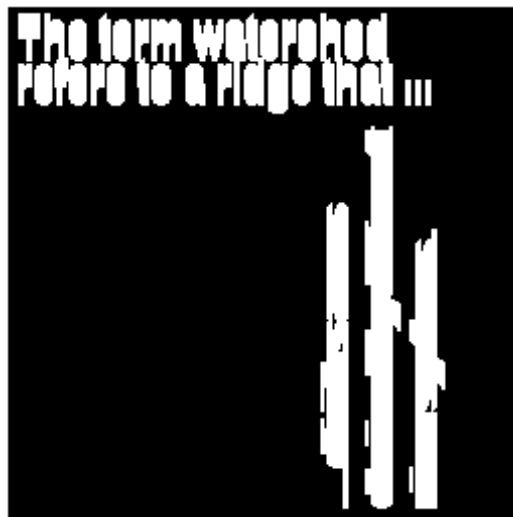
Παράδειγμα :

Εφαρμογή τελεστή διαστολής σε μια δυαδική εικόνα με ένα δομικό στοιχείο (structure element) τύπου κάθετης γραμμής (vertical line).

```
originalImage = imread('text.png');
SE = strel('line',11,90);
dilatedImage = imdilate(originalImage,SE);
imshow(originalImage)
title('Original')
figure
imshow(dilatedImage)
title('Dilated') [8]
```



Εικόνα 1.5 : Αρχική



Εικόνα 1.6 : Διαστολή

1.3.4.3. Άνοιγμα (Opening)

Η βασική λειτουργία που επιτελεί ο τελεστής μορφολογικού ανοίγματος (opening), είναι μία διάβρωση (erosion) η οποία ακολουθείται από μια διαστολή (dilation) [7].

Η εντολή στο MATLAB για τον τελεστή του ανοίγματος είναι η ακόλουθη :

$$IM2 = imopen(IM,SE);$$

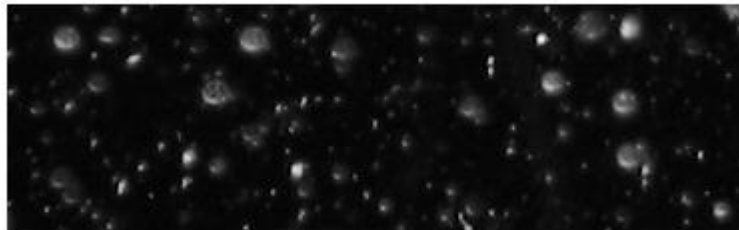
Η παραπάνω εντολή εφαρμόζει τον τελεστή ανοίγματος σε μία εικόνα (IM) η οποία είναι σε κλίμακα του γκρι ή σε μία δυαδική εικόνα και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

Η μεταβλητή SE είναι ένα δομικό στοιχείο (structure element) το οποίο επιστρέφεται από τη συνάρτηση strel (στην οποία θα αναφερθούμε αργότερα).

Παράδειγμα :

Εφαρμογή τελεστή ανοίγματος σε μια εικόνα κλίμακας του γκρι με ένα δομικό στοιχείο (structure element) τύπου δίσκου ακτίνας 5 pixels.

```
I = imread('snowflakes.png');
imshow(I)
se = strel('disk',5);
I_opened = imopen(I,se);
Figure
imshow(I_opened,[]) [8]
```



Εικόνα 1.7 : Αρχική



Εικόνα 1.8 : Άνοιγμα

1.3.4.4. Κλείσιμο (Closing)

Η βασική λειτουργία που επιτελεί ο τελεστής μορφολογικού κλεισίματος (closing), είναι μία διαστολή (dilation) η οποία ακολουθείται από μια διάβρωση (erosion) [7].

Η εντολή στο MATLAB για τον τελεστή του κλεισίματος είναι η ακόλουθη :

```
IM2 = imclose(IM,SE);
```

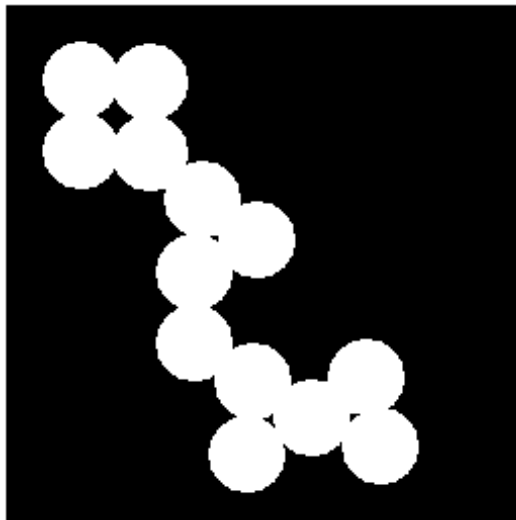
Η παραπάνω εντολή εφαρμόζει τον τελεστή μορφολογικού κλεισίματος σε μία εικόνα (IM) η οποία είναι σε κλίμακα του γκρι ή σε μία δυαδική εικόνα και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

Η μεταβλητή SE είναι ένα δομικό στοιχείο (structure element) το οποίο επιστρέφεται από τη συνάρτηση strel (στην οποία θα αναφερθούμε αργότερα).

Παράδειγμα :

Ο τελεστής κλείσιμο χρησιμοποιείτε για να ενώσει τους κύκλους της εικόνας 1.9 γεμίζοντας τα κενά μεταξύ τους και εξομαλύνοντας της εξωτερικές ακμές τους, χρησιμοποιώντας ένα δομικό στοιχείο (structure element) τύπου δίσκου ακτίνας 10 pixels.

```
originalBW = imread('circles.png');  
imshow(originalBW);  
se = strel('disk',10);  
closeBW = imclose(originalBW,se);  
figure  
imshow(closeBW) [8]
```



Εικόνα 1.9 : Αρχική



Εικόνα 1.10 : Κλείσιμο

1.3.4.5. Μετασχηματισμός Top-hat

Ο μετασχηματισμός Top-hat εφαρμόζει αρχικά τον μορφολογικό τελεστή άνοιγμα σε μια εικόνα και στην συνέχεια αφαιρεί αυτήν την εικόνα από την αρχική.

Η εντολή στο MATLAB για τον μετασχηματισμό Top-hat είναι η ακόλουθη :

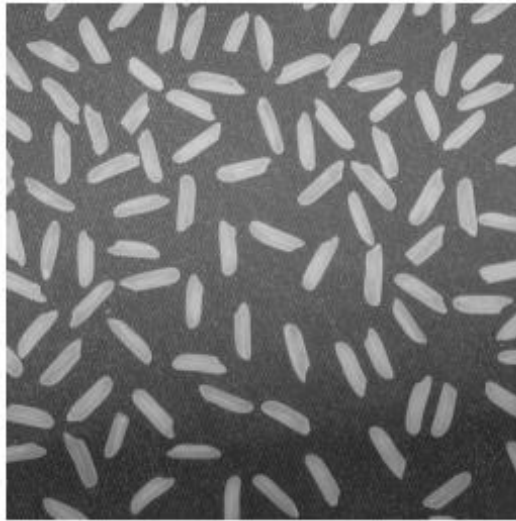
$$IM2 = imtophat(IM,SE);$$

Η παραπάνω εντολή εφαρμόζει τον μετασχηματισμό Top-hat σε μία εικόνα (IM) η οποία είναι σε κλίμακα του γκρι ή σε μία δυαδική εικόνα και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

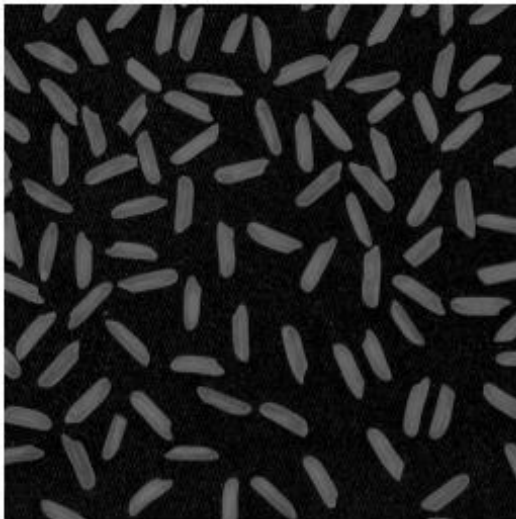
Παράδειγμα :

Ο μετασχηματισμός Top-hat μπορεί να χρησιμοποιηθεί για την διόρθωση του άνισου φωτισμού του φόντου μιας εικόνας, όπως φαίνεται παρακάτω.

```
I = imread('rice.png');
imshow(I)
se = strel('disk',12);
J = imtophat(I,se);
Figure
imshow(J) [8]
```



Εικόνα 1.11 : Αρχική



Εικόνα 1.12 : Top-hat

1.3.4.6. Μετασχηματισμός Bottom-hat

Ο μετασχηματισμός Bottom-hat εφαρμόζει αρχικά τον μορφολογικό τελεστή κλείσιμο σε μια εικόνα και στην συνέχεια αφαιρεί την αρχική εικόνα από αυτήν.

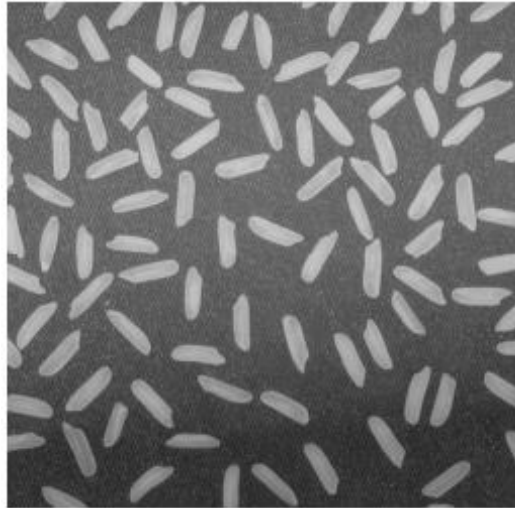
Η εντολή στο MATLAB για τον μετασχηματισμό Bottom-hat είναι η ακόλουθη :

$$IM2 = imbothat(IM,SE);$$

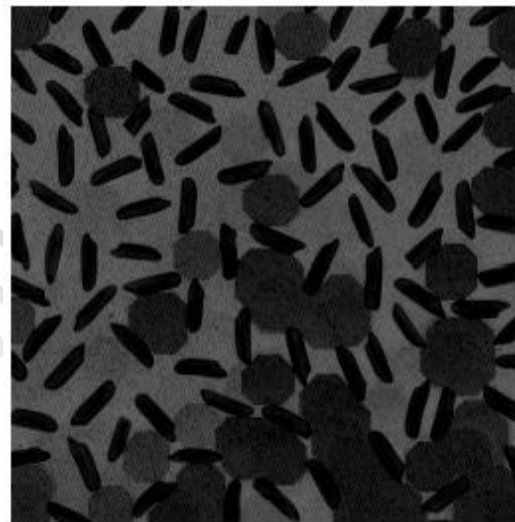
Η παραπάνω εντολή εφαρμόζει τον μετασχηματισμό Bottom-hat σε μία εικόνα (IM) η οποία είναι σε κλίμακα του γκρι ή σε μία δυαδική εικόνα και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

Παράδειγμα :

```
I = imread('rice.png');  
imshow(I)  
se = strel('disk',12);  
J = imbothat(I,se);  
Figure  
imshow(J) [8]
```



Εικόνα 1.13 : Αρχική

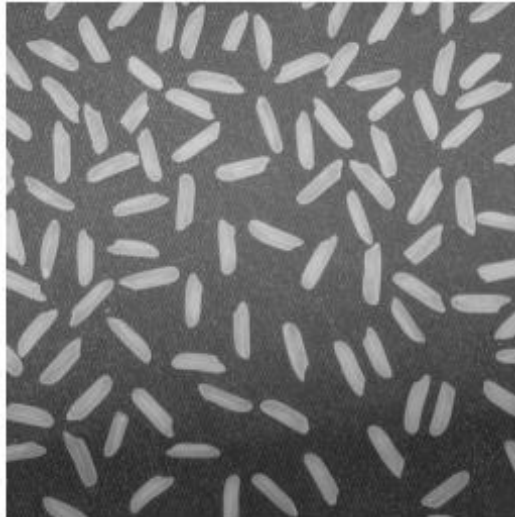


Εικόνα 1.14 : Bottom-hat

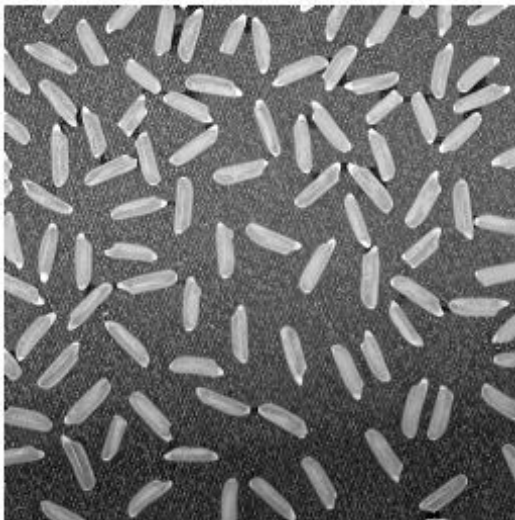
Οι μετασχηματισμοί Top-hat και Bottom-hat μπορούν να συνδυαστούν για να αυξήσουν την αντίθεση (contrast) μίας εικόνας.

```
I = imread('rice.png');  
imshow(I)
```

```
se = strel('disk',3);  
J = imsubtract(imadd(I,imtophat(I,se)), imbothat(I,se));  
Figure  
imshow(J) [8]
```



Εικόνα 1.15 : Αρχική



Εικόνα 1.16 : Top-hat και Bottom-hat

1.3.4.7. Γέμισμα οπών

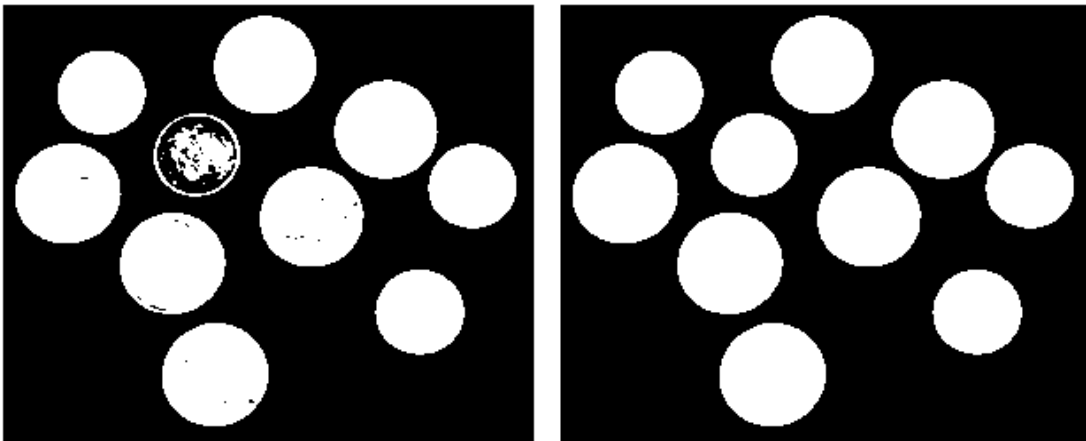
Εντολή στο MATLAB :

```
BW2 = imfill(BW,'holes');
```

Η παραπάνω εντολή γεμίζει τα κενά που υπάρχουν σε μία δυαδική εικόνα (BW) και επιστρέφει την επεξεργασμένη εικόνα στην μεταβλητή IM2.

Παράδειγμα :

```
BW4 = im2bw(imread('coins.png'));
BW5 = imfill(BW4,'holes');
imshow(BW4)
figure
imshow(BW5) [8]
```



Εικόνα 1.17 : Γέμισμα οπών

1.3.4.8. Αφαίρεση αντικειμένων από τα σύνορα

Εντολή στο MATLAB :

```
IM2 = imclearborder(IM);
```

Η παραπάνω εντολή αποκόπτει από την εικόνα (IM) αντικείμενα που είναι στα σύνορα της εικόνας και είναι πιο φωτεινά από το φόντο. Μπορεί να χρησιμοποιηθεί σε εικόνες κλίμακας του γκρι ή δυαδικές εικόνες.

Παράδειγμα :



Εικόνα 1.18 : Αρχική



Εικόνα 1.19 : Αφαίρεση αντικειμένων από τα σύνορα

1.3.4.9. Αφαίρεση μικρών αντικειμένων

Άλλη μία χρήσιμη λειτουργία είναι η αφαίρεση των μικρών αντικειμένων από μια εικόνα, η εντολή στο MATLAB είναι :

$$BW2 = bwareaopen(BW, P);$$

Η παραπάνω εντολή αφαιρεί από μία δυαδική εικόνα όλα τα στοιχεία (αντικείμενα) που συνδέονται μεταξύ τους και έχουν συνολικά λιγότερα από P pixels.

Παράδειγμα :

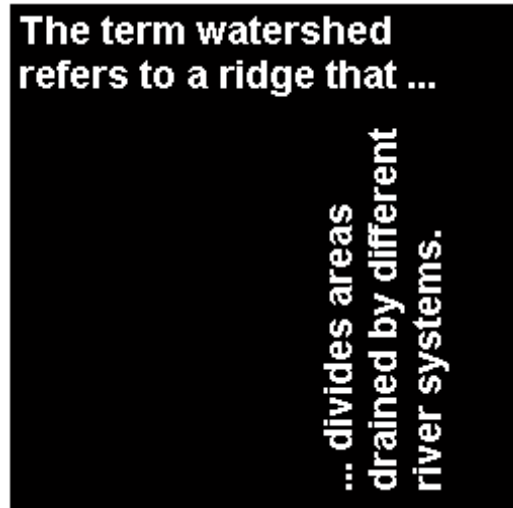
Σε αυτό το παράδειγμα αφαιρούνται όλα τα αντικείμενα από μία εικόνα που περιέχουν λιγότερα από 50 pixels.

$$BW = imread('text.png');$$

```

BW2 = bwareaopen(BW, 50);
imshow(BW);
Figure
imshow(BW2) [8]

```



Εικόνα 1.20 : Αρχική



Εικόνα 1.21 : Αφαίρεση μικρών αντικειμένων

1.3.5. Κατάτμηση εικόνας

Κατάτμηση μιας εικόνας είναι η διαδικασία κατά την οποία η εικόνα χωρίζεται στα μέρη (αντικείμενα) που την αποτελούν και στο φόντο της. Αποτελεί ένα πολύ σημαντικό και δύσκολο κομμάτι στην επεξεργασία εικόνας. Μία μέθοδος κατάτμησης εικόνας είναι η κατωφλίωση [7].

1.3.5.1. Κατωφλίωση

Η κατωφλίωση είναι ένα πολύ χρήσιμο εργαλείο που βοηθάει στην κατάτμηση μιας εικόνας, εφαρμόζεται σε εικόνες κλίμακας του γκριζου και βοηθάει στο να ξεχωρίσουν τα αντικείμενα μιας εικόνας από το φόντο της. Το αποτέλεσμα της κατωφλίωσης είναι μια νέα δυαδική εικόνα (δηλαδή εικόνα που αποτελείτε από 0 και 1), στην οποία το φόντο της αρχικής εικόνας (σε κλίμακα γκριζου) έχει μαυρίσει ενώ τα αντικείμενα της εικόνας εμφανίζονται με λευκό χρώμα ή το αντίθετο.

Εντολή στο MATLAB για απλή κατωφλίωση :

$$BW = \text{im2bw}(I, \text{level});$$

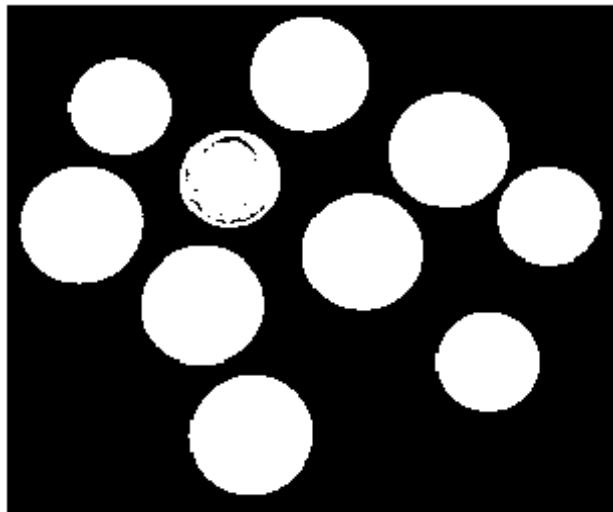
Η παραπάνω εντολή μετατρέπει μια εικόνα κλίμακας γκριζου (I) σε δυαδική εικόνα. Στην εικόνα εξόδου (BW) έχουν αντικατασταθεί όλα τα pixels της εικόνας εισόδου που έχουν φωτεινότητα μικρότερη από «level», με την τιμή 1 (λευκά) και όλα τα υπόλοιπα με την τιμή 0 (μαύρα).

Παράδειγμα :

```
I = imread('coins.png');  
imshow(I)  
I2=im2bw(I,0.4);  
figure  
imshow(I2) [8]
```



Εικόνα 1.23 : Αρχική



Εικόνα 1.24 : Κατωφλίωση

1.3.6. Χρήσιμες συναρτήσεις επεξεργασίας εικόνας στο MATLAB

1) Εισαγωγή εικόνας στο MATLAB :

Η εντολή για την εισαγωγή μιας εικόνας στο MATLAB είναι :

```
imread('filename')
```

Η εντολή αυτή διαβάζει μια εικόνα έγχρωμη ή σε κλίμακα του γκρι. Εάν το αρχείο (filename) δεν είναι στον τρέχοντα φάκελο, πρέπει καθοριστεί το πλήρες όνομα διαδρομής.

2) Εμφάνιση της εικόνας :

Η εντολή που χρησιμοποιείτε για την εμφάνιση μιας εικόνας είναι :

```
imshow(filename)
```

3) Μετατροπή RGB εικόνας σε εικόνα αποχρώσεων του γκρι :

```
rgb2gray(rgb_image)
```

Η εντολή rgb2gray μετατρέπει μία RGB εικόνα σε κλίμακα του γκριζου, διατηρώντας τη φωτεινότητά της.

4) Αποθήκευση εικόνας :

Η εντολή για την αποθήκευση μίας εικόνας σε αρχείο είναι :

```
imwrite (A, όνομα_αρχείου, μορφή)
```

Η εντολή `imwrite` (A, όνομα αρχείου, FMT) γράφει την εικόνα 'A' στο αρχείο που ορίστηκε από το 'όνομα_αρχείου' σε μορφή που καθορίζεται από 'μορφή'.

5) Αλλαγή μεγέθους εικόνας :

Η εντολή του MATLAB για την αλλαγή του μεγέθους μίας εικόνας είναι :

```
imresize(image, [rows cols])
```

Η εντολή `imresize` επιστρέφει μία εικόνα που έχει τον αριθμό των γραμμών και των στηλών που καθορίζονται από [rows cols].

6) Εντοπισμός ορίων μιας περιοχής σε δυαδική εικόνα :

```
bwboundaries(BW_image)
```

Η εντολή `bwboundaries` επιστρέφει τη σειρά και τη στήλη των συντεταγμένες των συνοριακών pixels όλων των αντικειμένων σε μια εικόνα.

7) Δημιουργία μορφολογικού δομικού στοιχείου :

```
str_e = strel(shape, param)
```

Η παραπάνω εντολή δημιουργεί ένα δομικό στοιχείο, `str_e`, του τύπου που καθορίζεται από το 'shape'.

8) Ιδιότητες των περιοχών των εικόνων :

```
regionprops(BW_image, properties)
```

Η παραπάνω εντολή μετράει ένα σύνολο ιδιοτήτων για κάθε αντικείμενο της δυαδικής εικόνας `BW_image`. [8]

1.4. Οπτική αναγνώριση χαρακτήρων (OCR)

1.4.1. Εισαγωγή

Η Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition - OCR) είναι η ηλεκτρονική ή μηχανική μετατροπή των σαρωμένων εικόνων χειρόγραφου ή τυποποιημένου κειμένου σε κωδικοποιημένη μορφή χαρακτήρων κειμένου ηλεκτρονικού υπολογιστή, δηλαδή αποτελεί μια προσπάθεια προσομοίωσης της ανθρώπινης όρασης από τον υπολογιστή. Σκοπός έχει να εντοπίσει και να αναγνωρίσει όλα τα αλφαριθμητικά σύμβολα μέσα σε μία εικόνα.

Χρησιμοποιείται για την εξαγωγή δεδομένων από κάποιες μορφές έντυπου, όπως είναι τα έγγραφα, τα βιβλία, τα γράμματα αλληλογραφίας, οι αποδείξεις πωλήσεων κτλ. Δηλαδή είναι μέθοδος με την οποία ψηφιοποιούνται έντυπα κειμένων έτσι ώστε να είναι προσπελάσιμα ηλεκτρονικά, να μπορούν αποθηκεύονται σε μικρότερο χώρο, να είναι προσβάσιμα από τον παγκόσμιο ιστό και να μπορούν να χρησιμοποιηθούν σε διάφορες διαδικασίες όπως η

αυτόματη μετάφραση, η εξόρυξη δεδομένων ή η μετατροπή κειμένου σε φωνή από τον ηλεκτρονικό υπολογιστή.

Η οπτική αναγνώριση χαρακτήρων αποτελεί πεδίο έρευνας για διάφορους τομείς όπως η αναγνώριση προτύπων, της κατανόησης των εικόνων από τον ηλεκτρονικό υπολογιστή, της τεχνητής νοημοσύνης κτλ.

Οι πρώτες εφαρμογές της οπτικής αναγνώρισης χαρακτήρων βασίζονταν στην επεξεργασία των γραμμάτων από μια φωτογραφία, τα οποία και έψαχναν μεμονωμένα μέσα στα προς έλεγχο κείμενα. Με τον καιρό δημιουργήθηκαν πιο εξελιγμένα συστήματα τα οποία εξετάζουν τις λέξεις σαν λαμβάνοντας υπ' όψιν τους και το νόημα των λέξεων. Έχουν πραγματοποιηθεί πολλές εργασίες και προγράμματα οπτικής αναγνώρισης χαρακτήρα όλα αυτά τα χρόνια, το πεδίο έρευνας είναι ακόμα ανοιχτό. [9][10]

1.4.2. Είδη εφαρμογών

Η τεχνολογία οπτικής αναγνώρισης χαρακτήρων μπορεί να ενσωματωθεί σε διάφορες εφαρμογές όπως σε desktop εφαρμογές, σε online εφαρμογές (παγκόσμιος ιστός, διαδικτυακό νέφος (cloud computing) , φορητές συσκευές (mobile devices)) η διαδικτυακή οπτική αναγνώριση χαρακτήρων έχει γίνει η νέα τάση του χώρου από το 2000 και μετά, εφαρμογές προσαρμοσμένες στην περίπτωση (invoice OCR, business-card OCR, ID card OCR, screenshot OCR, driver license OCR κτλ.). [9]

1.4.3. Χρήσεις OCR συστημάτων

Τα συστήματα οπτικής αναγνώρισης χαρακτήρων έχουν μια πληθώρα εφαρμογών κάποιες από αυτές της εφαρμογές είναι : βοηθήματα ανάγνωσης για τυφλούς, αυτόματη αναγνώριση διευθύνσεων για χρήση σε ταχυδρομεία, αναγνώριση πινακίδων αυτοκινήτων, άμεση επεξεργασία χειρόγραφων κειμένων, αναγνώριση υπογραφής κτλ [10].

1.4.4. Η μηχανή οπτικής αναγνώρισης χαρακτήρων Tesseract-OCR

1.4.4.1. Εισαγωγή και Ιστορία

Το Tesseract είναι μία μηχανή οπτικής αναγνώρισης χαρακτήρων (OCR) ανοιχτού κώδικα που αναπτύχθηκε από την HP μεταξύ 1984 και 1994. Το Tesseract εμφανίστηκε από το πουθενά στην UNLV του 1995 στον ετήσιο έλεγχο ακρίβειας OCR και έλαμψε με τα αποτελέσματά της και στη συνέχεια εξαφανίστηκε κάτω από την ίδια μυστικοπάθεια με την οποία είχε αναπτυχθεί. Το Tesseract ξεκίνησε ως ένα διδακτορικό ερευνητικό έργο στο εργαστήριο της HP στο Μπρίστολ, και προωθήθηκε ως πρόσθετο λογισμικού ή υλικού για τη γραμμή επίπεδων σαρωτών της HP.

Μετά από ένα κοινό πρόγραμμα μεταξύ των εργαστηρίων της HP στο Μπρίστολ, και του τμήματος σάρωσης της HP στο Κολοράντο, η Tesseract απέκτησε σημαντικό προβάδισμα στην ακρίβεια της έναντι των άλλων εμπορικών μηχανών OCR, αλλά δεν έγινε προϊόν. Το επόμενο στάδιο της ανάπτυξης της ήταν πίσω στο εργαστήριο της HP στο Bristol ως έρευνα του OCR στη συμπύεση. Στο τέλος αυτού του έργου, στα τέλη του 1994, η ανάπτυξη διακόπηκε εντελώς. Ο μηχανή στάλθηκε στο UNLV για τον ετήσιο έλεγχο ακρίβειας OCR μηχανών του 1995, όπου απέδειξε την αξία της έναντι των άλλων εμπορικών μηχανών της εποχής.

Στα τέλη του 2005, η HP κυκλοφόρησε την Tesseract και είναι πλέον διαθέσιμη δωρεάν στο <http://code.google.com/p/tesseract-ocr> [11].

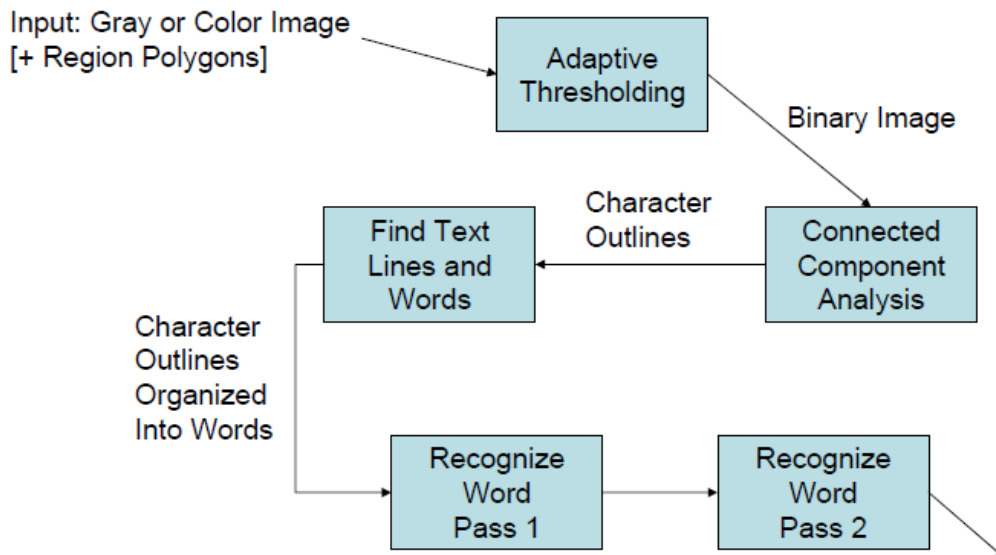
1.4.4.2. Αρχιτεκτονική

Η επεξεργασία ακολουθεί κάποια παραδοσιακά βήματα, αλλά μερικά από τα στάδια ήταν ασυνήθιστα για εκείνη την εποχή, και πιθανόν να παραμείνουν έτσι ακόμα και τώρα. Το πρώτο βήμα είναι μια ανάλυση των συνδεδεμένων στοιχείων στο οποίο αποθηκεύονται τα περιγράμματα των στοιχείων. Αυτό ήταν μια υπολογιστικά ακριβή απόφαση εκείνη την εποχή, αλλά είχε ένα σημαντικό πλεονέκτημα : με τον έλεγχο της ενσωμάτωσης των περιγραμμάτων, και των αριθμό των «παιδιών» και «εγγονιών» των περιγραμμάτων, είναι απλό να ανιχνευτεί το αντίστροφο κείμενο (inverse text) και να αναγνωρισθεί το ίδιο εύκολα όπως το μαύρο σε άσπρο κείμενο. Η Tesseract ήταν ίσως η πρώτη μηχανή OCR που ήταν σε θέση να αναγνωρίσει λευκό σε μαύρο κείμενο. Σε αυτό το στάδιο, τα περιγράμματα συγκεντρώνονται και ενσωματώνονται σε 'blobs'.

Τα 'blobs' είναι οργανωμένα σε γραμμές κειμένου, οι γραμμές και οι περιοχές αναλύονται για τον εντοπισμό σταθερού βήματος ή αναλογικού κειμένου. Οι γραμμές κειμένου διασπώνται σε λέξεις με διαφορετικό τρόπο ανάλογα με το είδος του χαρακτήρα διαχωρισμού. Το κείμενο με σταθερό βήμα τεμαχίζεται αμέσως από τα κύτταρα του χαρακτήρα. Το αναλογικό κείμενο χωρίζεται σε λέξεις χρησιμοποιώντας καθορισμένα διαστήματα και συγκεχυμένα διαστήματα.

Η αναγνώριση συνέχεια προχωρά ως μια διαδικασία δύο περασμάτων. Στο το πρώτο πέρασμα, γίνεται μια προσπάθεια να αναγνωρίσει μία μία λέξη με τη σειρά. Κάθε λέξη που είναι ικανοποιητική περνά σε έναν προσαρμοστικό ταξινομητή ως δεδομένα εκπαίδευσης. Ο προσαρμοστικός ταξινομητής, στη συνέχεια, μπορεί να αναγνωρίσει με μεγαλύτερη ακρίβεια κείμενο χαμηλότερα στη σελίδα.

Δεδομένου ότι ο προσαρμοστικός ταξινομητής μπορεί να έχει μάθει κάτι χρήσιμο πάρα πολύ αργά για να κάνει μια συνεισφορά στο επάνω μέρος της σελίδας, γίνεται ένα δεύτερο πέρασμα στη σελίδα, στο οποίο οι λέξεις που δεν είχαν αναγνωρισθεί αρκετά καλά αναγνωρίζονται ξανά [11].



Εικόνα 1.25 : Αρχιτεκτονική Tesseract-OCR [12]

2. Κεφάλαιο 2 : Υλοποίηση συστήματος αλληλεπίδρασης ανθρώπου υπολογιστή με την χρήση της “ Tesseract-OCR ”

2.1. Προγράμματα και Εργαλεία

Σε αυτό το κεφάλαιο θα παρουσιαστεί αναλυτικά ο τρόπος υλοποίησης της εφαρμογής, τα εργαλεία και τα προγράμματα τα οποία χρησιμοποιήθηκαν.

Η εφαρμογή αναπτύχθηκε στην γλώσσα προγραμματισμού Visual Basic.NET (VB.NET) η οποία αποτελεί κομμάτι του ολοκληρωμένου περιβάλλοντος ανάπτυξης εφαρμογών (IDE) της Microsoft, Visual Studio. Για την ψηφιακή επεξεργασία των εικόνων χρησιμοποιήθηκε η γλώσσα προγραμματισμού MATLAB και για την οπτική αναγνώριση χαρακτήρων (OCR) χρησιμοποιήθηκε η “ Tesseract OCR ” η οποία αναπτύχθηκε από την Hewlett Packard και θεωρείται μία από τις πιο ακριβής μηχανές OCR ανοικτού κώδικα που διατίθενται σήμερα. Επίσης έγινε χρήση της Emgu CV που είναι μία cross platform .Net wrapper της βιβλιοθήκης Open CV, η οποία επιτρέπει να χρησιμοποιηθούν οι συναρτήσεις της Open CV από γλώσσες όπως C#, VB, VC++, IronPython κ.α. Η Open CV (Open Source Computer Vision) είναι μια βιβλιοθήκη η οποία περιλαμβάνει συναρτήσεις επεξεργασίας εικόνων και υπολογιστικής όρασης.

Άλλα υλικά που χρησιμοποιήθηκαν είναι :

Web Camera :



Εικόνα 2.1

Μαγνητικός πίνακας ζωγραφικής :



Εικόνα 2.2

2.2. Περιγραφή συστήματος

Όπως αναφέρθηκε και προηγουμένως το σύστημα αποτελείται από μια web camera, η οποία είναι συνδεδεμένη με τον υπολογιστή μέσω θύρας USB, έναν μαγνητικό πίνακα ζωγραφικής και το λογισμικό το οποίο είναι γραμμένο στην γλώσσα προγραμματισμού VB.NET. Η web camera είναι στραμμένη κατάλληλα έτσι ώστε να κοιτάει τον μαγνητικό πίνακα ζωγραφικής.

Αρχικά θα παρουσιάζετε στον μαθητή ο σωστός τρόπος γραφής του αριθμού και στην συνέχεια θα ζητείτε από τον μαθητή να γράψει τον αριθμό στον μαγνητικό πίνακα ζωγραφικής. Αφού ο μαθητής γράψει τον αριθμό, θα φωτογραφίζετε ο μαγνητικός πίνακας (μέσω της κάμερας) και στην συνέχεια θα εξετάζετε αν ο αριθμός στην φωτογραφία είναι ο αριθμός ο οποίος ζητείτε. Αν ο μαθητής έχει γράψει σωστά τον αριθμό θα εμφανίζετε μήνυμα επιβράβευσης αλλιώς θα του ζητείτε να ξανά-προσπαθήσει.

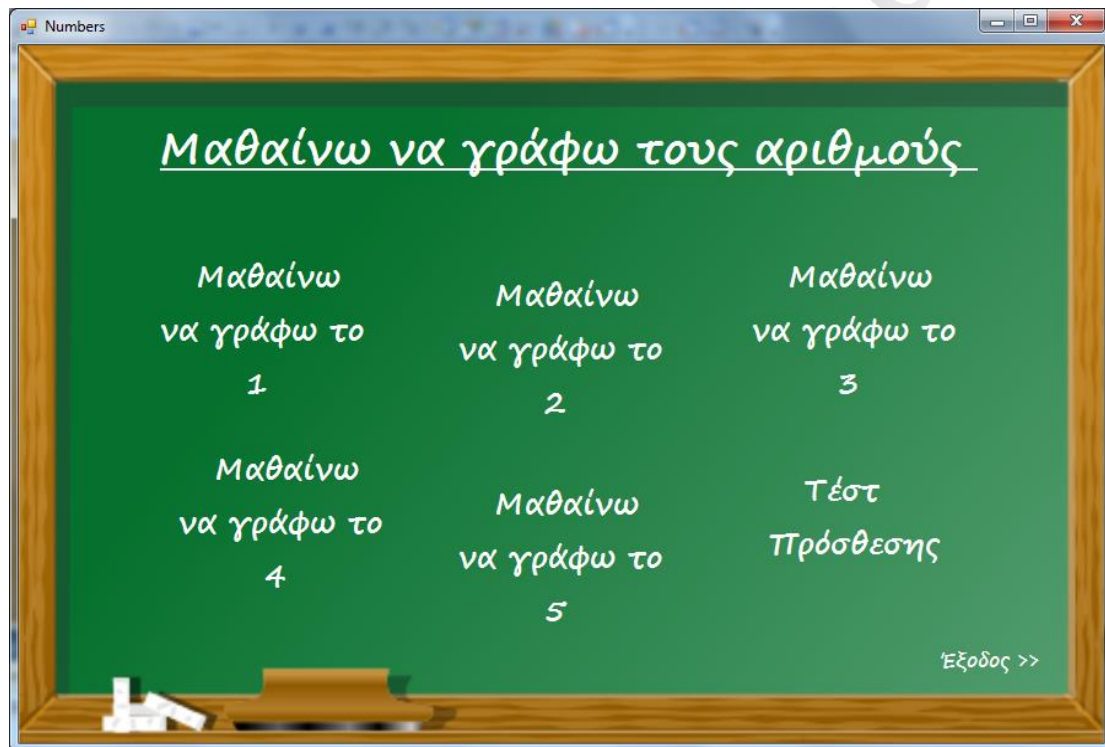
2.3. Βιβλιογραφική ανασκόπηση (OCR στην εκπαίδευση)

Η Lisa Antony ανέπτυξε μια εφαρμογή αναγνώρισης χειρογραφής η οποία έχει εφαρμογή σε ευφυή συστήματα διδασκαλίας για την εκμάθηση της άλγεβρας και την επίλυση των εξισώσεων [13]. Ο Bruce R. Maxim ανέπτυξε ένα περιβάλλον παιχνιδιού για την εκμάθηση της γραφής σε παιδιά του δημοτικού σχολείου με την χρήση επιτραπέζιου υπολογιστή χρησιμοποιώντας το σύστημα χειρογραφής του D'Nealian [14]. Ο Zhi-Hui Hu παρουσίασε ένα ευφύες σύστημα διδασκαλίας της κινέζικης γλώσσας με την χρήση χειρογραφής με αυτόματη αναγνώριση λαθών χρησιμοποιώντας την μέθοδο attributed relational graph (ARG) [15]. Ο Mohd Nizam Bin Saad παρουσίασε μια εφαρμογή με το όνομα HLN (handwriting leaning number) για την που βοηθάει στην εκμάθηση των αριθμών (0-9) που βασίζεται στην αναγνώριση χειρογραφής και απευθύνεται σε παιδιά προσχολικής ηλικίας [16].

2.4. Υλοποίηση εφαρμογής στην VB.NET

Για την εκπαίδευση των μαθητών στον σωστό τρόπο γραφής των αριθμών δημιουργήθηκε εφαρμογή στην γλώσσα προγραμματισμού VB.NET η οποία αποτελείται από 8 φόρμες, την αρχική στην οποία ο μαθητής μπορεί να επιλέξει τον αριθμό που επιθυμεί να μάθει να γράφει (1-5) και ένα τεστ μαθηματικών (πρόσθεσης), άλλες πέντε φόρμες, μία για κάθε αριθμό (1-5), μία φόρμα σαν τεστ πρόσθεσης και μία βοηθητική φόρμα η οποία θα χρησιμεύσει στην επίλυση προβλημάτων επεξεργασία εικόνας.

Η αρχική φόρμα της εφαρμογής φαίνεται παρακάτω :



Εικόνα 2.3

Όπως είπαμε και προηγουμένως, στην αρχική φόρμα υπάρχουν 7 επιλογές, μία για κάθε αριθμό από 1-5 και άλλη μία επιλογή η οποία είναι ένα τεστ πρόσθεσης των αριθμών 1 με 5, όπως φαίνεται και στην εικόνα 2.3 . Η επιλογή “Έξοδος” χρησιμοποιείται για τον τερματισμό της εφαρμογής.

Επιλέγοντας “ Μαθαίνω να γράφω το 1 ” ανοίγει μία νέα φόρμα η οποία φαίνεται παρακάτω :




Εικόνα 2.4

Σκοπός της συγκεκριμένης φόρμας είναι να διδάξει στον μαθητή τον σωστό τρόπο γραφείς του αριθμού 1. Αρχικά παρουσιάζετε στο πινακάκι της φόρμας ο αριθμός 1 και στην συνέχεια εμφανίζονται βέλη τα οποία δείχνουν τον τρόπο γραφείς του αριθμού όπως φαίνεται και στην εικόνα 2.4. Ακολούθως ζητείτε από τον μαθητή να γράψει τον αριθμό 1 στον μαγνητικό πίνακα.

Στο δεξί μέρος της φόρμας φαίνεται η εικόνα από την web camera η οποία είναι στραμμένη μπρος τον μαγνητικό πίνακα, μαζί με άλλες τρεις επιλογές.

Η πρώτη επιλογή, είναι η επιλογή "Ρυθμίσεις" με την οποία ανοίγει μια άλλη φόρμα η οποία χρησιμοποιείται για την ρύθμιση κάποιων παραμέτρων που αφορούν την επεξεργασία της εικόνας ώστε να γίνει όσο πιο σωστά γίνεται η αναγνώριση του αριθμού από την φωτογραφία, η οποία προήρθε απ'την web camera. Θα αναφερθούμε εκτενέστερα σε αυτή την φόρμα αργότερα.

Η δεύτερη επιλογή, είναι η επιλογή με το εικονίδιο της κάμερας , η οποία χρησιμοποιείτε για να τραβήξουμε φωτογραφία από τον μαγνητικό πίνακα ζωγραφικής, μέσω της web camera και στην συνέχεια να γίνει η επεξεργασία της εικόνας, ώστε να διαπιστωθεί αν ο αριθμός που έχει γράψει ο μαθητής είναι ο αριθμός 1.

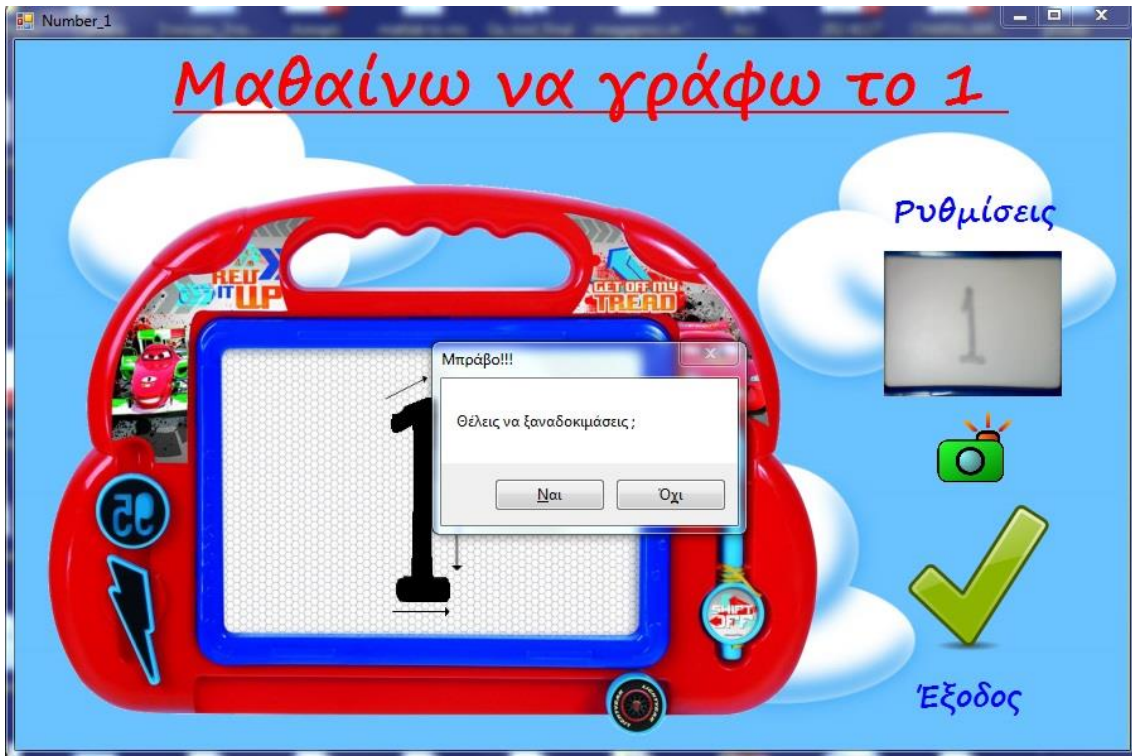
Η επιλογή "Έξοδος" χρησιμοποιείτε για τον τερματισμό της φόρμας και την επιστροφή στην αρχική φόρμα (εικόνα 2.3.).

Ο μαθητής αφού γράψει τον αριθμό 1 στον μαγνητικό πίνακα, πατάει στην επιλογή με το εικονίδιο της κάμερας όπως φαίνεται και στην εικόνα 2.5.



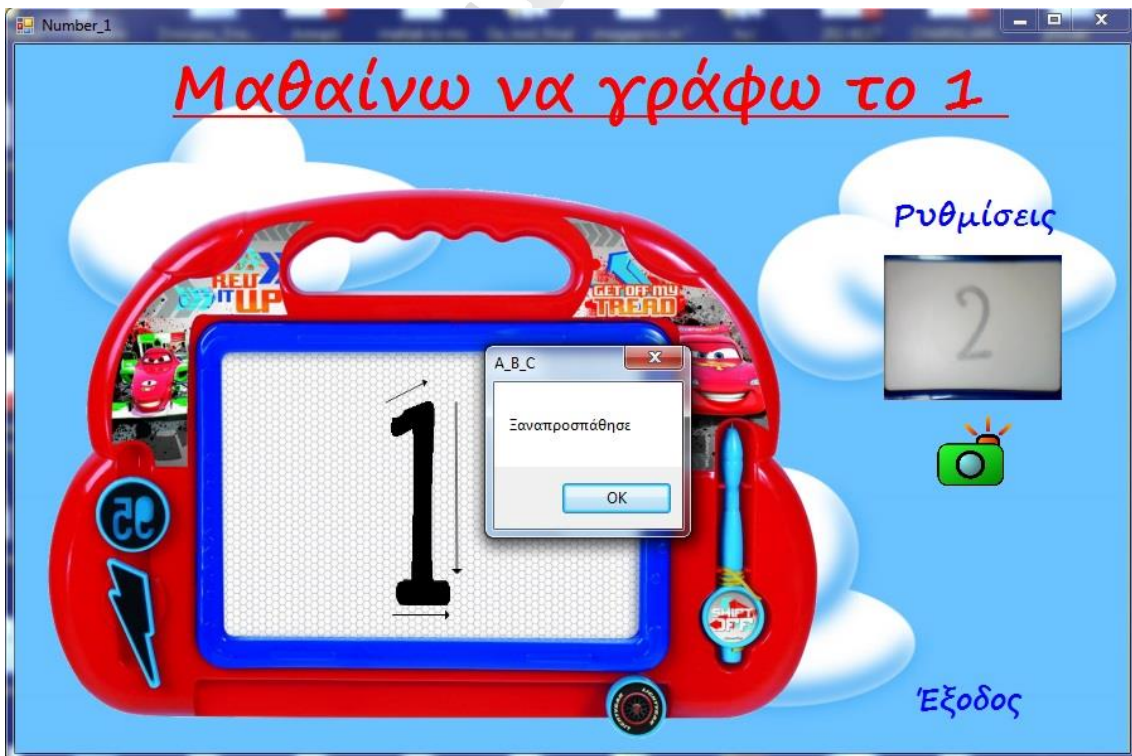
Εικόνα 2.5

Αν ο αριθμός που έχει γράψει είναι σωστός τότε, εμφανίζετε στην φόρμα το σύμβολο «tick» ✓, και ένα μήνυμα επιβράβευσης, όπως φαίνετε και στην εικόνα 2.6.



Εικόνα 2.6

Αν ο αριθμός που έγραψε ο μαθητής δεν είναι σωστός τότε εμφανίζεται μήνυμα που του ζητά να ξαναπροσπαθήσει και αυτό επαναλαμβάνεται μέχρι να γράψει σωστά τον αριθμό.



Εικόνα 2.7

Τα ίδια ισχύουν και για τις φόρμες με τους υπόλοιπους αριθμούς (2-5).
Για τον αριθμό δύο :



Εικόνα 2.8

Για τον αριθμό Τρία :



Εικόνα 2.9

Για τον αριθμό τέσσερα :



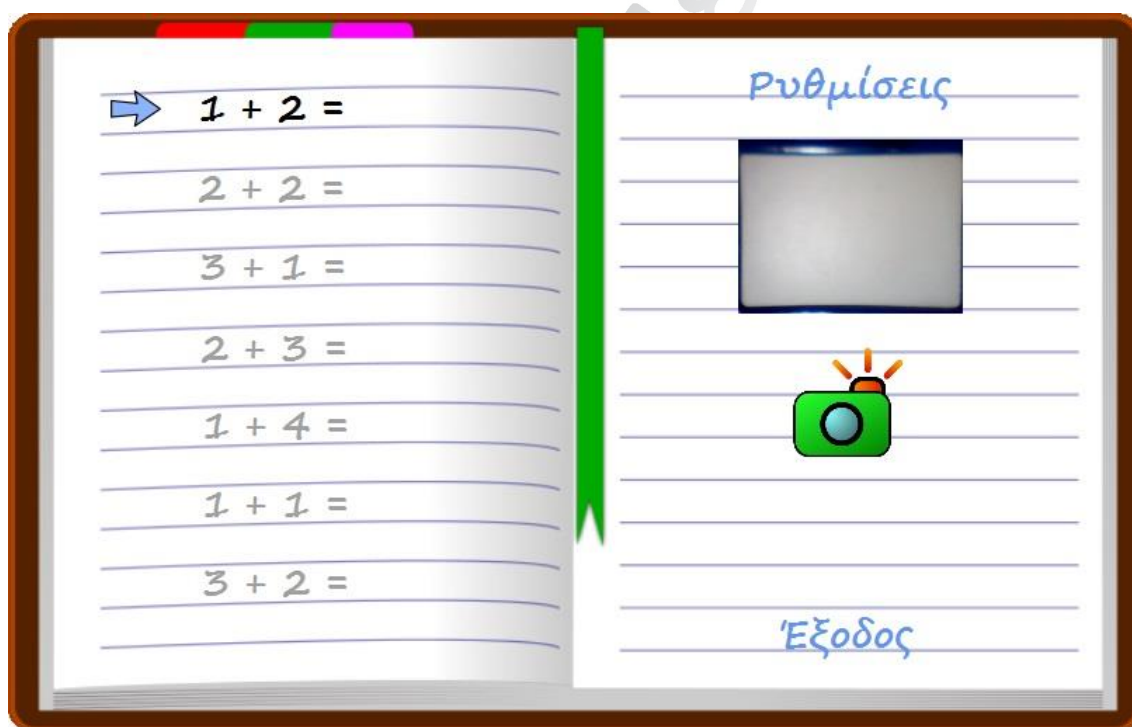
Εικόνα 2.10

Για τον αριθμό πέντε :



Εικόνα 2.11


Επιλέγοντας από την αρχική φόρμα “ Τεστ Πρόσθεσης ”, ανοίγει η φόρμα που φαίνεται στην εικόνα 2.12.



Εικόνα 2.12

Αυτή η φόρμα είναι ένα τέστ πρόσθεσης στην οποία ο μαθητής καλείτε να δώσει τις λύσεις στις πράξεις που φαίνονται και στην εικόνα 2.12.

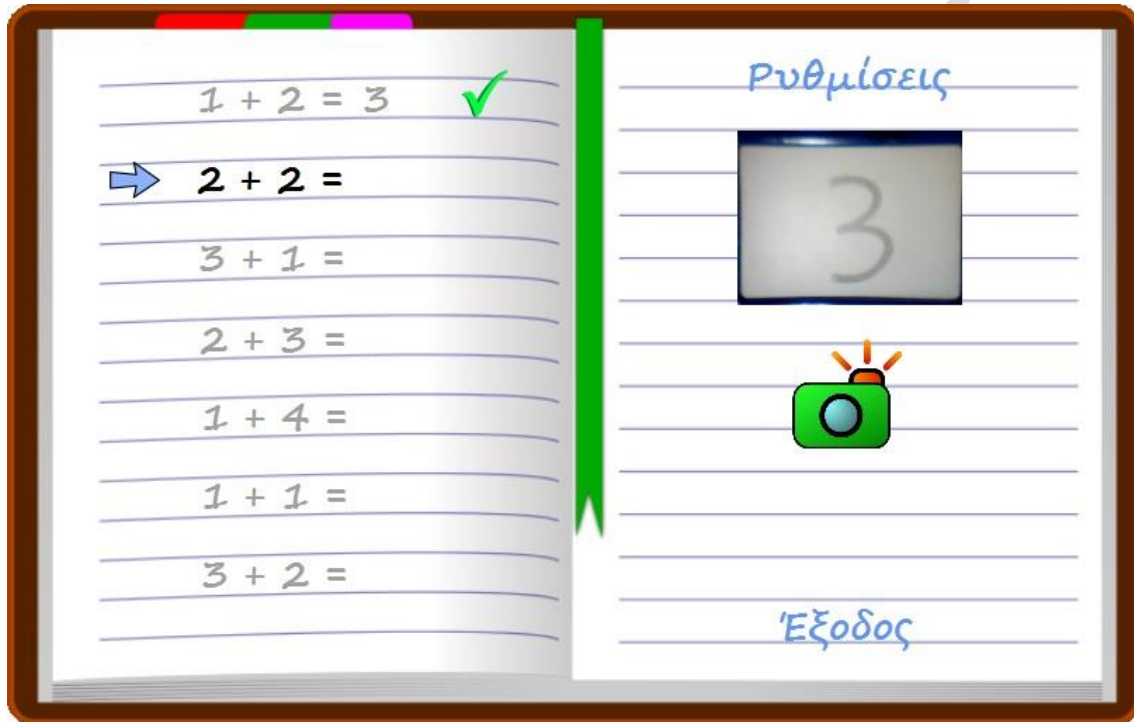
Όπως και στις προηγούμενες φόρμες στο δεξί μέρος της φόρμας φαίνεται η εικόνα από

την web camera, η επιλογή “Ρυθμίσεις”, η επιλογή  η οποία χρησιμοποιείτε για να τραβήξουμε φωτογραφία και στην συνέχεια να γίνει η επεξεργασία ώστε να διαπιστωθεί αν ο

αριθμός που έχει γράψει ο μαθητής είναι ο σωστός και η επιλογή “Έξοδος” με την οποία κλείνει η φόρμα.

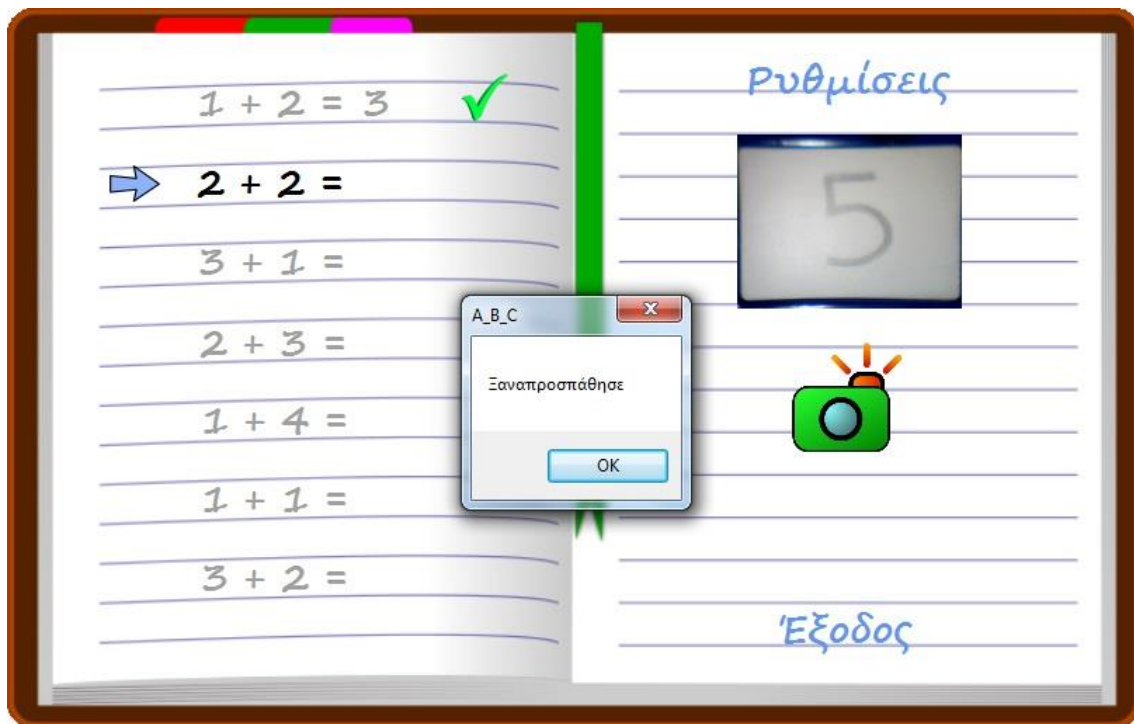
Στο αριστερό μέρος της φόρμας βρίσκονται οι πράξεις πρόσθεσης τις οποίες καλείτε να λύσει ο μαθητής. Κάθε φορά ένα βελάκι δείχνει ποιά πράξη πρέπει να απαντηθεί, όταν ο

μαθητής απαντήσει σωστά σε μία πράξη εμφανίζεται το σύμβολο «tick» ✓ και προχωράει στην επόμενη όπως φαίνεται και στην εικόνα 2.13.



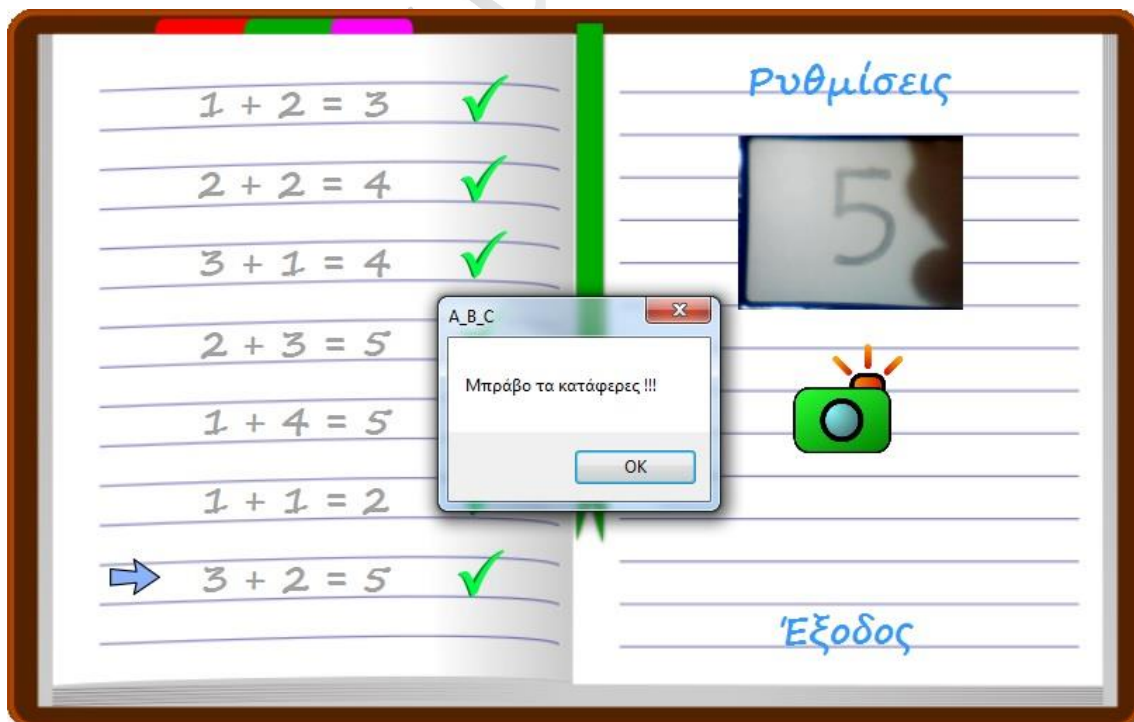
Εικόνα 2.13

Αν ο μαθητής απαντήσει λάθος σε μια πράξη τότε εμφανίζεται μήνυμα που του ζητάει να ξαναπροσπαθήσει.



Εικόνα 2.14

Όταν όλες οι πράξεις απαντηθούν σωστά τότε εμφανίζεται ένα μήνυμα επιβράβευσης όπως φαίνεται και στην εικόνα 2.15.



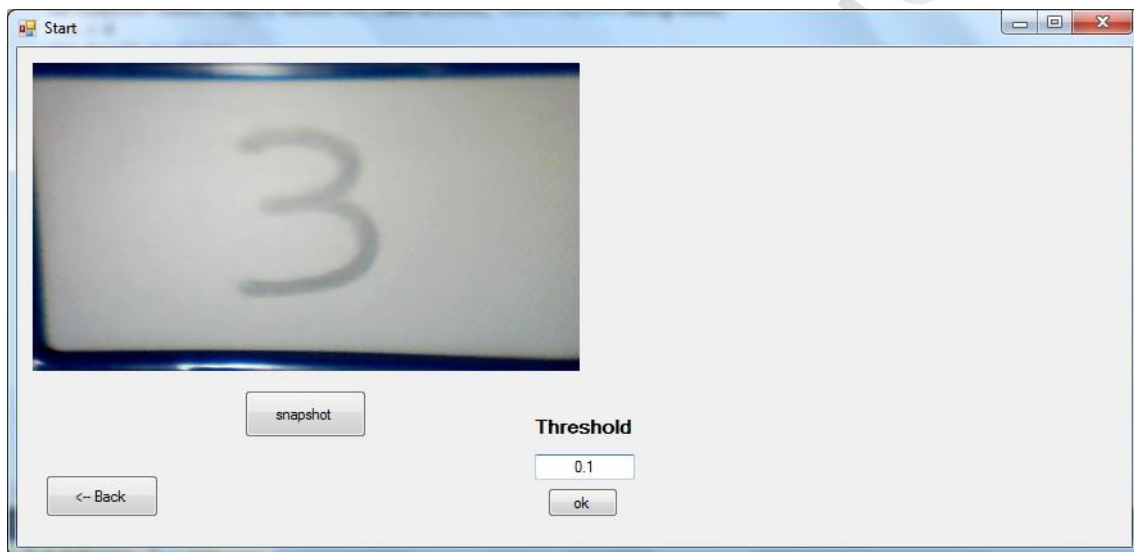
Εικόνα 2.15

Τέλος, σε όλες τις προηγούμενες φόρμες, στο δεξί μέρος υπάρχει η επιλογή “Ρυθμίσεις” η οποία όπως προείπαμε οδηγεί σε μια βοηθητική φόρμα που δημιουργήθηκε για να μπορούμε να ρυθμίζουμε εύκολα κάποιες παραμέτρους που αφορούν την επεξεργασία της φωτογραφίας, ώστε να είναι όσο το δυνατό πιο ακριβής η αναγνώριση του αριθμού.

Σκοπός της συγκεκριμένης φόρμας είναι η επιλογή του κατάλληλου κατωφλίου (Threshold), ώστε να είναι όσο το δυνατό πιο καθαρή η εικόνα του αριθμού κατά την μετατροπή της εικόνας σε ασπρόμαυρη (δυναδική) εικόνα, δηλαδή εικόνα που αποτελείται από 0 και 1.

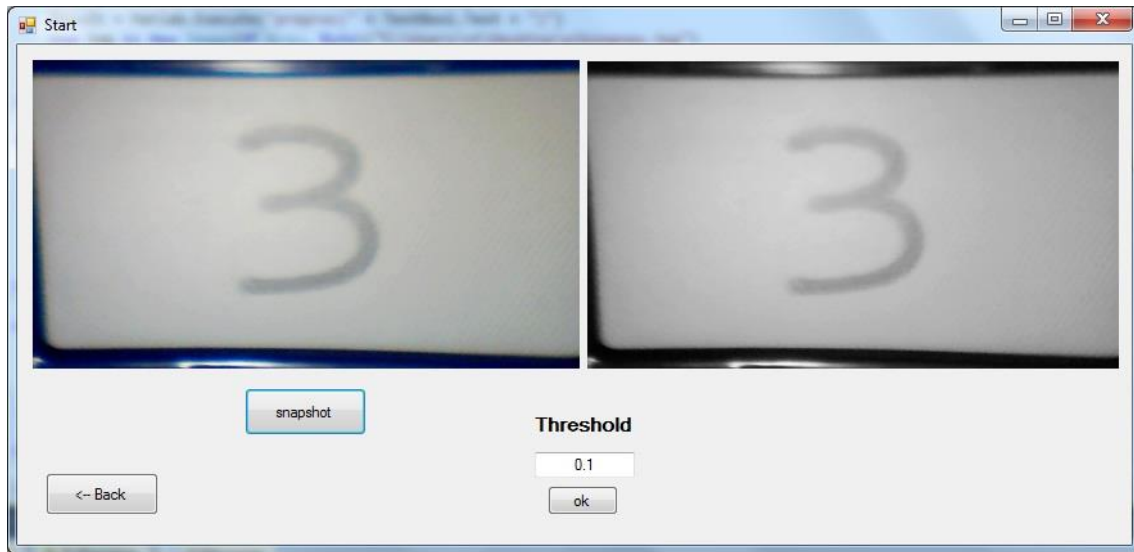
Ο κώδικας για την επεξεργασία της φωτογραφίας έχει υλοποιηθεί στην γλώσσα προγραμματισμού MATLAB και θα εξηγηθεί αργότερα.

Πατώντας στην επιλογή “Ρυθμίσεις” ανοίγει η φόρμα που φαίνεται στην παρακάτω εικόνα.



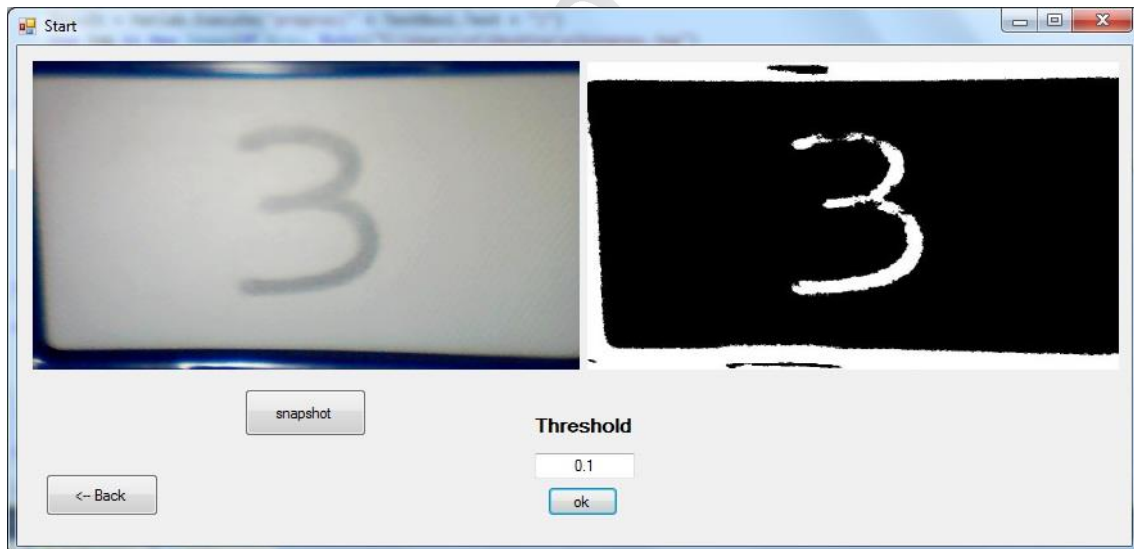
Εικόνα 2.16

Όπως φαίνεται σε αυτή την φόρμα (εικόνα 2.16), πάνω αριστερά βλέπουμε την εικόνα από την web camera η οποία είναι στραμμένη μπρός τον μαγνητικό πίνακα ζωγραφικής (στον οποίο, στο παράδειγμά μας, έχουμε γράψει τον αριθμό 3), κάτω από αυτή, υπάρχει η επιλογή «snapshot» με την οποία η web camera ‘παίρνει’ ένα στιγμιότυπο και το εμφανίζει στο δεξί μέρος της φόρμας όπως φαίνεται στην εικόνα 2.17.

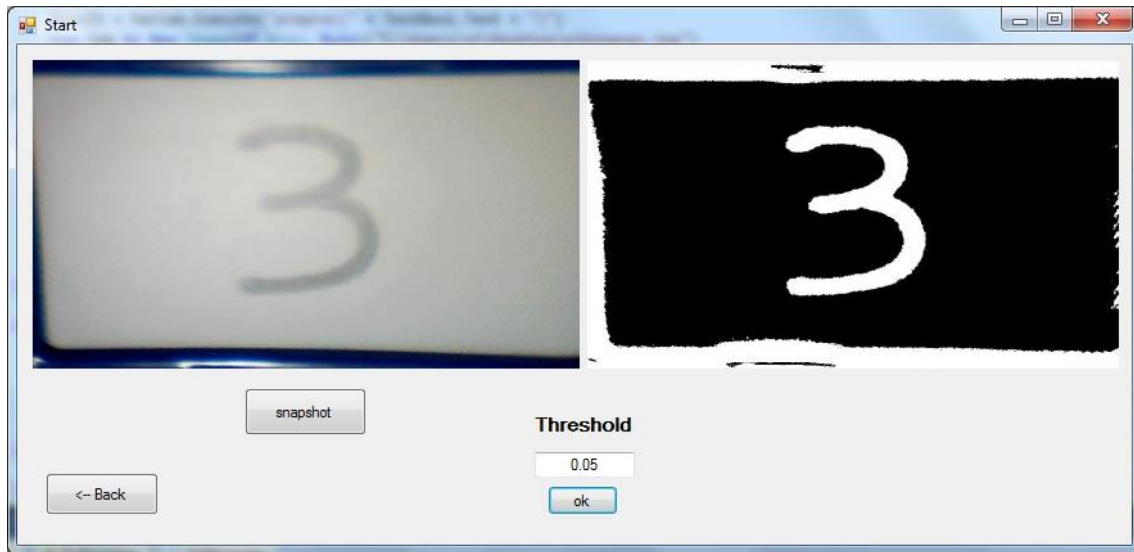
**Εικόνα 2.17**

Στην μέση της φόρμας υπάρχει η επιλογή "Threshold" η οποία είναι ένα κατώφλι που χρησιμοποιείτε για την μετατροπή της εικόνας σε ασπρόμαυρη (δυναδική) εικόνα.

Η προκαθορισμένη τιμή της μεταβλητής αυτής είναι το 0,1. Πατώντας "ok", εφαρμόζουμε αυτό το κατώφλι στην εικόνα.

**Εικόνα 2.18**

Όπως βλέπουμε στην εικόνα 2.18, ο αριθμός 3 δεν φαίνεται καθαρά, αλλάζοντας το κατώφλι σε 0,05 έχουμε το αποτέλεσμα φαίνεται στην παρακάτω εικόνα.



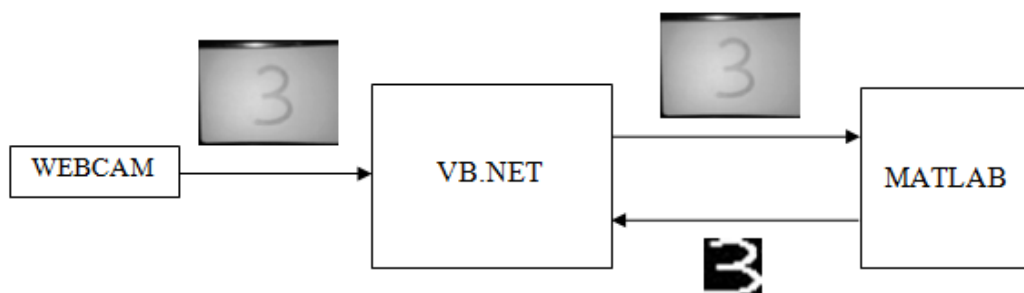
Εικόνα 2.19

Τώρα ο αριθμός 3 είναι καθαρός (χωρίς κενά), το περίγραμμα δεν μας απασχολεί αρκεί ο αριθμός να φαίνεται καθαρά.

Η επιλογή “Back”, χρησιμοποιείτε για την επιστροφή στην φόρμα από την οποία οδηγηθήκαμε σε αυτή την φόρμα, αφού έχει προηγηθεί η επιλογή του κατάλληλου κατωφλίου.

2.5. Υλοποίηση συστήματος εντοπισμού και απομόνωσης της χρήσιμης πληροφορίας (αριθμού) από την αρχική εικόνα στο MATLAB

Σε αυτό το μέρος της πτυχιακής, θα παρουσιαστούν αναλυτικά τα βήματα και οι τεχνικές της ψηφιακή επεξεργασία εικόνας που χρησιμοποιήθηκαν και σκοπό έχουν να καθarisουν την εικόνα ώστε να διακρίνεται όσο το δυνατό περισσότερο η χρήσιμη πληροφορία (αριθμός) καθώς και να εντοπιστεί και να απομονωθεί η χρήσιμη πληροφορία από την αρχική φωτογραφία, η οποία στην συνέχεια θα σταλεί σαν είσοδος στην μηχανή οπτικής αναγνώρισης χαρακτήρων.

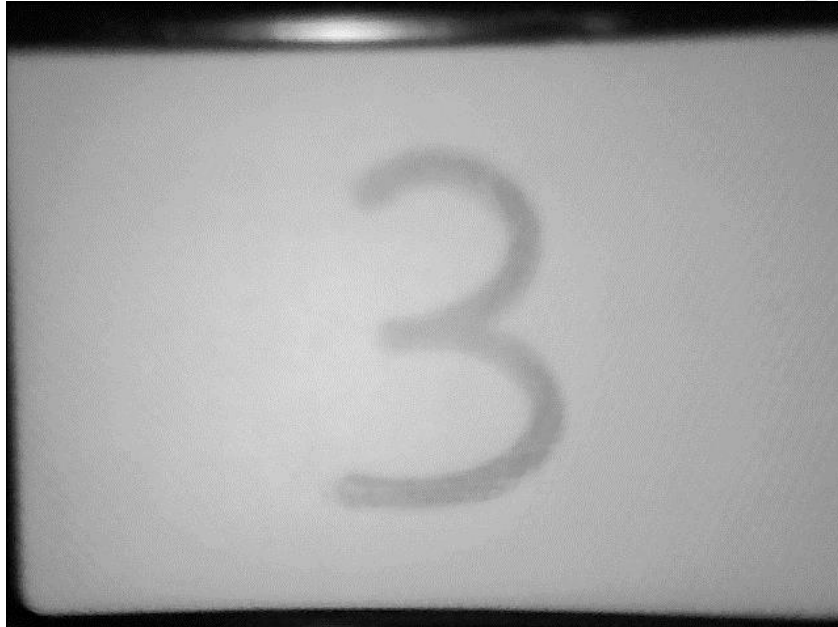


Εικόνα 2.20

Στην παραπάνω εικόνα (εικόνα 2.20), φαίνεται γραφικά η πορεία της εικόνας από την στιγμή που εισέρχεται μέσα στο σύστημα μέσω της κάμερας, καθώς και ο ρόλος του δημιουργία συστήματος αλληλεπίδρασης ανθρώπου υπολογιστή...

προγράμματος MATLAB, ο οποίος είναι να καταφέρει να απομονώσει την χρήσιμη πληροφορία από την αρχική φωτογραφία, έτσι ώστε να είναι πιο εύκολη η διαδικασία της αναγνώρισης του αριθμού από την μηχανή οπτικής αναγνώρισης χαρακτήρων *tesseract-ocr*.

Όπως είπαμε και πριν, όταν τραβηχτεί η φωτογραφία με τον αριθμό μέσω της κάμερας στέλνεται από την VB.NET στο MATLAB για να γίνει η επεξεργασία της. Παρακάτω θα δείξουμε ένα παράδειγμα επεξεργασίας μίας εικόνας με τον αριθμό 3 από το MATLAB.



Εικόνα 2.21

Μόλις το MATLAB παραλάβει την φωτογραφία, ξεκινάει την επεξεργασία της ώστε να εξάγει την χρήσιμη πληροφορία (τον αριθμό 3) από αυτήν.

Αρχικά εφαρμόζετε στην εικόνα ο μετασχηματισμός Top-hat (περισσότερες πληροφορίες για τον μετασχηματισμό top-hat υπάρχουν στο πρώτο κεφάλαιο - θεωρία), χρησιμοποιώντας ένα δομικό στοιχείο (structure element) τύπου δίσκου ακτίνας 35 pixels, για την διόρθωση του οποιουδήποτε άνισου φωτισμού στο φόντο της εικόνας, όπως φαίνεται και στην εικόνα 2.22.

**Εικόνα 2.22**

Στην συνέχεια ακολουθεί η διαδικασία της κατωφλίωσης της εικόνας, που σκοπό έχει να ξεχωρίσουν τα αντικείμενα μιας εικόνας από το φόντο της, στην δική μας περίπτωση, ο αριθμός 3 από το μαύρο φόντο.

Επιλέγοντας για κατώφλι την τιμή 0.1 παίρνουμε το παρακάτω αποτέλεσμα

**Εικόνα 2.23**

Όπως βλέπουμε και στην εικόνα 2.23 ο αριθμός 3 δεν φαίνεται καθαρά αλλά με σπασίματα, αυτό δεν το θέλουμε.

Αλλάζοντας το κατώφλι στον αριθμός 0,05 το αποτέλεσμα φαίνεται στην παρακάτω εικόνα



Εικόνα 2.24

Τώρα βλέπουμε ότι ο αριθμός 3 φαίνεται καθαρά και χωρίς σπασίματα, οπότε και καταλαβαίνουμε ότι η σωστή επιλογή κατώφλιου είναι πολύ σημαντική για την συνέχεια της διαδικασίας.

Έχοντας επιλέξει το κατάλληλο κατώφλι συνεχίζουμε εφαρμόζοντας στην εικόνα τον μορφολογικό τελεστή κλείσιμο (closing), χρησιμοποιώντας ένα δομικό στοιχείο (structure element) τύπου δίσκο ακτίνας 5 pixels, για την εξομάλυνση των εξωτερικών ακμών των αντικειμένων, όπως βλέπουμε και στην εικόνα 2.25.



Εικόνα 2.25

Στη συνέχεια η επεξεργασία προχωρεί καθαρίζοντας τα αντικείμενα της εικόνας που βρίσκονται στα σύνορα της εικόνας, προσπαθώντας να μείνει στην εικόνα μόνο ο αριθμός 3. Αυτό επιτυγχάνετε στο MATLAB με την εντολή “imclearborder”.



Εικόνα 2.26

Όπως βλέπουμε στην εικόνα 2.26 η εικόνα έχει σχεδόν καθαρίσει και ο αριθμός 3 είναι το μόνο μεγάλο αντικείμενο που έχει απομείνει στην εικόνα.

Εκτός από τον αριθμό 3, στα αριστερά της εικόνας φαίνεται ότι έχουν μείνει ακόμα κάποια μικρά αντικείμενα, διότι αυτά δεν ακουμπούσαν τα σύνορα της εικόνας. Χρησιμοποιώντας την εντολή “bwareaopen”, αφαιρούμε τα μικρά αντικείμενα που υπάρχουν στην εικόνα, τα οποία αποτελούνται από λιγότερα από 500 λευκά pixels.



Εικόνα 2.27

Όπως φαίνεται στην εικόνα 2.27, ο αριθμός 3 είναι το μόνο αντικείμενο που έχει μείνει στην εικόνα. Στην συνέχεια αφού υπολογιστούν τα όρια της περιοχής των αντικειμένων (boundaries) στην εικόνα με την εντολή “bwboundaries”, καθώς και το εμβαδόν και το ελάχιστο ορθογώνιο το οποίο τα περικλείει, με την εντολή “regionprops”, αποκόπτεται το αντικείμενο (αριθμός 3) από την εικόνα με την εντολή “imcrop”.



Εικόνα 2.28

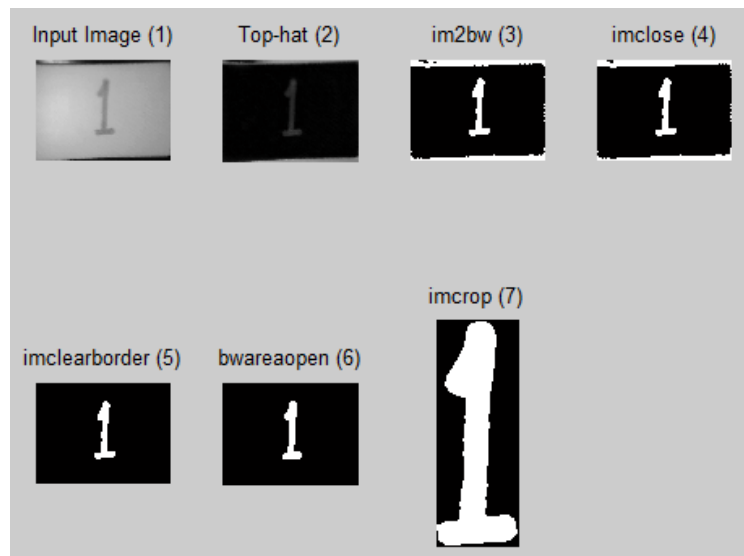
Στην συνέχεια η εικόνα υπόκειται σε αλλαγή μεγέθους (20x18 pixels) και αποστέλλεται πίσω στην VB.NET και στην μηχανή οπτικής αναγνώρισης χαρακτήρων όπου και θα γίνει η αναγνώριση του αριθμού.

2.5.1. Δοκιμές

Έγιναν τριών ειδών δοκιμές για τον έλεγχο της σωστής λειτουργίας της εφαρμογής, δοκιμές με εικόνες χωρίς θόρυβο (χωρίς σκιά ή ξένα αντικείμενα), δοκιμές με σκιά και δοκιμές με ξένα αντικείμενα.

Δοκιμές χωρίς θόρυβο :

Δοκιμή 1_n :



Εικόνα 2.29

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :

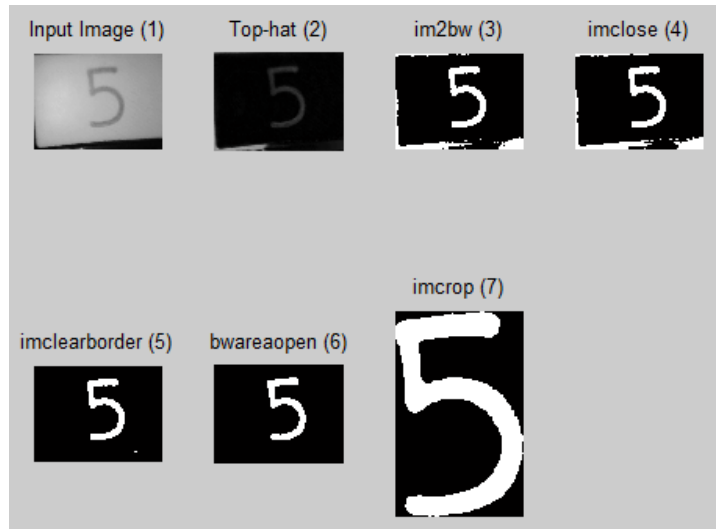
Δοκιμή 2₁ :



Εικόνα 2.30

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :

Δοκιμή 3_n :

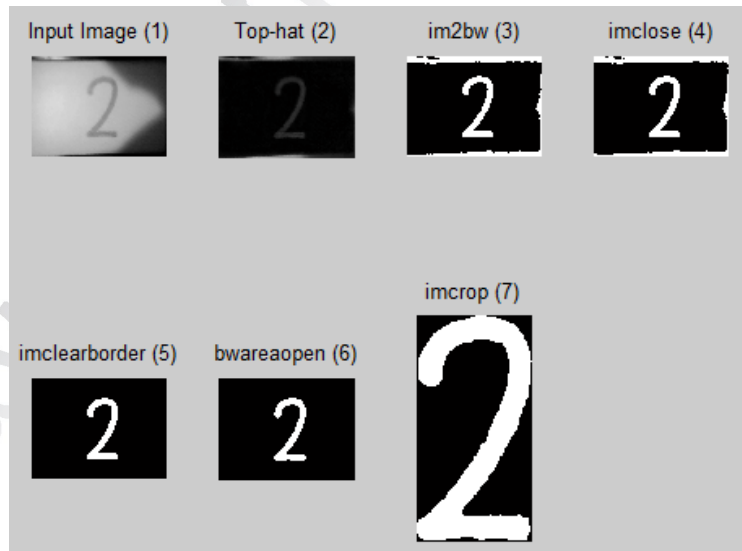


Εικόνα 2.31

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :

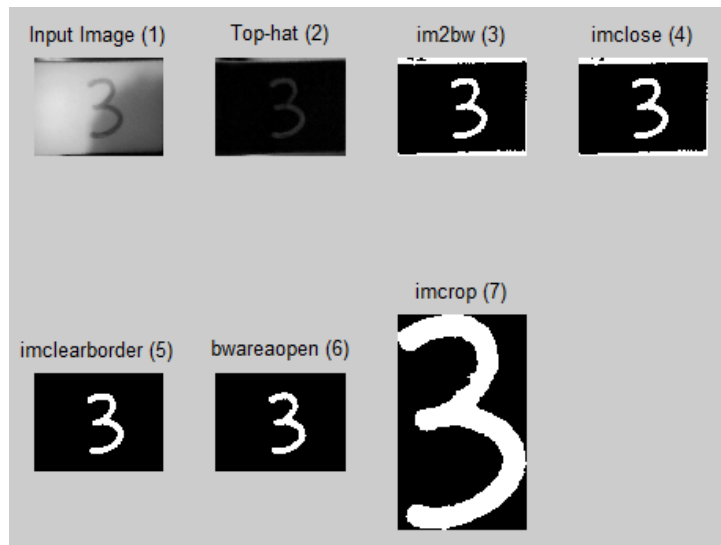
Δοκιμές με σκιά :

Δοκιμή 1_n :



Εικόνα 2.32

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :


Δοκιμή 2_η:

Εικόνα 2.33

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι : 

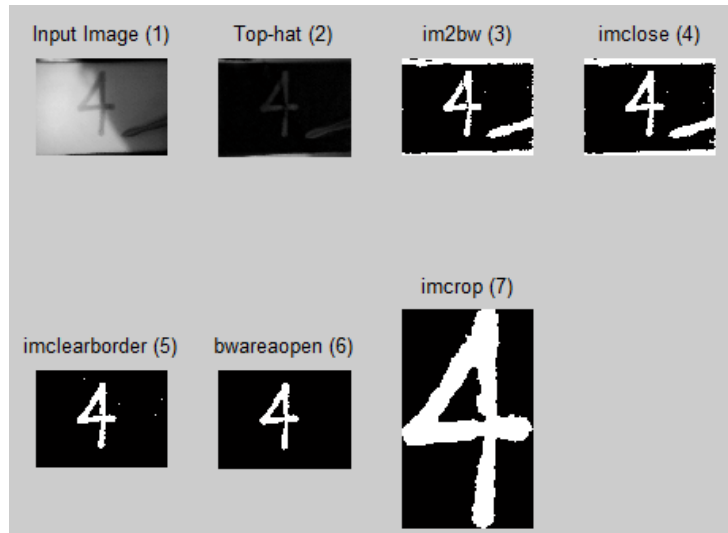
Δοκιμή 3_η:

Εικόνα 2.34

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι : 

Δοκιμές με σκιά και ξένα αντικείμενα :

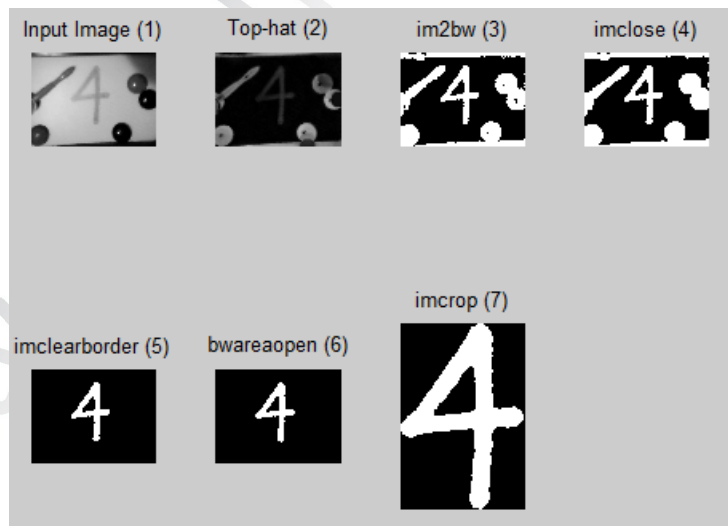
Δοκιμή 1_η:



Εικόνα 2.35

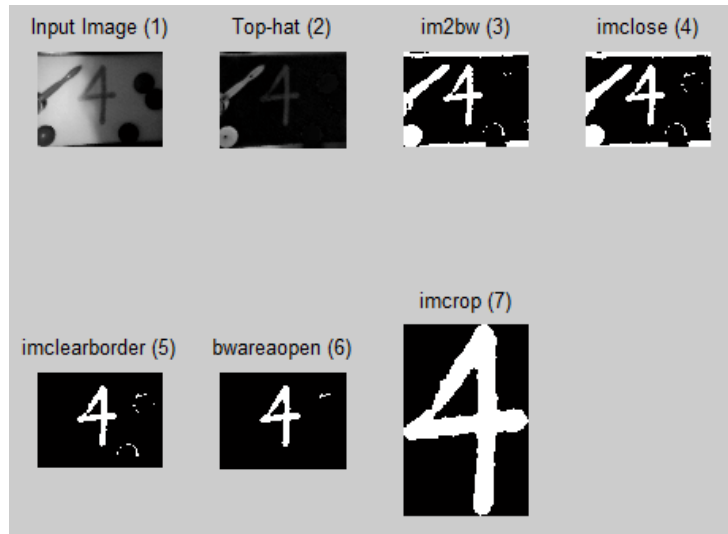
Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :

Δοκιμή 2_η:



Εικόνα 2.36

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :

Δοκιμή 3_η:

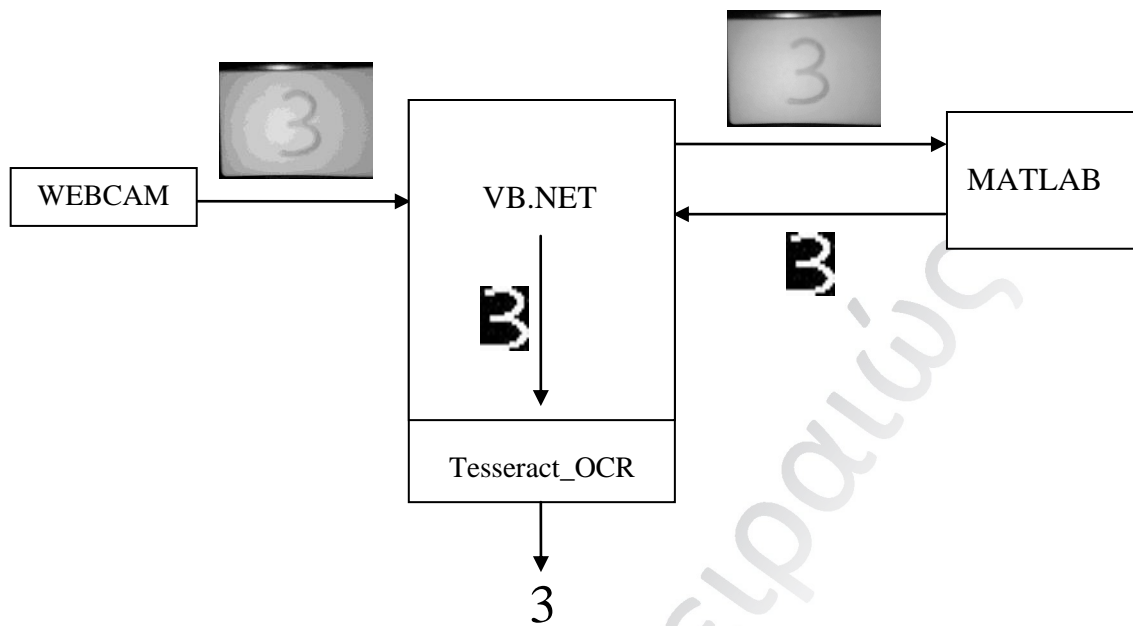
Εικόνα 2.37

Στην συνέχεια με την εντολή `imresize` γίνεται αλλαγή του μεγέθους της εικόνας σε 20x18 pixels, η τελική εικόνα είναι :

Όπως μπορούμε να δούμε από τις δοκιμές που παρουσιάστηκαν, το σύστημα μπορεί να εντοπίσει τον αριθμό ακόμα και σε εικόνες που περιέχουν διάφορα είδη θορύβου όπως σκιά, ξένα αντικείμενα ή και συνδυασμό αυτών.

2.6. Αναγνώριση χαρακτήρων με την χρήση της Tesseract-OCR

Μετά την ολοκλήρωση της επεξεργασίας της εικόνας από το MATLAB, η νέα (επεξεργασμένη) εικόνα αποστέλλεται πίσω στην VB.NET στην οποία θα γίνει η αναγνώριση του αριθμού, με την χρήση της μηχανής οπτικής αναγνώρισης χαρακτήρων Tesseract-OCR (περισσότερες πληροφορίες στο κεφάλαιο 1 - θεωρία), όπως φαίνεται και στην εικόνα 2.38.



Εικόνα 2.38

Όπως αναφέρεται και στο θεωρητικό μέρος της πτυχιακής εργασίας (κεφάλαιο 1), η Tesseract είναι ίσως η πιο ακριβής μηχανή οπτική αναγνώρισης χαρακτήρων (OCR) ανοιχτού κώδικα. Ήταν μία από τις 3 κορυφαίες μηχανές στη δοκιμή Ακρίβεια 1995 UNLV. Μεταξύ των ετών 1995 και 2006 είχε γίνει λίγη δουλειά σε αυτό, αλλά από τότε έχει βελτιωθεί σε μεγάλο βαθμό από την Google. Η Tesseract λειτουργεί σε Linux, Windows (με VC ++, VB.NET, VC# ή Cygwin) και Mac OSX. Η Tesseract σχεδιάστηκε αρχικά για να αναγνωρίζει μόνο αγγλικό κείμενο, αλλά στην συνέχεια τροποποιήθηκε κατάλληλα και έτσι σήμερα είναι σε θέση να αναγνωρίζει και άλλες γλώσσες όπως : ελληνικά, ουκρανικά, тайландέζικα, τούρκικα, σουηδικά, σέρβικα, πορτογαλικά, ιταλικά, κορεάτικα, γαλλικά, ινδικά, αρχαία ελληνικά, ιαπωνικά, κινέζικα, ρώσικα κτλ. Μπορεί κάποιος να χρησιμοποιήσει την tesseract για την αναγνώριση γραμμάτων ή λέξεων σε οποιαδήποτε από τις υποστηριζόμενες γλώσσες, κατεβάζονταν απλά τα δεδομένα εκπαίδευσης από την παρακάτω σελίδα : <https://code.google.com/p/tesseract-ocr/downloads/list> [17].

3. Κεφάλαιο 3 : Ασαφής λογική

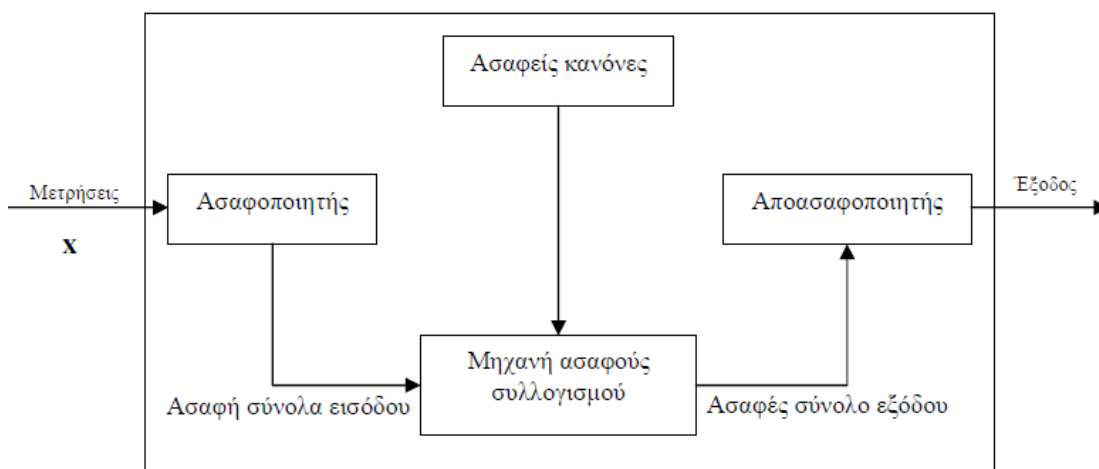
3.1. Τι είναι τα ασαφή συστήματα

Ο όρος «ασαφής λογική» εισήχθη από τον Lotfi A. Zadeh το 1965 [18]. και έχει εφαρμοστεί σε πολλούς τομείς, από τη θεωρία του ελέγχου μέχρι την τεχνητή νοημοσύνη. Η ασαφής λογική είχε, ωστόσο μελετηθεί από το 1920, ως εκτιμώμενη λογική κυρίως από τον Lukasiewicz και τον Tarski [19].

Στην κλασική αριστοτέλεια λογική μια λογική πρόταση μπορεί να πάρει μόνον δύο τιμές, αληθής ή ψευδής (1 ή 0), η ασαφής λογική είναι μια επέκταση της κλασικής αριστοτέλειας λογικής στην οποία μια πρόταση μπορεί να είναι αληθής με κάποιο βαθμό αληθείας, δηλαδή λέει ότι τα πράγματα συχνά δεν είναι «άσπρο-μαύρο» αλλά «αποχρώσεις του γκρι». Η ιδέα αυτή απετέλεσε επανάσταση στη θεωρία της λογικής η οποία μιμητέ τον τρόπο λειτουργίας της ανθρώπινης λογικής [21]. Η ασαφής λογική χρησιμοποιείται κυρίως για πολύπλοκα και σύνθετα προβλήματα για τα οποία δεν διαθέτουμε αρκετά αριθμητικά δεδομένα και μας βοηθάει να κατανοήσουμε την συμπεριφορά τέτοιων συστημάτων. Στα συστήματα ασαφούς λογικής συνδυάζονται δύο ειδών πληροφορίες η αριθμητική (είσοδος-μετρήσεις) και η πληροφορία που προέρχεται από την γνώση των εμπειρογνώμων και εκφράζεται με φυσική γλώσσα [21].

Τα συστήματα ασαφούς λογικής ανήκουν στην κατηγορία των ευφυών συστημάτων και εφαρμόζονται σε πάρα πολλά πρακτικά προβλήματα.

Στην παρακάτω εικόνα φαίνεται το διάγραμμα ενός συστήματος ασαφούς λογικής



Εικόνα 3.1 [21]

Όπως φαίνεται και στην εικόνα 3.1 τα δομικά στοιχεία ενός συστήματος ασαφούς λογικής είναι ο ασαφοποιητής (fuzzifier) οποίος χρησιμοποιείται για την ασαφοποίηση των τιμών της εισόδου, τα ασαφή σύνολα (fuzzy sets) που χρησιμοποιούνται για την αναπαράσταση των μεταβλητών εισόδου και εξόδου με λεκτικούς όρους, την βάση γνώσης (knowledge base) στην οποία υπάρχουν οι ασαφής κανόνες, τον μηχανισμό συμπερασμού (inference engine) ή μηχανισμός ασαφούς συλλογισμού ο οποίος χρησιμοποιώντας την βάση γνώσης εξάγει τα ασαφή σύνολα των συμπερασμάτων και τον αποασαφοποιητή (defuzzifier) ο οποίος μετατρέπει τα συμπεράσματα που εξάγει ο μηχανισμός συμπερασμού σε πραγματικούς αριθμούς.

3.2. Χρήση συστημάτων ασαφούς λογικής

Η ασαφής λογική εφαρμόζεται σε συστήματα ελέγχου, στην επεξεργασία σήματος και εικόνας, στην αναγνώριση προτύπων, στις τηλεπικοινωνίες, στην ιατρική, την οικονομία, την διοίκηση επιχειρήσεων, στη ρομποτική, στους ανελκυστήρες, στην οικολογία, στην στατιστική, στην ψυχολογία, στο σύστημα φρένων ABS, στα αυτόματα κιβώτια ταχυτήτων στα αυτοκίνητα, στην διαστημική τεχνολογία και σε πολλές άλλες εφαρμογές.

4. Κεφάλαιο 4 : Υλοποίηση συστήματος ασαφούς λογικής για την μοντελοποίηση των μαθητών

4.1. Στόχος και προγράμματα

Σε αυτό το κεφάλαιο θα ασχοληθούμε με την μοντελοποίηση των μαθητών, σε ένα διαδραστικό και προσαρμοστικό σύστημα διδασκαλίας μαθηματικών, για παιδιά με αυτισμό, βασισμένο στο αναλυτικό πρόγραμμα σπουδών για μαθητές με αυτισμό του Υπουργείου Εθνικής Παιδείας και Θρησκευμάτων. Σκοπός είναι η δημιουργία ενός συστήματος ασαφούς λογικής για την μοντελοποίηση των μαθητών, το οποίο θα δέχεται σαν εισόδους τις απαντήσεις του μαθητή και θα δίνει σαν έξοδο το ποσοστό επί τις εκατό (%), της ολοκλήρωσης του κεφαλαίου, καθώς και έναν χαρακτηρισμό που θα αφορά την επίδοση του μαθητή στο κεφάλαιο (πχ. Μέτρια, Καλά, Άριστα). Για την δημιουργία του συστήματος ασαφούς λογικής χρησιμοποιήθηκε το fuzzy logic toolbox του MATLAB.

Το εκπαιδευτικό λογισμικό που δημιουργήθηκε περιέχει τρεις ενότητες, η πρώτη ενότητα έχει σαν στόχο την διδασκαλία των αριθμών 1-3, η δεύτερη την διδασκαλία των αριθμών 4-5 και η τελευταία την διδασκαλία της πρόσθεσης των αριθμών (1-5). Στο τέλος κάθε ενότητας υπάρχει ένα τεστ, το οποίο αποτελείται από τρεις ασκήσεις και σκοπό έχει την αξιολόγηση των μαθητών στην κατανόηση της ύλης της κάθε ενότητας. Για την μοντελοποίηση των μαθητών χρησιμοποιούνται οι απαντήσεις των μαθητών στις τρεις ασκήσεις του τεστ της κάθε ενότητας.

4.2. Βιβλιογραφική ανασκόπηση

Ο Alenka Kavcic παρουσίασε έναν τρόπο για την περιγραφή της αβεβαιότητας της γνώσης του χρήστη, ο οποίος χρησιμοποιείται για τη μοντελοποίηση της γνώσης του χρήστη σε ένα web-base προσαρμοστικό εκπαιδευτικό σύστημα χρησιμοποιώντας ασαφή λογική. Το μοντέλο χρησιμοποιεί ασαφή σύνολα για την αναπαράσταση της γνώσης και ασαφείς κανόνες για την ενημέρωση του μοντέλου [22]. Η Constanza Huaraya παρουσίασε ένα μοντέλο που αποτυπώνει την εξειδικευμένη γνώση των καθηγητών και χρησιμοποιείται για την σχεδίαση ενός διαγνωστικού μοντέλου που βασίζεται στην ασαφή λογική το οποίο αποτελείται από τρεις μεταβλητές εισόδου, μία μεταβλητή εξόδου και 27 ασαφείς κανόνες [23]. Ο Dongming Xu παρουσίασε ένα σύστημα ασαφούς λογικής για την παρουσίαση της κατάστασης των γνώσεων του μαθητή καθώς και για την εξατομίκευση του λογισμικού σε ένα online εκπαιδευτικό σύστημα [24]. Ο Neetesh Saxena δημιούργησε ένα σύστημα ασαφούς λογικής το οποίο βοηθάει στην ανάλυση των επιδόσεων των φοιτητών σε ένα εκπαιδευτικό ίδρυμα χρησιμοποιώντας δύο εισόδους (βαθμολογία, παρουσία) και μία έξοδο (επίδοση) [25]. Ο Rafael Pedraza-Jiménez χρησιμοποίησε ασαφή λογική για την μοντελοποίηση των Φοιτητών στην πλατφόρμα InterMediActor [26]. Ο Gagan Goel προτείνει ένα σύστημα ασαφούς λογικής για την αναπαράσταση της μοντελοποίηση των φοιτητών (στη γεωμετρία) και η σύγκριση του με τον αλγόριθμο συμπληρωματικού συντελεστή μοντέλου (AFM) [27]. Ο Zoran Sevarac παρουσιάζει ένα νευρο-ασαφές σύστημα που μπορεί να χρησιμοποιηθεί για τη μοντελοποίηση των φοιτητών. Το προτεινόμενο σύστημα επιτρέπει την ταξινόμηση των φοιτητών και βασίζεται σε ποιοτικές παρατηρήσεις των χαρακτηριστικών τους και μπορεί να χρησιμοποιηθεί σε εφαρμογές e-learning [28]. Η Alenka Kavcic ανέπτυξε ένα σύστημα ασαφούς λογικής για την μοντελοποίηση των μαθητών το οποίο βοηθάει τους μαθητές στην εξατομίκευση της πλοήγησης τους μέσα στο πρόγραμμα σπουδών [29]. Η Katerina Kabassi παρουσίασε μια θεωρία ασαφής πολυκριτηριακή λήψη αποφάσεων που έχει χρησιμοποιηθεί για την αξιολόγηση ενός συστήματος e-learning με συναισθηματικές δυνατότητες αλληλεπίδρασης σε ένα σύστημα ηλεκτρονικής μάθησης που διδάσκει μια γλώσσα προγραμματισμού [30]. Η Konstantina Chrysafiadi ασχολήθηκε με την αξιολόγηση της αποτελεσματικότητας και της ακρίβειας του μοντέλου του μαθητή ενός εκπαιδευτικού περιβάλλοντος web-based για τη διδασκαλία του προγραμματισμού ηλεκτρονικών υπολογιστών. Το μοντέλο του μαθητή αντιπροσωπεύει τη δημιουργία συστήματος αλληλεπίδρασης ανθρώπου υπολογιστή...

γνώση του μαθητή και χρησιμοποιεί την ασαφή λογική προκειμένου να καθοριστεί και να ενημερωθεί το επίπεδο των γνώσεων του μαθητή κάθε φορά που αυτός αλληλεπιδρά με το σύστημα e-learning [31]. Η Konstantina Chrysafiadi περιγράφει την μοντελοποίηση του μαθητή μίας εκπαιδευτικής web-based εφαρμογής που διδάσκει τη γλώσσα προγραμματισμού Pascal χρησιμοποιώντας τεχνικές ασαφούς λογικής [32]. Η Konstantina Chrysafiadi παρουσιάζει μια νέα προσέγγιση για την εξατομικευμένη διδασκαλία στην web-based εκπαίδευση στον τομέα των γλωσσών προγραμματισμού βασισμένη σε fuzzy cognitive maps (FCMs) [33].

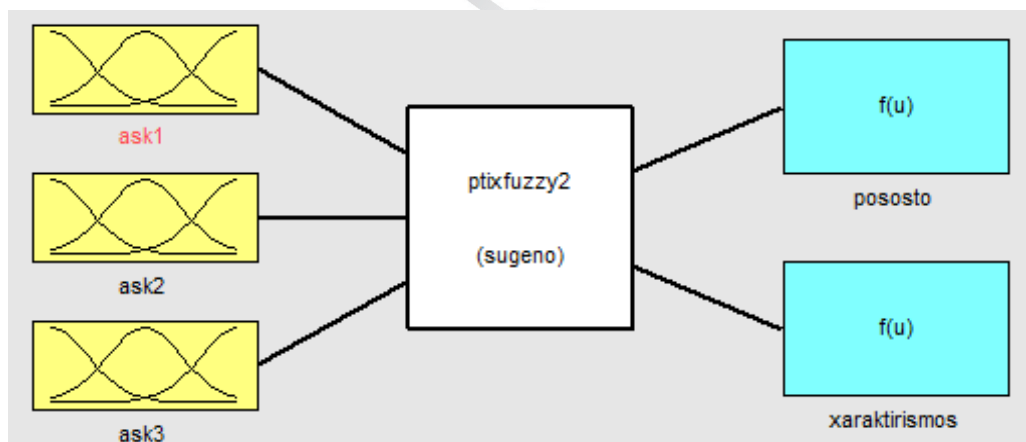
4.3. Υλοποίηση συστήματος ασαφούς λογικής στο MATLAB

Για την μοντελοποίηση των μαθητών δημιουργήθηκε σύστημα ασαφούς λογικής τύπου Takagi-Sugeno-Kang (TSK) ή αλλιώς Sugeno, μηδενικής τάξης, με ασαφοποιητή μονοσύνολο, μέθοδο συμπερασμού minimum (min), μέθοδο συνάθροισης maximum (max) και μέθοδο αποασαφοποίησης τον σταθμισμένο μέσο όρο (weighted average) με τύπο :

$$Output = \frac{\sum_{i=1}^N W_i Z_i}{\sum_{i=1}^N W_i}$$

Όπου N είναι το σύνολο των εισόδων, όπου W_i ο βαθμός πυροδότησης του i κανόνα και Z_i οι αντίστοιχες εξόδοι του ασαφούς συστήματος.

Το σύστημα ασαφούς λογικής που δημιουργήθηκε έχει τρεις εισόδους και δύο εξόδους όπως φαίνεται και στην εικόνα 4.1.

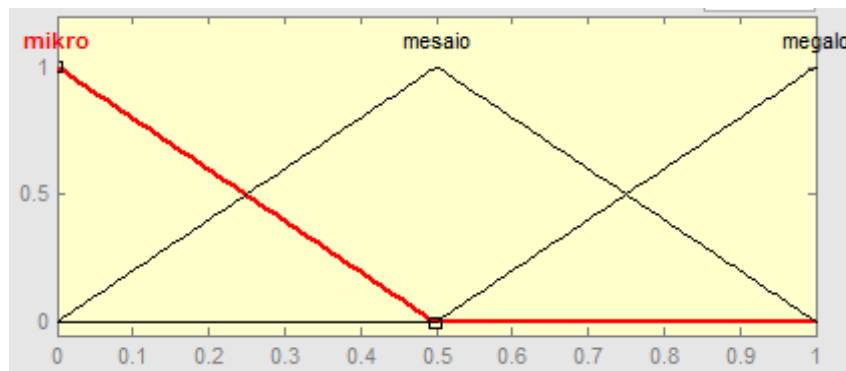


Εικόνα 4.1

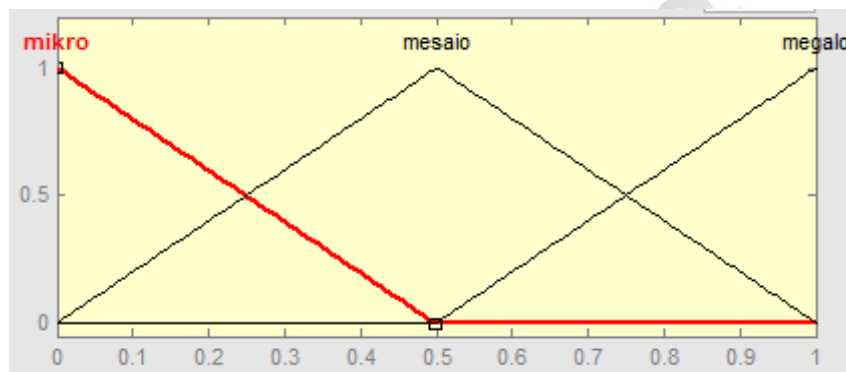
Οι εισόδοι (ask1, ask2, ask3) είναι τα ποσοστά ολοκλήρωσης των τριών ασκήσεων που έχει κάθε τεστ κανονικοποιημένα στην μονάδα.

Όπως βλέπουμε και στην εικόνα 4.1 το ασαφές σύστημα έχει δύο εξόδους. Η πρώτη έξοδος είναι το ποσοστό επί της εκατό (%) του τεστ και η δεύτερη είναι ένας χαρακτηρισμός (αριθμός 1-5) της επίδοσης του μαθητή για το τεστ πχ. Αν χαρακτηρισμός = 4 έχουμε “Καλά”, αν χαρακτηρισμός = 5 έχουμε “Άριστα” κτλ.

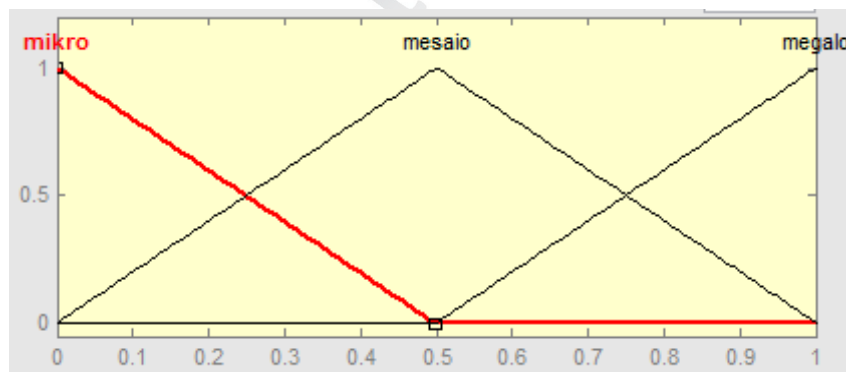
Παρακάτω φαίνονται οι συναρτήσεις συμμετοχής σε κάθε μία από της εισόδους :



Εικόνα 4.2 : Συναρτήσεις συμμετοχής της πρώτης εισόδου



Εικόνα 4.3 : Συναρτήσεις συμμετοχής της δεύτερης εισόδου



Εικόνα 4.4 : Συναρτήσεις συμμετοχής της τρίτης εισόδου

Όπως φαίνεται και στις παραπάνω εικόνες, οι γλωσσικές μεταβλητές που χρησιμοποιήθηκαν στις συναρτήσεις συμμετοχής των εισόδων είναι : μικρό, μεσαίο, μεγάλο.

Για την έξοδο “ποσοστό” οι συναρτήσεις συμμετοχής (membership functions) έχουν ως εξής :

Καθόλου = 0, Μικρό = 25, Μεσαίο = 50, Μεγάλο = 75, Πολύ Μεγάλο = 100

Για την έξοδο “χαρακτηρισμός” οι συναρτήσεις συμμετοχής (membership functions) έχουν ως εξής :

Τίποτα = 1, Λίγο = 2, Μέτρια = 3, Καλά = 4, Άριστα = 5

Οι 27 ασαφείς κανόνες που χρησιμοποιήθηκαν φαίνονται παρακάτω :

1. Αν (ask1 είναι μικρό) και (ask2 είναι μικρό) και (ask3 είναι μικρό) τότε (ποσοστό είναι καθόλου)(χαρακτηρισμός είναι τίποτα)
2. Αν (ask1 είναι μικρό) και (ask2 είναι μικρό) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι καθόλου)(χαρακτηρισμός είναι λίγο)
3. Αν (ask1 είναι μικρό) και (ask2 είναι μικρό) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
4. Αν (ask1 είναι μικρό) και (ask2 είναι μεσαίο) και (ask3 είναι μικρό) τότε (ποσοστό είναι καθόλου)(χαρακτηρισμός είναι λίγο)
5. Αν (ask1 είναι μικρό) και (ask2 είναι μεσαίο) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
6. Αν (ask1 είναι μικρό) και (ask2 είναι μεσαίο) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
7. Αν (ask1 είναι μικρό) και (ask2 είναι μεγάλο) και (ask3 είναι μικρό) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
8. Αν (ask1 είναι μικρό) και (ask2 είναι μεγάλο) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
9. Αν (ask1 είναι μικρό) και (ask2 είναι μεγάλο) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι καλά)
10. Αν (ask1 είναι μεσαίο) και (ask2 είναι μικρό) και (ask3 είναι μικρό) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
11. Αν (ask1 είναι μεσαίο) και (ask2 είναι μικρό) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
12. Αν (ask1 είναι μεσαίο) και (ask2 είναι μικρό) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
13. Αν (ask1 είναι μεσαίο) και (ask2 είναι μεσαίο) και (ask3 είναι μικρό) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
14. Αν (ask1 είναι μεσαίο) και (ask2 είναι μεσαίο) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
15. Αν (ask1 είναι μεσαίο) και (ask2 είναι μεσαίο) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι μεγάλο)(χαρακτηρισμός είναι καλά)
16. Αν (ask1 είναι μεσαίο) και (ask2 είναι μεγάλο) και (ask3 είναι μικρό) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
17. Αν (ask1 είναι μεσαίο) και (ask2 είναι μεγάλο) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μεγάλο)(χαρακτηρισμός είναι καλά)
18. Αν (ask1 είναι μεσαίο) και (ask2 είναι μεγάλο) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι πολύ_μεγάλο)(χαρακτηρισμός είναι καλά)
19. Αν (ask1 είναι μεγάλο) και (ask2 είναι μικρό) και (ask3 είναι μικρό) τότε (ποσοστό είναι μικρό)(χαρακτηρισμός είναι λίγο)
20. Αν (ask1 είναι μεγάλο) και (ask2 είναι μικρό) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
21. Αν (ask1 είναι μεγάλο) και (ask2 είναι μικρό) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι καλά)
22. Αν (ask1 είναι μεγάλο) και (ask2 είναι μεσαίο) και (ask3 είναι μικρό) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι μέτρια)
23. Αν (ask1 είναι μεγάλο) και (ask2 είναι μεσαίο) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι μεγάλο)(χαρακτηρισμός είναι καλά)
24. Αν (ask1 είναι μεγάλο) και (ask2 είναι μεσαίο) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι πολύ_μεγάλο)(χαρακτηρισμός είναι καλά)
25. Αν (ask1 είναι μεγάλο) και (ask2 είναι μεγάλο) και (ask3 είναι μικρό) τότε (ποσοστό είναι μεσαίο)(χαρακτηρισμός είναι καλά)
26. Αν (ask1 είναι μεγάλο) και (ask2 είναι μεγάλο) και (ask3 είναι μεσαίο) τότε (ποσοστό είναι πολύ_μεγάλο)(χαρακτηρισμός είναι καλά)
27. Αν (ask1 είναι μεγάλο) και (ask2 είναι μεγάλο) και (ask3 είναι μεγάλο) τότε (ποσοστό είναι πολύ_μεγάλο)(χαρακτηρισμός είναι άριστα)

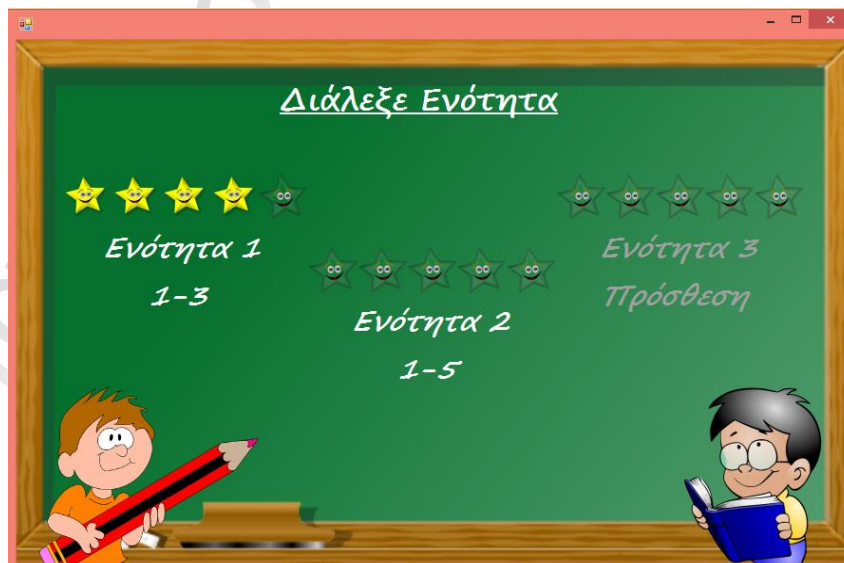
4.4. Χρήση στην VB.NET για την μοντελοποίηση των μαθητών

Η ολοκλήρωση του διαγωνίσματος κάθε ενότητας συνοδεύεται από την φόρμα που απεικονίζεται στην εικόνα 4.5. Πιο συγκεκριμένα απεικονίζονται ενδεικτικοί στόχοι η οποίοι συνοδεύονται από σχηματική αναπαράσταση (κίτρινα αστέρια) του πλήθους των σωστών απαντήσεων. Τα αποτελέσματα αυτά αποθηκεύονται στη βάση δεδομένων. Αναλυτικότερα αποθηκεύονται μεμονωμένα για κάθε στόχο στον πίνακα *exercises* ενώ τα συνολικά στον πίνακα *session* (ποσοστό %) και χαρακτηρισμός (αριθμός 1-5) στη βάση δεδομένων.



Εικόνα 4.5

Κατά την είσοδο του μαθητή στο λογισμικό, αντλείται η συνολική του επίδοση σε κάθε ενότητα από την βάση δεδομένων από την βάση δεδομένων, το οποίο είναι η έξοδος του ασαφούς συστήματος για την μοντελοποίηση των μαθητών που περιγραφικέ αναλυτικά προηγούμενος. Αν το ποσοστό επιτυχίας του είναι μεγαλύτερο από 60%, "ξεκλειδώνει" και η επόμενη ενότητα όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 4.6

Επίλογος

Η συγκεκριμένη μεταπτυχιακή διατριβή αποτελεί ένα κομμάτι ενός εκπαιδευτικού λογισμικού που δημιουργήθηκε για την εκπαίδευση μαθητών με αυτισμό στα μαθηματικά και βασίζεται στο αναλυτικό πρόγραμμα σπουδών για μαθητές με αυτισμό, του Υπουργείου Εθνικής Παιδείας και Θρησκευμάτων και πιο συγκεκριμένα αφορά τα τρία πρώτα κεφάλαια δηλαδή την εκμάθηση των αριθμών 1-3, την εκμάθηση των αριθμών 1-5 καθώς και την πράξη της πρόσθεσης. Η μεταπτυχιακή διατριβή αυτή ασχολείται με την διδασκαλία του σωστού τρόπου γραφής των αριθμών καθώς και με την μοντελοποίηση των μαθητών. Το κομμάτι της αναγνώρισης των χειρόγραφων χαρακτήρων αποτελεί ένα ολοκληρωμένο λογισμικό το οποίο θα μπορούσε να ενσωματωθεί σε οπουδήποτε εφαρμογή (για εκπαίδευση ή ψυχαγωγία) είναι γραμμένη στην γλώσσα προγραμματισμού VB.NET καθώς και σε visual C# και visual C++, κάνοντας τις απαραίτητες τροποποιήσεις.

Βιβλιογραφία

- [1] Καρυπίδη Μ. (2012), Η σημασία της μελέτης της αλληλεπίδρασης Ανθρώπου-Υπολογιστή (Human-Computer Interaction) κατά την κατασκευή ενός web-site ή μιάς καμπάνιας e-marketing, Πτυχιακή Εργασία, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης-Σχολή Διοίκησης και Οικονομίας-Τμήμα Εμπορίας και Διαφήμισης
- [2] Αβούρης, Ν. (2000), Εισαγωγή στην Επικοινωνία Ανθρώπου-Υπολογιστή. Αθήνα: Δίαυλος
- [3] Ιωαννίδης, Γ. και Λέπουρας, Γ. (2004-2005) : Σημειώσεις Επικοινωνίας Ανθρώπου-Μηχανής. Αθήνα: Εθνικό Και Καποδιστριακό Πανεπιστήμιο Αθηνών Τμήμα Πληροφορικής και Τηλεπικοινωνιών
- [4] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh and Mo Nours Arab, Human-Computer Interaction: Overview on State of the Art, MARCH 2008
- [5] Sebastian Stoliński, Wojciech Bieniecki, Application of OCR systems to processing and digitization of paper documents, Technical University of Łódź, Poland
- [6] Καλλιτάκης Γ. (2012), Αυτόματη Αναγνώριση Αντικειμένων Εικόνας, Πτυχιακή Εργασία, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης, Σχολή Τεχνολογικών Εφαρμογών Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων
- [7] Νικόλαος Ν. (2011), Ανάπτυξη εξελιγμένων αλγορίθμων για την ψηφιακή επεξεργασία και ανάλυση κυτταροπαθολογικών εικόνων με χρήση πλατφόρμας MATLAB, Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Συστημάτων Μετάδοσης Πληροφορίας και Τεχνολογίας Υλικών
- [8] MATLAB Help file, MATLAB Version 7.12.0 (R2011a with Image Processing toolbox), The MathWorks Inc.
- [9] Πέτρου Δ. (2013), Οπτική Αναγνώριση Χαρακτήρων (OCR). Περιγραφή ενός νέου αλγορίθμου και κατασκευή εφαρμογής Η/Υ που τον υλοποιεί για το σκοπό της Αυτόματης Αναγνώρισης Πινακίδων Κυκλοφορίας (ANPR), Πτυχιακή εργασία, Αλεξάνδρειο Τ.Ε.Ι. Θεσσαλονίκης, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Πληροφορικής
- [10] Παπαθανασίου Α. (2010), Μελέτη, σχεδιασμός και υλοποίηση αλγορίθμων εντοπισμού θέσης και αναγνώρισης χαρακτήρων σε τυπωμένες εικόνες, Διπλωματική Εργασία, Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών
- [11] Ray Smith (2007), An Overview of the Tesseract OCR Engine, Google Inc.
- [12] Ray Smith, Google Inc OSCON 2007, Πηγή από το Διαδίκτυο : <https://code.google.com/p/tesseract-ocr/downloads/detail?name=TesseractOSCON.pdf>
- [13] Lisa Anthony, Jie Yang, Kenneth R. Koedinger (2007), Adapting Handwriting Recognition for Applications in Algebra Learning, Human-Computer Interaction Institute Carnegie Mellon University
- [14] Bruce R. Maxim, Nilesh V. Patel, Nicholas D. Martineau, and Mark Schwartz (2007), Work In Progress - Learning via Gaming: An Immersive Environment for Teaching Kids Handwriting
- [15] Zhi-Hui Hu, Yun Xu, Liu-Sheng Huang and Howard Leung (APRIL 2009), A Chinese Handwriting Education System with Automatic Error Detection
- [16] Mohd Nizam Bin Saad, Abd Hadi Abd Razak and Azman Yasin (February 2012), The Adaptation of Handwriting Recognition System User Interface in Preschool Literacy Learning Courseware
- [17] Διαδίκτυο : <https://code.google.com/p/tesseract-ocr>
- [18] "Fuzzy Logic", Stanford Encyclopedia of Philosophy, Stanford University.
- [19] Pelletier, Francis Jeffry (2000). "Review of Metamathematics of fuzzy logics"
- [20] Π. Σπαγάκος, Ν. Σαμπαθιανάκης, Πτυχιακή εργασία, Τ.Ε.Ι Κρήτης, Σχολή Εφαρμοσμένων Επιστημών, Τμήμα ηλεκτρονικών Μηχανικών ΤΕ
- [21] Α. Ντούνης (2013) : Σημειώσεις Ευφυσούς Ελέγχου, ΤΕΙ Πειραιά, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών Αυτοματισμού ΤΕ, Τομέας Βιομηχανικής Πληροφορικής

- [22] Alenka Kavcic, Dealing with Uncertainty of User Knowledge: Fuzzy User Modelling in Educational Hypermedia, Faculty of Computer and Information Science University of Ljubljana
- [23] Constanza Huapaya, Proposal of Fuzzy Logic-based Students' Learning Assessment Model, Facultad de Ingeniería, UNMDP, Argentina
- [24] Dongming Xu, Huaqing Wang and Kaile Su (2002), Intelligent Student Profiling with Fuzzy Models, Department of Information Systems City University of Hong Kong Kowloon, Hong Kong
- [25] Neetesh Saxena, Kajal Kaushal Saxena (2010), Fuzzy Logic Based Students Performance Analysis Model for Educational Institutions, IMS Engg College, Ghaziabad (UP)-INDIA
- [26] Alenka Kavcic, Rafael Pedraza-Jiménez, Harold Molina-Bulla, Francisco J. Valverde Albacete, Jesús Cid-Sueiro, Angel Navia-Vázquez, Fuzzy Student Model in InterMediActor Platform
- [27] Gagan Goel, Sébastien Lallé, and Vanda Luengo (2011), Fuzzy Logic Representation for Student Modelling Case Study on Geometry
- [28] Zoran Sevarac, Neuro Fuzzy Reasoner for Student Modeling, Department of Information Systems, School of Business Administration FON, University of Belgrade, Serbia and Montenegro
- [29] Alenka Kavcic, Angel Navia-Vazquez, Rafael Pedraza-Jimenez (2003), Student Modelling Based on Fuzzy Inference Mechanisms
- [30] Katerina Kabassi, Efthimios Alepis, Maria Virvou (2011), Evaluating an Affective e-Learning System using a Fuzzy Decision Making Method
- [31] Konstantina Chrysafiadi, Maria Virvou (2012), Evaluating the integration of fuzzy logic into the student model of a web-based learning environment, Department of Informatics, University of Piraeus
- [32] Konstantina Chrysafiadi, Maria Virvou (2010), Modeling Student's Knowledge on Programming Using Fuzzy Techniques, Department of Informatics, University of Piraeus
- [33] Konstantina Chrysafiadi, Maria Virvou (2014), Fuzzy Logic for Adaptive Instruction in an E-learning Environment for Computer Programming, Department of Informatics, University of Piraeus