

**ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ**

**ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ**

**ΔΙΔΑΣΚΩΝ ΚΑΘΗΓΗΤΗΣ: ΤΣΑΛΙΚΑΚΗΣ ΔΗΜΗΤΡΙΟΣ**

**ΕΠΙΜΕΛΕΙΑ:**

**ΡΟΜΑΝ ΑΥΤΑΝΤΙΛΙΔΗΣ 295**

**ΘΟΔΩΡΗΣ ΜΠΑΗΣ 333**

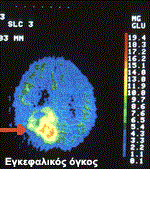
**6ο ΕΞΑΜΗΝΟ**

**Brain tumor segmentation**

ABSTRACT

This project deals with a special kind of image segmentations, usually used in medicine, well known as brain tumor segmentation. The process is being initialized by the main user, who sets the preferable image to be processed. After this, the initial image is converted to a grayscale image to minimize the consumption of system resources. The program continues its execution with an image threshold and a minimum object size to affect the initial image, both of them given as input by user. Before user is asked to input the index area to be counted as a pixel area, we set a normalization, because there are still some nosy and spam objects in our image. After all, we get the picture’s number of pixels followed by a final graphic display of the tumor region on the initial image. To sum up, this application refers to all classes of users, even to them who don’t have any idea of matlab; this attribute is excused by the strong feedback between user and application, and by the general purpose of the application; we can use it to find other kinds of tumor as well. We also made a video to help everyone who can’t handle a problem with this project.

ΓΕΝΙΚΗ ΑΝΑΦΟΡΑ ΣΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΜΕΛΕΤΗΣ

Ένας εγκεφαλικός όγκος, είναι οποιοσδήποτε ενδοκρανιακός όγκος που χαρακτηρίζεται από ανεξέλεγκτη διαίρεση κυττάρων, κανονικά είτε μέσα στον εγκέφαλο, στα κρανιακά νεύρα, στις μεμβράνες που καλύπτουν τον εγκέφαλο (μήνιγγες), στο κρανίο, στο βλεννογόνο και κωνοειδή αδένα ή μπορεί να εξαπλωθεί από κάποιον καρκίνο που εντοπίζεται σε άλλο μέρος του σώματος (μεταστατικοί όγκοι).

Χωρίζονται σε δύο κατηγορίες, τους καλοήθεις και τους κακοήθεις, ανάλογα με την ταχύτητα ανάπτυξής τους.

Ακόμη, το μέγεθος του εγκεφαλικού όγκου παίζει καθοριστικό ρόλο, τόσο στο είδος των συμπτωμάτων που θα προκληθούν, όσο και στη θεραπευτική αντιμετώπιση.

ΓΕΝΙΚΗ ΑΝΑΦΟΡΑ ΣΤΟ ΘΕΜΑ ΤΗΣ ΕΡΓΑΣΙΑΣ

Μιλώντας για τμηματοποίηση εικόνων ή αλλιώς segmentation αναφερόμαστε στη διατήρηση μόνο μερικών σημαντικών θέσεων στην περιοχή της αρχικής εικόνας. Αυτό γίνεται είτε για να εξομαλύνουμε οπτικά μια εικόνα,η οποία παρουσιάζει τυχόν ανωμαλίες σε ορισμένα σημεία, είτε επειδή απαιτούμε μόνο ένα κομμάτι της αρχικής, οπότε είναι απαραίτητο μέσω διάφορων γνωστών τεχικών(π.χ. κατωφλίωση), να αφαιρέσουμε τις πλειάδες των περιττών pixels, γεγονός που όχι μόνο βοηθάει εμάς για την εκάστοτε επεξεργασία που θέλουμε να κάνουμε πάνω στην εικόνα, αλλά και δεν επιβαρύνει το περαιτέρω πόρους συστήματος, οι οποίοι σε διαφορετική περίπτωση θα χρησιμοποιούνταν άσκοπα.

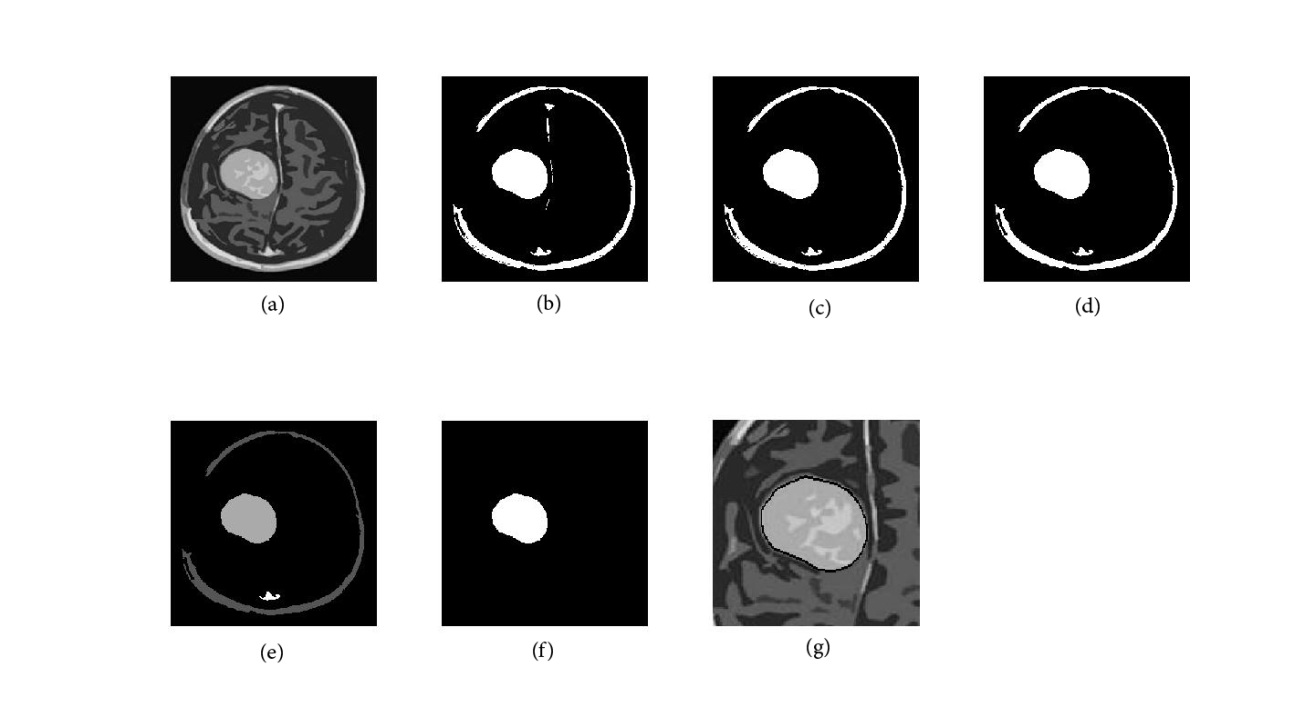
ΣΤΟΧΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Το αρχικό ζητούμενο της παρούσας εργασίας ήταν η ποσοτικοποίηση και η εμφανής διαχωρισιμότητα του όγκου μιας αρχικής εικόνας.

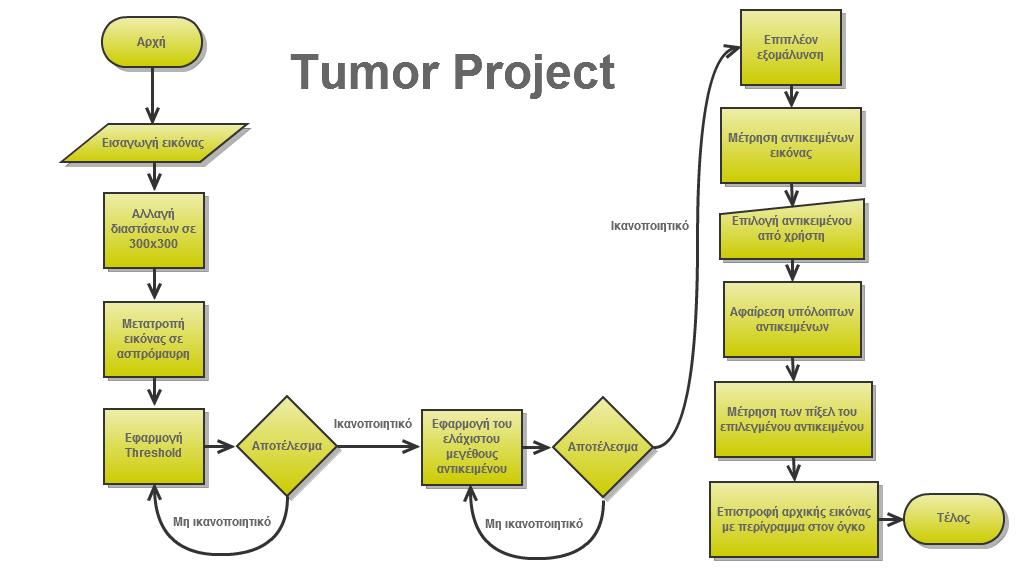
ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΕ ΜΟΡΦΗ ΨΕΥΔΟΚΩΔΙΚΑ

Αρχικά, ο χρήστης εισάγει την εικόνα που επιθυμεί να επεξεργαστεί, η οποία κατευθείαν μετατρέπεται σε ασπρόμαυρη. Ακολουθεί η εισαγωγή τιμής επιθυμητού κατωφλίου από το χρήστη, δηλαδή το όριο το οποίο πρόκειται να καθορίσει ποια pixels θα παραμείνουν και στο επόμενο στιγμιότυπο. Έπειτα, πάλι ο χρήστης εισάγει το επιθυμητό ελάχιστο μέγεθος αντικειμένου και έπειτα από μια εξομάλυνση που γίνεται στο τελευταίο στιγμιότυπο, ξεκινά η διαδικασία υπολογισμού των pixels του όγκου, αφού πρώτα ο χρήστης έχει ορίσει μέσω του data cursor την περιοχή που θέλει να δουλέψει η εφαρμογή.

Καθένα από τα ξεχωριστά στάδια παρουσιάζεται με ένα αυτόνομο στιγμιότυπο, ενώ τελικά, εμφανίζεται και η αρχική εικόνα με την περιοχή στην οποία εκτείνεται ο όγκος να παρουσιάζεται περιγεγραμμένη. Παραθέτουμε ένα παράδειγμα εκτέλεσης της εφαρμογής για την εικόνα br\_tumor1.jpg(η συγκεκριμένη, καθώς και οι br\_tumor2.jpg, br\_tumor3.jpg επισυνάπτονται στο συμπιεσμένο αρχείο της εργασίας) :



Ακολουθεί η παρουσίαση της εφαρμογής με τη μορφή διαγράμματος ροής :



ΠΕΡΙΓΡΑΦΗ ΚΩΔΙΚΑ

*Στο σημείο αυτό, θα παρουσιάσουμε τον τρόπο λειτουργίας της εφαρμογής περιγράφοντας τον κώδικά της και παράλληλα αναλύωντας κάθε σημείο, το οποίο θεωρούμε ότι είναι απαραίτητο να διασαφηνιστεί(παραθέτοντας ταυτόχρονα τα αντίστοιχα στιγμιότυπα, ώστε να μην δημιουργθούν τα παραμικρά λογικά κενά στον αναγνώστη που έρχεται πρώτη φορά σε επαφή με την εφαρμογή).*

Αρχικά ο χρήστης πληκτρολογεί το όνομα της εικόνας που πρόκειται να επεξεργαστεί, το οποίο εισάγεται μέσω της συνάρτησης input στη μεταβλητή name :



Aκολουθεί ο διαχωρισμός του ονόματος της εικόνας που εισάγει ο χρήστης, με τέτοιο τρόπο, ώστε να λαμβάνουμε ανεξάρτητα και το όνομα της εικόνας, αλλά και τον τύπο της. Η παραπάνω διαδικασία πραγματοποιηθήκε με τη συνάρτηση strread(x, ‘type type’, ‘delimiter’, ‘.’ ) , η οποία παίρνει σαν ορίσματα μια μεταβλητή(στη συγκεκριμένη περίπτωση, τη μεταβλητή στην οποία έχουμε αποθηκεύσει το όνομα της εικόνας που εισήγαγε ο χρήστης, του τύπους δεδομένων που αντιστοιχούν στις δύο νέες μεταβλητές που θα δημιουργήσουμε(τις βάζουμε σε έναν 2\*1 πίνακα), καθώς και έναν delimiter, ο οποίος στην προκειμένη περίπτωση θέλουμε να είναι η τελεία (‘.’). Με αυτό τον τρόπο, καταφέρνουμε να εξαλείψουμε κάθε ποσοστό πιθανής αποτυχίας εισαγωγής ονόματος εικόνας, καθώς, εφόσον, κάθε εικόνα ονοματίζεται έχοντας σαν κατάληξη τον τύπο εικόνας στον οποίο ανήκει, ο χρήστης δεν έχει να κάνει κάποια περαιτέρω ενέργεια, αφού αυτοματοποιήθηκε η όλη διαδικασία διαχωρισμού μέσω της παραπάνω συνάρτησης :



Διαβάζουμε, λοιπόν, την εικόνα(μέσω της γνωστής μας imread() ) και μέσω της imresize() με αρχική παράμετρο την ήδη δοθείσα εικόνα, αλλάζουμε τις διαστάσεις της σε 300\*300 :



Η όλη διαδικασία εξυπηρετεί την ισχύ της εφαρμογής για κάθε νέα εικόνα. Έτσι, ουσιαστικά ο χρήστης δεν χρειάζεται να παρέμβει καθόλου στον κώδικα, καθώς όλα γίνονται αυτοματοποιημένα.

Ακολουθεί η μετατροπή της εικόνας σε grayscale κλίμακα, προκειμένου να γίνει ευκολότερα η κατωφλίωση :

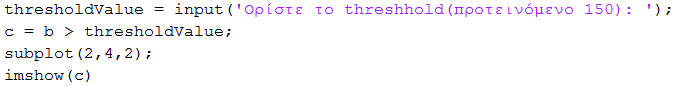


Στο σημείο αυτό, έχουμε μια ενέργεια, η οποία επαναλαμβάνεται πριν από κάθε εμφάνιση στιγμιοτύπου επεξεργασίας της αρχικής εικόνας, που ουσιαστικά εξυπηρετεί τη διαδοχική παράθεση στιγμιοτύπων επεξεργασίας της αρχικής εικόνας σε ένα κοινό επίπεδο(οπότε την παρουσιάζουμε σε αυτό το σημείο μόνο και όχι σε κάθε σημείο που εμφανίζεται). Πρόκειται για τη γνωστή μας από το εργαστήριο subplot(), η οποία σύμφωνα με τις παραμέτρους που έχουμε δώσει, ρυθμίζεται να μειώνει κάθε φορά στο μισό, το μέγεθος εικόνων, καθώς και να τις παρουσιάζει διαδοχικά κατά τον Χ-άξονα(έτσι, έπειτα από την αρχική παραμετροποίηση, για κάθε νέα εμφάνιση, απλώς αυξάνουμε κατά 1 μονάδα την τελευταία παράμετρό της) :

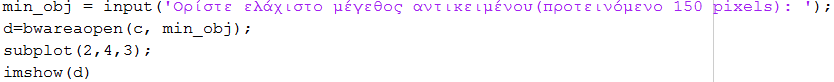


*Ακολουθούν 2 διαδοχικές καταστάσεις αλληλεπίδρασης της εφαρμογής με το χρήστη:*

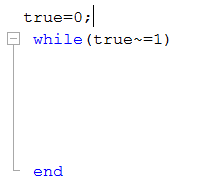
Αρχικά τίθεται στη μεταβλητή thresholdValue, η προτιμόμενη από το χρήστη τιμή κατωφλίου κι έπειτα αποθηκεύονται στη μεταβλητή-πίνακα c οι τιμές, οι οποίες επαληθεύουν την άτυπη ανίσωση, δηλαδή έχουν τιμή μεγαλύτερη της επιλεγμένης κατωφλίου(ακολουθεί εμφάνιση του αντίστοιχου στιγμιοτύπου) :



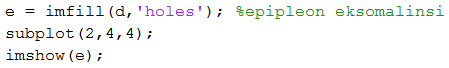
Ορίζουμε στη μεταβλητή min\_obj το ελάχιστο επιθυμητό μέγεθος αντικειμένου και μέσω της συνάρτησης bwareaopen(), η οποία εξομαλύνει την εικόνα που βρίσκεται στη μεταβλητή c από τυχόν ανωμαλίες εμφάνισης(σελίδα 3, εικόνα (b)) θέτουμε d τη νέα εικόνα-πίνακά μας :



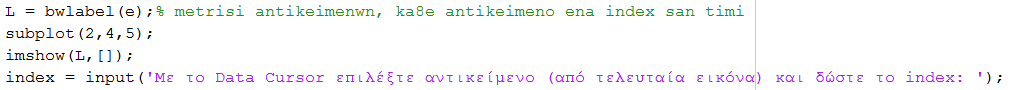
*Να σημειωθεί ότι, στις 2 παραπάνω ενέργειες, προς αποφυγή τυχόν μη επιτρεπτών τιμών εισόδου, καθώς και άρση απόφασης ορισμού τιμών εισόδου επιλέξαμε να τοποθετήσουμε ενα flag(σημαία ελέγχου), έτσι ώστε μόνο όταν δωθεί έγκυρη τιμή και είναι σίγουρος ο χρήστης ότι θέλει να συνεχίσει, να προχωρήσει η εφαρμογή :*



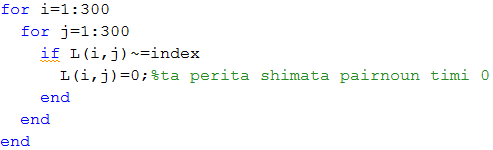
Ακολουθεί μια επιπλέον εξομάλυνση της d, γεμίζοντας όσα κενά έχουν απομείνει(π.χ. διάσπαρτες κουκίδες) και εμφάνιση της βελτιωμένης εικόνας :



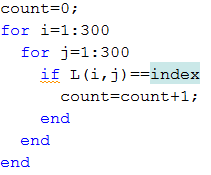
Είναι η στιγμή που πρέπει να θέσουμε την προς υπολογισμό pixels περιοχή, κάτι που ουσιαστικά θα γίνει επιλέγοντας μέσω data cursor(στο αντίστοιχο figure) τη συγκεκριμένη περιοχή και εισάγοντας το εμφανιζόμενο index στη δική μας μεταβλητή index:



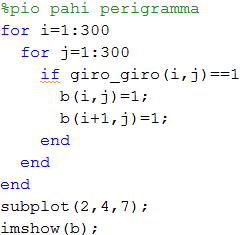
Έτσι, χρειαζόμαστε ένα for-loop, το οποίο θα συγκρατεί τις τιμές που βρίσκονται εντός της επιθυμητής περιοχής, μετατρέποντάς τες σε 1, ενώ τις υπόλοιπες σε 0 :



Για να μετρήσουμε, όμως τα συνολικά pixels της επιλεχθείσας περιοχής, πρέπει μέσω ενος αθροιστή να αθροίσουμε όλους τους συνδυασμούς i,j που ικανοποιούν τη μεταβλητή index, δηλαδή που βρίσκονται εντός της παραπάνω περιοχής :



Εμφανίζουμε το μέγεθος όγκου μέσω της fprintf και στη συνέχεια εφηύραμε το εξής τέχνασμα για λίγο πιο παχύ περίγραμμα της περιοχής του όγκου : αφού θέσαμε  , «μαυρίσαμε» κάθε pixel της συγκεκριμένης περιοχής, καθώς και το δεξί διπλανό του, προκειμένου να γίνει ελάχιστα εντονότερη(περισσότερο ορατή δηλαδή) η περιγραμματοποίηση του όγκου στην αρχική εικόνα(καθώς αυτή εμφανίζουμε στο τέλος) :



ΣΥΜΠΕΡΑΣΜΑΤΑ

Αρχικά, αξίζει να σημειώσουμε το γεγονός ότι, εκτιμούμε πως οι αρχικοί στόχοι καλύφθηκαν στο έπακρο, καθώς η εφαρμογή χαρακτηρίζεται από :

* Λειτουργικότητα
* Αποδοτικότητα
* Φιλικότητα προς το χρήστη
* Μεταφερσιμότητα

Ειδικότερα, το 3ο στοιχείο καλύπτει ακόμη και την περίπτωση που κάποιος χρήστης δεν έχει καμία ιδέα από matlab, καθώς τον καθοδηγεί μέσω κατάλληλων ελέγχων για κάθε ένα στάδιο μετασχηματισμού της αρχικής εικόνας-υπάρχει και σχετικό βοηθητικό βίντεο στα συνημμένα της εφαρμογής. Ακόμη, έτσι καλύπτει και την περίπτωση που δοθεί κάποια μη λειτουργική για την εκάστοτε εισαχθείσα εικόνα-ακτινογραφία παράμετρος, δίνοντας του τη δυνατότητα να αλλάξει την αρχική επιλογή της παραμέτρου.

Το 4ο στοιχείο δικαιολογείται από το γεγονός ότι η εφαρμογή δημιουργήθηκε με τέτοιο τρόπο, ώστε να μπορεί να αποτελέσει μια ξεχωριστή οντότητα κάποιου προγράμματος(ή και σαν απλή αυτόνομη εφαρμογή εξυπηρέτησης ιατρών),π.χ. μια συνάρτηση, καθώς η οποιαδήποτε επεξεργασία που γίνεται μέσω αυτής, λαμβάνει χώρα σε μια εικόνα-ακτινογραφία την οποία εισάγει ο χρήστης σαν παράμετρο, κάτι που σημαίνει ότι μπορεί να χρησιμοποιηθεί και για εύρεση όγκου και στα υπόλοιπα σημεία του σώματος.

Εφόσον, λοιπόν, η υλοποίηση του αρχικού ζητούμενου ήταν επιτυχής, προχωρήσαμε στις παρακάτω προσθήκες-βελτιώσεις της εφαρμογής, οι οποίες τελικά συνετέλεσαν στην υποστήριξη και των 5 προαναφερθέντων γνωρισμάτων της. Οι επιπλέον υλοποιήσεις της εφαρμογής μας περιγράφονται παρακάτω :

1. Εισαγωγή εικόνας από το χρήστη
2. Αυτοματοποιημένος διαχωρισμός ονόματος από τύπο εικόνας
3. Χρήση συγκεκριμένου μεγέθους επεξεργασίας εισαγόμενης εικόνας
4. Προβολή διαδοχικών σταδίων της επεξεργασμένης εικόνας, στο ίδιο επίπεδο
5. Εισαγωγή flags(για καλύτερο έλεγχο του χρήστη)

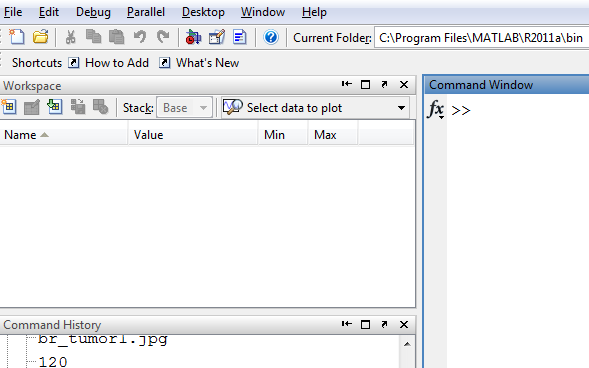
ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ

Θα δώσουμε ένα δικό μας παράδειγμα χρήσης της εφαρμογής.

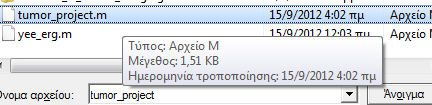
Αρχικά, προσέχουμε, ώστε όλος ο φάκελος του προγράμματός μας να βρίσκεται στο απόλυτο path του workspace της matlab :



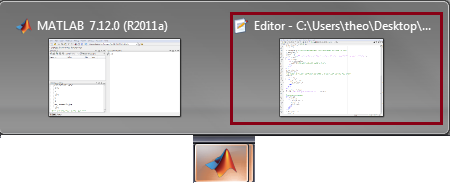
Αφού βεβαιωθούμε(ή έστω προσθέσουμε το απόλυτο μονοπάτι στο οποίο βρίσκεται ο φάκελος της εφαρμογής μας, στους χώρους εργασίας της matlab), προχωρούμε σε άνοιγμα της matlab.

Ανοίγουμε τη matlab : 

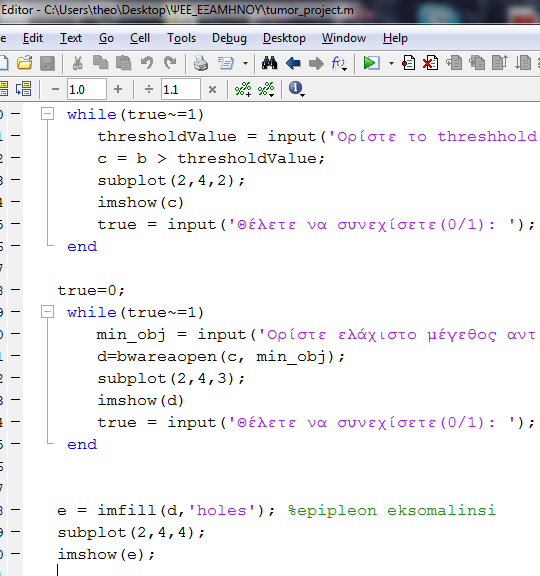
Ανοίγουμε το αρχείο tumor\_project.m, από το μενού File🡪Open :



Παρατηρούμε στη γραμμή εργασιών ότι έχει ανοίξει ένα νέο παράθυρο matlab :



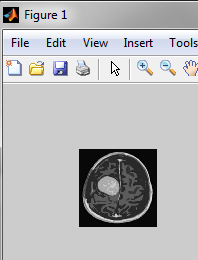
Μεταφερόμαστε σε αυτό το παράθυρο και παρατηρούμε τον κώδικα της εφαρμογής(παραθέτουμε ένα κομμάτι του) :



*Το νέο παράθυρο έχει ανοίξει με τον editor της matlab, καθώς, δημιουργήσαμε μια γενικού σκοπού εφαρμογή, δηλαδή πρόκειται για συνάρτηση της matlab( το υποδηλώνει άλλωστε και η κατάληξη .m  )*

Πατάμε F5 για να ξεκινήσει η εφαρμογή, επιλόγοντας “Add to path” από το εμφανιζόμενο πλαίσιο διαλόγου(όπως διευκρινίσαμε νωρίτερα) και παρατηρούμε την εφαρμογή να έχει ξεκινήσει ζητώντας όνομα εικόνας :



Δίνουμε “br\_tumor1.jpg” και παρατηρούμε την φόρτωση της εικόνας στο Figure 1 :

Η εφαρμογή συνεχίζει ζητώντας μας να ορίσουμε τιμή κατωφλίου, γνωστοποιώντας μας ότι το προτεινόμενο κατώφλι έχει τιμή 150. Εμείς ας δώσουμε 120 για το συγκεκριμένο παράδειγμα :



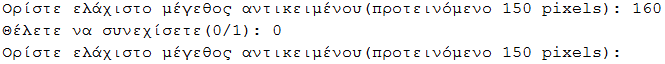
Στο συγκεκριμένο σημείο τοποθετήσαμε τον πρώτο από τους ελέγχους που προαναφέραμε, για μεγαλύτερη αποδοτικότητα και αξιοπιστία της εφαρμογής. Έτσι, ο χρήστης πατάει το «1» αν είναι σίγουρος για την τιμή που νωρίτερα εισήγαγε, ή το 0 αν θέλει να εισάγει ξανά την τιμή. Εμείς θα πατήσουμε το «1» :



Παρατηρούμε, λοιπόν, ότι η εφαρμογή συνεχίζει, ζητώντας μας να ορίσουμε το ελάχιστο μέγεθος αντικειμένου, πάλι προτείνοντας το 150, ενώ παράλληλα έχει εμφανιστεί και η κατωφλιομένη εκδοχή της αρχικής εικόνας, ακριβώς δίπλα από αυτή, για λόγους καλύτερης κατανόησης της λειτουργικότητας της(και αυτό γίνεται για κάθε νέα επεξεργασία της εικόνας) :



Προχωρούμε στην εισαγωγή ελαχίστου μεγέθος αντικειμένου, δημιουργώντας ουσιαστικά ένα λάθος, δηλαδή αντί για 150, δίνουμε 160. Συνειδητοποιούμε ότι δεν ήταν αυτή η τιμή που θέλαμε να δώσουμε. Επομένως, στη συγκεκριμένη ερώτηση, πατάμε «0», ώστε να μας δοθεί ξανά η δυνατότητα να εισάγουμε νέα τιμή :

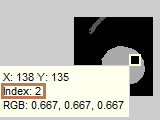


Παρατηρούμε το νέο γράφημα, ακολουθούμενο από την επιπλέον εξομάλυνση που κάνει μόνη της η εφαρμογή :



Δίνουμε 150 και έπειτα «1», ώστε να συνεχίσουμε. Μας εμφανίζεται το ακόλουθο μήνυμα :

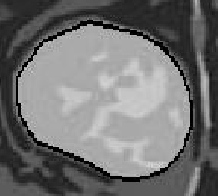


Οπότε, μεταφερόμαστε στο Figure 1-αν θέλουμε κάνουμε όσο zoom in χρειαζόμαστε, ώστε να διευκολυνθούμε στην επιλογή, μέσω των αντίστοιχων κουμπιών του παραθύρου  - παραθέτουμε και το δικό μας zoom in : )και με το Data Cursor (  )επιλέγουμε την περιοχή της οποίας τον όγκο επιθυμούμε να υπολογίσουμε(στη συγκεκριμένη περίπτωση μιλάμε για τη γκρι περιοχή), ώστε να μας εμφανιχτεί η τιμή index της.

Εισάγουμε τη συγκεκριμένη τιμή στο πλαίσιο διαλόγου της εφαρμογής(«2» στην προκειμένη περίπτωση). Παρατηρούμε ότι όντως εμφανίζεται το μέγεθος του όγκου:



Τέλος, μεταφερόμαστε ξανά στο Figure 1 και παρατηρούμε ότι η περιοχή του όγκου έχει γίνει πλέον απόλυτα διαχωρίσιμη, τόσο στο στην index περιοχή του τελευταίου γραφήματος (  ), όσο και στην αρχική εικόνα :



*Είναι πιθανόν στην προκειμένη περίπτωση να μην είναι 100% ορατό(γι’ αυτό άλλωστε μαυρίσαμε και το κάθε διπλανό οριακό pixel, προκειμένου να φαίνεται λίγο πιο καθαρά στην τελική εικόνα ο όγκος), καθώς μικρύναμε την κλίμακα εμφάνισης γραφήματος, ώστε μέσω subplot να εμφανίζονται τα διαδοχικά γραφήματα σε ένα μοναδικό γράφημα, σε μεγαλύτερες κλίμακες ανάλυσης όμως φαίνεται ξεκάθαρα ο διαχωρισμός στην αρχική εικόνα.*

ΠΑΡΑΤΗΡΗΣΕΙΣ

* Η εργασία έγινε στα πλαίσια της εξεταστικής Σεπτεμβρίου, χωρίς να γίνει χρήση καμιάς από τις παρεχόμενες βοήθειες(δημοσιευμένο υλικό σχετικά με το θέμα, βοηθητικοί κώδικες για υλοποίηση των ζητούμενων λειτουργιών, κλπ)του διδάσκοντα, για λόγους απλότητας των ζητουμένων της, αλλά κυρίως εκμάθησης της matlab
* Προτινόμενο ζουμ προβολής αναφοράς 🡪 130%

ΒΙΒΛΙΟΓΡΑΦΙΑ

www.google.gr

http://el.wikipedia.org

http://www.medlook.net.cy

www.matlab.com