

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΕΡΓΑΣΙΑΣ

Απαλλακτική Εργασία του μαθήματος «Ανάλυση Εικόνας» για το ακαδημαϊκό έτος 2020-2021

Θέμα:

**Αυτόματος Χρωματισμός Ασπρόμαυρης Εικόνας με Χρήση
Τεχνικών Μηχανικής Μάθησης.**

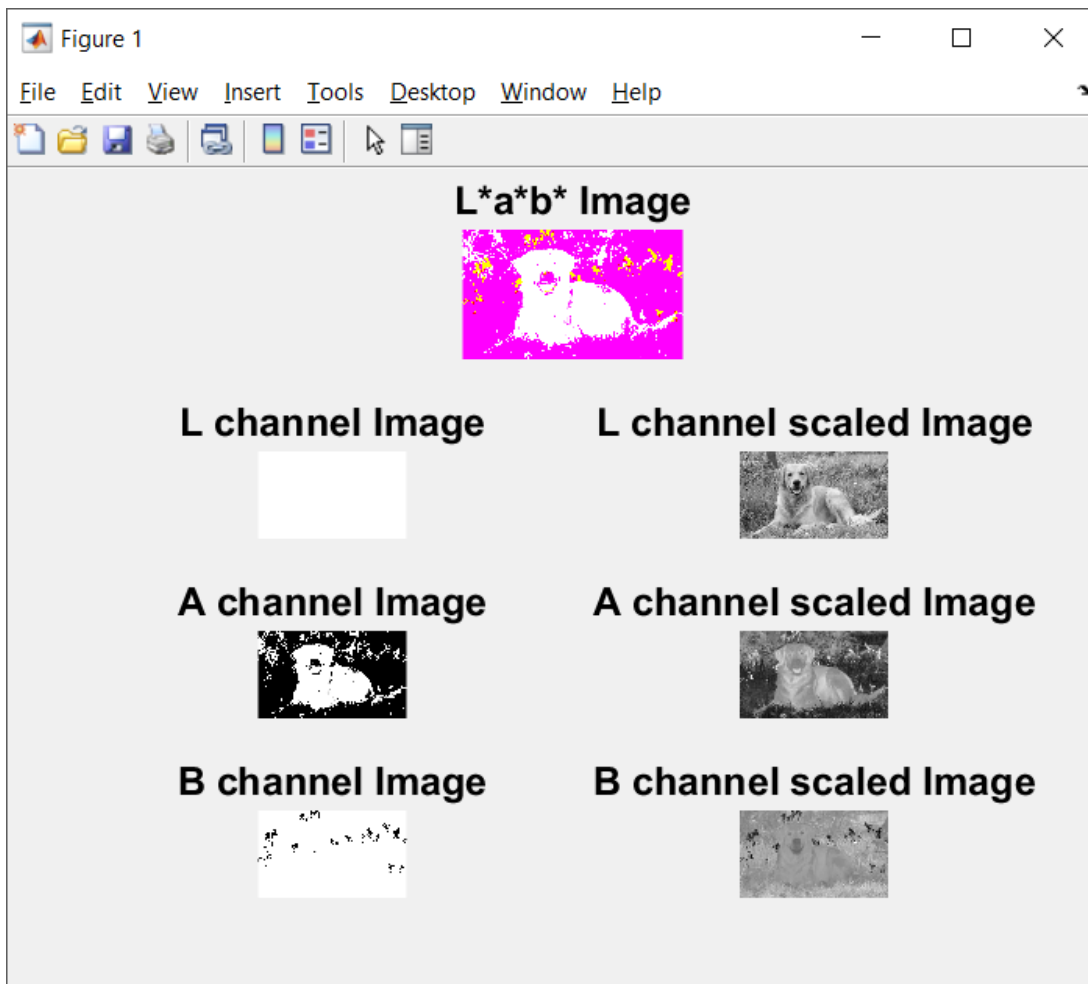
Επιμελούμενοι φοιτητές:

Παναγιώτης Δημητρέλλος – Π17026

Χαράλαμπος Φωτογιάννης – Π17156

1) Αναπαράσταση Εικόνας στον Χρωματικό Χώρο Lab .

- Μετατρέπουμε την πηγαία RGB εικόνα σε εικόνα $L^*a^*b^*$ μέσω της συνάρτησης `rgb2lab`
- Παίρνουμε το κάθε κανάλι ' L^* ', ' a^* ' και ' b^* ' για την $L^*a^*b^*$ εικόνα
- Εμφανίζουμε καθένα από τα κανάλια ατομικά και ρυθμίζουμε την απεικόνιση βασισμένη στο εύρος των τιμών των pixels



2) Διακριτοποίηση του Χρωματικού Χώρου Lab με βάση ένα σύνολο συναφών εικόνων εκπαίδευσης.

Σε αυτό το σημείο έχουμε χρησιμοποιήσει δύο σύνολα εκπαίδευσης: το πρώτο είναι για μια μεμονωμένη εικόνα και το χρησιμοποιήσαμε για testing , και το δεύτερο κάνει αναφορά σε 3 εικόνες και είναι αυτό που θα αναλύσουμε παρακάτω.

Έτσι λοιπόν,

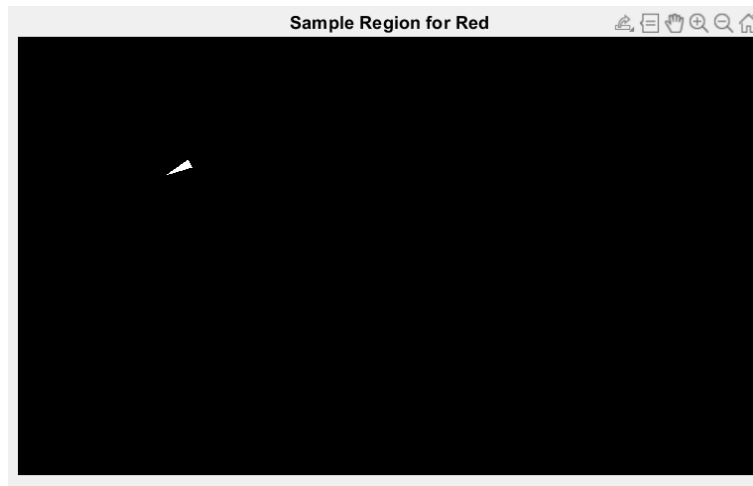
- Παίρνουμε ένα σύνολο εκπαίδευσης συναφών έγχρωμων εικόνων



- Υπολογίζουμε τα δείγματα χρώματος στον χρωματικό χώρο L^*a^*b για κάθε περιοχή.

Η προσέγγισή μας είναι να επιλέξουμε μια μικρή περιοχή δείγματος για κάθε χρώμα και να υπολογίσουμε τον μέσο χρώμα κάθε περιοχής δείγματος στο χώρο a^*b^* . Θα χρησιμοποιήσουμε αυτούς τους χρωματικούς δείκτες για να ταξινομήσουμε κάθε pixel.

Για παράδειγμα για την εικόνα source το μέσο χρώμα της κόκκινης περιοχής δείγματος στο διάστημα a^*b^* είναι το ακόλουθο:



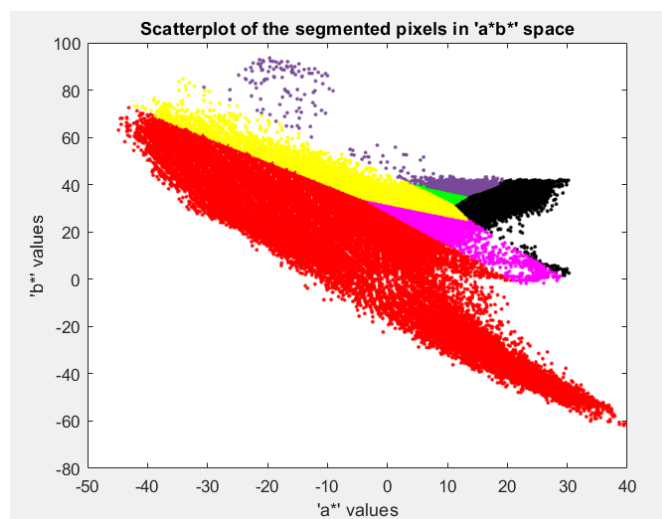
- Ταξινομούμε κάθε εικονοστοιχείο χρησιμοποιώντας τον κανόνα του πλησιέστερου γείτονα.

Τα αποτελέσματα εμφάνισης της ταξινόμησης πλησιέστερου γείτονα είναι τα εξής:



- Εμφανίζουμε τις τιμές 'a *' και 'b *' των ετικετών χρωμάτων:

Μπορούμε να δούμε πόσο καλά η ταξινόμηση του πλησιέστερου γείτονα διαχώρισε τους διαφορετικούς πληθυσμούς χρωμάτων σχεδιάζοντας τις τιμές «a *» και «b *» pixel που ταξινομήθηκαν σε ξεχωριστά χρώματα. Για σκοπούς εμφάνισης, επισημάνουμε κάθε σημείο με την έγχρωμη ετικέτα του.



Στην συνέχεια:

Διαβάζουμε την εικόνα και την μετατρέπουμε σε χώρο χρωμάτων $L^*a^*b^*$.

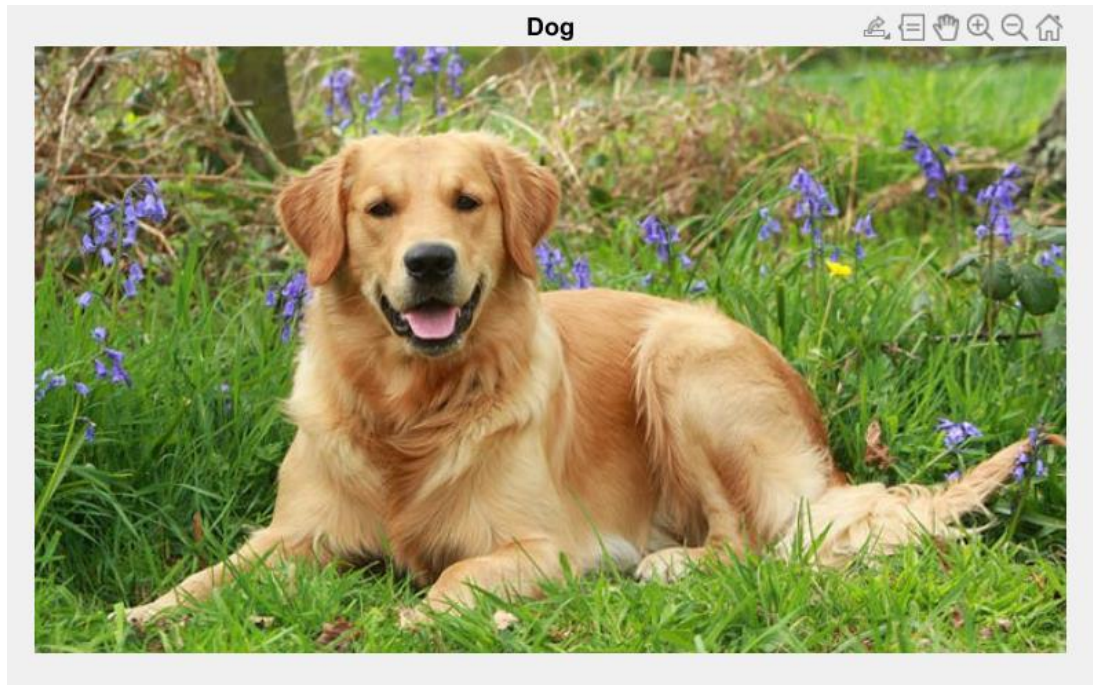
Κάνουμε εξαγωγή καναλιών a^* και b^* και αναδιαμόρφωση.

Κάνουμε τμηματοποίηση χρησιμοποιώντας k-means και εμφανίζουμε το αποτέλεσμα.

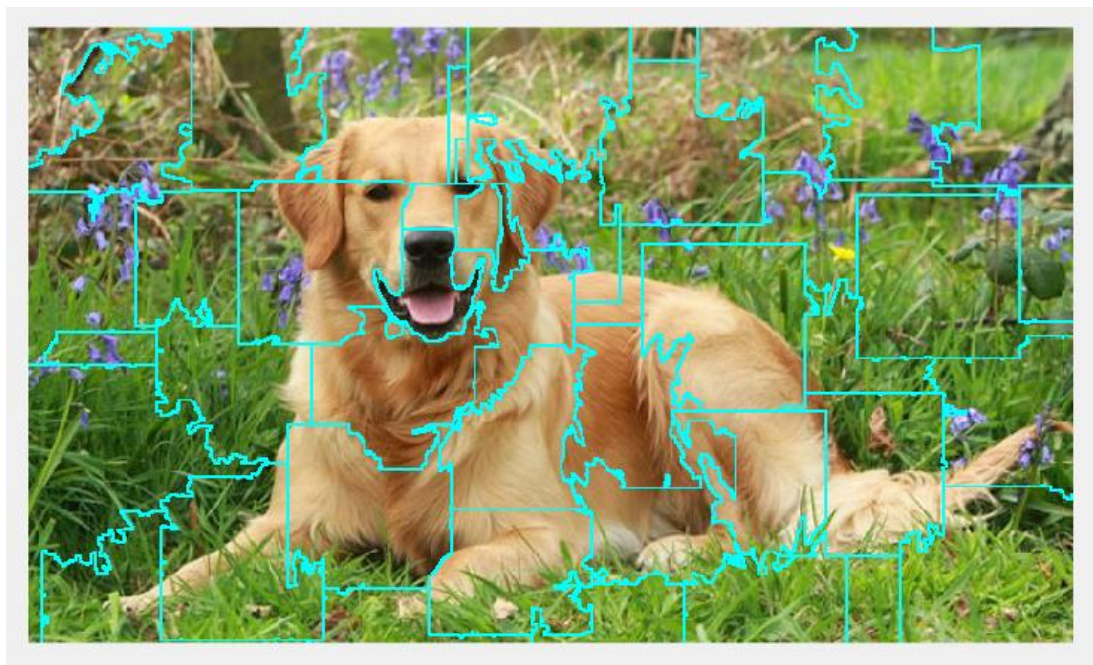


3) Κατάτμηση Εικόνας σε Superpixels σύμφωνα με τον αλγόριθμο SLIC

Έχουμε την αρχική εικόνα:



Υπολογίζουμε τα superpixels της εικόνας με την χρήση του αλγορίθμου SLIC και εμφανίζουμε τα όρια του superpixel που επικαλύπτονται στην αρχική εικόνα:



Στην συνέχεια ορίζουμε το χρώμα κάθε εικονοστοιχείου στην εικόνα εξόδου στο μέσο χρώμα RGB της περιοχής του superpixel.



4) Εξαγωγή Χαρακτηριστικών Υφής (SURF Features & Gabor Features) ανά Super Pixel.

1. Εξαγωγή Χαρακτηριστικών Υφής (SURF Features) ανά Super Pixel.

Θα χρησιμοποιήσω τη συνάρτηση `label2idx` για τον υπολογισμό των δεικτών των pixel σε κάθε σύμπλεγμα `superpixel`. Αυτό θα μου επιτρέψει να αποκτήσω πρόσβαση στις τιμές των κόκκινων, πράσινων και μπλε στοιχείων χρησιμοποιώντας γραμμική ευρετηρίαση

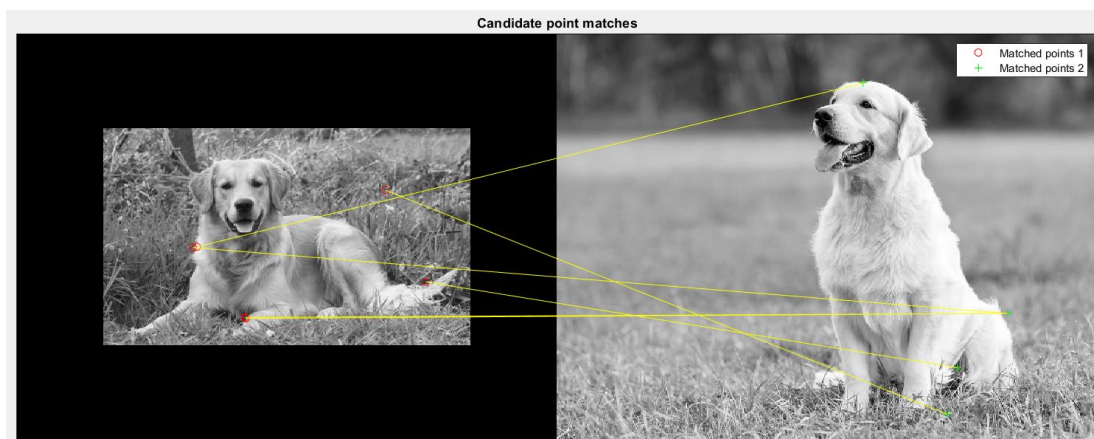
Έτσι λοιπόν για κάθε ένα από τα συμπλέγματα `N superpixel`, χρησιμοποιούμε γραμμική ευρετηρίαση για πρόσβαση στα κόκκινα, πράσινα και μπλε στοιχεία.

Ανακατασκευάζουμε τα αντίστοιχα εικονοστοιχεία ενώ ανιχνεύουμε τα χαρακτηριστικά SURF για την έκδοση κλίμακας του γκρι της αρχικής εικόνας και δείχνουμε τα 10 ισχυρότερα από αυτά, στην εικόνα εξόδου της κλίμακας του γκρι.

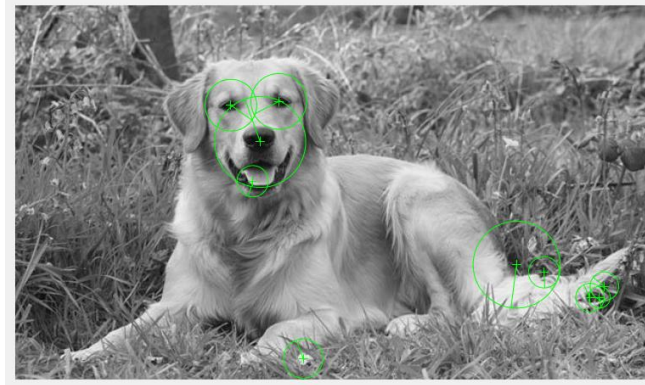


- Στο επόμενο βήμα βρίσκουμε αντίστοιχα σημεία μεταξύ δύο εικόνων χρησιμοποιώντας τις λειτουργίες SURF.

Οπτικοποιούμε τις υποψήφιες αντιστοιχήσεις:



Κάνουμε εξαγωγή των χαρακτηριστικών SURF από την εικόνα και οπτικοποιούμε τα 10 ισχυρότερα SURF χαρακτηριστικά, συμπεριλαμβανομένων των κλιμάκων και του προσανατολισμού τους που καθορίστηκαν κατά τη διαδικασία εξαγωγής περιγραφέα.

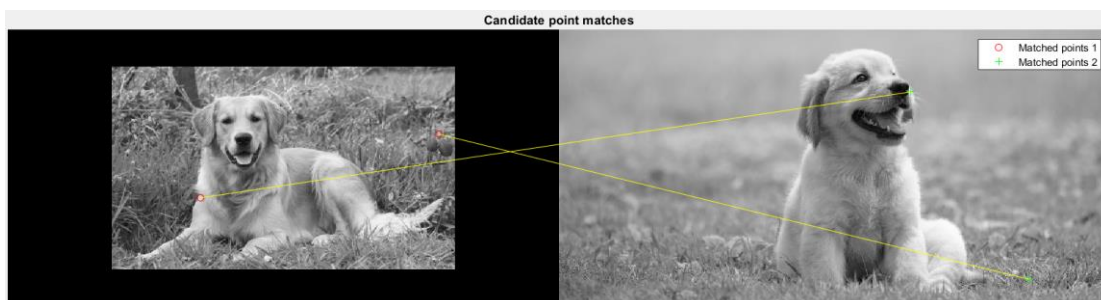
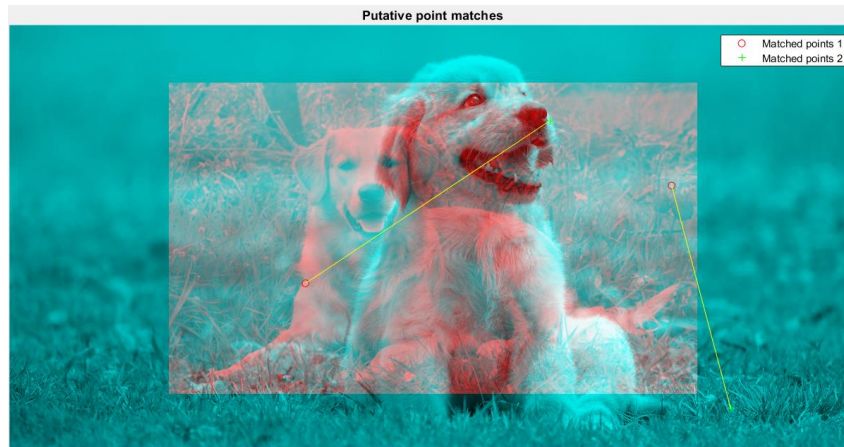


Σε δεύτερη φάση κάναμε εξαγωγή Χαρακτηριστικών Υφής (SURF Features) από μία ασπρόμαυρη εικόνα και από μια έγχρωμη και στην συνέχεια τα αντιστοιχίσαμε.

Κάνουμε εξαγωγή από την διαφορετική έγχρωμη εικόνα:



Αντιστοιχούμε τα χαρακτηριστικά και από τις δύο μεριές και οπτικοποιούμε τα υποψήφια χαρακτηριστικά.



2. Εξαγωγή Χαρακτηριστικών Υφής (Gabor Features) ανά Super Pixel.

Αρχικά έχουμε την εικόνα source :



- Σχεδιάζουμε μια σειρά φίλτρων Gabor που συντονίζονται σε διαφορετικές συχνότητες και προσανατολισμούς.

Το σύνολο συχνοτήτων και προσανατολισμών έχει σχεδιαστεί για να εντοπίζει διαφορετικά, περίπου ορθογώνια, υποσύνολα πληροφοριών συχνότητας και προσανατολισμού στην εικόνα εισαγωγής.

Με τακτικά δείγματα προσανατολισμών μεταξύ $[0, 150]$ μοίρες σε βήματα των 30 μοιρών.

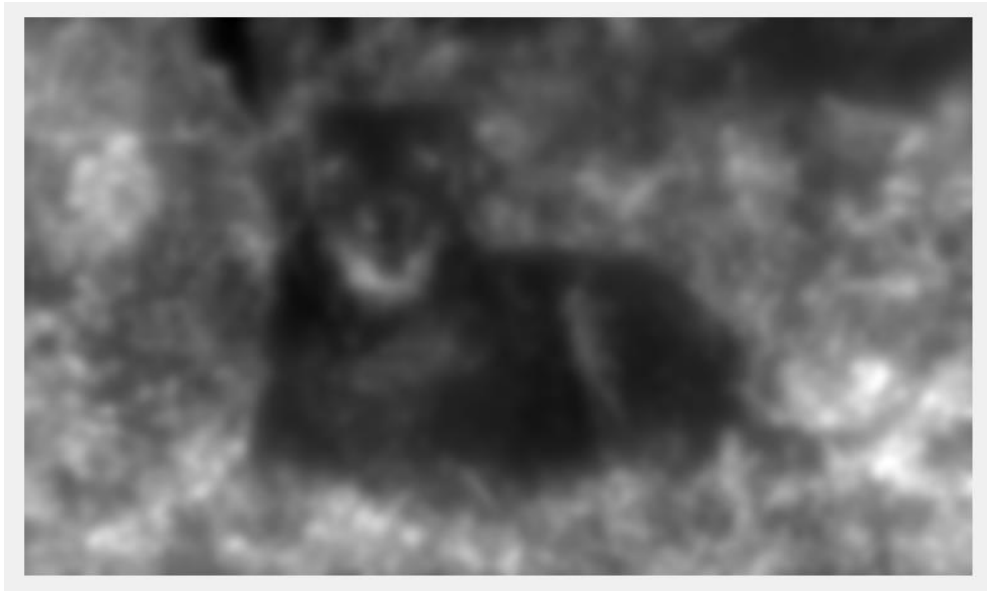
Και με δείγμα μήκους κύματος σε αυξανόμενες δυνάμεις του δύο και εκκίνηση από 4.

- Εξαγάγουμε χαρακτηριστικά μεγέθους Gabor από την εικόνα προέλευσης.

- Για να χρησιμοποιήσουμε τις αποκρίσεις μεγέθους Gabor ως χαρακτηριστικά για χρήση στην ταξινόμηση, απαιτείται κάποια μετα-επεξεργασία.

Αυτή η επεξεργασία δημοσιεύσεων περιλαμβάνει εξομάλυνση Gauss, προσθέτοντας επιπλέον χωρικές πληροφορίες στο σύνολο χαρακτηριστικών, αναδιαμόρφωση του συνόλου λειτουργιών μας στη φόρμα που αναμένεται από τις συναρτήσεις `rca` και `kmeans` και ομαλοποίηση των πληροφοριών χαρακτηριστικών σε μια κοινή διακύμανση και μέσο όρο.

Οπτικοποιούμε το σύνολο δυνατοτήτων:

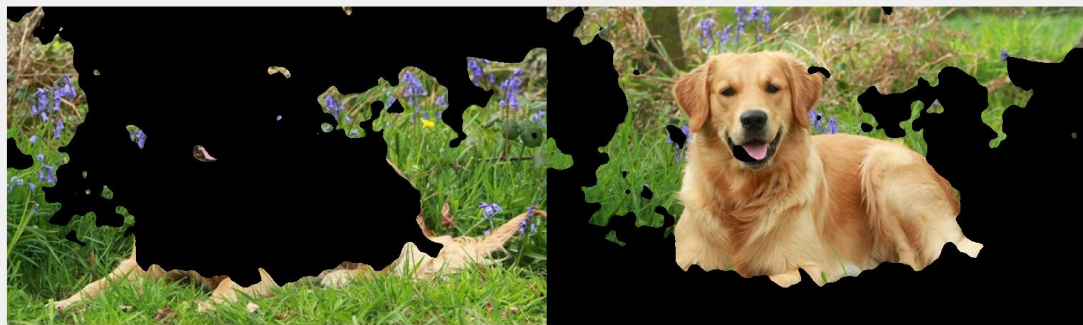


Ταξινομούμε τα χαρακτηριστικά υφής Gabor χρησιμοποιώντας kmeans:

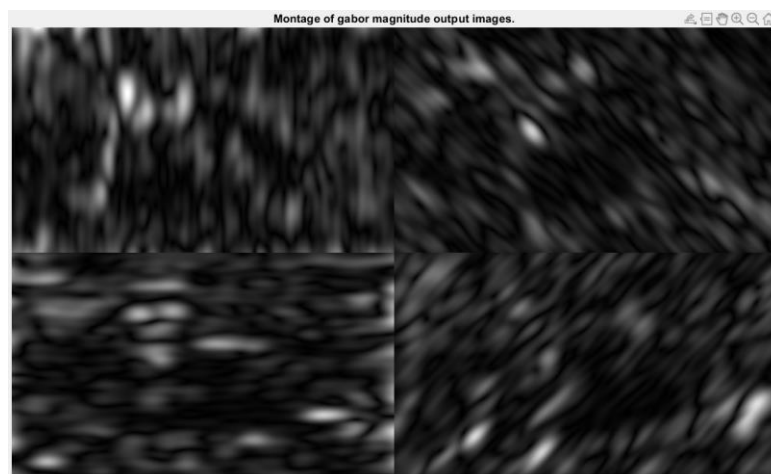


Οπτικοποιούμε την τμηματοποιημένη εικόνα χρησιμοποιώντας imshowpair.

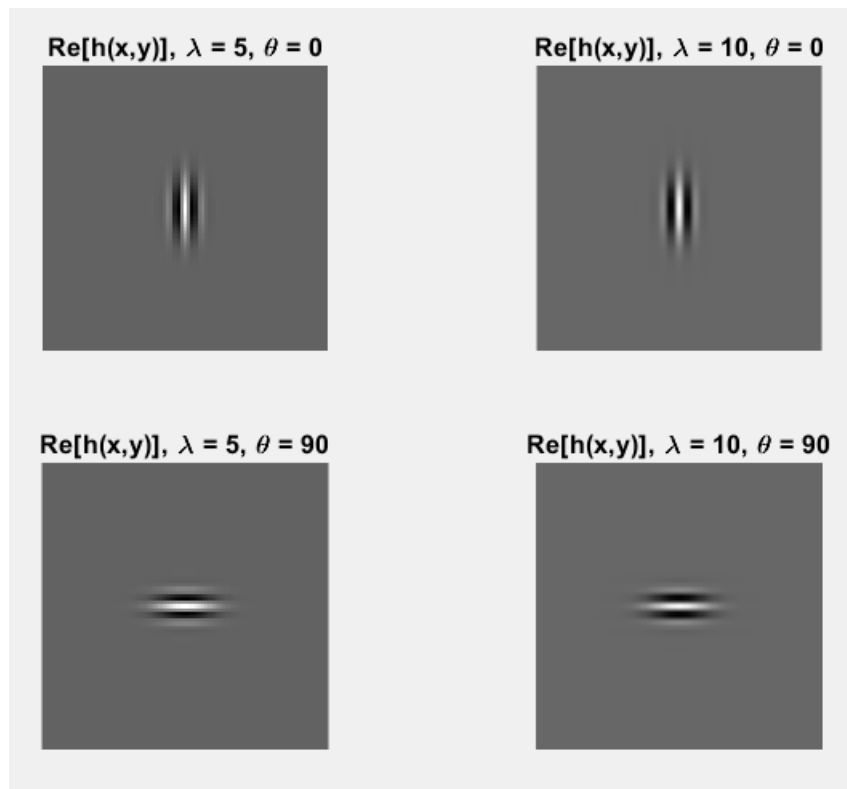
Εξετάζουμε τις εικόνες προσκηνίου και φόντου που προκύπτουν από τη μάσκα BW που σχετίζεται με την ετικέτα matrix L.



Κατασκευή σειράς φίλτρου Gabor και εφαρμογή στην εικόνα εισαγωγής:



Επόμενο βήμα είναι η κατασκευή φίλτρου Gabor Array και οπτικοποιούμε το μήκος κύματος και τον προσανατολισμό:

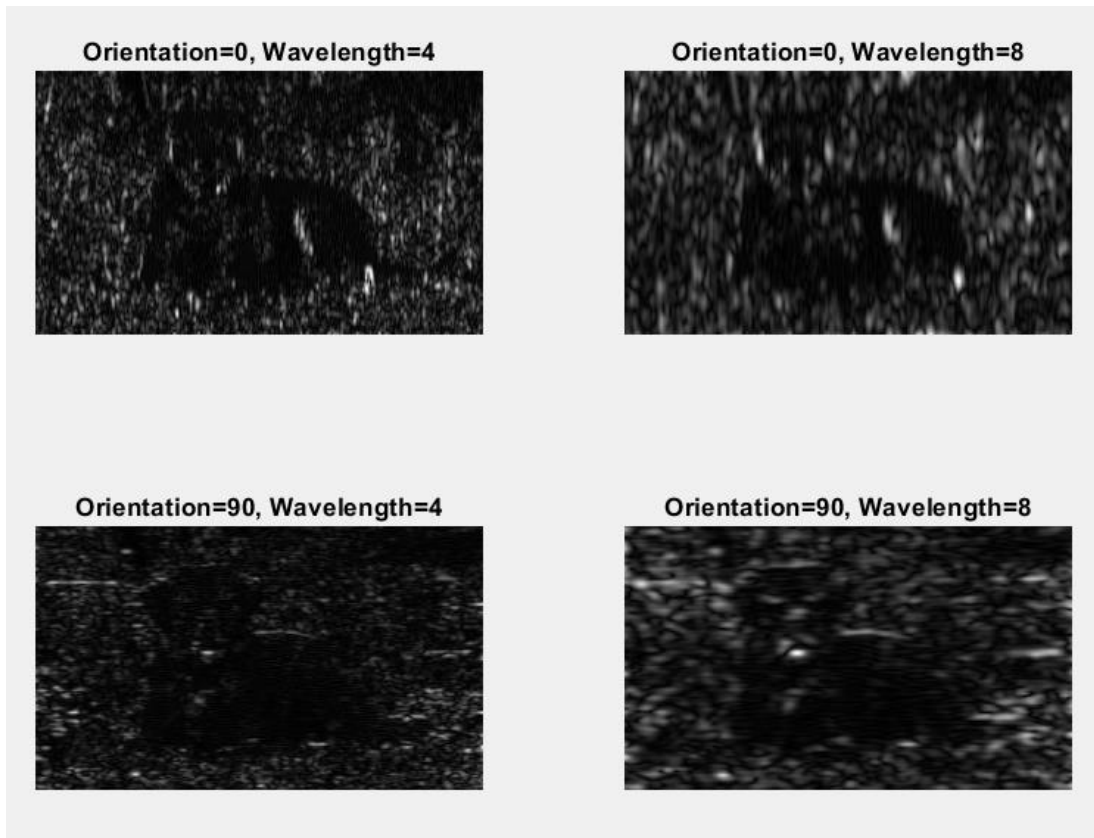


Στην συνέχεια εφαρμόζουμε το φίλτρο Gabor στην εικόνα εισαγωγής:

Εμφανίζουμε την αρχική εικόνα με γραφικές παραστάσεις του μεγέθους και της φάσης που υπολογίζονται από το φίλτρο Gabor.



Τέλος κάνουμε εφαρμογή συστοιχίας φίλτρων Gabor στην εικόνα εισαγωγής:



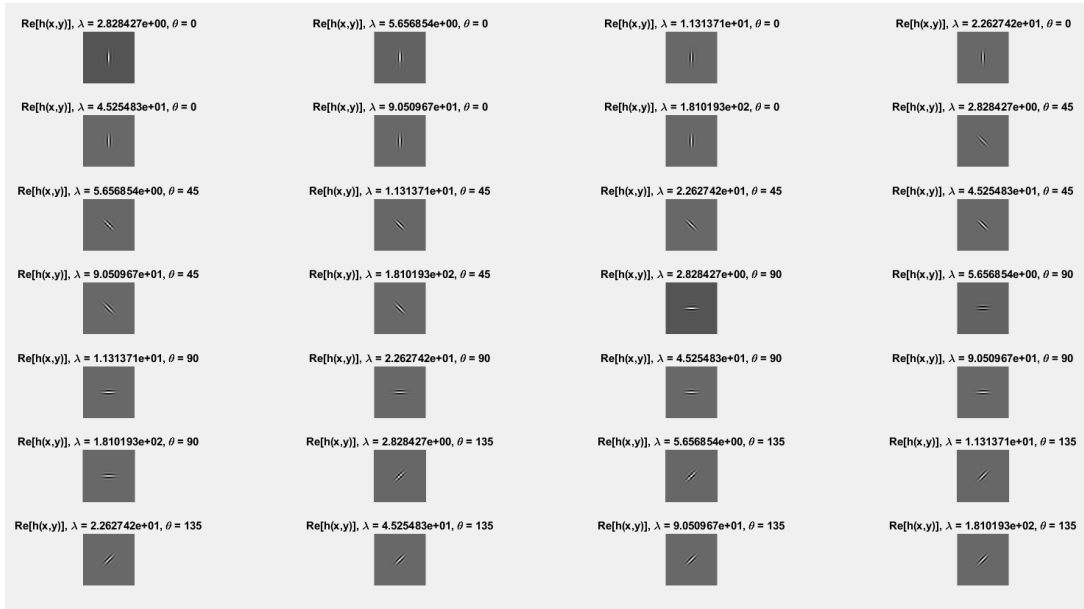
Το σχήμα δείχνει την απόκριση μεγέθους για κάθε φίλτρο.

2.2) Κάνουμε εξαγωγή των χαρακτηριστικών Gabor από μια έγχρωμη εικόνα .

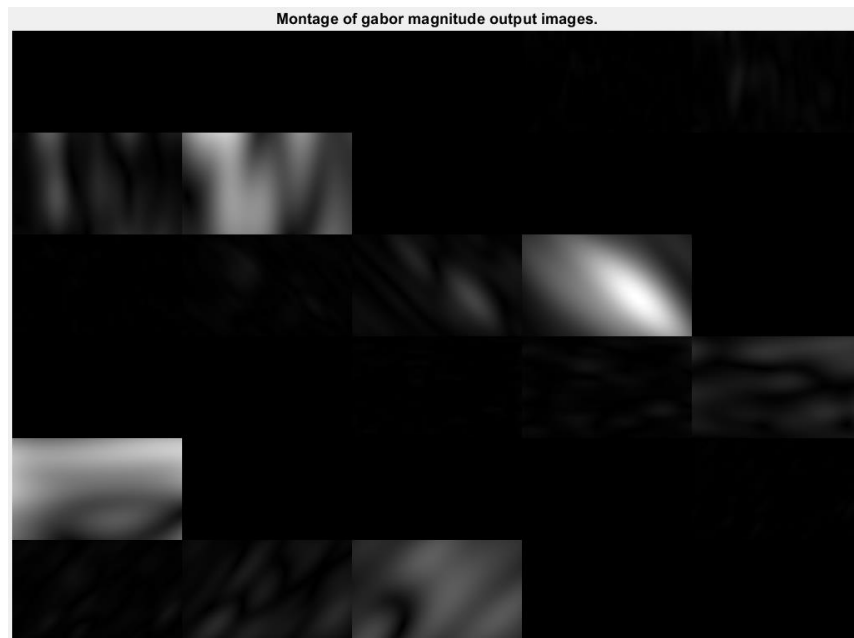
Αρχικά αυτό το κάνουμε έτσι ώστε να εξάγουμε τα χαρακτηριστικά από τα έγχρωμα superpixels για να εκπαιδεύσουμε το SVM.

Σχεδιάζουμε μια σειρά φίλτρων Gabor που συντονίζονται σε διαφορετικές συχνότητες και προσανατολισμούς.

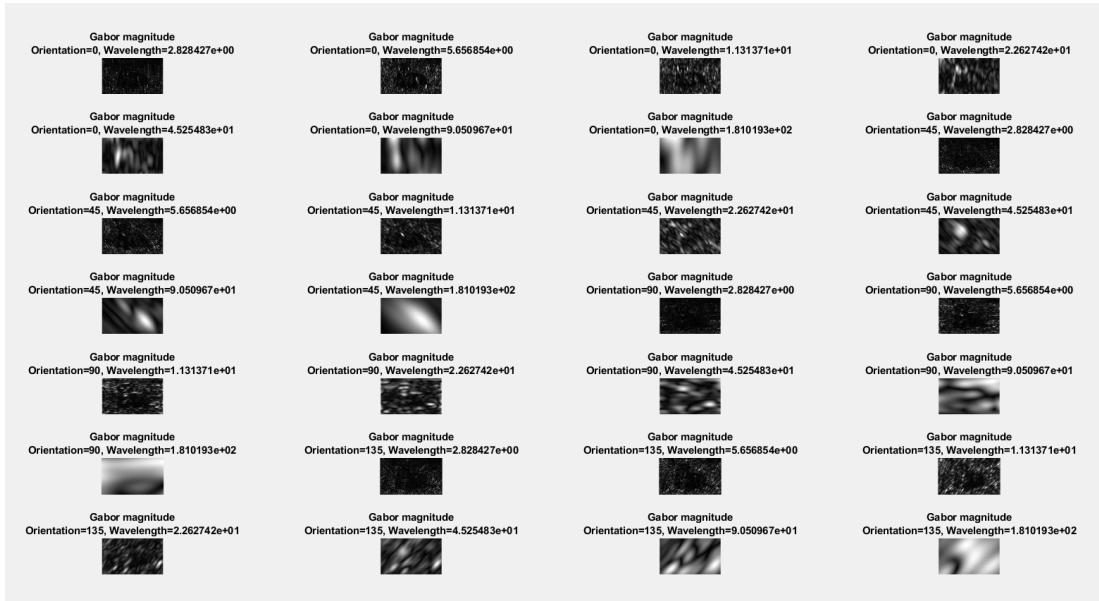
Οπτικοποιούμε το πραγματικό μέρος του χωρικού πυρήνα της συνέλιξης κάθε φίλτρου Gabor στον πίνακα:



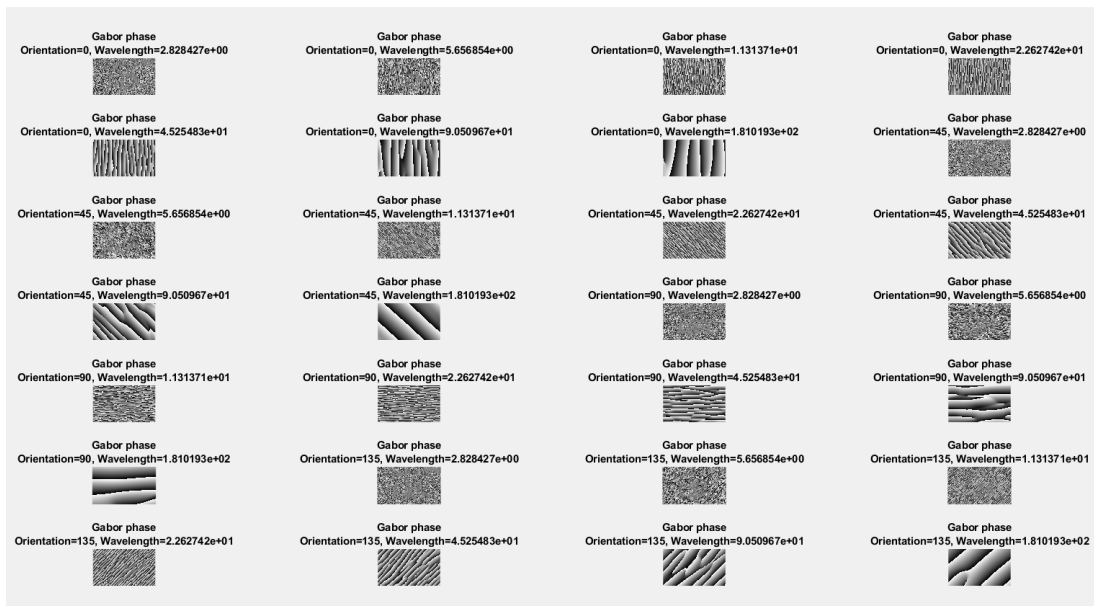
Εμφανίζουμε τα αποτελέσματα μεγέθους:



Εμφανίζουμε το μέγεθος που υπολογίζεται από το φίλτρο Gabor:

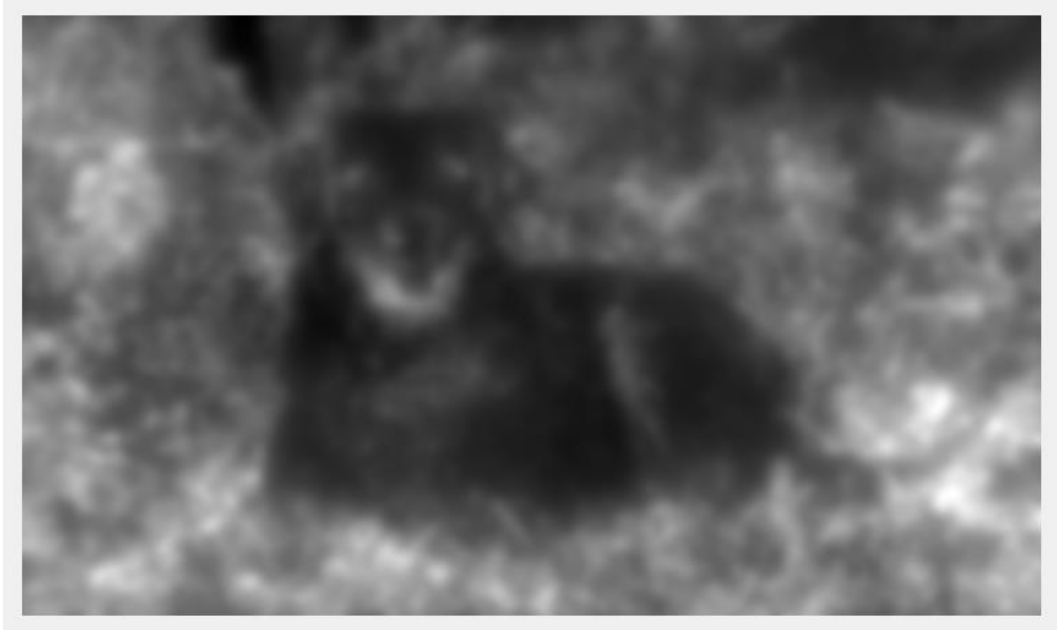


Εμφανίζουμε την φάση που υπολογίζεται από το φίλτρο Gabor:



Στο επόμενο βήμα κάνουμε μετα-επεξεργασία των εικόνων μεγέθους Gabor σε χαρακτηριστικά Gabor.

Οπτικοποιούμε το σύνολο δυνατοτήτων:



Ταξινομούμε τα χαρακτηριστικά υφής Gabor χρησιμοποιώντας kmeans:



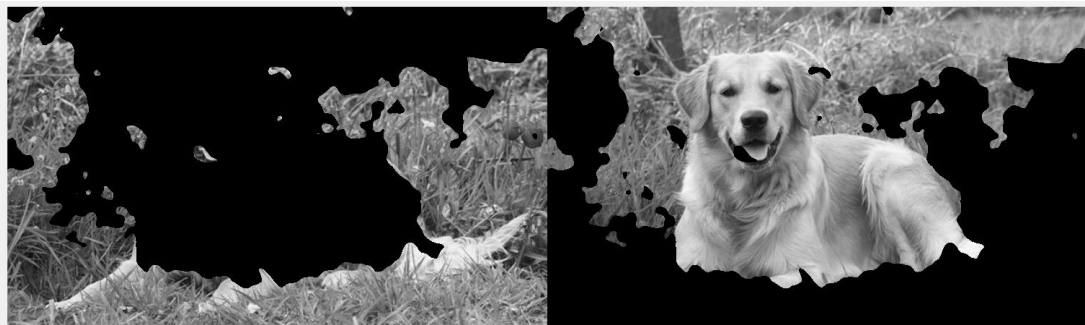
Οπτικοποιούμε την τμηματοποιημένη εικόνα χρησιμοποιώντας imshowpair.
Εξετάζουμε τις εικόνες προσκηνίου και φόντου που προκύπτουν από τη μάσκα BW που σχετίζεται με την ετικέτα matrix L.



2.3) Σε αυτό το σημείο κάνουμε εξαγωγή των χαρακτηριστικών Gabor από μια ασπρόμαυρη εικόνα .

Αυτό το κάνουμε για να εξάγουμε χαρακτηριστικά από τα ασπρόμαυρα superpixels και να τα περάσουμε στο εκπαιδευμένο SVM για να μαντέψει το χρώμα των superpixel το οποίο θα είναι ένα από τα διακριτά χρώματα.

Ακολουθούμε την ίδια διεργασία που προαναφέραμε για την έγχρωμη εικόνα.



5) Εκμάθηση Τοπικών Μοντέλων Πρόγνωσης Χρώματος με Χρήση Ταξινομητών SVM

5.1 Εκπαίδευουμε τους ταξινομητές SVM χρησιμοποιώντας πυρήνα Gaussian

Αυτό το παράδειγμα δείχνει πώς να δημιουργήσουμε έναν μη γραμμικό ταξινομητή με τη λειτουργία πυρήνα Gaussian.

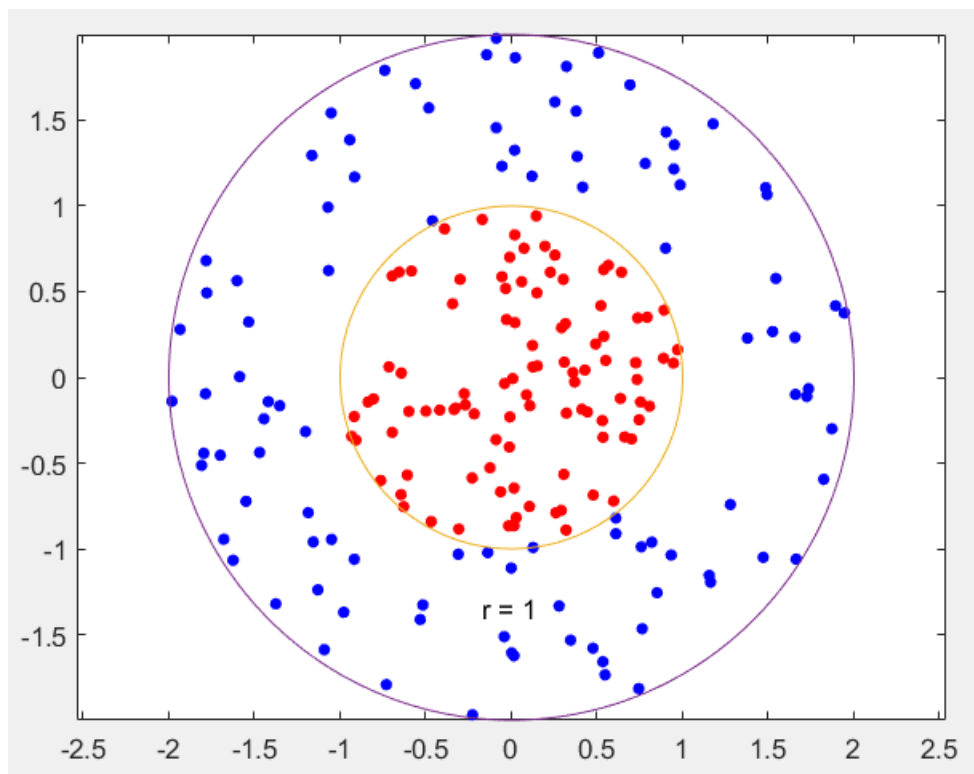
Πρώτα, δημιουργούμε μια κατηγορία σημείων μέσα στο δίσκο μονάδας σε δύο διαστάσεις και μια άλλη κλάση σημείων στον δακτύλιο από την ακτίνα 1 έως την ακτίνα 2.

Στη συνέχεια, δημιουργεί έναν ταξινομητή με βάση τα δεδομένα με τον πυρήνα λειτουργίας ακτινικής βάσης Gaussian. Ο προεπιλεγμένος γραμμικός ταξινομητής είναι προφανώς ακατάλληλος για αυτό το πρόβλημα, καθώς το μοντέλο είναι κυκλικά συμμετρικό. Ορίζουμε την παράμετρο περιορισμού πλαισίου σε 10^4 για να κάνουμε μια αυστηρή ταξινόμηση, που σημαίνει ότι δεν υπάρχουν εσφαλμένα ταξινομημένα σημεία.

Άλλες λειτουργίες του πυρήνα μπορεί να μην λειτουργούν με αυτόν τον αυστηρό περιορισμό του κουτιού, καθώς ενδέχεται να μην είναι σε θέση να παράσχει αυστηρή ταξινόμηση.

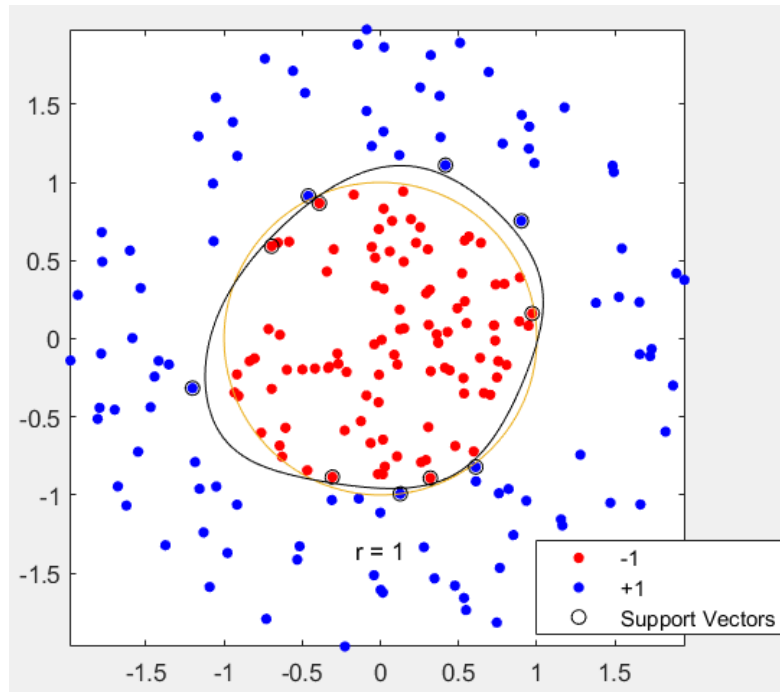
Ακόμα κι αν ο ταξινομητής rbf μπορεί να διαχωρίσει τις κλάσεις, το αποτέλεσμα μπορεί να είναι υπερβολικό.

Σχεδιάζουμε τα σημεία και τους κύκλους των ακτίνων 1 και 2 για σύγκριση:

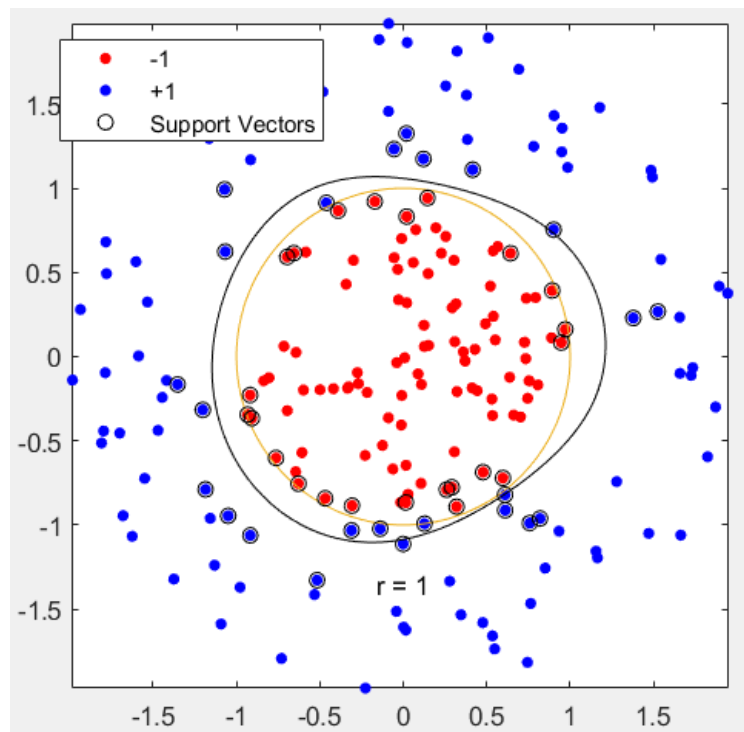


- Εκπαίδευουμε έναν ταξινομητή SVM με το KernelFunction σε «rbf» και το BoxConstraint σε Inf.

Σχεδιάζουμε τα δεδομένα και το όριο της απόφασης:



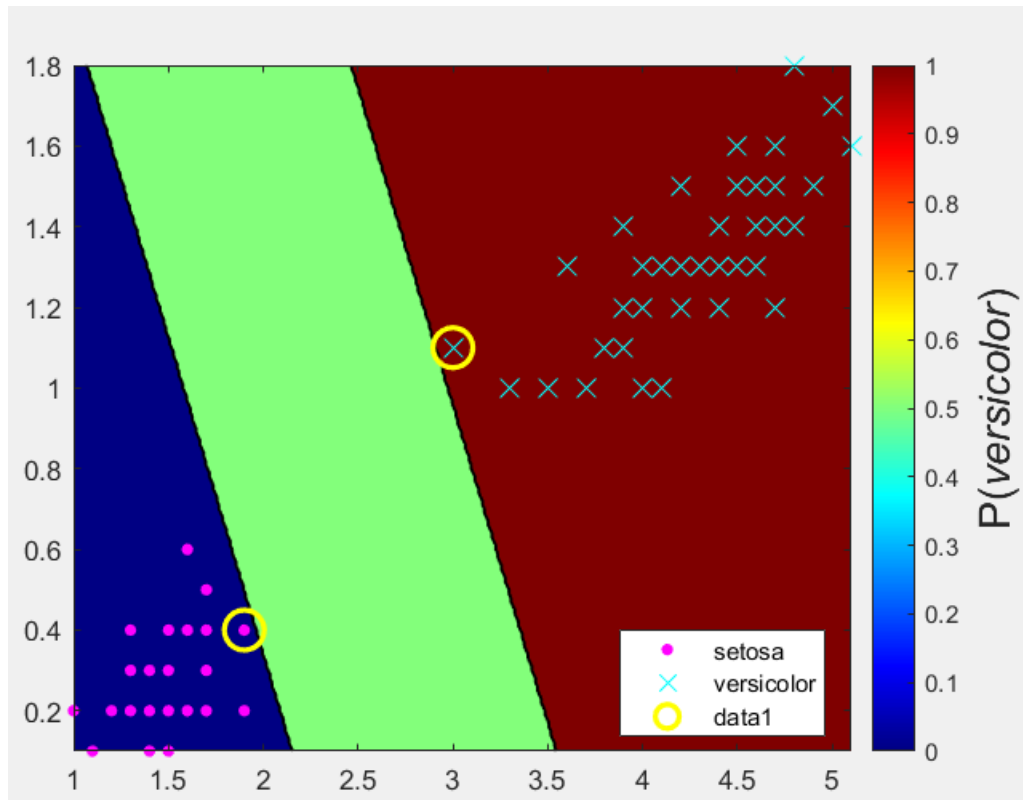
Το `fitsvm` δημιουργεί έναν ταξινομητή που βρίσκεται κοντά σε έναν κύκλο ακτίνας 1. Η διαφορά οφείλεται στα τυχαία δεδομένα εκπαίδευσης.



5.3 Σχεδίαση μεταγενέστερων περιοχών πιθανότητας για μοντέλα ταξινόμησης SVM

Αυτό το παράδειγμα δείχνει πώς να προβλέψουμε τις οπίσθιες πιθανότητες των μοντέλων SVM σε ένα πλέγμα παρατηρήσεων, και στη συνέχεια σχεδιασμός των οπίσθιων πιθανοτήτων πάνω από το πλέγμα. Η χάραξη οπίσθιας πιθανότητας εκθέτει όρια αποφάσεων.

Σε εκμάθηση δύο τάξεων, εάν οι τάξεις είναι διαχωρίσιμες, τότε υπάρχουν τρεις περιοχές: μία όπου οι παρατηρήσεις έχουν θετική τάξη οπίσθια πιθανότητα 0, μία όπου είναι 1, και μια όπου είναι η θετική τάξη προηγούμενη πιθανότητα.

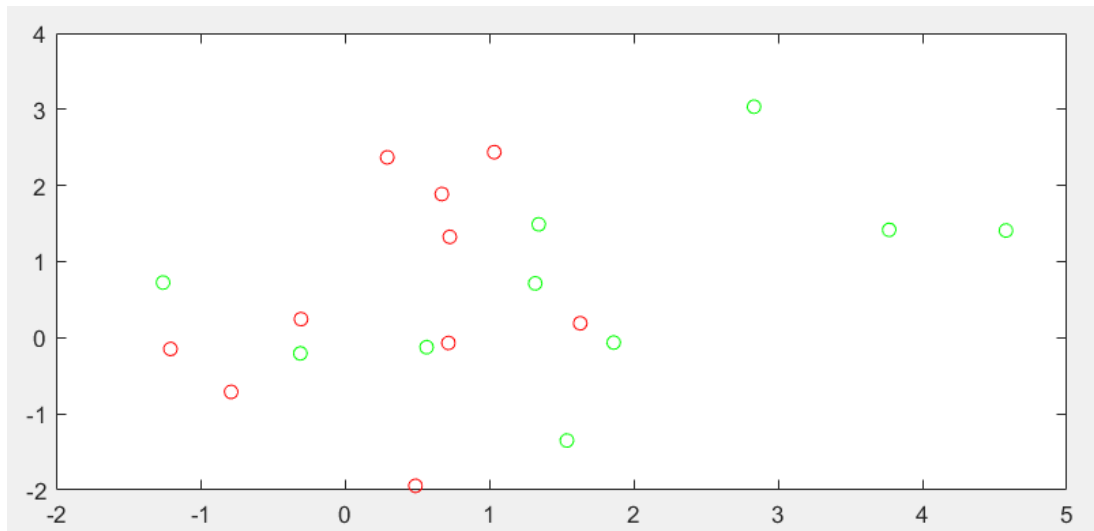


5.5 Βελτιστοποιούμε μια προσαρμογή ταξινόμησης SVM χρησιμοποιώντας τη βελτιστοποίηση Bayesian

Αυτό το παράδειγμα δείχνει πώς να βελτιστοποιήσουμε μια ταξινόμηση SVM χρησιμοποιώντας τη συνάρτηση `fitcsvm` και το ζεύγος τιμών-τιμών `OptimizeHyperparameters`.

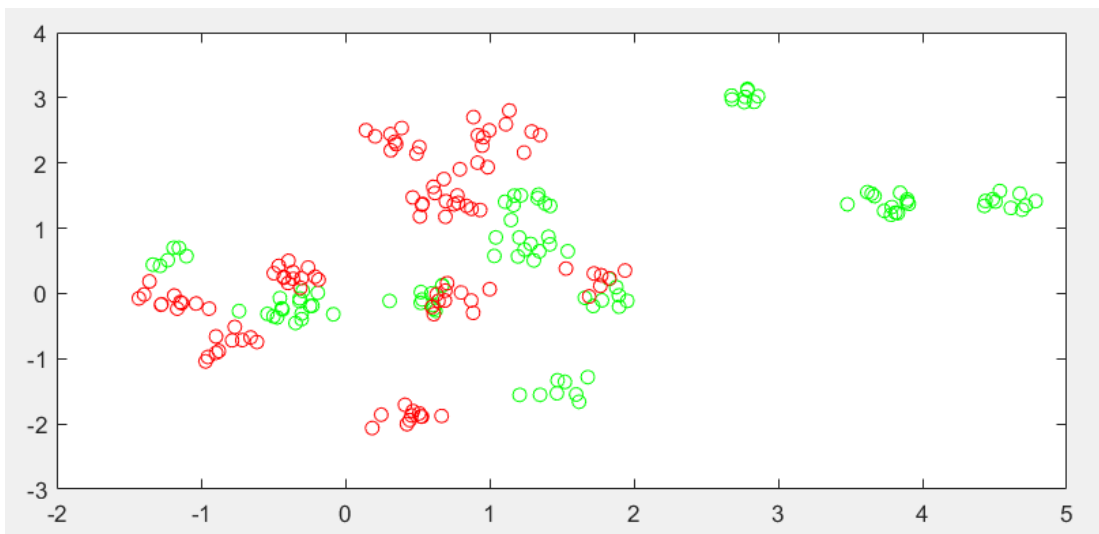
Η ταξινόμηση λειτουργεί σε τοποθεσίες σημείων από ένα μοντέλο μείγματος Gauss.

Το μοντέλο ξεκινά με τη δημιουργία 10 σημείων βάσης για μια "πράσινη" τάξη, διανεμημένη ως ανεξάρτητη κανονική 2-D με μέση τιμή (1,0) και διακύμανση μονάδας. Παράγει επίσης 10 σημεία βάσης για μια "κόκκινη" τάξη, διανεμημένη ως ανεξάρτητη 2-D κανονική με μέση τιμή (0,1) και διακύμανση μονάδας.

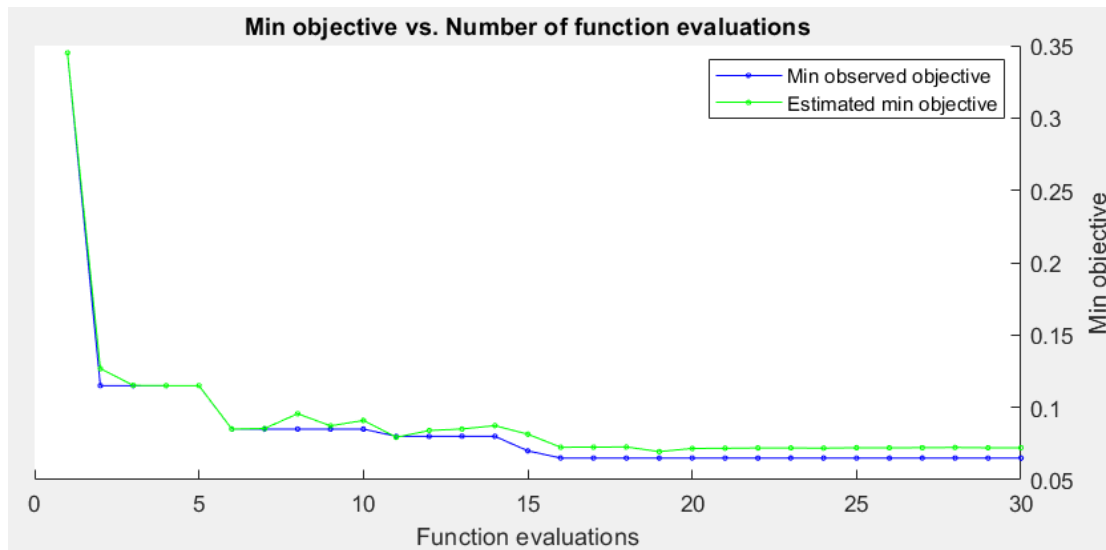


Δεδομένου ότι ορισμένα κόκκινα σημεία βάσης είναι κοντά στα πράσινα σημεία βάσης, μπορεί να είναι δύσκολο να ταξινομηθούν τα σημεία δεδομένων μόνο με βάση την τοποθεσία.

Για κάθε τάξη (πράσινο και κόκκινο), δημιουργούμε 100 τυχαία σημεία ως εξής:



Κάνουμε προετοιμασία δεδομένων για ταξινόμηση

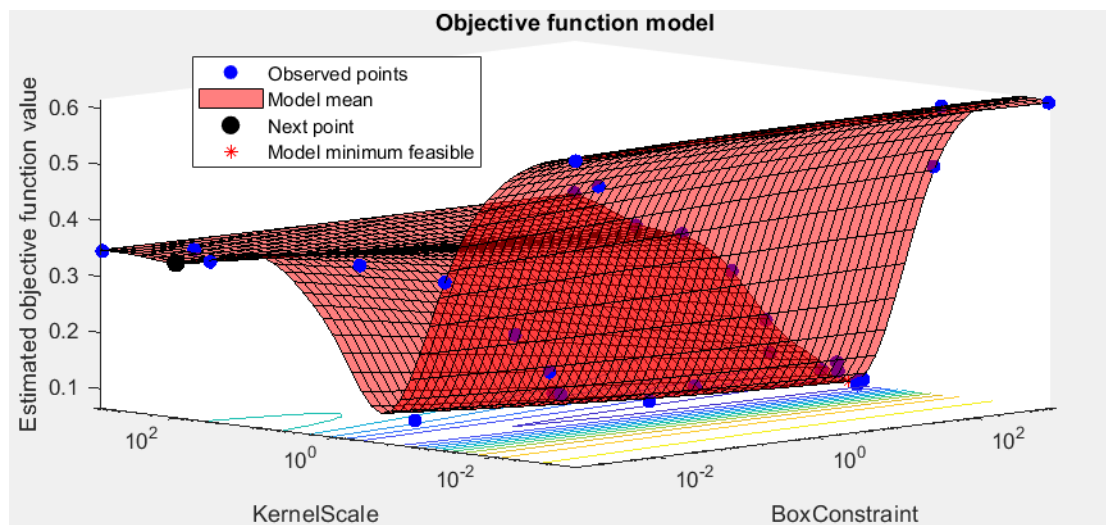


Βελτιστοποιούμε το Fit:

Για να βρούμε μια καλή εφαρμογή, που σημαίνει μια με χαμηλή απώλεια εγκυρότητας, ορίζουμε επιλογές για τη χρήση βελτιστοποίησης Bayesian.

Χρησιμοποιούμε το ίδιο διαμέρισμα διασταυρούμενης επικύρωσης c σε όλες τις βελτιστοποιήσεις.

Βρίσκουμε την απώλεια του βελτιστοποιημένου μοντέλου:



Οπτικοποιούμε τον βελτιστοποιημένο ταξινομητή.

