

Εργασία 3 Ψηφιακή Επεξεργασία Εικόνας 2018

Μελεζιάδης Ιωάννης

AEM 8760

Image2Graph.m

Υπάρχουν σχόλια που εξηγούν τις εντολές οπότε ο αναγνώστης καλείται να έχει ανοιχτό συγχρόνως και το αρχείο. Ο affinity πίνακας κατασκευάζεται παίρνοντας με τη σειρά όλα τα pixels κάθε γραμμής της αρχικής εικόνας. Σχετικά με το συνδυασμό των pixels και τον υπολογισμό των βαρών, υπάρχουν δυο όμοια κομμάτια κώδικα με σχολιασμό “calculations for pixel-*i*” που βρίσκουν τα pixels (που θέλω να βρω το μεταξύ τους βάρος) και ανάλογα με το πόσα κανάλια έχει η εικόνα υπολογίζω και το βάρος τους. Τέλος αφού έχει ολοκληρωθεί όλη η διαδικασία ελέγχω τη συμμετρία του πίνακα με την μεταβλητή `matrixSymmetry` της οποίας η τιμή θέλω να είναι 1.

mySpectralClustering.m

Το αρχείο είναι πλήρως σχολιασμένο και ακολουθεί τις οδηγίες από την εκφώνηση. Σε περίπτωση που εμφανιστεί error στην εντολή `[Vectors, Values] = eigs(L, k, 'sm');` Τότε το ‘sm’ πρέπει να αλλαχτεί σε ‘smallestabs’(το ίδιο και στο `nonRecursive.m`). Αυτό οφείλεται στις διαφορετικές εκδόσεις του matlab. Το ‘sm’ για παράδειγμα δουλεύει στην R2014b.

nonRecursive.m

Πρόκειται για την υλοποίηση της μη αναδρομικής μορφής της μεθόδου *n-cuts* που διαφοροποιείται από το αρχείο `mySpectralClustering.m` μόνο στο κομμάτι του υπολογισμού των ιδιοτιμών και ιδιοδιανυσμάτων που σε αυτή τη περίπτωση γίνεται με βάση το γενικευμένο πρόβλημα.

Ενότητα 1 (demo1.m)

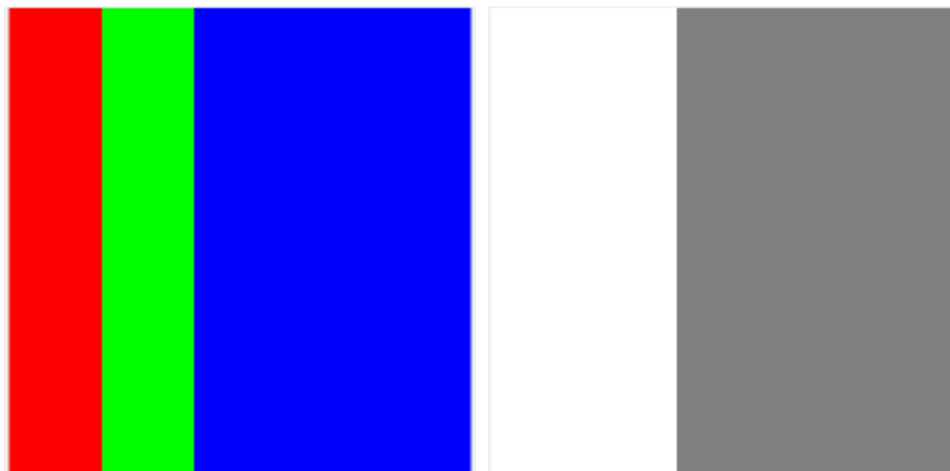
Παρακάτω παρουσιάζεται το αποτέλεσμα του script demo1.m :

```
The results for k = 2 are :  
    1    1    1    1    2    2    2    2    1    1    1    1  
  
The results for k = 3 are :  
    1    1    1    1    2    2    2    2    3    3    3    3  
  
The results for k = 4 are :  
    4    2    4    4    3    3    3    2    1    1    1    1
```

Φαίνεται ότι όσο αυξάνει το k τόσο περισσότερες περιοχές της αρχικής εικόνας αναγνωρίζονται και ομαδοποιούνται στα ανάλογα clusters.

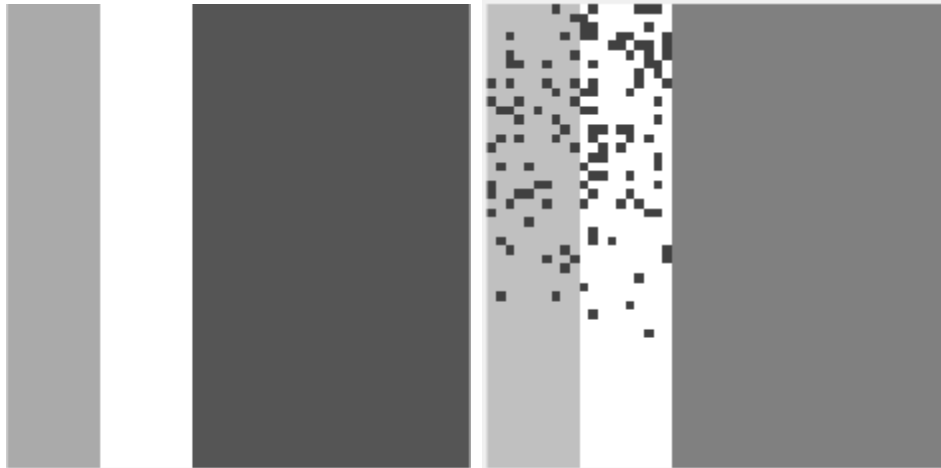
Ενότητα 2 (demo2.m)

Ο κώδικας αυτός αφού δημιουργήσει τους affinity πίνακες για τις 2 εικόνες, εκτελεί τις ίδιες λειτουργίες με το demo1.m με τη διαφορά ότι παρουσιάζεται και το αποτέλεσμα του clustering της εικόνας μετά από κάθε κάλεσμα της mySpectralClustering.m. Αυτό επιτυγχάνεται με το κομμάτι κώδικα με σχόλιο “visualize” το οποίο και εμφανίζεται για κάθε συνδυασμό εικόνας- k . Για την οπτικοποίηση των clusters αρχικά διαιρείται ο μονοδιάστατος πίνακας clusterIdx με το k ώστε να περιέχει τα clusters σε διαφορετικές αποχρώσεις του γκρι. Στη συνέχεια δημιουργείται από τον πίνακα clusterIdx ένας δυσδιάστατος πίνακας στο μέγεθος της αρχικής εικόνας με την εντολή reshape του matlab. Τα αποτελέσματα του demo2.m παρουσιάζονται παρακάτω :



Αρχική εικόνα(d2a)

$k = 2$



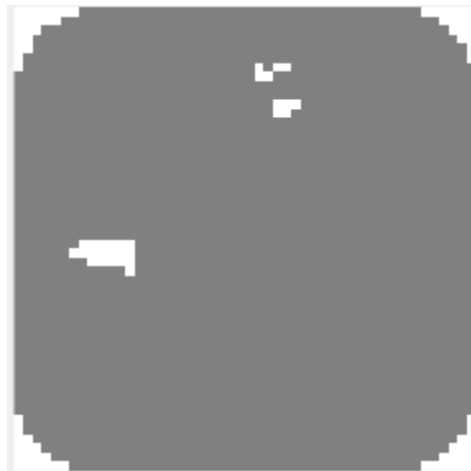
k = 3

k = 4

Για $k = 2$ βλέπουμε ότι δεν διαχώρισε δυο περιοχές και αυτό είναι αναμενόμενο. Αντίθετα για $k = 3$ τις διαχώρισε ακριβώς, ενώ για $k = 4$ φαίνεται να επέλεξε τυχαία σημεία από τις δυο πρώτες περιοχές για να χωρέσει κάπου και το τέταρτο cluster(που όμως δεν υπήρχε και για αυτό ο αλγόριθμος αποτυγχάνει).



Αρχική εικόνα(d2b)



k = 2



k = 3



k = 4



k = 5

Σε αυτήν την εικόνα που είχε περισσότερες περιοχές βλέπουμε ότι για $k = 2$ διαχώρισε τις πιο φωτεινές περιοχές (άσπρο) από τις υπόλοιπες. Στη συνέχεια βλέπουμε ότι όσο αυξάνει το k τόσο περισσότερες περιοχές τις εικόνας κατηγοριοποιεί και αναγνωρίζει.

Ενότητα 3 (demo3a.m)

Παρακάτω παρουσιάζονται τα αποτελέσματα του demo3a.m :



(d2a)k = 2

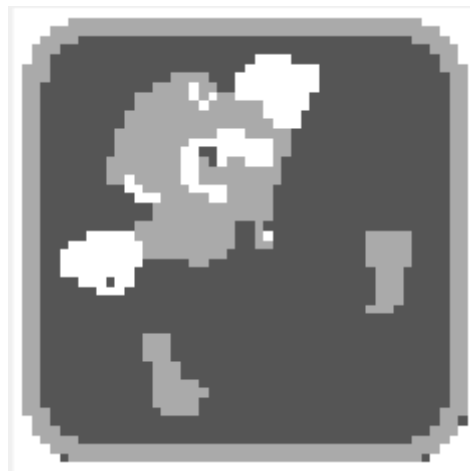
k = 3



k = 4



(d2b)k = 2



k = 3



$k = 4$



$k = 5$

Σύμφωνα με τα παραπάνω αποτελέσματα , η μη αναδρομική εκδοχή της μεθόδου n-cuts παράγει καλύτερα αποτελέσματα από την spectral clustering. Αυτό φαίνεται καθώς ομαδοποιεί περισσότερα pixels με αποτέλεσμα να αναγνωρίζεται πιο εύκολα και πιο γρήγορα(δηλαδή με μικρότερο k) η αρχική εικόνα.