Ψηφιακή Επεξεργασία Εικόνας

Εργασία 2

Όνομα: Πορλού Χάιδω

ΑΕΜ: 9372

---

1. Εικόνες ως γράφοι

Στο πρώτο κομμάτι της εργασίας καλούμαστε να κατασκευάσουμε την ρουτίνα **Image2Graph(imIn)**, η οποία δέχεται σαν είσοδο μια εικόνα με n κανάλια και επιστρέφει τον affinity πίνακα που περιγράφει ένα μη-κατευθυντικό γράφο G = (V,E).

Κάθε κελί του affinity πίνακα αποτελεί την τιμή του βάρους μιας ακμής του γράφου, και υπολογίζεται ως $A(i, j) = \frac{1}{e^d}$, όπου d(i, j) η ευκλείδια απόσταση της φωτεινότητας των καναλιών μεταξύ του i-οστού και j-οστού πίξελ. Σημειώνεται ότι ο γράφος που περιγράφει ο affinity πίνακας είναι fully connected (δεν υπάρχει μηδενικό κελί).

---

2. Εικόνες ως γράφοι

Στη συνέχεια, καλούμστε να υλοποιήσουμε τη μέθοδο Spectral Clustering, τα βήματα της οποίας είναι τα εξής:

1. Δεδομένης μιας εικόνας εισόδου, κατασκευάζουμε ένα μη κατευθυντικό γράφο, μέσω της παραπάνω διαδικασίας. Έστω **W** ο affinity πίνακας που περιγράφει τον γράφο.

2. Υπολογίζουμε το Λαπλασιανό πίνακα **L** ως **L** = **D** - **W**. Ο διαγώνιος πίνακας **D** ορίζεται ως: **D**(i, i) = Σ_j **W**(i, j).

3. Λύνουμε το πρόβλημα ιδιοτιμών **L**x = λx, και υπολογίζουμε τις k **μικρότερες** ιδιοτιμές καθώς και τα k ιδιοδιανύσματα που αντιστοιχούν σε αυτές τις ιδιοτιμές.

4. Σχηματίζουμε τον πίνακα **U** k που περιέχει τα ιδιοδιανύσματα u_1, ... , u_k σαν στήλες. Για i = 1, ... , n, έστω **yi** το διάνυσμα που αντιστοιχεί στην i-οστή γραμμή του **U**.

5. Ομαδοποιούμε τα σημεία **yi** με τον αλγόριθμο k-means στα clusters C1 .... ,Ck.

Κατασκευάζουμε λοιπόν την ρουτίνα mySpectralClustering(anAffinityMat, k), η οποία δέχεται σαν είσοδο έναν affinity πίνακα και τον αριθμό των clusters στον οποίο θέλουμε να τον διασπάσουμε, και έχει σαν έξοδο έναν πίνακα/στήλη Μ στοιχείων (αν ο affinity matrix είναι μεγέθους MxM), ο οποίος δείχνει τις ετικέτες των clusters στις οποίες ανήκουν οι κορυφές του γράφου.

Για παράδειγμα, αν καλέσουμε την παραπάνω συνάρτηση με όρισμα k = 2, κάθε κελί/θέση του πίνακα εξόδου θα περιέχει την τιμή 1 ή 2, ανάλογα φυσικά με την έξοδο του αλγορίθμου kmeans.

Demo 1

Στο πρώτο demo της εργασίας, καλούμαστε να παρουσιάσουμε τη λειτουργία της συνάρτησης που υλοποιήσαμε, mySpectralClustering(). Μας δίνεται ήδη ένας affinity πίνακας (d1a), τον οποίο θα χρησιμοποιήσουμε στη συνέχεια, ώστε να παρουσιάσουμε τα αποτελέσματα.

Πέραν των τιμών του clusterIdx που επιστρέφει η μέθοδος mySpectralClustering(), προχωρήσαμε και σε ένα visualization των αποτελεσμάτων, χρησιμοποιώντας

α. τον πίνακα με τα clusterId,

b. μία διαδικασία reshape (στο αρχικό μέγεθος της εικόνας),

c. μία διαδικασία transpose, και τέλος

d. την εμφάνιση της προκύπτουσας εικόνας μέσω της συνάρτησης MATLAB imshow(image, [0 k])*, όπου k ο αριθμός των cluster.

*το k παίζει το ρόλο του μέγιστου αριθμού που εμφανίζεται μέσα στον πίνακα, στον οποίο η συνάρτηση MATLAB imshow αντιστοιχίζει την μεγαλύτερη δυνατή φωτεινότητα (1), και ανάλογα δίνει τιμές σε όλες τις μικρότερες στάθμες ώστε να υπάρχει το καλύτερο δυνατό visualization.

Τα αποτελέσματα του demo για k = 2, 3, 4 clusters παρουσιάζονται στη συνέχεια, μέσω ενός πίνακα:

clusterIdx		
k = 2	**k = 3**	**k = 4**
1	1	3
1	1	1
1	1	1
1	1	3
2	3	4
2	3	4
2	3	4
2	3	4
2	2	2
2	2	2
2	2	2
2	2	2

Παρακάτω παρουσιάζεται και το visualization για κάθε περίπτωση:

k = 2

k = 3

k = 4

Φυσικά, αναφερόμαστε σε 4x3 εικόνες, οπότε δε μπορεί να φανεί κάτι ξεκάθαρα, αλλά προστέθηκε για λόγους πληρότητας. Γνωρίζοντας όμως αποτελέσματα που θα παρουσιασθούν παρακάτω (μέθοδος «διόρθωσης» με αναδρομικό ncuts), το σωστό clustering γίνεται όταν k = 3, και η $4^{η}$ απόχρωση που εμφανίζεται όταν k = 4 εμφανίζεται αυθαίρετα, μιας και το kmeans προσπαθεί να «χωρέσει» ενα $4^{ο}$ cluster ενώ δεν υπάρχει αντίστοιχη απόχρωση.

Θα μπορούσε επίσης να σημειωθεί ότι ένα cluster με 2 στοιχεία/πίξελ είναι πολύ μικρό ώστε να δώσει σημαντική πληροφορία (όταν k = 4).

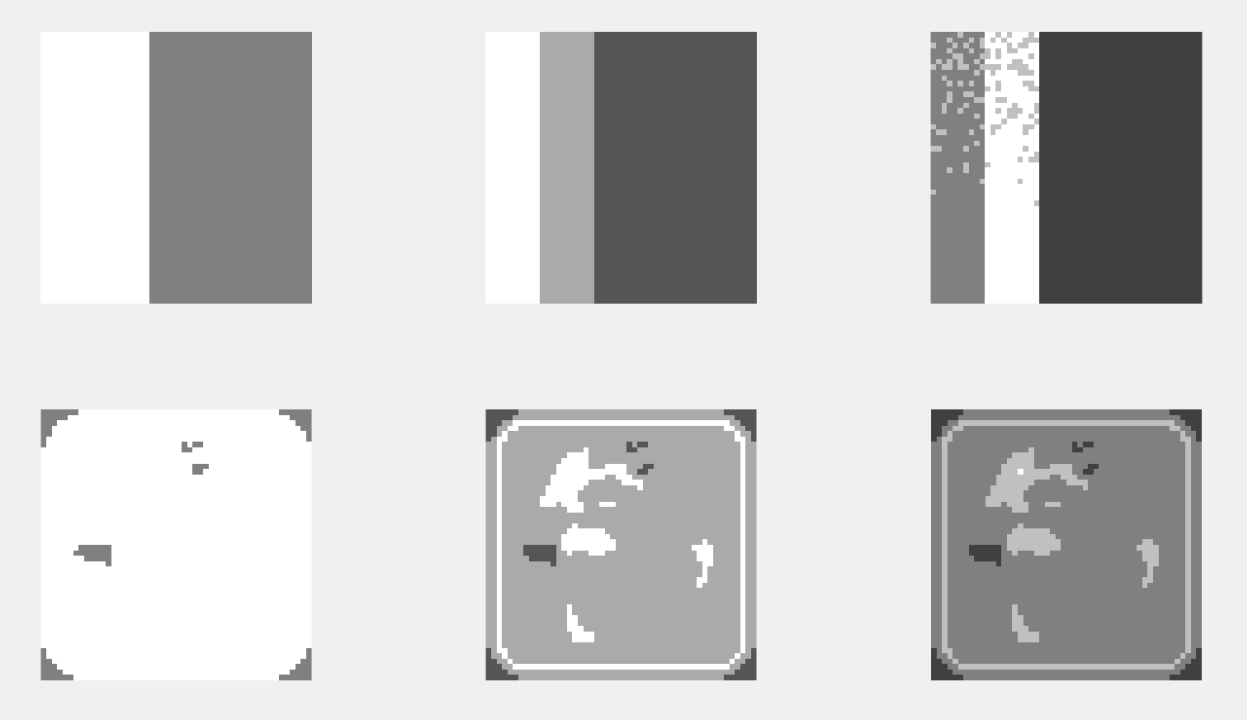
Demo 2

Στο πρώτο demo της εργασίας, καλούμαστε να παρουσιάσουμε τη λειτουργία της συνάρτησης mySpectralClustering(), σε συνδυασμό με την αρχική συνάρτηση Image2Graph(), η οποία δημιουργεί έναν affinity πίνακα με είσοδο μία εικόνα/πίνακα.

Καλείται λοιπόν αρχικά η Image2Graph(), και στη συνέχεια τρέχει η διαδικασία του spectral clustering για όλες τις περιπτώσεις (k = 2, k = 3, k = 4), και για τις 2 εικόνες (d2a & d2b).

Το visualization των παραπάνω παρουσιάζεται παρακάτω:

k = 2 k = 3 k = 4



Αξίζει να σημειωθεί ότι στην πρώτη εικόνα έχουμε σωστότερο clustering όταν k = 3, ενώ όταν προχωράμε σε k = 4, αρχίζουν και εμφανίζονται πίξελ σε λάθος θέσεις, καθώς ο αλγόριθμος kmeans προσπαθεί να «χωρέσει» μια $4^{η}$ απόχρωση σε μια εικόνα που είχε αρχικά μόνο 3. Έτσι, εμφανίζει πίξελ σε σχετικά τυχαίες θέσεις, δημιουργώντας ένα προβληματικό αποτέλεσμα.

Όσον αφορά την $2^{η}$ εικόνα, όταν έχουμε k = 2 δεν φαίνεται κάτι ιδιαίτερα ξεκάθαρα, ενώ για k = 3 και k = 4 το αποτέλεσμα είναι πολύ καλύτερο.

Σημειώνω εδώ ότι είχα πολλά προβλήματα με το rng(1) όταν το έβαζα μόνο στην αρχή κάθε demo, υποθέτω ότι κάθε seed επηρεαζόταν από τον προηγούμενο kmeans με κάποιον τρόπο, όπως αναφέρεται εδώ: **https://www.mathworks.com/help/matlab/math/generate-random-numbers-that-are-repeatable.html.** Είναι πιθανό να έχει να κάνει και με το version του MATLAB που χρησιμοποιώ (R2016a). In any case, τελικά τοποθέτησα την εντολή rng(1) κάθε φορά πριν από την κλήση της mySpectralClustering(). Εναλλακτικά δοκίμασα να την τοποθετήσω και πριν την εντολή kmeans() μέσα στη συνάρτηση, με παρόμοια αποτελέσματα.

---

3. Normalized-cuts

Στο τελευταίο κομμάτι της εργασίας καλούμαστε να υλοποιήσουμε την (αναδρομική και μη) μέθοδο normalized cuts. Τα βήματα της μεθόδου αυτής είναι παρόμοια με της μεθόδου spectral clustering, με μία διαφοροποίηση στο βήμα 3:

1. Δεδομένης μιας εικόνας εισόδου, κατασκευάζουμε ένα μη κατευθυντικό γράφο, μέσω της παραπάνω διαδικασίας. Έστω **W** ο affinity πίνακας που περιγράφει τον γράφο.

2. Υπολογίζουμε το Λαπλασιανό πίνακα **L** ως **L** = **D** - **W**. Ο διαγώνιος πίνακας **D** ορίζεται ως: $\mathbf{D}(i, i) = \Sigma_j \mathbf{W}(i, j)$.

3. Λύνουμε το **γενικευμένο** πρόβλημα ιδιοτιμών $\mathbf{L}x = \lambda \mathbf{D}x$, και υπολογίζουμε τις k **μικρότερες** ιδιοτιμές καθώς και τα k ιδιοδιανύσματα που αντιστοιχούν σε αυτές τις ιδιοτιμές.

4. Σχηματίζουμε τον πίνακα **U** k που περιέχει τα ιδιοδιανύσματα $u_1, \ldots, u_k$ σαν στήλες. Για i = 1, ... , n, έστω **yi** το διάνυσμα που αντιστοιχεί στην i-οστή γραμμή του **U**.

5. Ομαδοποιούμε τα σημεία **yi** με τον αλγόριθμο k-means στα clusters C1 .... ,Ck.

Αν τα παραπάνω βήμα τρέξουν μόνο **μία** φορά, έχουμε την μη-αναδρομική εκδοχή του αλγορίθμου. Αν, από την άλλη, θέλουμε να υλοποιήσουμε την αναδρομική εκδοχή, επαναλαμβάνουμε τα παραπάνω βήματα (δηλαδή ουσιαστικά διχοτομούμε κάθε cluster σε 2), κάνοντας κάθε φορά κατάλληλους ελέγχους, για να αποφασίσουμε πότε θα σταματήσει η διαδικασία.

Πιο συγκεκριμένα, αν ο αριθμός των κόμβων είτε με ετικέτα 1 είτε 2 που προκύπτουν είναι μικρότερος από ένα κατώφλι T1 **ή** αν η τιμή Ncut(A,B) είναι

μεγαλύτερη από ένα κατώφλι T2, τότε η διχοτόμηση των συγκεκριμένων κομματιών που προέκυψαν σταματά. Διαφορετικά, συνεχίζει η διχοτόμηση κανονικά, δηλαδή ξεκινούν ξανά από την αρχή τα βήματα 1-5 για κάθε διχοτομημένο κομμάτι. Η διαδικασία, φυσικά, ολοκληρώνεται όταν κανένα από τα κομμάτια/clusters που έχουν δημιουργηθεί δεν μπορούν να σπάσουν περαιτέρω.

Η τιμή Ncut υπολογίζεται με τον παρακάτω τρόπο:

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

$$assoc(A,V) = \sum_{u \in A, t \in V} \mathbf{W}(u,t)$$

Καλούμαστε, λοιπόν, να κατασκευάσουμε την ρουτίνα **myNCuts(anAffinityMat, k)**, η οποία περιγράφει την μη-αναδρομική εκδοχή του αλγορίθμου, δηλαδή ολοκληρώνει μια μόνο επανάληψη των βημάτων 1-5.

Στη συνέχεια, δημιουργούμε τη συνάρτηση **calculateNcut(anAffinityMat, clusterIdx)**, η οποία υπολογίζει την τιμή Ncut που θα χρησιμοποιήσουμε σαν έλεγχο στην αναδρομική εκδοχή της διαδικασίας normalized cuts συγκρίνοντας την με το T2, σε συνδυασμό με την τιμή T1.

Για την αναδρομική εκδοχή της μεθόδου υλοποιούμε την συνάρτηση **myNCutsRec(anAffinityMat, k, T1, T2, cluster)**, η οποία πέρα από τον affinity πίνακα και τον αριθμό των clusters που θα δημιουργούμε κάθε φορά (εδώ 2), παίρνει σαν ορίσματα και τις τιμές T1 και T2 που χρειάζονται για τον έλεγχο, καθώς και τη μεταβλητή cluster, η οποία δείχνει το όνομα/αριθμό του cluster στο οποίο βρισκόμαστε/καταλήξαμε. Χρησιμοποιείται για την διαφοροποίηση των clusters, διότι κανονικά οι μόνες τιμές που δέχεται ένα cluster είναι 1 ή 2, και πρέπει με κάποιο τρόπο να δέχονται διαφορετικές τιμές.

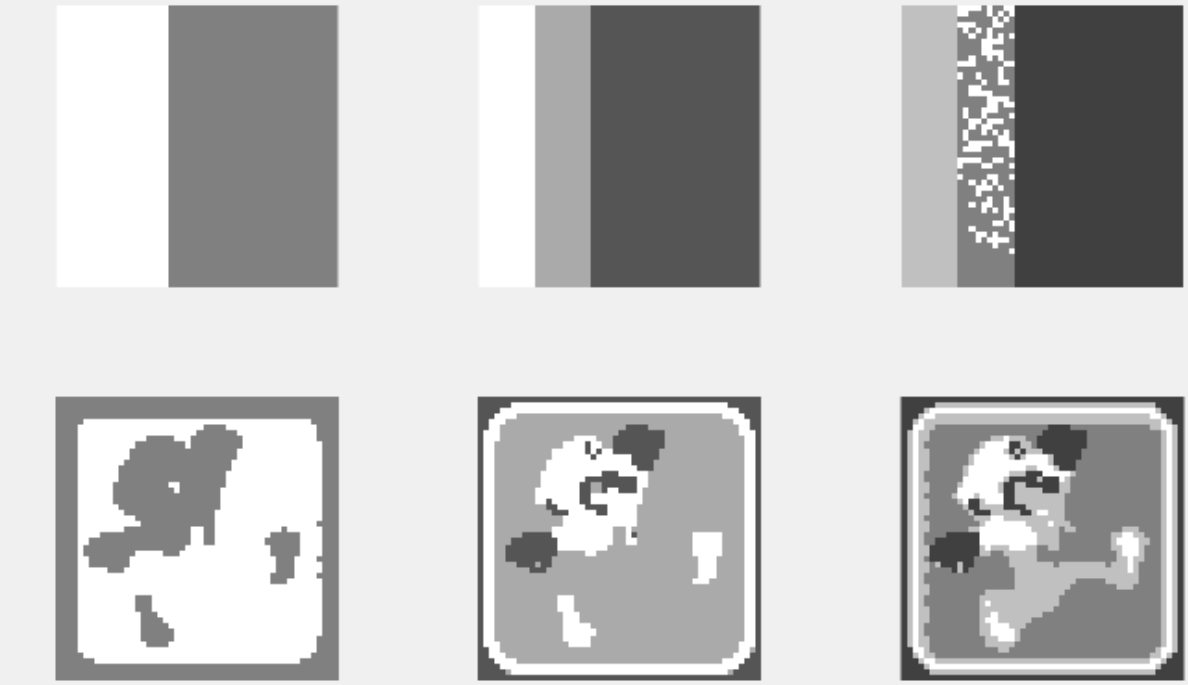
Demo 3a

Στο πρώτο demo 3, καλούμαστε να παρουσιάσουμε τη λειτουργία της συνάρτησης που υλοποιήσαμε, myNCuts(), και στη συνέχεια να παραθέσουμε τυχόν σχόλια συγκρίνοντας τις μεθόδους spectral clustering και μη-αναδρομικής ncuts.

Η λειτουργία παρουσιάζεται ξανά με χρήση των εικόνων d2a και d2b, χρησιμοποιώντας φυσικά πρώτα τη συνάρτηση Image2Graph() για την παραγωγή των affinity πινάκων.

Αρχικά παρατίθονται τα 6 αποτελέσματα της μη-αναδρομικής εκδοχής για k = 2, 3, 4:

k = 2 k = 3 k = 4



Στη συνέχεια παρατίθενται οι εικόνες με την μέθοδο spectral clustering (πάνω) και με την μη-αναδρομική μέθοδο Ncuts (κάτω), για χάρη σύγκρισης. Υλοποιούνται σε ξεχωριστό demo που δεν αποτελεί κομμάτι των αρχείων που απεστάλησαν.

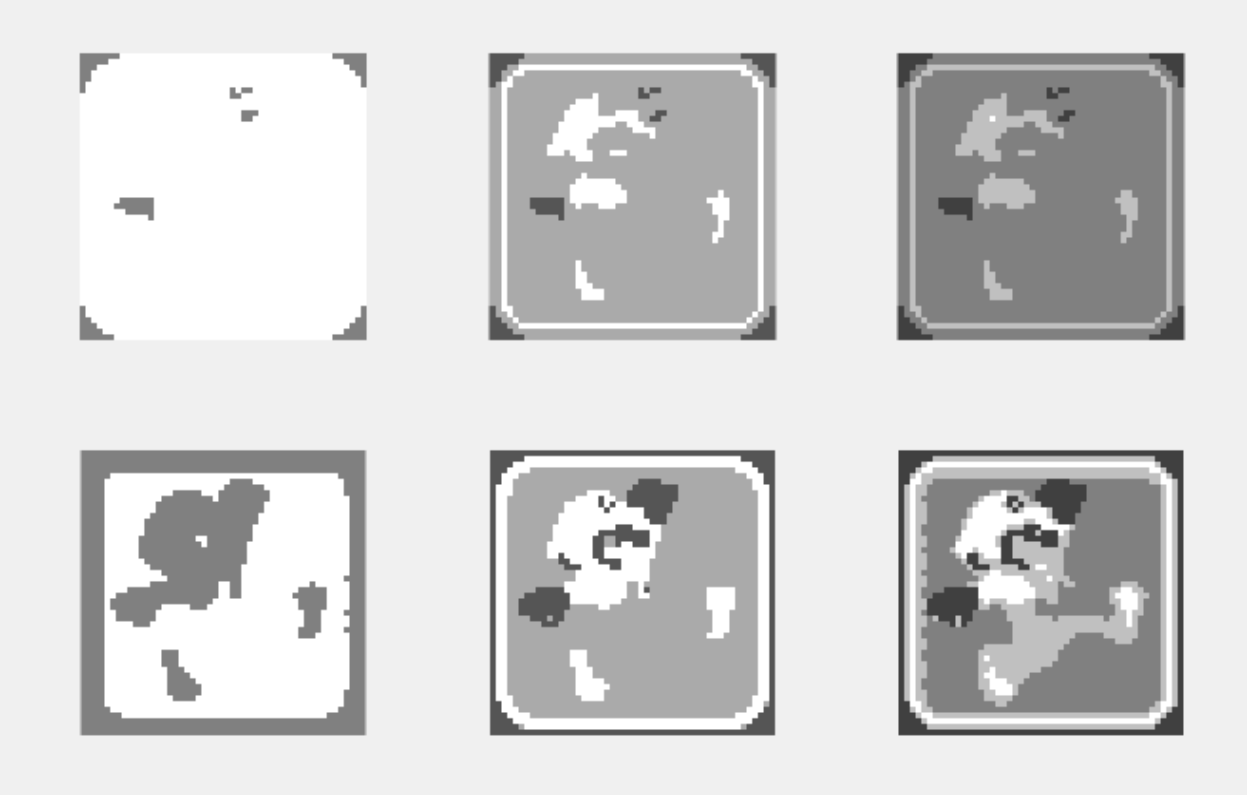
Εικόνα d2a

k = 2 k = 3 k = 4

Παρατηρείται πως το αποτέλεσμα για k = 2 και k = 3 είναι ίδια, ενώ για k = 4 το αποτέλεσμα διαφοροποιείται. Πιο συγκεκριμένα, φαίνεται ότι στη μέθοδο ncuts τα «λάθος» πίξελ εμφανίζονται σε ένα μόνο cluster, το οποίο μπορεί ίσως να δικαιολογηθεί αν σκεφτούμε ότι γίνεται προσπάθεια διχοτόμησης μόνο σε αυτό το cluster από τον αλγόριθμο kmeans, ενώ στην 1η περίπτωση γίνεται διάσπαση των 2 clusters που πρϋπήρχαν σε 3 (άρα εισάγεται σφάλμα σε 2 clusters). Παρ'όλ'αυτά, η δικαιολόγηση αυτή δεν έχει κάποια βάση παραπάνω.

Εικόνα d2b

k = 2 k = 3 k = 4



Στην δεύτερη εικόνα παρατηρείται πως τα αποτελέσματα είναι πολύ πολύ καλύτερα με την μη-αναδρομική μέθοδο ncuts σε σχέση με την μέθοδο του spectral clustering, σε όλες τις περιπτώσεις (k = 2, 3, 4).

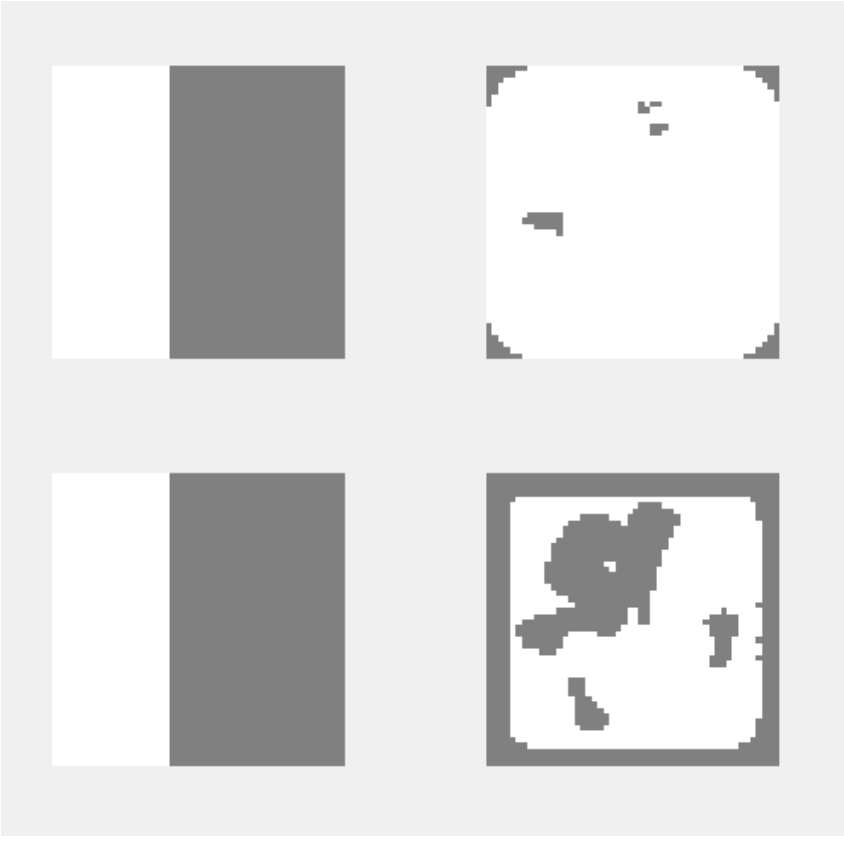
Demo 3b

Στο δεύτερο demo 3 καλούμαστε να τρέξουμε την αναδρομική εκδοχή της μεθόδου ncuts για *μία* μόνο φορά, πρακτικά δηλαδή να τρέξουμε την μη αναδρομική εκδοχή της, και να παρουσιάσουμε την τιμή ncut που προκύπτει για κάθε μία από τις δύο εικόνες. Τα clusters δηλαδή που θα δημιουργηθούν θα είναι μόνο δύο.

Η τιμές ncut που προκύπτουν για τις δύο υλοποιήσεις (με k = 2) είναι οι εξής:

Ncut = 0.5092 Ncut = 0.7853

Συγκριτικά με τη διαδικασία spectral clustering για k = 2, δεν παρατηρούνται διαφορές στην πρώτη εικόνα. Στην δεύτερη εικόνα, από την άλλη, όπου υπάρχουν αρχικά περισσότερες λεπτομέρειες, η μέθοδος ncuts φαίνεται να παρουσιάζει πολύ καλύτερα αποτελέσματα, όπως φαίνεται και παρακάτω.



Demo 3c

Στο τρίτο και τελευταίο demo 3 καλούμαστε να τρέξουμε την ολοκληρωμένη αναδρομική εκδοχή της μεθόδου ncuts, δίνοντας ως ορίσματα τα δοσμένα από την εκφώνηση T1 και T2. Επειδή για T2 = 0.20 δεν γίνεται περαιτέρω διάσπαση (εκτός της πρώτης), επιλέγουμε αιθαίρετα για την πρώτη εικόνα d2a το T2 = 0.51 και για τη δεύτερη εικόνα d2b το T2 = 0.79, ώστε να φανεί πιο ξεκάθαρα η αναδρομική διάσπαση.

Τα πρώτα αποτελέσματα με την αναδρομή όπως έχει περιγραφεί στην εκφώνηση (δηλαδή διάσπαση στα 2 κάθε φορά που ncut < T2 για την προηγούμενη επανάληψη) παρουσιάζονται παρακάτω:

Ο αλγόριθμος φαίνεται να δουλεύει σχετικά καλά, αξίζει όμως να σημειωθεί πως στην πρώτη εικόνα συνεχίζουν να υπάρχουν προβλήματα διάσπασης και «τυχαία» σφάλματα όταν προσπαθεί να εισαχθεί κάποια απόχρωση που δεν προϋπήρχε.

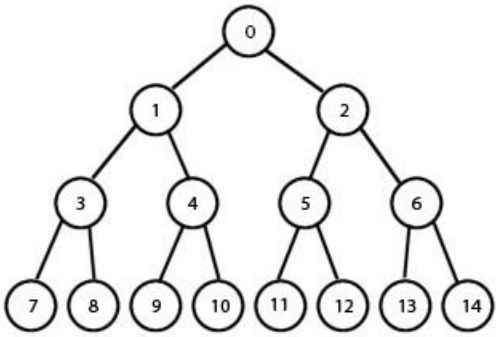
Ο λόγος που συμβαίνει το παραπάνω είναι ότι, αφού γίνει ο έλεγχος για το ncut π.χ. για τα αρχικά cluster 1 και 2 (και ο έλεγχος μέσω του T1 φυσικά), γίνεται κατευθείαν η διάσπαση τους σε 2 clusters το καθένα, δηλαδή σύνολο 4. Υπάρχει όμως περίπτωση να χρειάζεται διάσπαση με σύνολο 3 clusters, όπως συμβαίνει στην πρώτη εικόνα.

Σε αυτή την περίπτωση, εισάγουμε μια μικρή διαφοροποίηση στον κώδικα. Πιο συγκεκριμένα, ο έλεγχος για το T2 συμβαίνει αμέσως μετά το αποτέλεσμα του αλγόριθμου kmeans, και αν δεν περάσει ο έλεγχος σε εκείνο το σημείο τότε, αντί

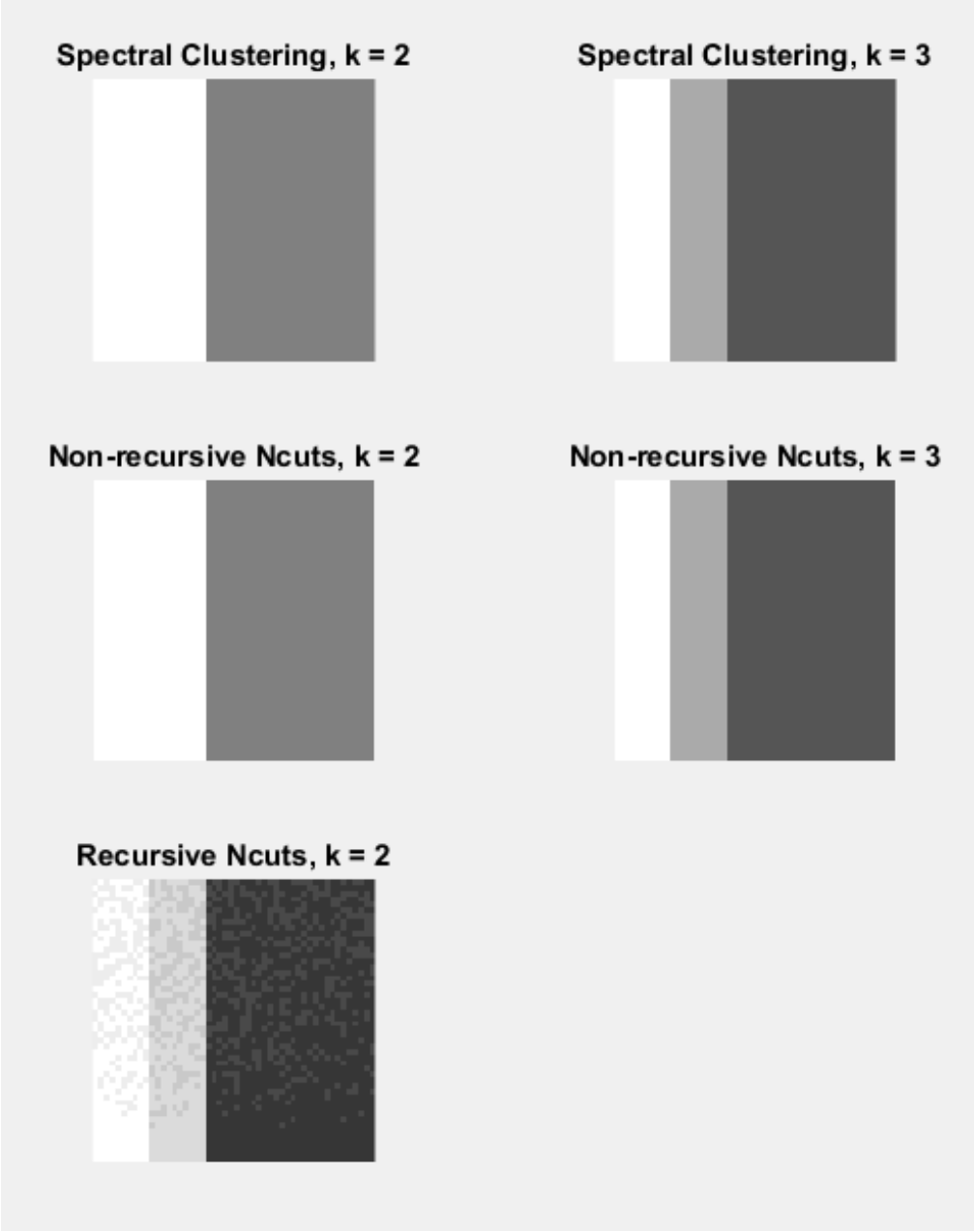
να ελεγχθούν τα clusters για το T1, «αναιρείται» η διάσπαση που προηγήθηκε, καθώς δίνεται σε κάθε ένα από τα 2 clusters η ίδια τιμή. Με αυτό τον τρόπο, διορθώνονται τα σφάλματα που προκύπτουν όταν μία από τις 2 διασπάσεις τελικά δεν «έπρεπε» να γίνει εξαρχής.

Τα αποτελέσματα του παραπάνω παρουσιάζονται εδώ:

Αξίζει να σημειωθεί βέβαια ότι, για χάρη «διόρθωσης» σε περιπτώσεις όπως αυτή που προαναφέρθηκε, φαίνεται να χάνεται σημαντική πληροφορία σε εικόνες όπως η $2^{η}$ (όπου τα clusters που προκύπτουν στην πρώτη περίπτωση είναι πολύ περισσότερα). Το ορθό λοιπόν εδώ είναι λίγο πιο «υποκειμενικό», καθώς πρέπει να αποφασίσουμε ποια εκδοχή θα χρησιμοποιηθεί ανάλογα με το τι τελικά θέλουμε από το αποτέλεσμα (ορθότητα αποχρώσεων, λεπτομέρεια, ακρίβεια), και φυσικά ανάλογα με τη μορφή των εικόνων εισόδου. Εδώ σημειώνεται ότι ο κώδικας που έχω αποστείλει δημιουργεί μία εικόνα στην οποία οι τιμές των αποχρώσεων αποτελούν τα «τελικά» φύλλα του παρακάτω δένδρου. Για παράδειγμα, για την εικόνα d2a οι αποχρώσεις της προκύπτουσας εικόνας (για τον κλασσικο ncuts με τα «σφάλματα») είναι οι: 3, 4, 11, 12, 13, 14.

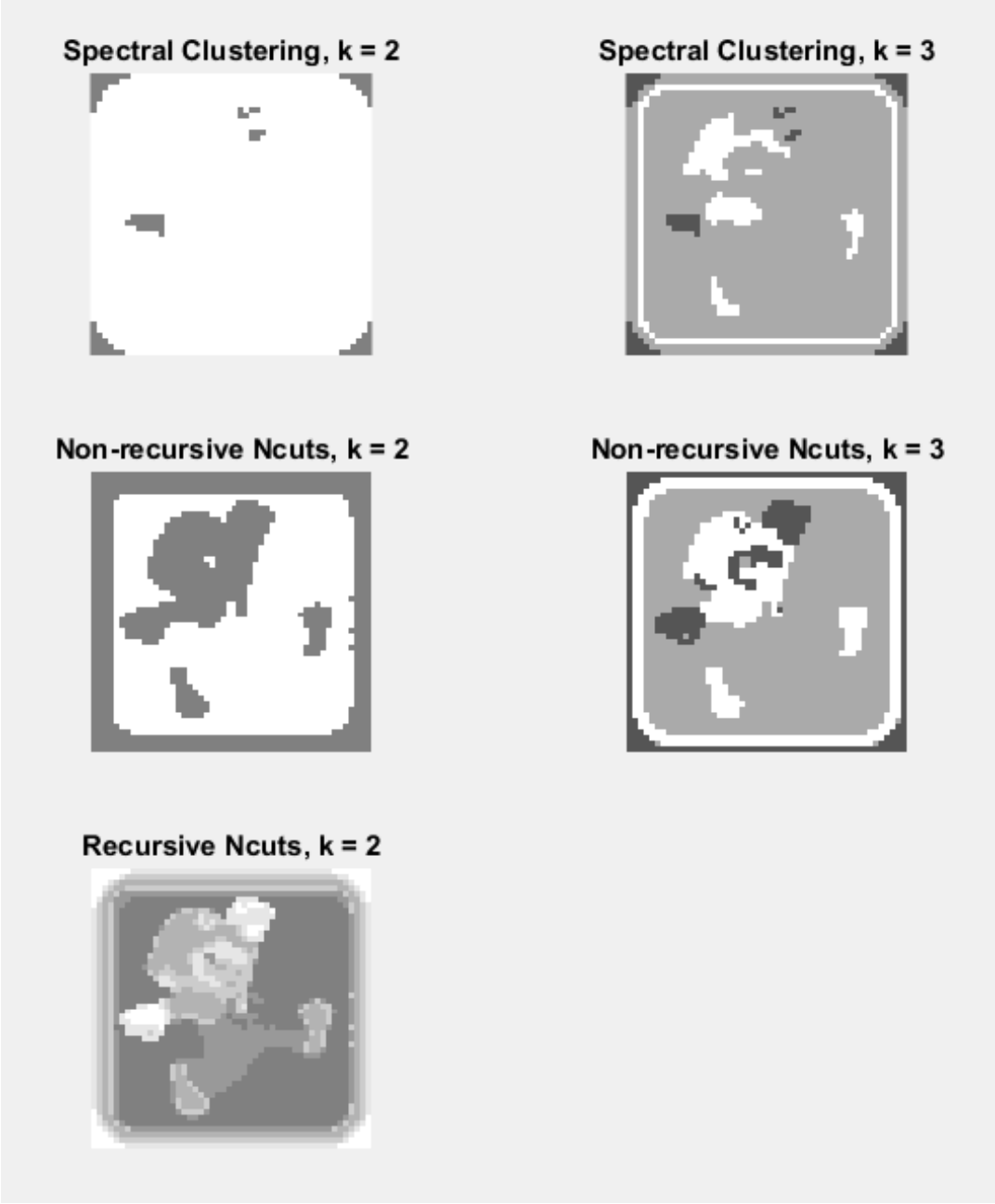


Στη συνέχεια παρουσιάζεται η σύγκριση της 1ης/κλασσικής εκδοχής ncuts με τη διαδικασία spectral clustering για k = 2 και k = 3, καθώς και με την μη αναδρομική διαδικασία ncuts για k = 2 και k = 3 για την εικόνα d2a.



Η μέθοδος του spectral clustering και η μη αναδρομική ncuts έχουν ακριβώς τα ίδια αποτελέσματα, όπως είναι και αναμενόμενο. Η αναδρομική εκδοχή της μεθόδου ncuts, από την άλλη, εμφανίζει κάποια προβλήματα που αναλύθηκαν επαρκώς παραπάνω.

Τέλος, παρουσιάζεται η σύγκριση της 1ης/κλασσικής εκδοχής ncuts με τη διαδικασία spectral clustering για k = 2 και k = 3, καθώς και με την μη αναδρομική διαδικασία ncuts για k = 2 και k = 3 για την εικόνα d2b.



Σε αυτή την περίπτωση, μπορούμε να δούμε πολύ καθαρά την βελτίωση από κάθε μέθοδο στην επόμενη. Οι μέθοδοι spectral clustering και non-recursive ncuts για k = 2 έχουν τα χειρότερα αποτελέσματα, το οποίο είναι και αναμενόμενο λόγω της λεπτομέρειας της συγκεκριμένης εικόνας.

Αδιαμφισβήτητα, το καλύτερο αποτέλεσμα έχει η αναδρομική εκδοχή της μεθόδου ncuts (εδώ τρέχει για T2 = 0.79 και T1 = 5). Η λεπτομέρεια δίνεται σε ικανοποιητικό βαθμό σε σχέση με τις υπόλοιπες διαδικασίες (να σημειωθεί πως εδώ δημιουργούνται k = 6 clusters μέσω της αναδρομής).