

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών

Ψηφιακή Επεξεργασία Εικόνας

Εργασία 2^η – Εαρινό Εξάμηνο 2020/2021

Image Segmentation



Καβελίδης Φραντζής Δημήτριος – ΑΕΜ 9351

Επικοινωνία: kavelids@ece.auth.gr

23/5/2021

Θέμα:

Η δεύτερη εργασία του μαθήματος αφορά την υλοποίηση των παρακάτω:

1. Αναπαράσταση των εικόνων ως γράφους
2. Image Segmentation με τη μέθοδο *spectral clustering*
3. Image Segmentation με τη μέθοδο *normalized cuts* (n-cuts / αναδρομικά και μη αναδρομικά)

Τα δεδομένα για τους σκοπούς της εργασίας, δηλαδή ο ένας γράφος – Affinity πίνακας και οι 2 εικόνες, δίνονται στο αρχείο `dip_hw_2.mat`

Ανάλυση:

1. Εικόνες ως γράφοι

Στο πρώτο μέρος της εργασίας καλούμαστε να φτιάξουμε μία συνάρτηση (**Image2Graph**) η οποία παίρνει σαν είσοδο μία εικόνα διαστάσεων $M \times N$ και επιστρέφει στην έξοδο τον Affinity πίνακα που περιγράφει έναν μη κατευθυντικό fully connected γράφο $G = (V, E)$ χρησιμοποιώντας ως μετρική-συνάρτηση κόστους την

$$A(i, j) = \frac{1}{e^{d(i, j)}}$$

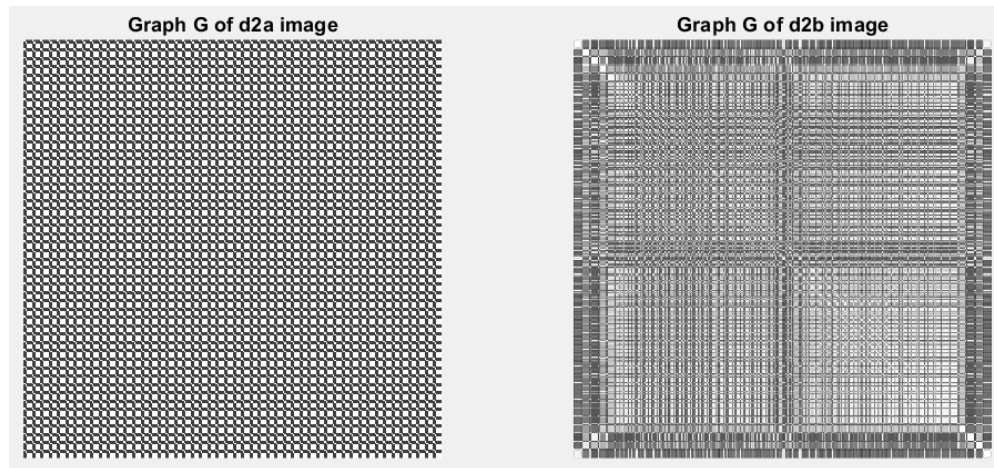
όπου $d(i, j)$ η Ευκλείδεια απόσταση της φωτεινότητας μεταξύ των καναλιών μεταξύ του i -οστού και του j -οστού pixel της εικόνας.

Υλοποίηση / Λειτουργία συνάρτησης Image2Graph:

Αρχικά, παίρνουμε τις διαστάσεις της εικόνας και φτιάχνουμε έναν άδειο πίνακα με διαστάσεις $(M \times N) \times (M \times N)$. Στη συνέχεια μέσα σε 4 for loops φτιάχνουμε τα indices για να γεμίσουμε τα κελιά του πίνακα με τα βάρη της συνάρτησης, συγκρίνοντας κάθε φορά το i -pixel με το j -pixel. Για λόγους ταχύτητας εκτέλεσης, εκμεταλλευόμαστε την ιδιότητα συμμετρίας του πίνακα Affinity ώστε να μην τρέξουμε όλες τις εντολές σε κάθε for.

Σημείωση: Η παραπάνω υλοποίηση είναι σίγουρα η λιγότερο φιλοσοφημένη, αλλά δοκιμάζοντας και άλλες υλοποιήσεις όπως η μετατροπή της εικόνας σε πίνακα cell ή χρησιμοποιώντας κάποιες έτοιμες ρουτίνες της MATLAB, φαίνεται πως για τις μικρές διαστάσεις των εικόνων της παρούσας εργασίας είναι η γρηγορότερη υλοποίηση (από αυτές που δοκιμάστηκαν), παρόλο που υστερεί σε χρόνο στην μετατροπή μεγάλων εικόνων.

Ενδεικτικά, παρουσιάζεται η απεικόνιση των γραφών που προκύπτουν για τις εικόνες d2a, d2b:



Εικόνα 1 – Απεικόνιση των γραφών των εικόνων d2a, d2b

Σαφώς, η απεικόνιση των τιμών των γραφών ως εικόνα δεν έχει κάποια ουσιαστική σημασία για τις ανάγκες της εργασίας μας.

2. Spectral Clustering

Για να κάνουμε κατάτμηση της εικόνας μας με την μέθοδο *spectral clustering* ακολουθούμε τα παρακάτω βήματα:

- i. Κατασκευή του μη κατευθυντικού γραφού G με τη χρήση της `Image2Graph.m` ο οποίος έχει ως έξοδο τον Affinity πίνακα W .
- ii. Υπολογισμός του Λαπλασιανού πίνακα $L = D - W$, με $D(i, i) = \sum_j W(i, j)$.
- iii. Υπολογισμός των k μικρότερων ιδιοτιμών του L και των αντίστοιχων ιδιοδιανυσμάτων.
- iv. Σχηματισμός του πίνακα ιδιοδιανυσμάτων U .
- v. Ομαδοποίηση με τον αλγόριθμο k -means.

Υλοποίηση της συνάρτησης `mySpectralClustering`:

Η συνάρτηση παίρνει ως όρισμα τον Affinity πίνακα από την έξοδο της `Image2Graph`, καθώς και τον αριθμό των clusters (k) που θέλουμε να σχηματιστούν. Για την υλοποίηση λοιπόν ακολουθούμε τα παραπάνω βήματα ii.-v.

Demo 1

Στο συγκεκριμένο αρχείο (`demo1.m`) παρουσιάζονται τα αποτελέσματα της συνάρτησης `mySpectralClustering` χρησιμοποιώντας τον Affinity πίνακα `d1a` του δοσμένου αρχείου για $k = 2, k = 3, k = 4$. Εύκολα βλέπουμε (χρησιμοποιώντας την συνάρτηση `size`) ότι ο `d1a` έχει διαστάσεις 12×12 , επομένως καταλαβαίνουμε ότι οι διαστάσεις της εικόνας που (θεωρητικά) αναπαριστά, έχουν γινόμενο το 12. Αυθαίρετα, για visualization θα θεωρήσουμε ότι η αρχική

εικόνα είχε διαστάσεις 4·3 εφόσον δεν γνωρίζουμε τις πραγματικές διαστάσεις. Προφανώς αυτή η διάταξη δεν είναι μοναδική.

Αρχικά το όρισμα που χρησιμοποιήθηκε κατά την εύρεση των μικρότερων ιδιοτιμών ήταν το ‘smallestreal’.

Τα αποτελέσματα φαίνονται παρακάτω:

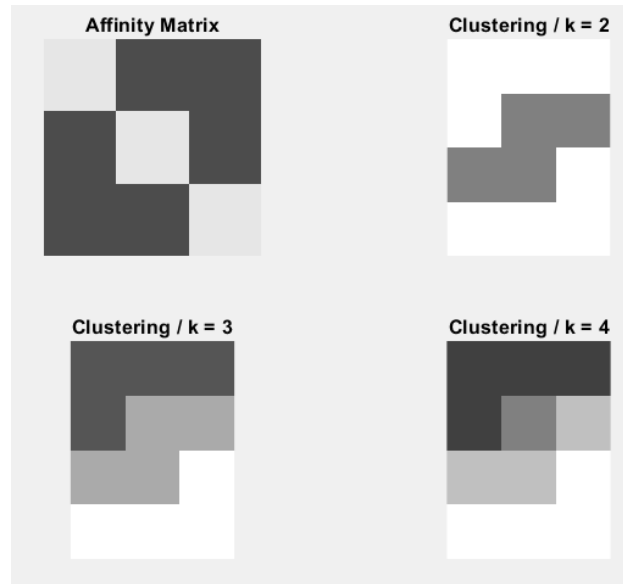


Εικόνα 2 – Demo 1 / Visualization / ‘smallestreal’

Οι δε τιμές στην κονσόλα είναι:

```
Time for mySpectralClustering to run with k = 2 : Elapsed time is 0.256300 seconds.
ans =
    1    1    1    1    1    1    1    1    2    2    2    2
Time for mySpectralClustering to run with k = 3 : Elapsed time is 0.031158 seconds.
ans =
    1    1    1    1    2    2    2    2    3    3    3    3
Time for mySpectralClustering to run with k = 4 : Elapsed time is 0.028526 seconds.
ans =
    1    1    1    1    2    3    3    3    4    4    4    4
```

Στη συνέχεια όμως, χρησιμοποιήθηκε τελικά το όρισμα `diff (= eps)` που φαίνεται να είναι το σωστότερο και αυτό που προτείνεται στη βιβλιογραφία:



Εικόνα 3 – Demo 1 / Visualization / $diff (= eps)$

Οι δε τιμές στην κονσόλα:

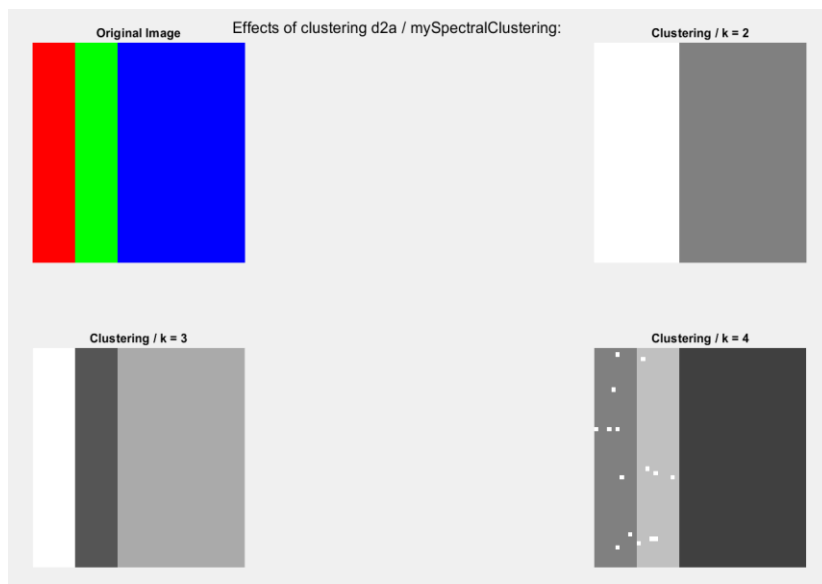
```
Time for mySpectralClustering to run with k = 2 : Elapsed time is 0.259140 seconds.
ans =
     2     2     2     2     1     1     1     1     2     2     2     2
Time for mySpectralClustering to run with k = 3 : Elapsed time is 0.029429 seconds.
ans =
     1     1     1     1     2     2     2     2     3     3     3     3
Time for mySpectralClustering to run with k = 4 : Elapsed time is 0.023606 seconds.
ans =
     1     1     1     1     2     3     3     3     4     4     4     4
```

Όπως φαίνεται, το πόσες αποχρώσεις-ομάδες-clusters βλέπουμε είναι ίσο με τον αριθμό k , επομένως πετυχαίνουμε όντως την κατάντηρηση των δεδομένων που αντιπροσωπεύει ο γράφος (έστω της εικόνας 4.3).

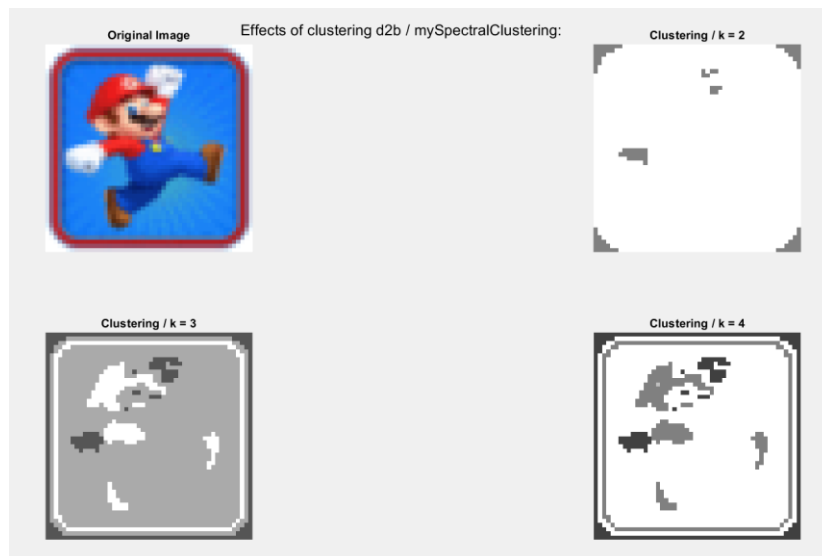
Demo 2

Στο συγκεκριμένο αρχείο (demo2.m) παρουσιάζονται τα αποτελέσματα της συνάρτησης `mySpectralClustering` χρησιμοποιώντας τις εικόνες `d2a` `d2b` του αρχείου για $k = 2$, $k = 3$, $k = 4$.

Τα αποτελέσματα φαίνονται παρακάτω:



Εικόνα 4 – Clustering στην d2a για $k=2,3,4$ / Spectral Clustering / Μετρική 1



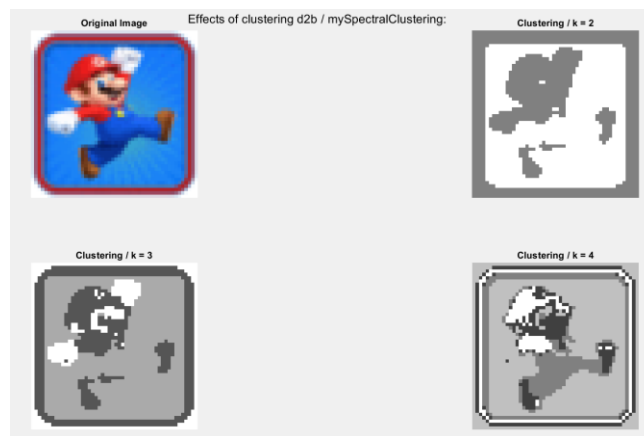
Εικόνα 5 – Clustering στην d2b για $k=2,3,4$ / Spectral Clustering / Mario 1 / Μετρική 1

Στην πρώτη εικόνα, εύκολα παρατηρούμε ότι για $k = 2$, $k = 3$ η συνάρτηση χωρίζει τέλεια σε clusters, θεωρώντας το πράσινο και το κόκκινο μαζί ως μία ομάδα (cluster) και το μπλε ξεχωριστά (για $k = 2$), και χωρίζοντας τέλεια σε 3 χρώματα (για $k = 3$). Όταν τώρα σε μία εικόνα που έχει μόνο 3 χρώματα κάνουμε clustering με $k = 4$, θα παρατηρήσουμε κάποια pixel να σκορπίζονται στην εικόνα. Αυτό είναι λογικό γιατί για $k > \# \text{χρωματικές_περιοχές_εικόνας}$ ο αλγόριθμος δεν έχει κάποιο σαφές κριτήριο για να δημιουργήσει την 4^η ομάδα κι έτσι υπάρχει μία τυχαία κατανομή αυτών των pixels στην εικόνα για να «χωρέσει» αυτό το 4^ο group.

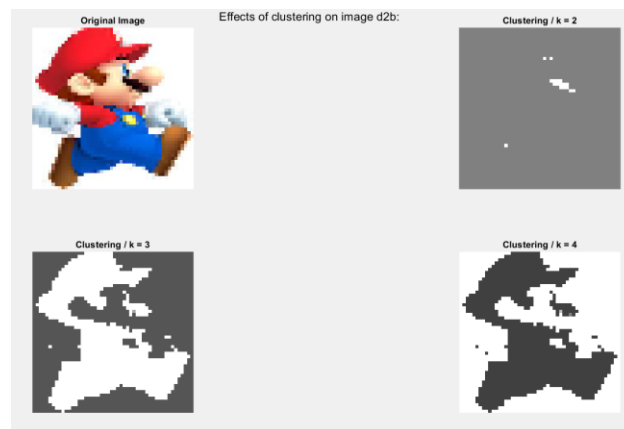
Εν γένει, ο αλγόριθμος που χρησιμοποιούμε κάνει κατάτμηση της εικόνας χρησιμοποιώντας κάποια μετρική που βρίσκει μεγάλες αλλαγές από το ένα pixel στο άλλο στην φωτεινότητα εννοώντας ότι

ελέγχει το βάρος του κάθε pixel μέσω του γράφου και ανάλογα με αυτό ομαδοποιεί τα pixels στα clusters. Αυτό βέβαια σημαίνει ότι για διαφορετική μετρική / συνάρτηση κόστους που θα χρησιμοποιήσουμε πριν την κατάτμηση θα πάρουμε και διαφορετικά αποτελέσματα στο clustering.

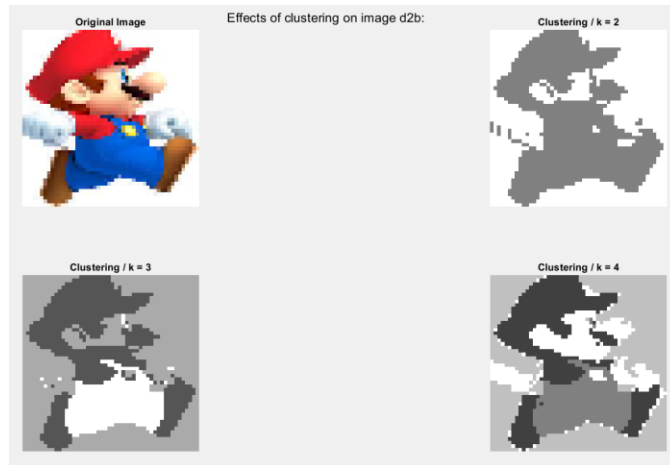
Αν για παράδειγμα αλλάξουμε τον τρόπο που ορίζουμε τον διαγώνιο πίνακα D , και τον ορίσουμε ως $D(i, i) = \sum_i W(i, i)$, δηλαδή τον διαγώνιο που περιέχει τα διαγώνια στοιχεία του Affinity πίνακα, τότε ο Λαπλασιανός πίνακας $L = D - W$ θα έχει στην διαγώνιο μηδενικά στοιχεία/βάρη. Αυτό ουσιαστικά είναι μία τροποποίηση της μετρικής που χρησιμοποιούμε. Τα αποτελέσματα που παίρνουμε με αυτήν είναι αρκετά ενδιαφέροντα. Επίσης εξ' ορισμού, επειδή παίρνουμε μόνο τα διαγώνια στοιχεία, ο χρόνος εκτέλεσης με αυτόν τον πίνακα είναι σημαντικά γρηγορότερος. Σημειώνεται ότι κατά την χρήση της συγκεκριμένης μετρικής χρησιμοποιούμε την eigS με όρισμα 'smallestreal' αντί του diff (= eps) που χρησιμοποιούμε με την μετρική της εκφώνησης. Παρακάτω φαίνονται τα πειράματα και η σύγκριση αυτής της μετρικής (έστω **Μετρική 2**) με αυτήν που προτείνεται στην εκφώνηση (έστω **Μετρική 1**) σε διάφορες εικόνες:



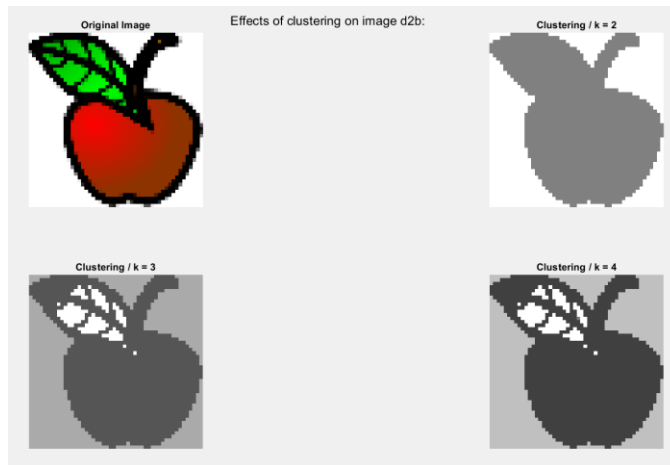
Εικόνα 6 - mySpectralClustering / Μετρική 2 / Mario 1



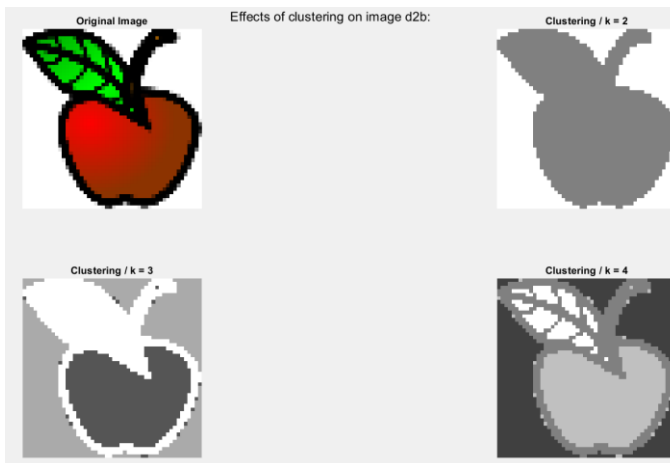
Εικόνα 7 - mySpectralClustering / Μετρική 1 / Mario 2



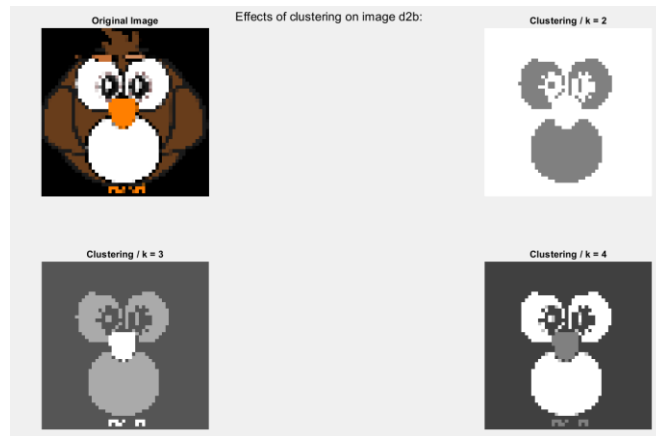
Εικόνα 8 - mySpectralClustering / Μετρική 2 / Mario 2



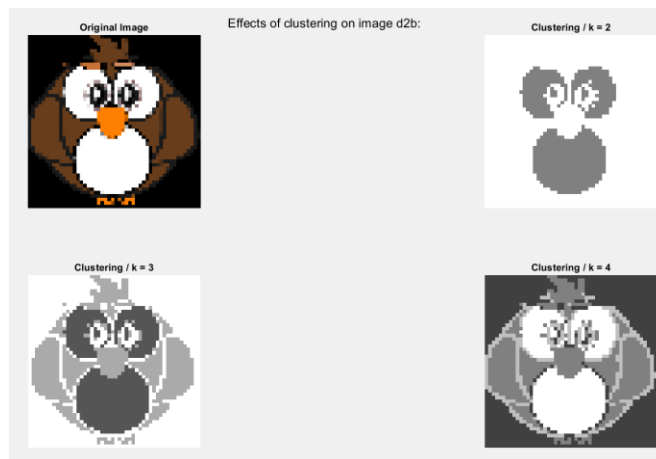
Εικόνα 9 - mySpectralClustering / Μετρική 1 / Μήλο



Εικόνα 10 - mySpectralClustering / Μετρική 2 / Μήλο



Εικόνα 11 - mySpectralClustering / Μετρική 1 / Κουκουβάγια



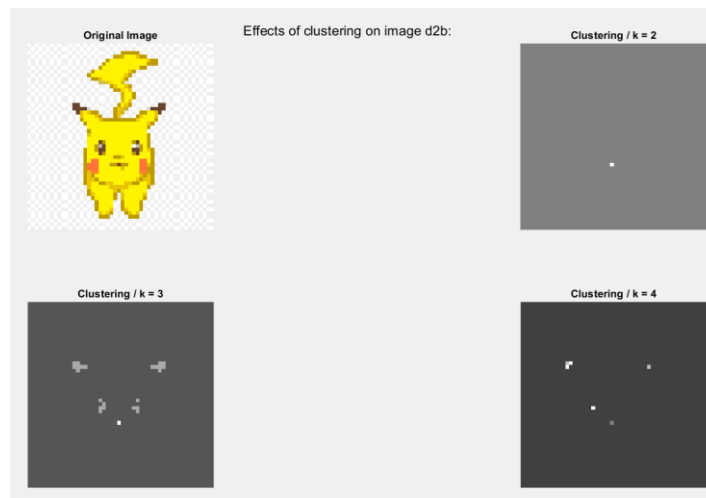
Εικόνα 12 - mySpectralClustering / Μετρική 2 / Κουκουβάγια



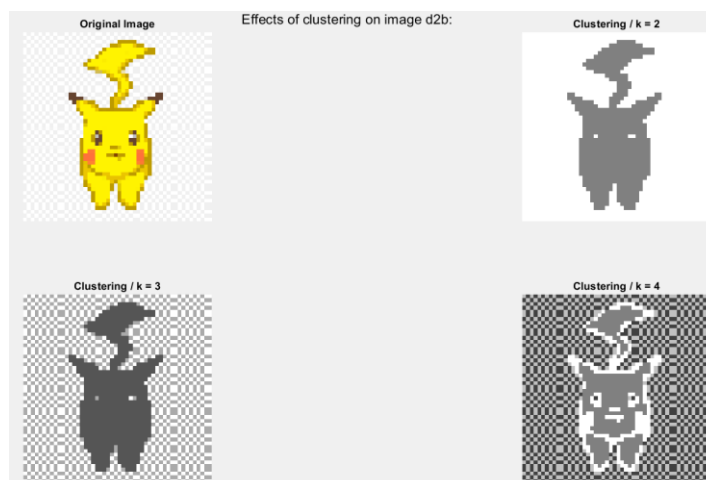
Εικόνα 13 - mySpectralClustering / Μετρική 1 / Λουλούδια 1^{ης} εργασίας



Εικόνα 14 - *mySpectralClustering* / Μετρική 2 / Λουλούδια 1^{ης} εργασίας



Εικόνα 15 - *mySpectralClustering* / Μετρική 1 / Πικάτσου



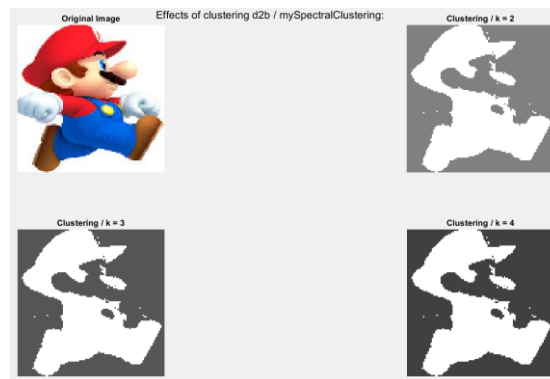
Εικόνα 16 - *mySpectralClustering* / Μετρική 2 / Πικάτσου

Για τις ανάγκες αυτής της εργασίας χρησιμοποιήσαμε μικρές εικόνες των 50*50 pixels. Γενικά, φαίνεται πως κάποιες μετρικές αποδίδουν καλύτερα σε κάποια είδη εικόνων από κάποιες άλλες, έτσι η Μετρική 2 φαίνεται να ανταποκρίνεται καλύτερα σε όλες τις εικόνες που δοκιμάστηκαν σε αυτό το μέγεθος. Αυτό που φαίνεται ωστόσο με αυτήν είναι πως η συγκεκριμένη Μετρική (2) φαίνεται στα πειράματά μας να βρίσκει καλύτερα τα περιγράμματα των εικόνων, παρόλο που αυτό δεν είναι ένα συμπέρασμα που μπορούμε να γενικεύσουμε. Σε κάποιες εικόνες τα αποτελέσματα είναι σημαντικά καλύτερα, ενώ σε άλλες δεν διαφέρουν πολύ. Για να είμαστε απόλυτα συνεπείς, μπορεί καταχρηστικά να χρησιμοποιούμε τον όρο Spectral Clustering και την myspectralclustering για να χρησιμοποιήσουμε την μετρική μας (για λόγους ευκολίας), αλλά στην πραγματικότητα, έχουμε αλλάξει ενεργά την μέθοδο, δηλαδή πλέον δεν μιλάμε για «κλασικό» Spectral Clustering. Να σημειωθεί επίσης ότι πολλά από τα παραπάνω πειράματα έχουν γίνει χωρίς την `rng(1)` δηλαδή τον έλεγχο της τυχαιότητας καθώς έγινε έρευνα μίας «μέσης» συμπεριφοράς.

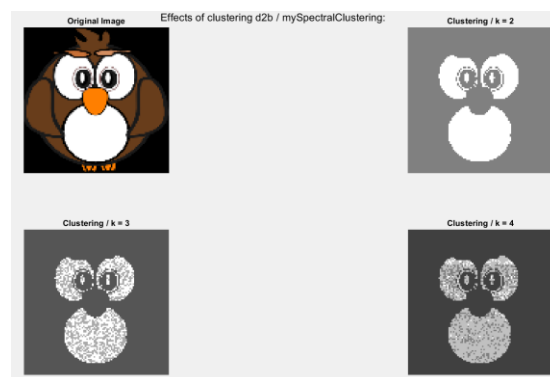
Άλλες μετρικές που μπορούν να χρησιμοποιηθούν [1] είναι:

1. $A(x, y) = \frac{1}{e^{d(x, y)}}$, με $d(x, y) = \sum_i^n |x_i - y_i|^2$ (Squared Euclidean Distance)
2. $d(x, y) = 1 - (\cos^{-1}(\text{similarity})/\pi)$
 $\text{similarity} = \frac{x \cdot y}{||x|| \cdot ||y||}$ (Cosine Similarity)
3. $A(x, y) = \frac{1}{e^{d(x, y)}}$, με $d(x, y) = \sum_i^n |x_i - y_i|^2 / 2\sigma^2$ (Gaussian Distance)

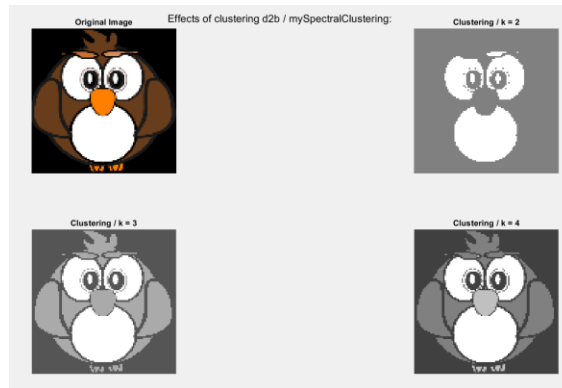
Ενδεικτικά, κάποιοι πειραματισμοί με την Squared Euclidean Distance:



Εικόνα 17 - mySpectralClustering / Μετρική με Squared Euclidean Distance / Mario 2

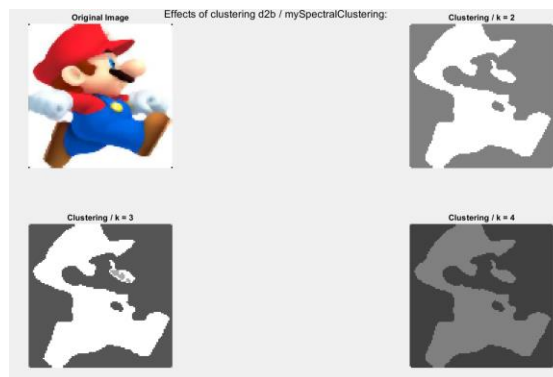


Εικόνα 18 - mySpectralClustering / Μετρική με Squared Euclidean Distance / Mario 2



Εικόνα 19 - mySpectralClustering / Μετρική με Squared Euclidean Distance με $D(i, i) = \sum_i W(i, i)$ / Κουκουβάγια

Πέρα από την επιλογή της μετρικής που χρησιμοποιούμε, μία άλλη συνηθισμένη πρακτική, τουλάχιστον για κάποιες κατηγορίες εικόνων που ενδέχεται να έχουν αναμενόμενο θόρυβο, είναι το smoothing της εικόνας με low pass filtering. Αυτό βέβαια συνεπάγεται ότι αν θέλουμε μεγάλη ακρίβεια στα γειτονικά pixels, αναμένουμε για παράδειγμα πολλά edges, αυτό δεν είναι απαραίτητα καλή πρακτική. Φυσικά οι πειραματισμοί μπορούν να επεκταθούν και στις τιμές του φίλτρου, για τους πειραματισμούς όμως της εργασίας δημιουργήθηκε η συνάρτηση myFilter με ένα median φίλτρο και σταθερές παραμέτρους. Παρακάτω φαίνονται κάποια από τα αποτελέσματα που φάνηκε να έχουν διαφορά έχοντας κάνει αυτήν την προ-επεξεργασία:



Εικόνα 20 - mySpectralClustering / Μετρική 1- Φιλτραρισμένη εικόνα / Mario 2

Εδώ είναι ξεκάθαρο ότι κάποια pixel που εμφανιζόντουσαν εκεί όπου θα περιμέναμε τα χέρια και ήταν προηγουμένως διάσπαρτα, τώρα έχουν εξαφανιστεί και το clustering είναι πιο ξεκάθαρο στο σώμα του Mario.



Εικόνα 21 - mySpectralClustering / Μετρική 2 - Φιλτραρισμένη εικόνα / Λουλούδια πρώτης εργασίας

Η διαφορά φαίνεται και εδώ, αφού καταλήγουμε σε ένα πιο smooth clustering με λιγότερα σκορπισμένα pixels και πιο ξεκάθαρα clusters.

Ανακεφαλαίωση των «ρυθμίσεων»:

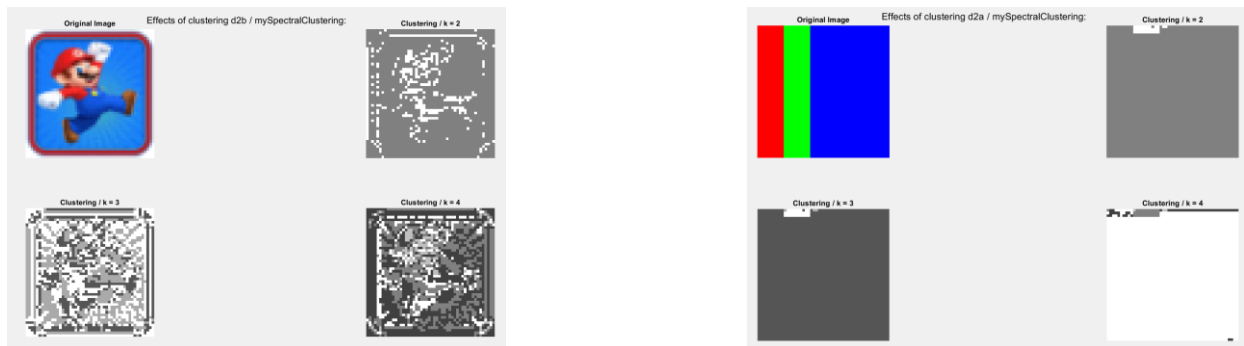
Μετρική 1: Ακολουθούμε τα βήματα όπως στην εκφώνηση με $D(i, i) = \sum_j W(i, j)$ και υπολογισμό των ιδιοτιμών με όρισμα $\text{diff} (= \text{eps})$.

Μετρική 2: Ακολουθούμε τα βήματα όπως στην εκφώνηση με διαφορετικό διαγώνιο πίνακα από αυτόν της εκφώνησης: $D(i, i) = \sum_i W(i, i)$ και υπολογισμό των ιδιοτιμών με όρισμα 'smallestreal'.

Για φιλτράρισμα / low pass filtering : Χρησιμοποιείται η συνάρτηση myFiter – median filter

Για δοκιμές με άλλες μετρικές που αναφέρθηκαν: Ουσιαστικά η Μετρική 1 με άλλη μετρική/απόσταση $d(i, j)$ κατά τον υπολογισμό του Affinity πίνακα.

Σημείωση: Περίεργως, όταν χρησιμοποιήθηκε η Μετρική 2 με όρισμα $\text{diff} (= \text{eps})$ τα αποτελέσματα δεν ήταν καθόλου ικανοποιητικά, ενώ με όρισμα 'smallestreal' ήταν σταθερά πολύ ικανοποιητικά. Αυτός ήταν και ο λόγος που χρησιμοποιήθηκε αυτή η αρχικοποίηση. Δίνεται παρόλα αυτά ενδεικτικά μία εικόνα στην οποία χρησιμοποιήθηκε η Μετρική 2 με όρισμα $\text{diff} (= \text{eps})$:



Εικόνα 22 – Αποτελέσματα με τη Μετρική 2 και όρισμα $\text{diff} (= \text{eps})$ / Αποτυχία

3. Normalized-cuts

Στην ενότητα αυτή, θα κάνουμε κατάτμηση της εικόνας με την χρήση της state-of-the-art μεθόδου, την normalized-cuts ή αλλιώς n-cuts.

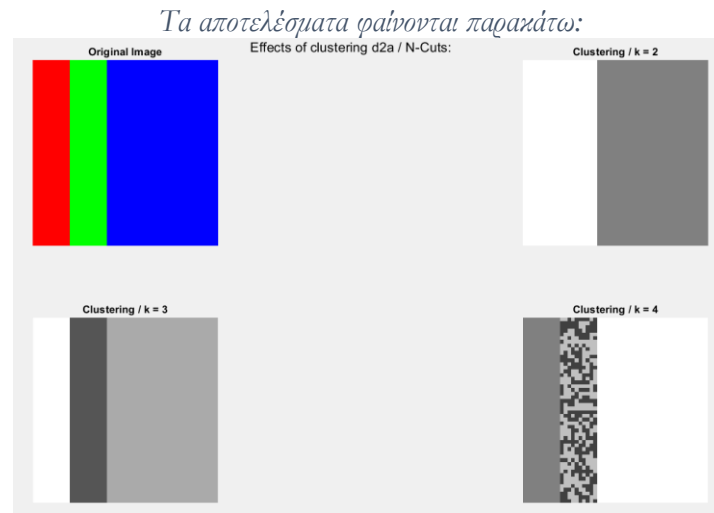
Μη αναδρομική εκδοχή της n-cuts / Λειτουργία συνάρτησης myNCuts:

Η μη αναδρομική εκδοχή έχει ακριβώς τα ίδια βήματα με την μέθοδο Spectral Clustering, με τη διαφορά ότι λύνουμε το **γενικευμένο** πρόβλημα ιδιοτιμών, δηλαδή το $Lx = \lambda Dx$

Demo 3a

Στο συγκεκριμένο αρχείο (demo3a.m) παρουσιάζονται τα αποτελέσματα της συνάρτησης **myNCuts** χρησιμοποιώντας τις εικόνες d2a d2b του αρχείου για $k = 2, k = 3, k = 4$.

Η δε συνάρτηση myNCuts είναι μία υλοποίηση της μη αναδρομικής εκδοχής της μεθόδου n-cuts.



Εικόνα 23 – Clustering στην d2a για $k=2,3,4$ / myNCuts – Μη αναδρομική / Μετρική 1

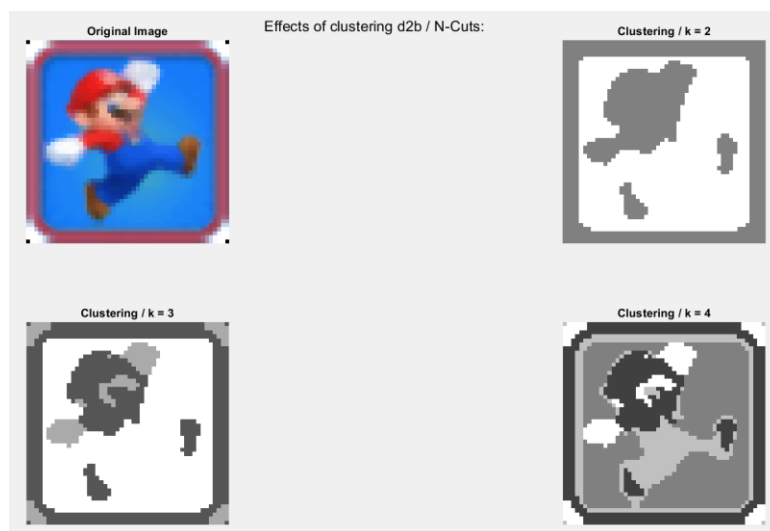


Εικόνα 24 – Clustering στην d2a για $k=2,3,4$ / myNCuts – Μη αναδρομική / Μετρική 1

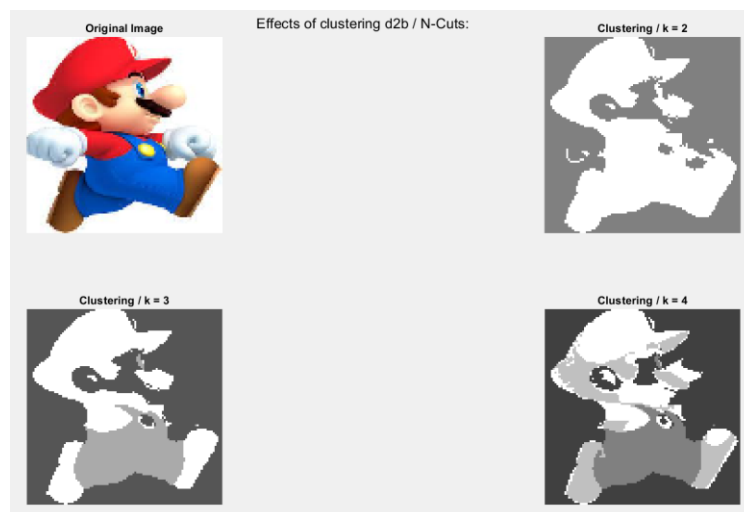
Όπως είναι αναμενόμενο, με την παραδοχή που έχουμε κάνει στην Μετρική 2, δεδομένου ότι ο Affinity πίνακας έχει πάντα στην διαγώνιο όλα τα στοιχεία ίσα με τη μονάδα, ο D θα είναι $D = I_{MN \times MN}$, κι έτσι το γενικευμένο πρόβλημα ιδιοτιμών, στην πραγματικότητα είναι το απλό πρόβλημα ιδιοτιμών:

$$Lx = \lambda Dx \Leftrightarrow Lx = \lambda x.$$

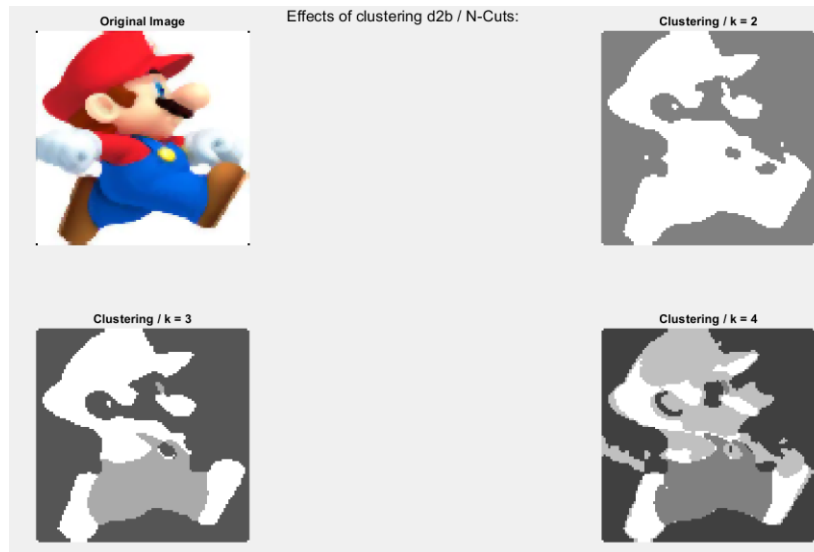
Επομένως, δεν κερδίζουμε τίποτα περισσότερο από την μετρική αυτή (Μετρική 2) χρησιμοποιώντας την ncuts αφού στην πραγματικότητα χρησιμοποιούμε την **mySpectralClustering**. Για την ακρίβεια, χάνουμε μάλιστα σε χρόνο καθώς το γενικευμένο πρόβλημα ιδιοτιμών που απαιτούμε να λυθεί στην MATLAB με την συνάρτηση `eigs` παίρνει (σημαντικά) περισσότερο χρόνο παρόλο που στην ουσία λύνουμε το απλό πρόβλημα ιδιοτιμών. Παρακάτω φαίνονται λοιπόν διάφορα πειράματα μόνο με την Μετρική 1 και low pass filtering/smoothing πριν το clustering:



Εικόνα 25 – Clustering στην d2a για k=2,3,4 / myNCuts – Μη αναδρομική / Μετρική 1/ Φιλτραρισμένη εικόνα



Εικόνα 26 – myNCuts – Μη αναδρομική / Μετρική 1 / Mario 2



Εικόνα 27 – myNCuts – Μη αναδρομική / Μετρική 1 / Mario 2 – Φιτραρισμένη εικόνα

Σύγκριση → Μη αναδρομική N-Cuts -vs- Spectral Clustering:

Είναι σαφές ότι η N-Cuts κάνει πολύ καλύτερη δουλειά από την Spectral Clustering (τουλάχιστον με την Μετρική 1 της εκφώνησης), αφού χωρίζει πολύ πιο ικανοποιητικά σε clusters τις δοσμένες εικόνες. Ειδικά στην εικόνα d2b, τα αποτελέσματα με την μη αναδρομική n-cuts ήταν σημαντικά καλύτερα, αφού ξεχώριζε το σώμα του Mario, ενώ με Spectral Clustering ομαδοποιούνταν μόνο pixels από το κεφάλι και τα άκρα του. Επομένως παρόλο που στην πολύ απλή εικόνα d2a δεν βλέπουμε σημαντική διαφορά (παρά μόνο στον τρόπο με τον οποίο προσπαθεί να χωρέσει το 4^ο cluster σε κάποια από τις χρωματικές ομάδες), σε πιο περίπλοκες εικόνες η N-Cuts μας δίνει πολύ πιο χρήσιμη πληροφορία για την εικόνα με την ομαδοποίηση που κάνει.



Εικόνα 28 - myNCuts – Μη αναδρομική / Μετρική 1 / Λουλούδια πρώτης εργασίας



Εικόνα 29 - myNCuts – Μη αναδρομική / Μετρική 1 / Λουλούδια πρώτης εργασίας / Φιλτραρισμένη εικόνα

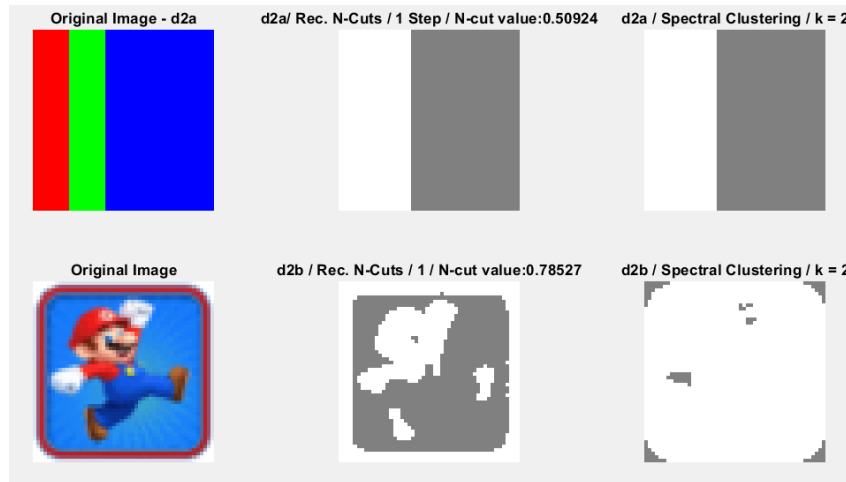
Εύκολα μπορούμε να δούμε κι εδώ ότι η n-cuts υπερτερεί σημαντικά (τουλάχιστον στο σύνολο των φωτογραφιών που έγιναν οι πειραματισμοί). Όχι μόνο ξεχωρίζει καλύτερα, αλλά σε κάποιες εικόνες όπως βλέπουμε έχει κάνει καλύτερο clustering ακόμα και με μικρότερο k .

Αναδρομική N-Cuts / Λειτουργία της συνάρτησης myNCutsRecursive:

Η αναδρομική εκδοχή της μεθόδου τρέχει την myNCuts για $k = 2$, δηλαδή ακολουθεί τα πρώτα 5 βήματα και στην σπάει την εικόνα στα 2. Στην συνέχεια, έχοντας υπολογίσει την ncut τιμή γίνεται ο έλεγχος για τα κατώφλια. Αν ικανοποιείται η συνθήκη τότε σταματάει και επιστρέφει τις ετικέτες των 2 clusters, την ncut τιμή και το βάθος/αναγνωριστικό του φύλλου του αναδρομικού δέντρου. Αν δεν ικανοποιείται τότε προχωράει σπάζοντας σε 2 μέρη το cluster το οποίο πέρασε τον έλεγχο και συνεχίζει έτσι αναδρομικά κάνοντας τελικά αλληπάλληλες ενώσεις. Η συγκεκριμένη συνάρτηση λειτουργεί για λόγους ευκολίας και αποτελεσματικότητας με cell arrays ενώ στο τέλος επιστρέφει πάλι ένα διάνυσμα με τις ετικέτες των clusters για να υπάρχει συνέπεια ως προς τις υπόλοιπες συναρτήσεις της εκφώνησης. Για την αναδρομή χρησιμοποιείται μία βοηθητική συνάρτηση σχεδόν πανομοιότυπη με την myNCutsRecursive η οποία λέγεται **PartitionNCuts** και ουσιαστικά τρέχει την αναδρομή της myNCutsRecursive.

Demo 3b

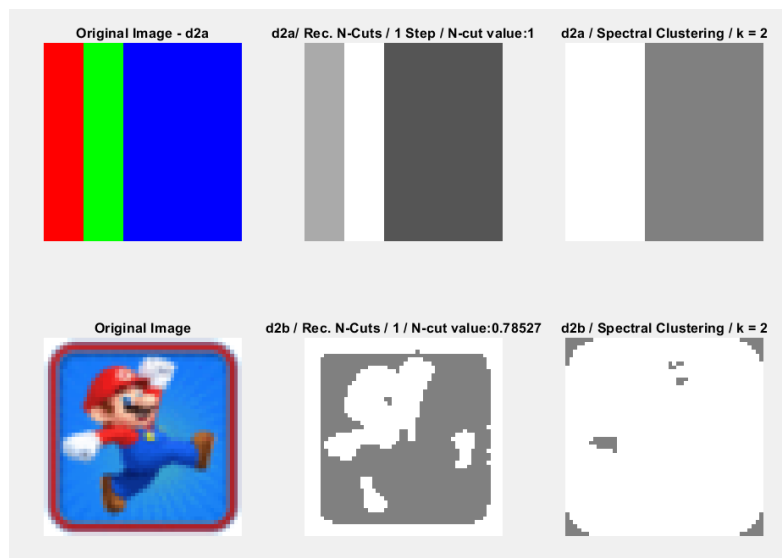
Στο συγκεκριμένο αρχείο (demo3b.m) παρουσιάζονται τα αποτελέσματα της συνάρτησης **myNCutsRecursive**, δηλαδή της αναδρομικής μεθόδου για ένα μόνο βήμα. Όπως είναι σαφές η συνάρτηση αυτή όταν λειτουργεί για ένα μόνο βήμα, έχει ακριβώς την ίδια λειτουργία με την **myNCuts** για $k = 2$. Έτσι, όπως είναι αναμενόμενο, έχουμε ακριβώς τα ίδια αποτελέσματα με αυτά του demo3a για $k = 2$. Στα κατώφλια $T^1 = 5$, $T^2 = 0.2$ που δίνονται, η συνάρτηση θα προχωρήσει μόνο σε ένα βήμα, δηλαδή θα σπάσει μόνο μία φορά σε 2 clusters την εικόνα. Τα αποτελέσματα φαίνονται παρακάτω:



Εικόνα 30 – myNCutsRecursion για $T^2 = 0.2$ / d2a και d2b / N-cuts Recursion -vs- Spectral Clustering

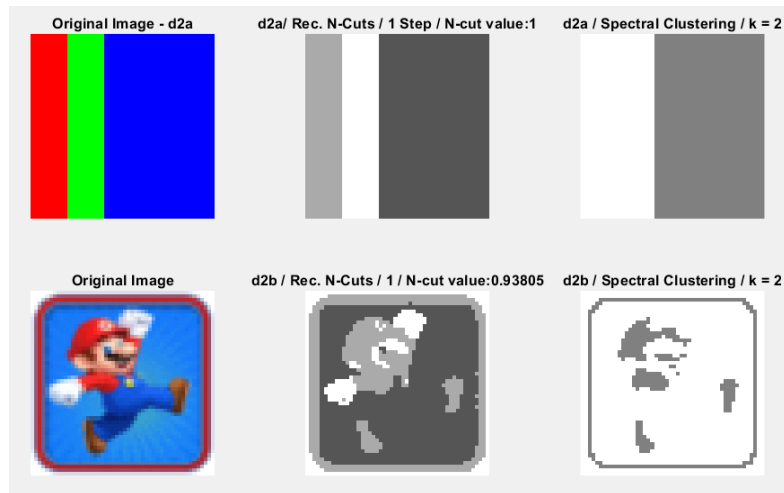
Η n-cuts όπως αναφέρθηκε και προηγουμένως, φαίνεται να ξεχωρίζει καλύτερα τα μέλη του Mario, όπως φαίνεται δηλαδή στην εικόνα, η n-cuts έχει ομαδοποιήσει τα pixels σε όλο το πάνω μέρος του σώματος (κεφάλι και πόδια). Ίσως μάλιστα το συγκεκριμένο visualization αδικεί κιόλας το πόσο υπερέχει.

Αν τώρα ελέγξουμε την τιμή που δίνει η μετρική της ncut, δηλαδή η **calculateNcut**, είτε μέσω της αναδρομικής μεθόδου (παίρνοντας μόνο την πρώτη τιμή), είτε μέσω της μη αναδρομικής μεθόδου, θα δούμε ότι αυτή είναι $\cong 0.785 > 0.2$ για την εικόνα d2b και $\cong 0.509 > 0.2$ για την εικόνα d2a και για αυτό το λόγο η αναδρομή σταματάει εκεί. Αν θελήσουμε να κάνουμε κατάτμηση ξανά επομένως, δηλαδή να προχωρήσουμε σε επόμενη αναδρομή, πρέπει να ξεπεράσουμε αυτήν την τιμή, επομένως πρέπει να δώσουμε νέα τιμή στο T^2 . Για παράδειγμα για τιμές ανάμεσα στις δύο αυτές ncut values, θα γίνει κατάτμηση στην d2a αλλά όχι στην d2b:



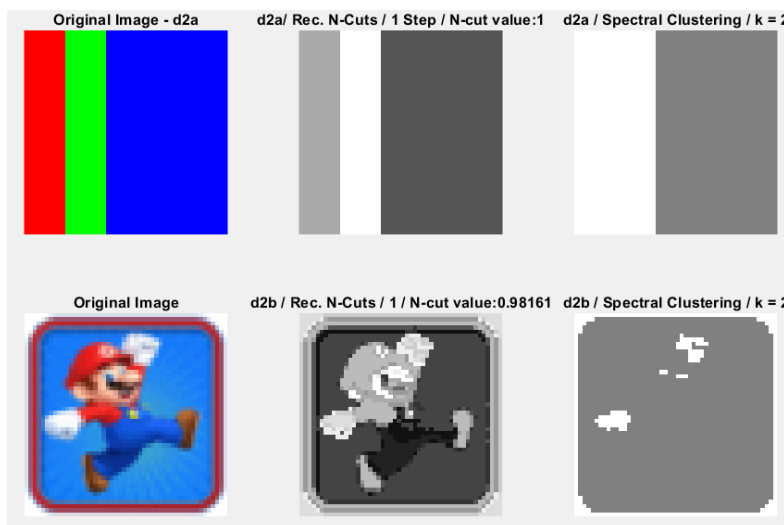
Εικόνα 31 – myNCutsRecursion για $T^2 = 0.7$ / d2a και d2b / N-cuts Recursion -vs- Spectral Clustering

Αν πάλι τώρα ξεπεράσουμε αυτό το κατώφλι:



Εικόνα 32 – myNCutsRecursion για $T^2 = 0.8$ / d2a και d2b / N-cuts Recursion -vs- Spectral Clustering

Έτσι παρατηρούμε ότι έχει γίνει και πάλι κατάτμηση της εικόνας, δηλαδή προχωρήσαμε ένα επίπεδο παρακάτω στο νέο μας δέντρο αναδρομών. Ένα περίεργο φαινόμενο που παρατηρήθηκε ήταν πως χωρίς καμία αλλαγή του κώδικα και της $\text{rng}(1)$, το αποτέλεσμα της spectral clustering άλλαξε ακριβώς μόλις το T^2 ξεπέρασε το κατώφλι 0.785, ενώ για δοκιμές σε τιμές μικρότερες από αυτή παρέμενε ίδιο. Αν τώρα μεγαλώσουμε κι άλλο την τιμή της T^2 η διαδικασία αυτή θα συνεχίσει να κάνει κατάτμηση της εικόνας σε ακόμα περισσότερα επίπεδα:

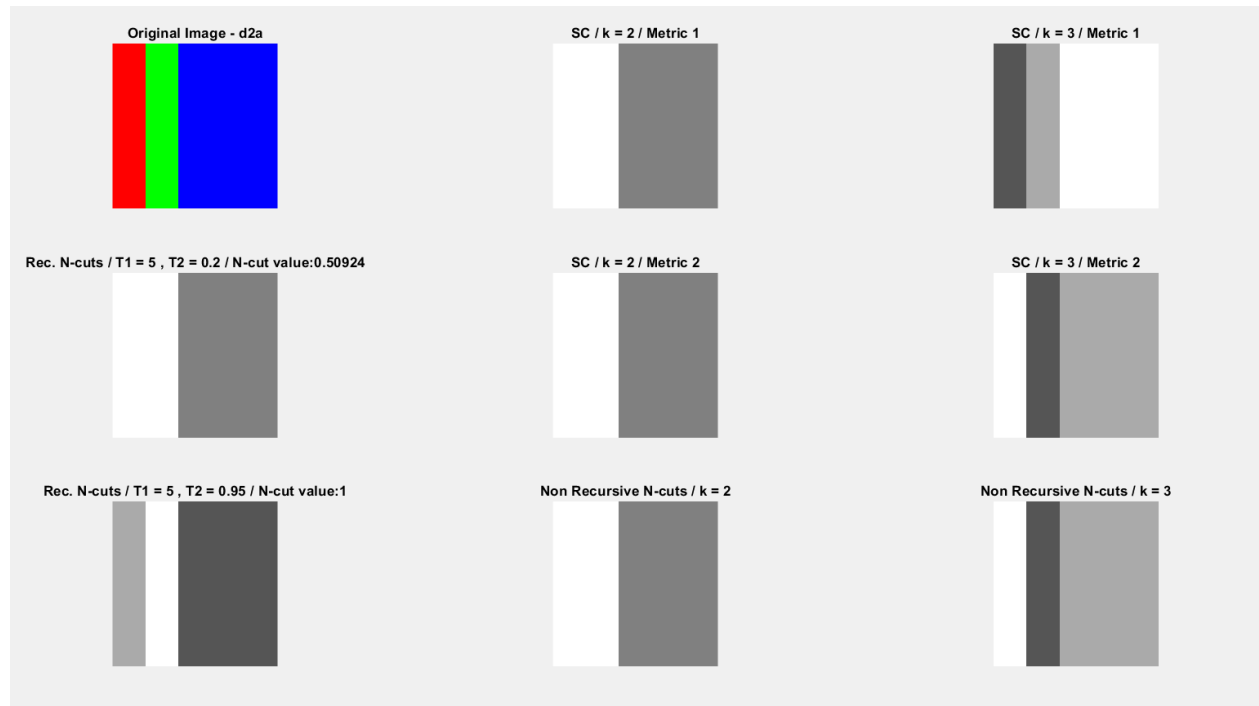


Εικόνα 33 – myNCutsRecursion για $T^2 = 0.95$ / d2a και d2b / N-cuts Recursion -vs- Spectral Clustering

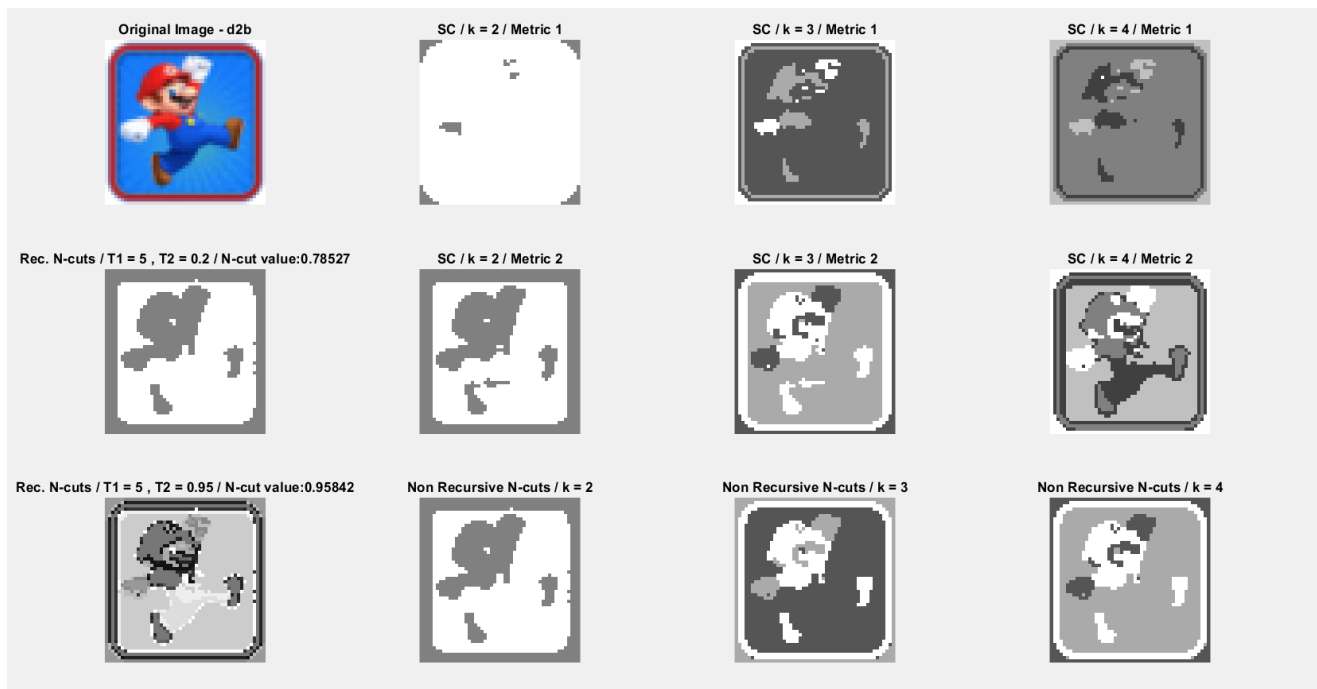
Παρατηρούμε ότι στην d2b έχουμε ήδη πετύχει πολύ ικανοποιητικό clustering, μάλιστα έχουμε πλέον τόσα clusters που «κινδυνεύουμε» να χαθεί πληροφορία σε μία πιο «blocky εικόνα». Τα clusters δηλαδή έχουν ήδη αρχίσει να είναι πολλά δίνοντας μεν μεγαλύτερη λεπτομέρεια αλλά λειτουργώντας εις βάρος της ξεκάθαρης ομαδοποίησης των αντικειμένων της εικόνας. Από την άλλη, βλέπουμε στην d2a να έχει γίνει τέλειο clustering και πλέον το κατώφλι να είναι πια στο 1, τιμή που αν τη φτάσουμε ή την ξεπεράσουμε θα μας δώσει ανεπιθύμητα αποτελέσματα.

Demo 3c

Στο συγκεκριμένο αρχείο (demo3b.m) παρουσιάζονται τα αποτελέσματα τις συνάρτησης **myNCutsRecursive**, δηλαδή τις αναδρομικής μεθόδου, σε μία ολοκληρωμένη παρουσίαση της εκτέλεσης της, καθώς και η σύγκρισή της με τις προηγούμενες μεθόδους.



Εικόνα 34 – Συγκρίσεις μεθόδων Clustering που χρησιμοποιήθηκαν στην εργασία / d2a



Εικόνα 35 – Συγκρίσεις μεθόδων Clustering που χρησιμοποιήθηκαν στην εργασία / d2b

Όπως βλέπουμε, στην εικόνα d2a που είναι μία απλή εικόνα με 3 χρωματικές περιοχές, οι υπόλοιπες μέθοδοι είναι τελείως ισάξίες, ενώ με την κατάλληλη επιλογή κατωφλίου η αναδρομική N-Cuts κάνει επίσης την κατάτμηση. Πιο ουσιαστικές διαφορές βλέπουμε σε πιο περίπλοκες εικόνες όπως η d2b όπου φαίνονται τα εξής:

- Στο δοσμένο κατώφλι T^2 , η αναδρομική N-Cuts είναι (όπως είδαμε και πριν) ισάξια με την μη αναδρομική N-Cuts για $k = 2$ αφού ουσιαστικά έχει γίνει μόνο μία φορά κατάτμηση.
- Για $k = 2$, η εικόνα από την μέθοδο SC με τη Μετρική 2 καλύπτει μεγαλύτερη περιοχή από αυτήν των άλλων δύο εικόνων σε αυτό το k , παρόλα αυτά δεν είναι κάποια φοβερά χρήσιμη πληροφορία αυτή που δίνει γιατί δεν εμφανίζει όλο το πόδι του Mario αλλά μόνο ένα μέρος του. Παρόλα αυτά τόσο η κατάτμηση αυτή όσο και η κατάτμηση με τη myNCuts μη αναδρομικά υπερτερούν κατά πολύ από την Spectral Clustering (με τη Μετρική 1).
- Για $k = 3$, ισχύουν ακριβώς τα ίδια με $k = 2$, παρόλο που τώρα η Spectral Clustering έχει αρχίσει να έχει καλύτερα αποτελέσματα.
- Για $k = 4$, η διαφορά είναι πολύ μεγάλη, καθώς η SC με τη Μετρική 2 έχει κάνει πολύ ικανοποιητική κατάτμηση ξεχωρίζοντας σχεδόν όλο το σώμα. Επόμενη έρχεται η μη αναδρομική N-Cuts και τελευταία η Spectral Clustering.
- Η μη αναδρομική N-Cuts θα λέγαμε ότι για μεγάλο κατώφλι (πολύ κοντά στο 1) κατασκευάζει πολύ καλά μία κβαντισμένη εκδοχή της αρχικής μας εικόνας, κάτι που δεν θέλουμε απαραίτητα στην κατάτμηση εικόνας καθώς θέλουμε τα αντικείμενα που ομαδοποιούνται να φαίνονται ξεκάθαρα και όχι λόγω της ύπαρξης περισσότερων clusters (υψηλότερης ακρίβειας) να γίνονται δυσδιάκριτα. Για λίγο μικρότερα κατώφλια φαίνεται να υπερτερεί από τις υπόλοιπες μεθόδους εκτός από την μέθοδο που χρησιμοποιείται η Μετρική 2 για $k = 4$ (ευδιάκριτα clusters).

Ενδεικτικά, θα δείξουμε τώρα τα Levels της αναδρομής – leaves του δέντρου για τις αναδρομικές μεθόδους που χρησιμοποιήσαμε:

```
Level1a =  
  
1x2 cell array  
  
{'R-A'}    {'R-B'}
```

Το παραπάνω είναι δέντρο της myNCutRecursive για $T^2 = 0.2$ όπου γίνεται μόνο ένα βήμα, επομένως έχουμε μόνο τα παιδιά/φύλλα A και B - Εικόνα d2a

```
Level2a =  
  
1x3 cell array  
  
{'R-A'}    {'R-B-A'}    {'R-B-B'}
```

Το παραπάνω είναι δέντρο της myNCutRecursive για $T^2 = 0.95$ όπου γίνεται 2 βήματα, επομένως έχουμε τα παιδιά/φύλλα A και BA ,BB - Εικόνα d2a

```

Level1b =

1x2 cell array

{'R-A'}    {'R-B'}

```

Το παραπάνω είναι δέντρο της myNCutRecursive για $T^2 = 0.2$ όπου γίνεται μόνο ένα βήμα, επομένως έχουμε μόνο τα παιδιά/φύλλα A και B - Εικόνα d2b

```

Level2b =

1x15 cell array

Columns 1 through 3

{'R-A-A-A-A-A'}    {'R-A-A-A-A-B'}    {'R-A-A-A-B-A'}

Columns 4 through 6

{'R-A-A-A-B-B-A-A'}    {'R-A-A-A-B-B-A-B'}    {'R-A-A-A-B-B-B'}

Columns 7 through 10

{'R-A-A-B'}    {'R-A-B-A-A'}    {'R-A-B-A-B'}    {'R-A-B-B-A'}

Columns 11 through 14

{'R-A-B-B-B'}    {'R-B-A'}    {'R-B-B-A-A'}    {'R-B-B-A-B'}

Column 15

{'R-B-B-B'}

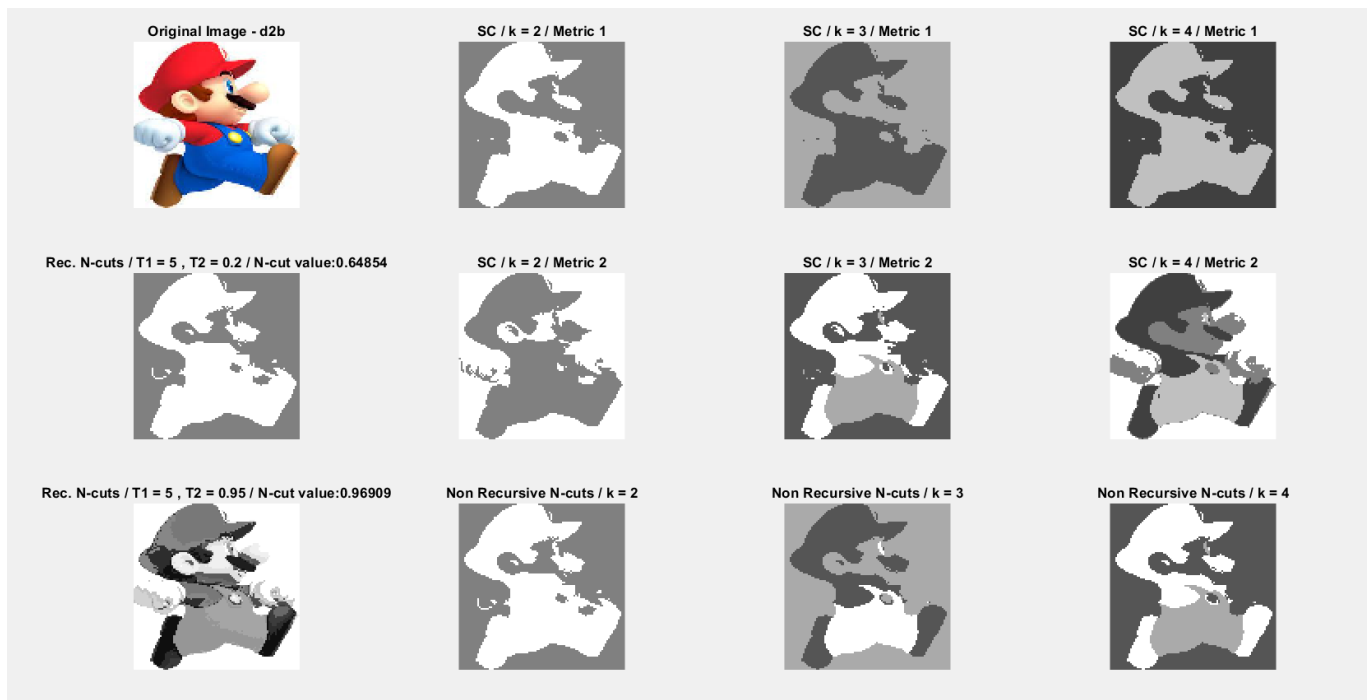
```

Το παραπάνω είναι δέντρο της myNCutRecursive για $T^2 = 0.95$ όπου γίνονται πολλά αναδρομικά βήματα (15 φύλλα / 15 clusters) - Εικόνα d2b

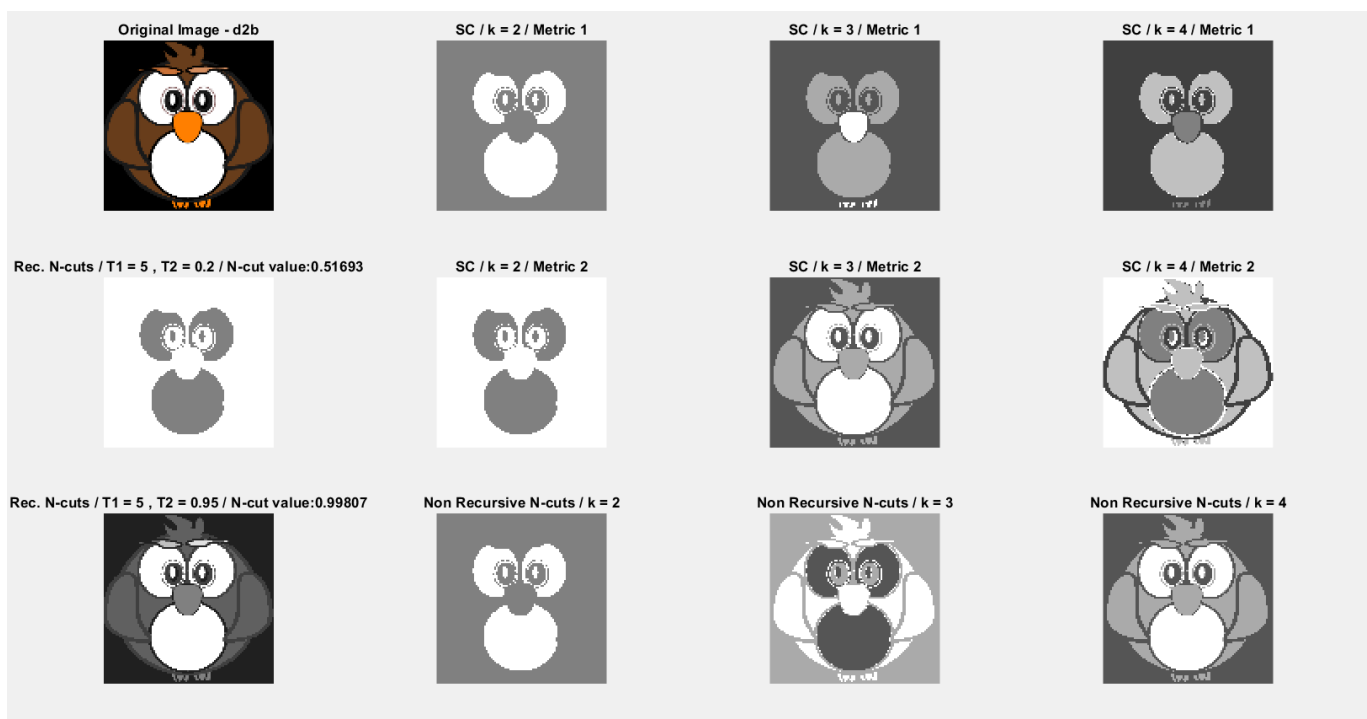
Σημείωση:

Στα πειράματά μας φάνηκε ότι το T^1 επηρεάζει πολύ λιγότερο την κατάταμσή μας και πως πρέπει να το αυξήσουμε σημαντικά για να αλλάξει / χαλάσει το αποτέλεσμα. Έτσι, θεωρείται πως σε ένα πραγματικό πρόβλημα δεν θα χρειαστεί να αυξηθεί τόσο και επομένως δεν δόθηκε ιδιαίτερη βαρύτητα στην περαιτέρω έρευνα γύρω από αυτό.

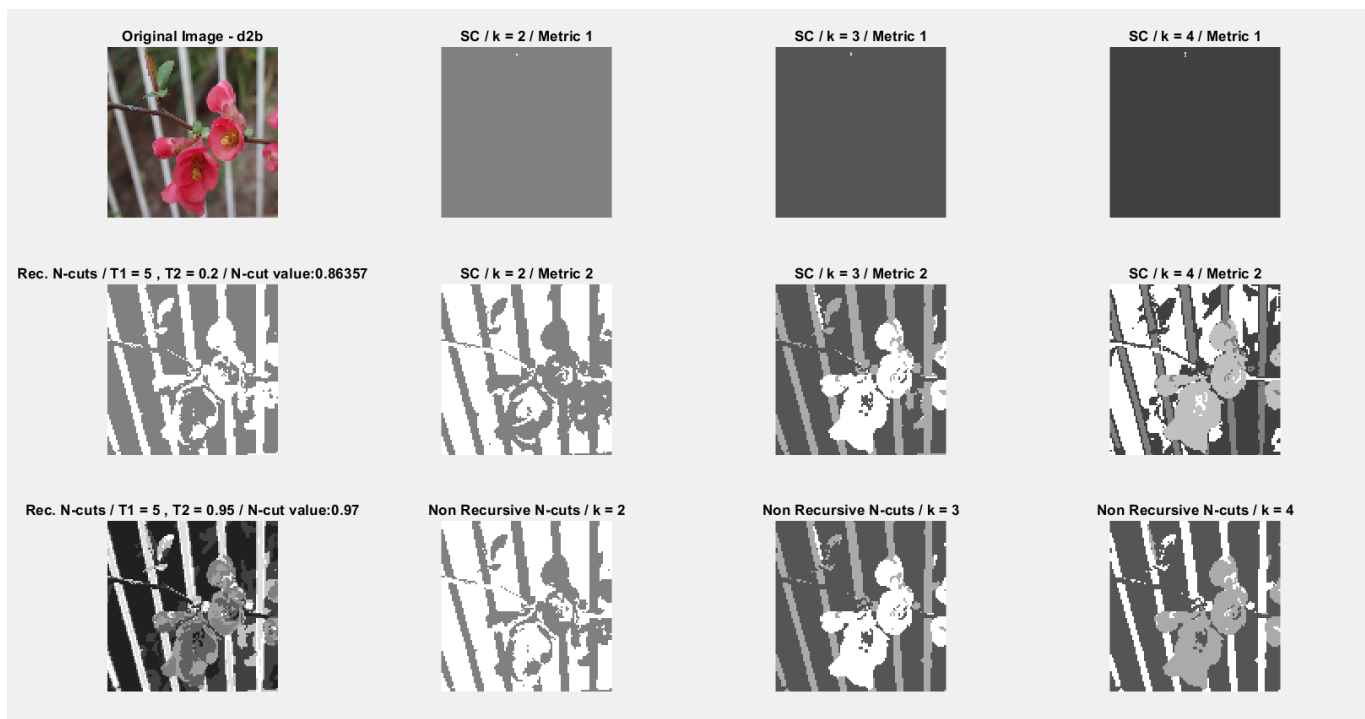
Τέλος, παρουσιάζονται κάποια πειράματα ακόμα με άλλες εικόνες:



Εικόνα 36 – Συγκρίσεις μεθόδων Clustering που χρησιμοποιήθηκαν στην εργασία / Mario 2



Εικόνα 7 – Συγκρίσεις μεθόδων Clustering που χρησιμοποιήθηκαν στην εργασία / Mario 2



Εικόνα 38 – Συγκρίσεις μεθόδων Clustering που χρησιμοποιήθηκαν στην εργασία / Mario 2

Σύνοψη – Συμπεράσματα – Future work:

Στην εργασία αυτή έγιναν διάφοροι πειραματισμοί με μεθόδους κατάτμησης εικόνας. Είναι ξεκάθαρο ότι η state-of-the-art μέθοδος N-Cuts φαίνεται όντως να υπερτερεί στις περισσότερες φωτογραφίες. Επίσης παρουσιάστηκε μία τροποποίηση της μεθόδου Spectral Clustering η οποία φαίνεται να έχει πολύ καλά αποτελέσματα στο σύνολο των εικόνων στις οποίες έγιναν οι πειραματισμοί. Επιπλέον, δοκιμάστηκαν (επιφανειακά) κι άλλες μετρικές και μέθοδοι και πρακτικές όπως low pass filtering / smoothing της εικόνας πριν την κατάτμηση. Η παρούσα εργασία ήταν μία πολύ ενδιαφέρουσα μαθησιακή διαδικασία και γέννησε ιδέες για clustering σε βιοϊατρικές εικόνες, εφαρμογές σε δορυφόρους και άλλα. Ο πειραματισμός με τις διαφορετικές μετρικές βοήθησε στην καλύτερη κατανόηση των διάφορων τεχνικών που χρησιμοποιούνται στην επεξεργασία εικόνας. Το συμπέρασμα ως προς την χρήση της εκάστοτε μετρικής είναι πως σύμφωνα με το πρόβλημα που έχουμε να λύσουμε και την γνώση μας για αυτό (μπορεί για παράδειγμα να ασχολούμαστε με εικόνες συγκεκριμένης φύσης), κάνουμε και την ανάλογη επιλογή της μετρικής που πιστεύουμε ότι ανταποκρίνεται ικανοποιητικότερα.

Αναφορές:

- [1] Modh Jigar S, Shah Brijesh, Shah Satish k, “A New K-mean Color Image Segmentation with Cosine Distance for Satellite Images”, IJEAT, vol.1 Issue 5 June 2012