

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Ψηφιακή Επεξεργασία Εικόνας

Εργασία 2

Κούτση Χριστίνα
ΑΕΜ: 9871
email: cvkoutsis@ece.auth.gr

Μάιος, 2022

1 Εικόνες ως γράφοι

Στην πρώτη ενότητα της εργασίας μας ζητάται να κατασκευάσουμε μία συνάρτηση η οποία μετατρέπει μία δοσμένη εικόνα σε γράφο. Η συνάρτηση αυτή θα πρέπει να βγάζει σωστά αποτελέσματα ανεξαρτήτως της εικόνας εισόδου.

Έστω η $M \times N$ εικόνα εισόδου. Ο **affinity** πίνακας θα είναι διάστασης $(M \times N) \times (M \times N)$ και θα περιγράφει την συσχέτιση κάθε **pixel** της δοσμένης εικόνας με όλα τα υπόλοιπα $M \times N$ **pixel** της εικόνας. Στην συγκεκριμένη άσκηση, η συσχέτιση μεταξύ των **pixels** περιγράφεται από την σχέση

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

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

- Για $n = 1$

Για 1 κανάλι, συνεπώς για ασπρόμαυρη εικόνα, θα έχουμε:

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

- Για $n = 3$

Για 3 κανάλια, θα έχουμε:

$$d(i,j) = \sqrt{(img(i,1) - img(j,1))^2 + (img(i,2) - img(j,2))^2 + (img(i,3) - img(j,3))^2}$$

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

Για $i = j$ θα έχουμε $d(i,i) = 0$ και $A = \frac{1}{e^{d(i,i)}} = 1$. Για τον λόγο αυτό ο **affinity** πίνακας θα είναι και τετραγωνικός.

2 Graph Spectral Clustering

Στη δεύτερη ενότητα της εργασίας μας ζητάται η κατασκευή της συνάρτησης `myGraphSpectralClustering`, η οποία εφαρμόζοντας τα 5 βήματα της μεθόδου **Graph Spectral Clustering**, μας επιστρέφει τον πίνακα `clusterIdx`, ο οποίος περιέχει τις ετικέτες των `clusters` στα οποία ανήκουν τα `pixels`.

2.1 Demo 1

Για το **demo 1** χρησιμοποιήθηκε ο δοσμένος **affinity** πίνακας **d1a** και πραγματοποιήθηκαν 3 υλοποιήσεις της ρουτίνας `myGraphSpectralClustering` για $k = [2,3,4]$. Τα αποτελέσματα φαίνονται παρακάτω:

$k = 2$	$k = 3$	$k = 4$
2	1	1
2	1	1
2	1	1
2	1	1
1	2	2
1	2	3
1	2	3
1	2	3
2	3	4
2	3	4
2	3	4
2	3	4

- $k = 2$

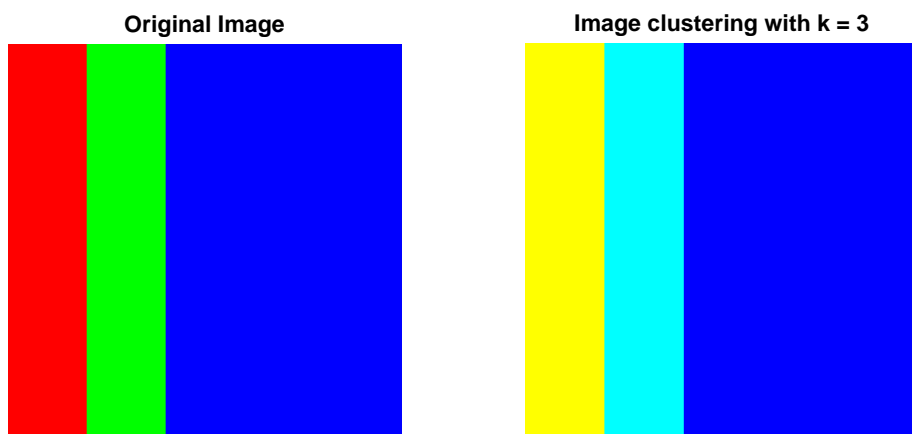
Έχουμε διαχωρισμό σε 2 clusters με το cluster με `id = 2` να έχει περισσότερα μέλη

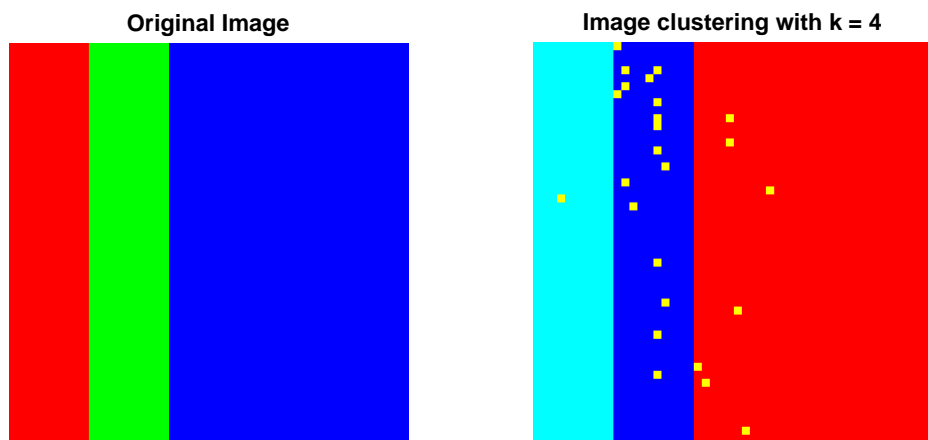
- $\chi = 3$
Παρατηρώ ότι έχουμε 3 clusters ίδιου μεγέθους
- $\chi = 4$
Παρατηρώ ότι δεν υπάρχει μεγάλη βελτίωση στην χρήση ενός παραπάνω cluster, καθώς το cluster με $\text{id} = 2$ έχει ένα μόνο μέλος

2.2 Demo 2

Για το demo 2 εφαρμόζω την ρουτίνα `myGraphSpectralClustering` στις εικόνες εισόδου `d2a` και `d2b`. Τα αποτελέσματα είναι τα εξής:

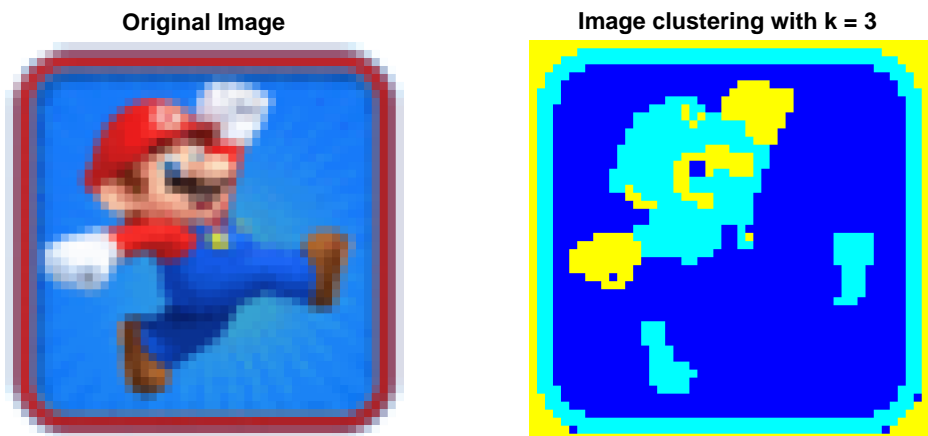
- Εικόνα `d2a`

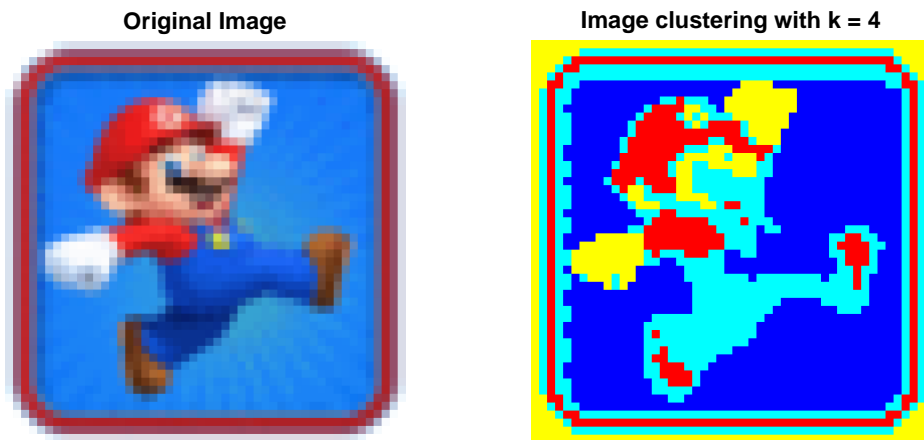




Παρατηρώ ότι η προσθήκη ενός ακόμα **cluster** δεν παρέχει μεγάλη βελτίωση στην εικόνα, καθώς μόνο μερικά **pixels** ανήκουν στο νέο **cluster**.

- Εικόνα d2b





Εδώ παρατηρώ ότι η προσθήκη ενός ακόμα **cluster** προσέφερε εμφανώς μεγαλύτερη λεπτομέρεια στην εικόνα.

3 Normalized-cuts

Στο ερώτημα αυτό μας ζητείται η υλοποίηση της συνάρτησης **nCutValue**, η οποία υπολογίζει την μετρική **N-cut**. Η μετρική αυτή θα χρησιμοποιηθεί αργότερα για την "απόφαση" περεταίρω χωρισμού του γράφου ή όχι.

3.1 Demo 3a

Στο **demo 3a** μας ζητάται η αναδρομική εκτέλεση της μεθόδου **n-cuts** για ένα βήμα. Τα αποτελέσματα που έχουμε είναι τα εξής:

- Εικόνα d2a

Original Image

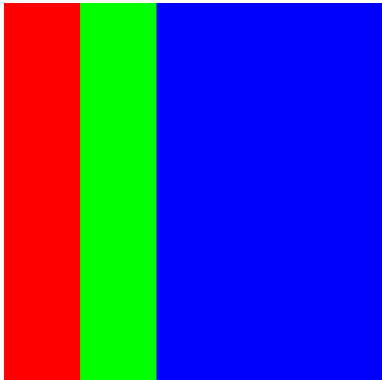
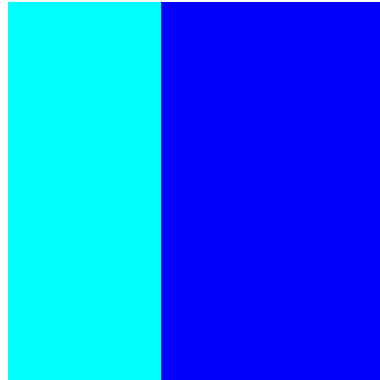


Image clustering with $k = 2$
n-cut value: 0.50924

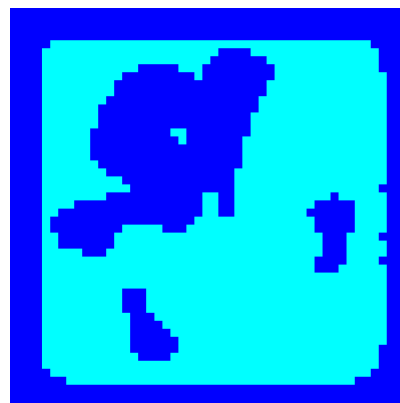


- Εικόνα d2b

Original Image



Image clustering with $k = 2$
n-cut value: 0.78529



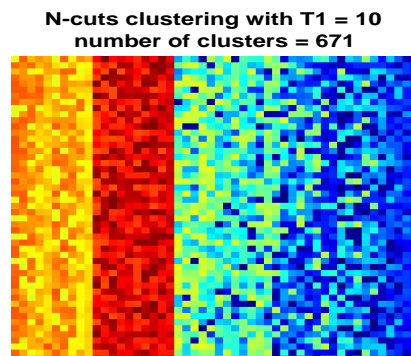
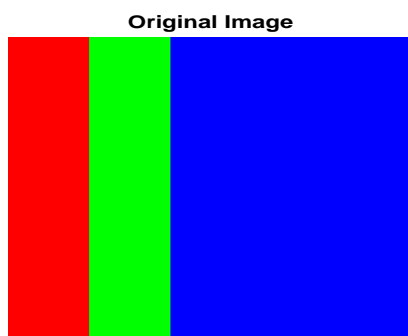
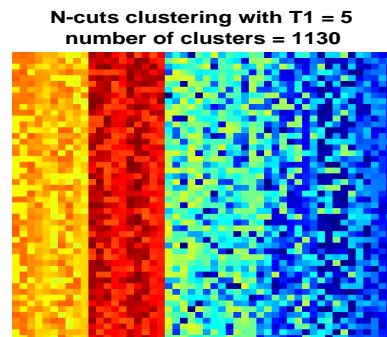
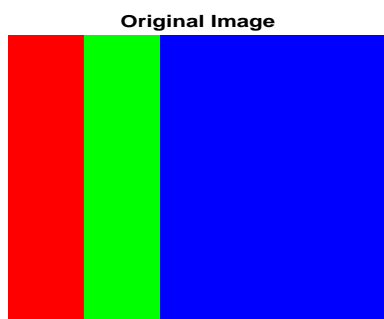
3.2 Demo 3b

Στο ερώτημα αυτό παρουσιάζεται η αναδρομική μέθοδος **n-cuts**. Για την μέθοδο αυτή χρησιμοποιήθηκαν τα κατώφλια $T1$ και $T2$ για τα οποία δοκιμάστηκαν διάφορες τιμές. Συγκεκριμένα και για τις δύο $T1 = [5,10,20,30]$ και $T2 = [0.8,0.9,0.95,1]$. $T2 = [0.9,0.95,0.98,1]$

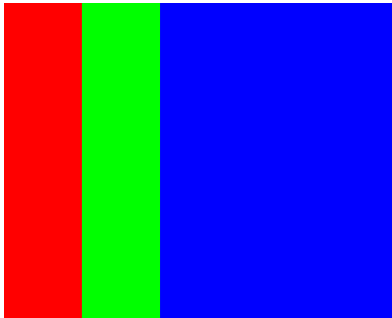
- Εικόνα d2a

Για την εικόνα d2a δοκιμάστηκαν τα κατώφλια $T1 = [5,10,20,30]$ και $T2 = [0.8,1]$

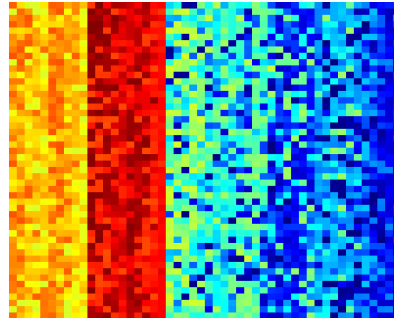
– Image clustering με συνθήκη $T1$



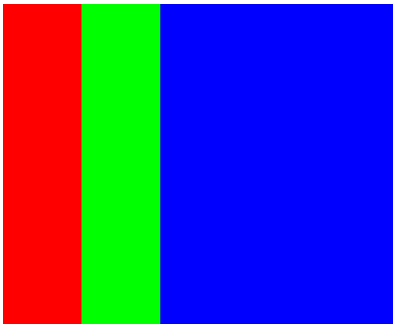
Original Image



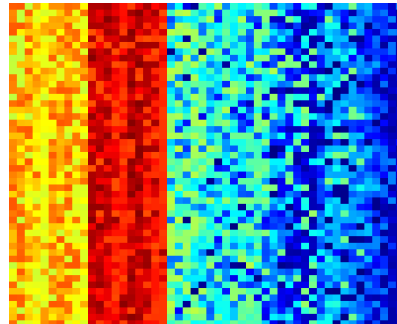
N-cuts clustering with $T1 = 20$
number of clusters = 260



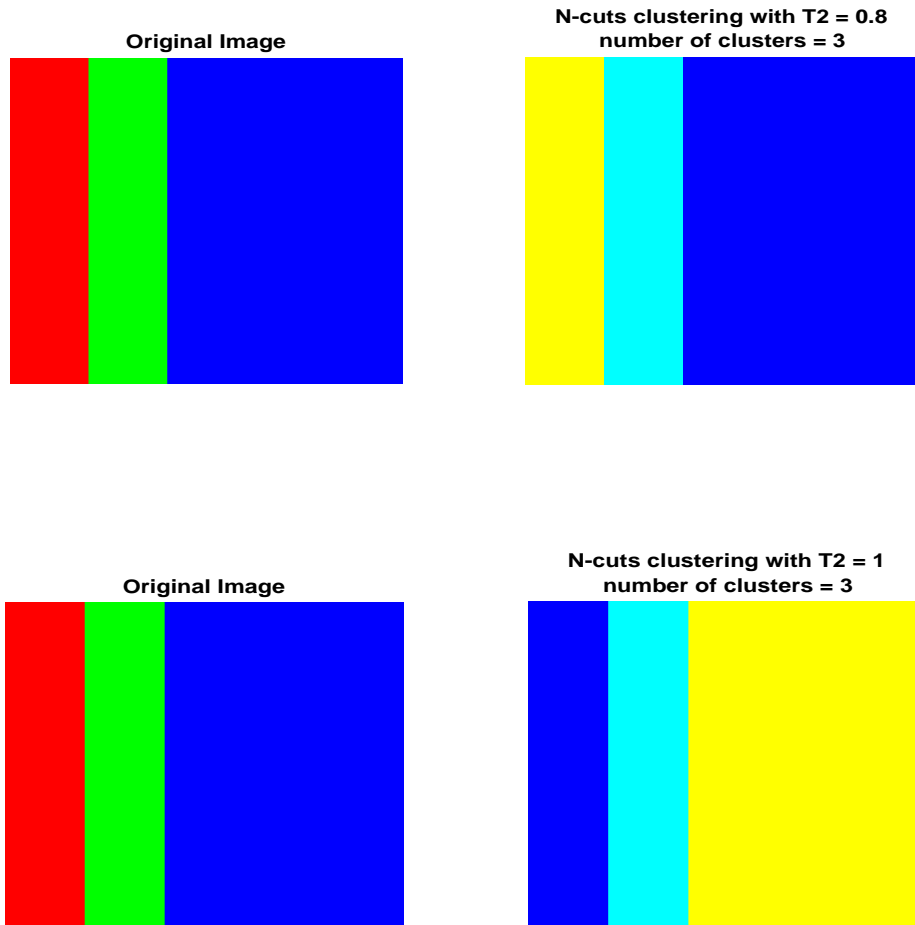
Original Image



N-cuts clustering with $T1 = 30$
number of clusters = 211



– Image clustering με συνθήκη T2



Παρατηρώ ότι η εικόνα έχει βέλτιστο **clustering** για αριθμό **clusters** ίσο με 3. Ελέγχοντας την μετρική **n-cut**, αυτό γίνεται αντιληπτό από τον αλγόριθμο και δεν προχωράει σε περεταίρω διαχωρισμό σε **clusters**. Αντιθέτως, όταν χρησιμοποιούμε την συνθήκη **T1** αυτό δεν γίνεται αντιληπτό και ο αλγόριθμος τερματίζει μόνο όταν ο αριθμός των κόμβων που ανήκουν σε ένα **cluster** είναι μικρότερος από το κατώφλι **T1**. Επομένως προχωράει σε άσκοπους διαχωρισμούς σε **clusters**, οι οποίοι δημιουργούν θόρυβο στην εικόνα

- Εικόνα d2b

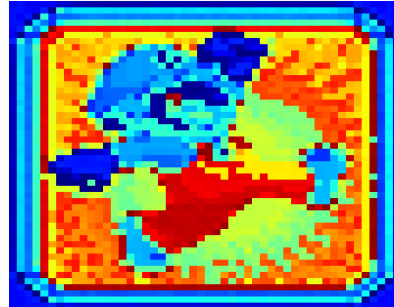
Για την εικόνα d2a δοκιμάστηκαν τα κατώφλια $T1 = [5,10,20,30]$ και $T2 = [0.9,0.95,0.98,1]$

- Image clustering με συνθήκη T1

Original Image



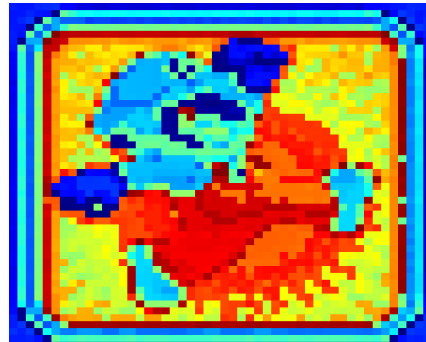
N-cuts clustering with $T1 = 5$
number of clusters = 876



Original Image



N-cuts clustering with $T1 = 10$
number of clusters = 483



Original Image



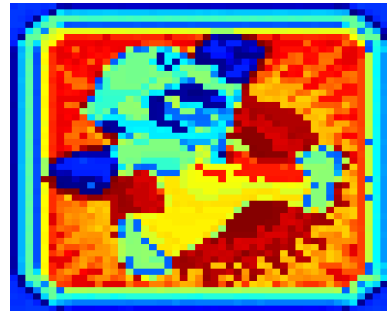
N-cuts clustering with $T1 = 20$
number of clusters = 241



Original Image



N-cuts clustering with $T1 = 30$
number of clusters = 168

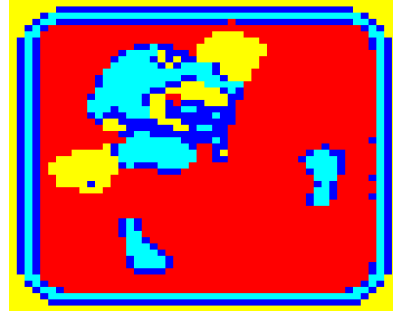


– Image clustering με συνθήκη T_2

Original Image



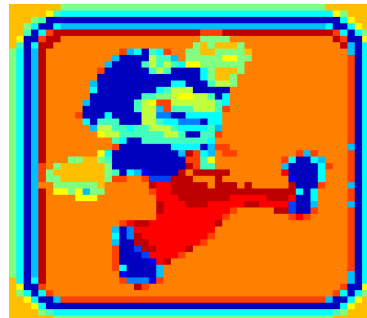
N-cuts clustering with $T_2 = 0.9$
number of clusters = 4



Original Image



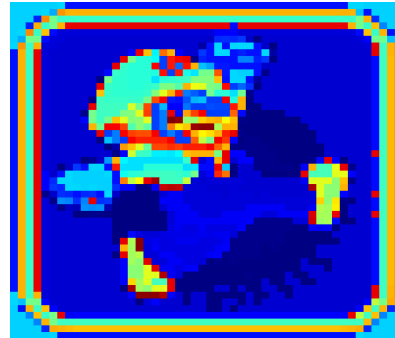
N-cuts clustering with $T_2 = 0.95$
number of clusters = 15



Original Image



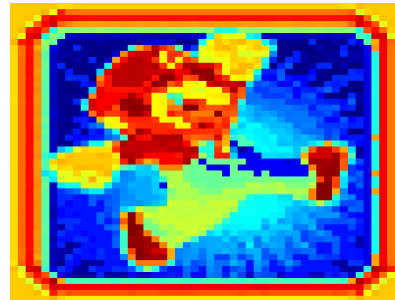
N-cuts clustering with $T2 = 0.98$
number of clusters = 132



Original Image



N-cuts clustering with $T2 = 1$
number of clusters = 1600



Παρατηρώ ότι για αρκετα μεγάλες τιμές του $T2$ ($T2= 0.9,0.95$) η συνθήκη τερματισμού ικανοποιείται για μικρό αριθμό **clusters**. Ο αριθμός των **clusters** αυξάνεται δραματικά όσο το $T2$ πλησιάζει στην μονάδα και μεγιστοποιείται για $T2 = 1$. Στη συγκεκριμένη εικόνα όσο μεγαλύτερο το $T1$ και το $T2$ τόσο περισσότερα τα **clusters** της τελικής εικόνας και συνεπώς μεγαλύτερη ευκρίνεια.

4 Superpixel Segmentation

Στο 4ο ερώτημα της εργασίας μας ζητάται ο διαχωρισμός της εικόνας σε **superpixels** εφαρμόζοντας την μέθοδο **SLIC** και στη συνέχεια η εφαρμογή της μη αναδρομικής και αναδρομικής μεθόδου **n-cuts**. Η διαδικασία αυτή απαιτεί τα εξής βήματα:

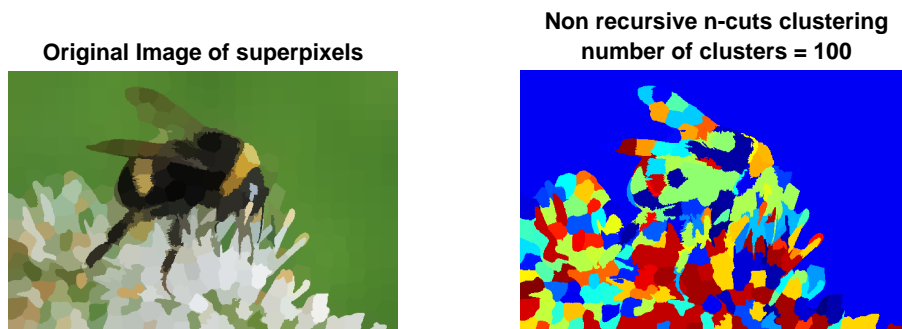
1. Εφαρμογή της μεθόδου **SLIC** για την εύρεση του πίνακα **labels** ο οποίος περιέχει την ετικέτα του **superpixel** στο οποίο ανήκει το κάθε **pixel**.
2. Εύρεση της περιγραφής του κάθε **superpixel** μέσω της συνάρτησης **superpixelDescriptor**, η οποία μας επιστρέφει την τελική εικόνα, χωρισμένη σε **superpixels**.
3. Μετατροπή της εικόνας των **superpixels** σε γράφο
Σε αυτό το βήμα πραγματοποιήθηκε η μετατροπή της εικόνας των **superpixels** σε **affinity** πίνακα, με την βοήθεια της συνάρτησης **Image2Graph**. Επειδή όμως η εικόνα των **superpixels** είναι διάστασης $M \times N$ και τελικά το μέγεθος του **affinity** πίνακα θα προέκυπτε $(M \times N) \times (M \times N)$, το οποίο δεν είναι διαχειρίσιμο, θα εκμεταλλευτούμε τον διαχωρισμό της εικόνας σε **superpixels** και θα τροποποιήσουμε την συνάρτηση **Image2Graph**. Συγκεκριμένα, εφόσον για εικόνα **k** **superpixels** θα έχουμε **k** διαφορετικές εκφράσεις τις εικόνας, μπορούμε να δώσουμε ως όρισμα στην **Image2Graph** την ακολουθία αυτών των "μοναδικών" **pixels**. Έτσι, ο **affinity** πίνακας που θα προκύψει θα είναι πολύ μικρότερος και συνεπώς πιο διαχειρίσιμος.
4. Εφαρμογή της μεθόδου **n-cut** (αναδρομική ή μη αναδρομική) για τον **affinity** πίνακα που βρήκαμε παραπάνω και για τα "μοναδικά" **pixels** της εικόνας **superpixel**.
5. Ανάθεση της τιμής του **cluster** στο οποίο βρήκαμε ότι ανήκουν τα "μοναδικά" **pixels** και στα υπόλοιπα **pixels** τα οποία ανήκουν στο ίδιο **superpixel**.

Αρχικά εκτελώ τα βήματα 1 και 2 και βρίσκω την εικόνα **superpixels**. Το αποτέλεσμα είναι το εξής:



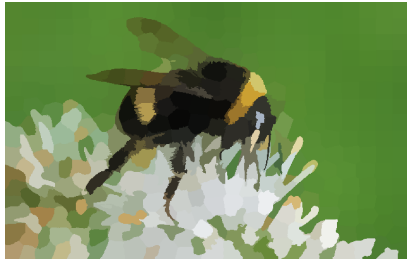
- Μη αναδρομική μέθοδος **n-cuts**

Για την μη αναδρομική εκτέλεση της μεθόδου επιλέγω $k = 100$. Στη συνέχεια θα προσπαθήσω να πετύχω τον ίδιο ή παρόμοιο αριθμό **cluster** επιλέγοντας τις κατάλληλες συνθήκες T1 και T2. Εφαρμόζω λοιπόν την μη αναδρομική μέθοδο **n-cuts** για $k = 100$:

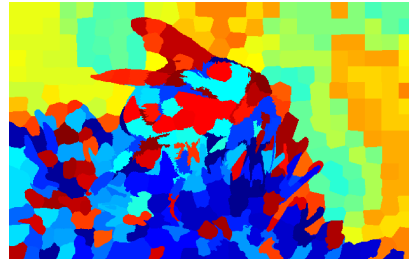


- Αναδρομική μέθοδος **n-cuts** με συνθήκη τερματισμού $T1$
Για την αναδρομική μέθοδο, παρατηρώ ότι ο αλγόριθμος τερματίζει κοντά 100 clusters για $T1 = 7$. Το αποτέλεσμα του αλγορίθμου είναι:

Original Image of superpixels

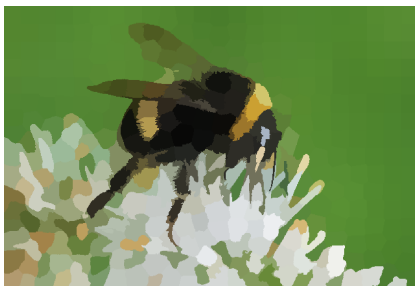


N-cuts clustering with $T1 = 7$
number of clusters = 100



- Αναδρομική μέθοδος **n-cuts** με συνθήκη τερματισμού $T2$
Για την αναδρομική μέθοδο, παρατηρώ ότι ο αλγόριθμος τερματίζει κοντά 100 clusters για $T2 = 10^{-3}$. Το αποτέλεσμα του αλγορίθμου είναι:

Original Image of superpixels



N-cuts clustering with $T2 = 0.001$
number of clusters = 107

