Ψηφιακή Επεξεργασία Εικόνας
Εργασία 2

Κουτρουμπής Γεώργιος, ΑΕΜ: 9668

2022

Περιεχόμενα

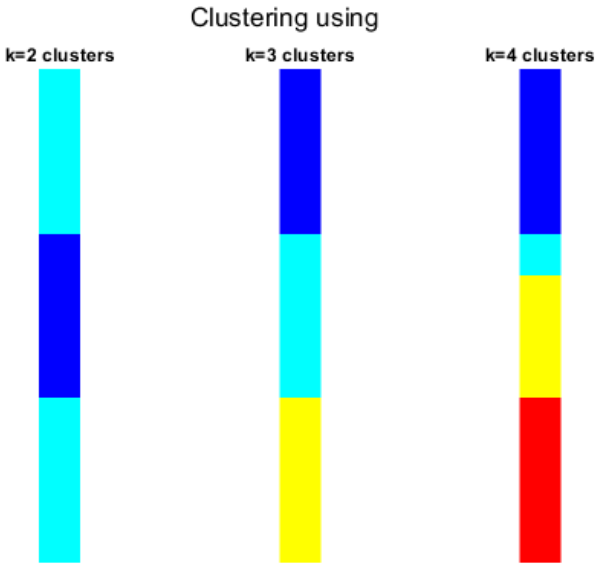
1 Εικόνες ως γράφοι

Για να πετύχουμε το image segmentation με τους τρόπους που ζητήθηκαν, και θα παρουσιαστούν και στη συνέχεια, αρχικά χρειάζεται η εικόνα να εκφραστεί ως γράφος. Συγκεκριμένα, από την εικόνα, δημιουργείται ένας affinity matrix, ο οποίος είναι ένας συμμετρικός πίνακας, $(M*N)x(M*N)$, όπου M, N οι διαστάσεις της εικόνας, με κάθε στοιχείο (i,j) να περιέχει την τιμή $\frac{1}{e^{d(i,j)}}$, όπου $d(i,j)$ η Ευκλείδια απόσταση της φωτεινότητας των καναλιών μεταξύ των pixels i και j.
Καθώς ο πίνακας είναι συμμετρικός, αρκεί να υπολογιστούν οι τιμές για τον κάτω τριγωνικό πίνακα, οπότε μετά για κάθε στοιχείο (i,j), το στοιχείο (j,i) θα έχει ίδια τιμή.
Στην υλοποίηση της συνάρτησης, κάθε στοιχείο της κύριας διαγωνίου έχει τιμή ίση με το 0.

2 Graph Spectral Clustering

Με την υλοποίηση και χρήση της Graph Spectral Clustering μεθόδου που περιγράφεται στην εκφώνηση, μπορεί μια εικόνα να χωριστεί σε clusters. Σύμφωνα με τον αλγόριθμο που περιγράφεται, έχοντας έναν affinity πίνακα W, βρίσκουμε το degree πίνακα D, και σχηματίζουμε τον μή κανονικοποιημένο Λαπλασιανό πίνακα $L = D - W$. Λύνοντας το γενικευμένο πρόβλημα ιδιοτιμών $Lx = \lambda Dx$, με τη βοήθεια της eigs του MATLAB, και σχηματίζοντας τον πίνακα που έχει ως στήλες τα k μικρότερα ιδιοδιανύσματα, είμαστε σε θέση να χωρίσουμε την εικόνα σε clusters. Αυτό πραγματοποιείται με τη χρήση της συνάρτησης kmeans του MATLAB, όπου επιστρέφει τελικά τη ταμπέλα του cluster που ανήκει κάθε pixel.

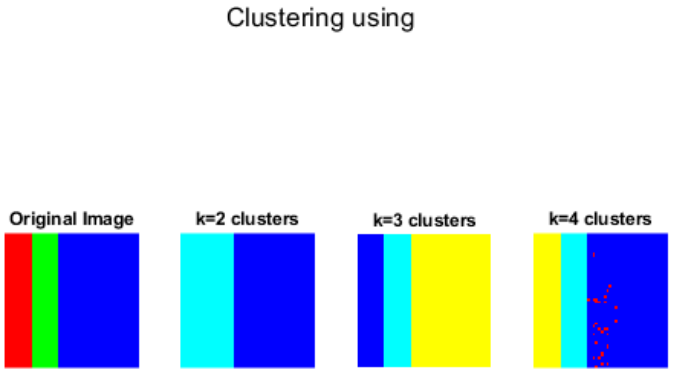
3 Demo 1



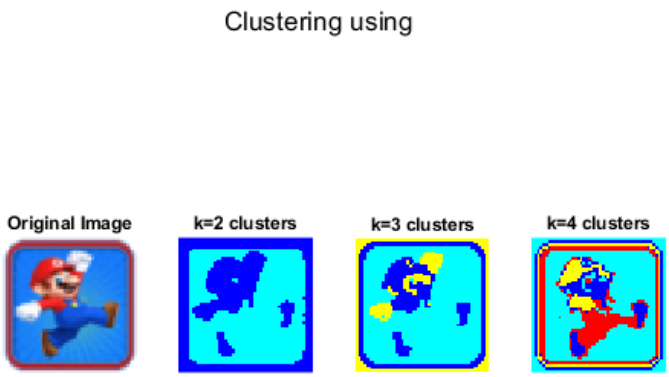
Εικόνα 1. *Αποτέλεσμα spectral clustering για διάφορα k στον δοκιμαστικό affinity πίνακα*

Παραπάνω παρουσιάζεται μια οπτικοποίηση των αποτελεσμάτων του spectral clustering στον δοκιμαστικό affinity πίνακα d1a. Πρακτικά πρόκειται για μια εικόνα με συνολικά 12 pixels, αγνώστου διάστασης. Παρατηρώντας τα αποτελέσματα για k=2,3,4 μπορούμε να συμπεράνουμε ότι η εικόνα αποτελείται από 3 χρώματα, καθώς για k=3 έχουμε το καλύτερο clustering. Καλύτερο καθώς για k=4, παρατηρούμε ότι κάποιο από τα 3 υπάρχοντα clusters "εξαναγκάζεται" να χωριστεί σε δύο (πράγματι αν αφαιρέσουμε το rng(1), βλέπουμε ότι αλλάζει το 4ο cluster τυχαία, ενώ για k=2,3 μένουν ίδια). Αντίστοιχα, για k=2, βλέπουμε ότι από τα 3 clusters που συμπαιρένουμε ότι υπάρχουν, τα 2 ενώνονται, καθώς έχουμε ορίσει να χωριστεί η εικόνα σε 2 clusters.

4 Demo 2



Εικόνα 2. *Αποτέλεσμα spectral clustering για διάφορα k στην δοκιμαστική εικόνα d2a*



Εικόνα 3. *Αποτέλεσμα spectral clustering για διάφορα k στην δοκιμαστική εικόνα d2b*

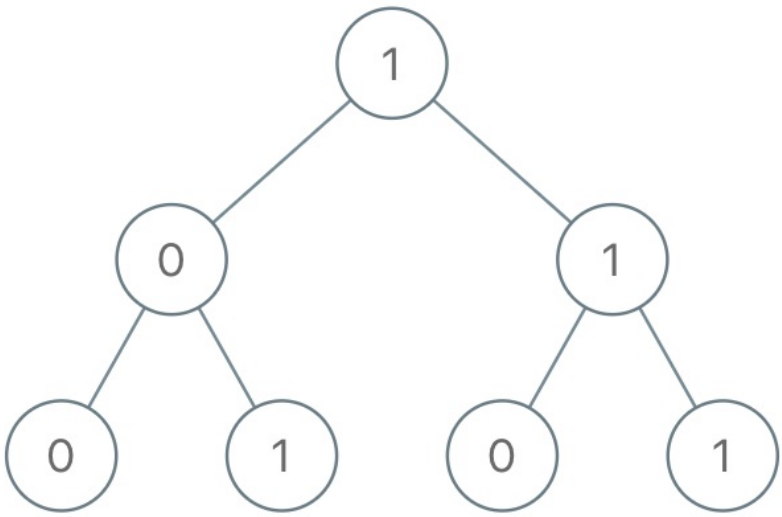
Παραπάνω παρουσιάζεται μια οπτικοποίηση των αποτελεσμάτων του spectral clustering στις δύο δοκιμαστικές εικόνες d2a, d2b.

Για την πρώτη εικόνα, παρατηρούμε συμπεριφορά παρόμοια με αυτήν στο Demo 1. Για k=2, οι 2 λωρίδες θεωρούνται 1 cluster, καθώς η μπλε λωρίδα πρέπει να θεωρηθεί από μόνη της 1 cluster. Για k=3, το clustering είναι τελείως σωστό, καθώς για κάθε 1 λωρίδα υπάρχει 1 cluster. Για k=4, δημιουργείται τυχαία ένα 4ο cluster, το οποίο δεν δίνει καμία πληροφορία, καθώς "εξαναγκάζεται" ο αλγόριθμος να φτιάξει περισσότερα clusters από όσα χρειάζονται.

Στην δεύτερη εικόνα, παρατηρούμε ότι με την αύξηση του αριθμού των clusters, παράγονται καλύτερα αποτελέσματα, μιας και η συγκεκριμένη έχει παραπάνω πληροφορία από την 1η. Για k=2, γίνεται μια σχετικά καλή αποτύπωση της φιγούρας του Mario, με το μπλε overall να θεωρείται ίδιο cluster με το φόντο. Για k=3, βλέπουμε ότι γίνεται διαχωρισμός των χαρακτηριστικών του προσώπου και των γαντιών, ενώ για k=4, πλέον τα overalls αποτελούν ξεχωριστό cluster, ενώ πλέον τα γάντια ενσωματώνονται με το το φόντο. Αυτό έχει να κάνει με τη τυχαιότητα του k-means, καθώς ακόμα και τα 4 clusters δεν είναι αρκετά για τη συγκεκριμένη φωτογραφία, η οποία αποτελείται από περισσότερα ξεχωριστά κομμάτια. Μάλιστα τρέχοντας τον αλγόριθμο χωρίς το rng(1), παρατηρούμε διαφορετικά clustering της εικόνας. Σε όλες τις περιπτώσεις όμως γίνεται διακριτοποίηση της φιγούρας, και διάφορων χαρακτηριστικών, με τυχαιότητα.

5 Ncuts

Υλοποιήθηκε συνάρτηση που υπολογίζει την τιμή ncut value, καθώς και συνάρτηση που υλοποιεί την ανδρομική έκδοση του αλγορίθμου ncuts. Για ένα "χωρισμό" που γίνεται με τη χρήση της myGraphSpectralClustering για k=2, υπολογίζεται όπως στο paper. Αυτή η τιμή πρακτικά εκφράζει την συσχέτιση μεταξύ των στοιχείων σε κάθε cluster/partition, αλλά και τη μη-συσχέτιση μεταξύ των δύο clusters/partitions. Για αυτό και όταν αυτή η τιμή είναι πάνω από κάποιο όριο, λειτουργεί ως κανόνας διακοπής του αναδρομικού αλγορίθμου, καθώς σημαίνει ότι έχει επιτευχθεί καλό "χώρισμα".
Για την αναδρομική έκδοση του ncuts αλγορίθμου, ακολουθήθηκε διαδικασία παρόμοια με την κατασκευή ενός δυαδικού δέντρου. Αρχίζοντας με τον αρχικό affinity πίνακα της εικόνας, αυτός χωρίζεται στα 2 με τη χρήση spectral clustering, k=2, και γίνεται έλεγχος αν η ncut τιμή που προκύπτει είναι μεγαλύτερη από ένα όριο ή αν ο αριθμός των πίξελ σε κάθε cluster είναι μικρότερος από ένα άλλο όριο. Αν καμία από τις δύο συνθήκες δεν ισχύει, τα 2 "χωρίσματα"/clusters που προκύπτουν, χωρίζονται και αυτά το καθένα σε 2, και ακολουθείται αναδρομικά η ίδια διαδικασία.
Για την αρίθμηση του κάθε cluster, ξεκινάμε αρχικά με την ολόκληρη εικόνα να έχει την ταμπέλα "0" ή "1". Σε κάθε χώρισμα, προστίθεται αριστερά από αυτήν την ταμπέλα το "0" για το ένα cluster/partition και το "1" για το άλλο, δηλ. θα προκύψουν οι ταμπέλες, αν ξεκινήσαμε με "0", "00" και "10". Στη συνέχεια, ακολουθείται αναδρομικά η ίδια διαδικασία. Καταλήγουμε με ένα binary string για κάθε pixel, όπου μετατρέποντάς το σε αριθμό του δεκαδικού συστήματος, προκύπτουν τελικά οι ταμπέλες του clustering.

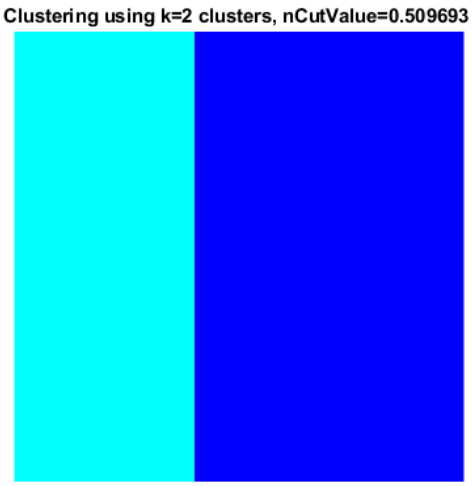


Εικόνα 4. *Παράδειγμα δυαδικού δέντρου για labeling των clusters*

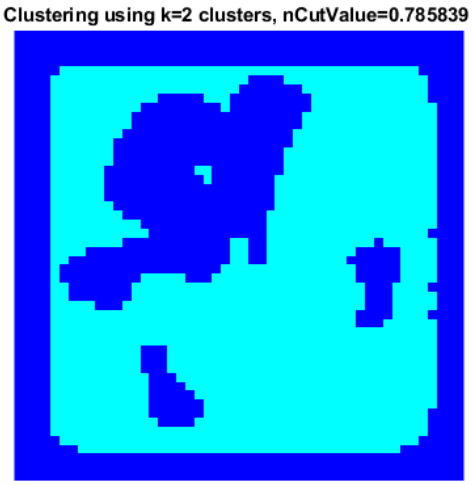
Βλέποντας ως παράδειγμα την εικόνα 4, ξεκινώντας ως αρχική ταμπέλα το "1", και ακολουθώντας την παραπάνω διαδικασία, θα προέκυπταν 4 clusters, με ταμπέλες, από αριστερά προς τα δεξιά, "001", "101", "011", "111", ή 2, 5, 3, 7. (Δε μας ενδιαφέρει να είναι σε κάποια σειρά οι αριθμοί, απλώς να είναι μοναδικές οι ταμπέλες για κάθε cluster που προκύπτει)

6 Demo 3

6.1 3a



Εικόνα 5. *Ένα βήμα για αναδρομικό ncuts στην εικόνα d2a*



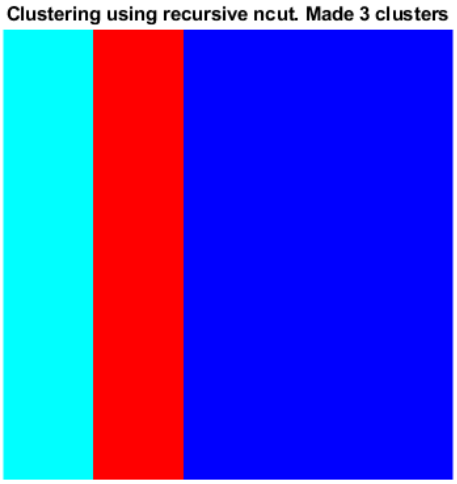
Εικόνα 6. *Ένα βήμα για αναδρομικό ncuts στην εικόνα d2b*

Παρατηρούμε ότι ένα βήμα του αναδρομικού ncuts, είναι πρακτικά ένα βήμα του μη-αναδρομικού ncuts ή spectral clustering για k=2. Συνεπώς, τα αποτελέσ-

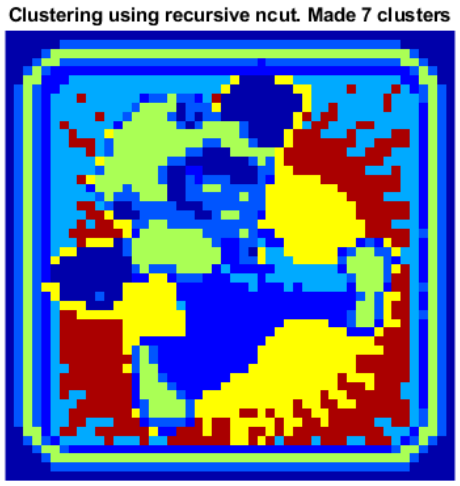
ματα που παίρνουμε είναι ίδια με αυτά του Demo 2, για k=2. Γνωρίζουμε ότι η εικόνα d2a πρέπει να χωριστεί άλλη μια φορά, ώστε να προκύψουν 3 clusters, άρα συμπεραίνουμε ότι η τιμή ncut=0.51 προκύπτει είναι αρκετά κοντά στο επιθυμητό threshold για να πετύχουμε καλό clustering (δηλ. να χωριστούν και οι 3 λωρίδες). Πράγματι, θα δούμε ότι στο 3b, η τιμή στην οποία επιτυγχάνεται αυτό είναι κοντά στο 0.6.

Για την εικόνα d2b, παρατηρούμε τιμή ncut=0.78. Τιμή πιο ψηλή από αυτήν στην d2a, μιας και παρατηρούμε ότι η φιγούρα και το φόντο της εικόνας διαφέρουν πολύ παραπάνω, από ότι λωρίδες RGB στην πρώτη εικόνα. Άρα λογικό είναι, αυτό το χώρισμα, να είναι "καλύτερο" (μας δίνει παραπάνω πληροφορία), από αυτό στην πρώτη εικόνα. Μας φανερώνεται επίσης ότι για όρια στην ncut value θα πρέπει να βάλουμε μια μεγαλύτερη τιμή, σε σχέση με την πρώτη εικόνα. Πράγματι, για τιμή κοντά στο 1 επιτυγχάνεται ικανοποιητικό clustering.

6.2 3b



Εικόνα 7. *Αναδρομικό ncuts στην εικόνα d2a*



Εικόνα 8. *Αναδρομικό ncuts στην εικόνα d2b*

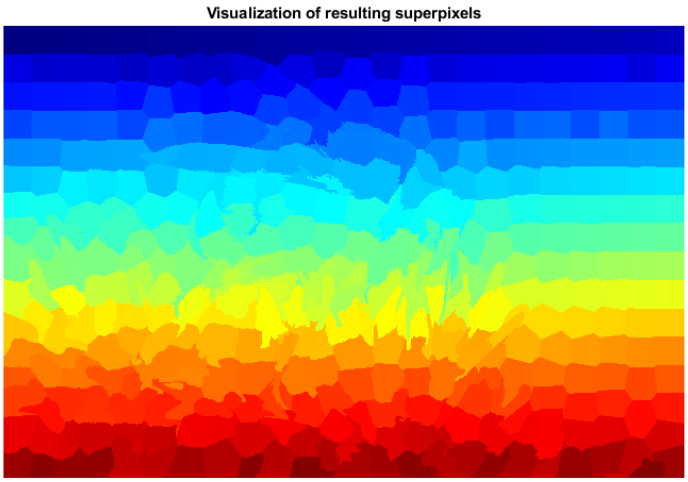
Παραπάνω παρουσιάζονται τα αποτελέσματα του αναδρομικού ncuts αλγορίθμου για τις δοκιμαστικές εικόνες d2a, d2b.

Για την εικόνα d2a, με τα thresholds που δίνονται στην εκφώνηση $t_1 = 5, t_2 = 0.6$, επιτυγχάνεται "καθαρό" clustering των τριών λωριδών. Μεγαλύτερο ρόλο σε αυτόν τον χωρισμό, όμως, έπαιξε το 2ο όριο, δηλαδή αυτό της τιμής ncut value. Άλλωστε παρατηρήθηκε ότι και με αύξηση του t_1, πάλι προκύπτει το ίδιο αποτέλεσμα. Αυτό συμβαίνει καθώς ήδη από το πρώτο χώρισμα προκύπτουν 2 clusters, που το πρώτο χωρίζεται μία ακόμη φορά, για να προκύψουν οι 2 αριστερά λωρίδες, ενώ το αριστερά έχει ήδη επιτύχει καλή ncut τιμή, και δεν χωρίζεται άλλο. Όπως είχαμε δει από το 3a, η τιμή του ορίου δε θα είχε μεγάλη διαφορά από την ncut που προκύπτει γαι 1 βήμα, καθώς υπήρχε ήδη πολύ καλή συσχέτιση εντός του δεξιά cluster, και μεγάλη διαφοροποίηση μεταξύ των 2 clusters.

Για την εικόνα d2b, χρησιμοποιήθηκαν όρια $t_1 = 300, t_2 = 1$. Όπως παρατηρήθηκε ήδη από το 3a, θα θέλαμε μεγάλο threshold για την ncut value, καθώς από το 1ο κιόλας βήμα η τιμή ήταν κοντά στο 0.8. Αντίστοιχα, για την τιμή t_1, που εκφράζει το ελάχιστο αριθμό από pixels σε ένα cluster, θέλουμε και αυτή να είναι υψηλή, καθώς τα διακριτά χαρακτηριστικά της εικόνας, όπως πχ τα γάντια, τα παππούτσια, το καπέλο, το φόντο, κλπ, όλα αποτελούνται από πολλά pixels. Ελέγχοντας για δίαφορες άλλες τιμές των ορίων, παρατηρήσαμε ότι κρατώντας σταθερό το όριο $t_2 = 1$, και μειώνοντας το t_1, δημιουργούνται πολλά clusters, ενώ κρατώντας σταθερό το $t_1 = 300$ και μειώνοντας το t_2, ή κρατώντας σταθερό το όριο $t_2 = 1$, και αυξάνοντας το t_1 δημιουργούνται λίγα (αυξάνοντας το t_2 με σταθερό το t_1 δεν είχε διαφορά).

7 Demo 4

Σε αυτό το κομμάτι έγινε η χρήση αναδρομικού και μη-αναδρομικού ncuts αλγορίθμου σε εικόνα που έχει προεπεξεργαστεί και μετατραπεί σε superpixels, οπότε ουσιαστικά παρουσιάζεται μια ολοκληρωμένη διαδικασία image segmentation.
640 x 425 Η εικόνα εισόδου που μας δίνεται για αυτό το demo έχει διαστάσεις 640x425. Αυτό σημαίνει ότι για να γινόταν αναπαράσταση της εικόνας ως γράφος, θα χρειαζόταν ένας πίνακας διαστάσεων $(640*425)x(640*425)$ ή $272000x272000$! Για αρχικοποίηση τέτοιου πίνακα, το MATLAB χρειάζεται 551.2GB, που μας δείχνει ότι για να κάνουμε image segmentation σε αυτήν την εικόνα είναι πρακτικά αδύνατο και υπολογιστικά πολύ κοστοβόρο, χωρίς την προεπεξεργασία της εικόνας. Χωρίζοντας την εικόνα πρώτα σε superpixels μπορούμε να επιτύχουμε την αναπαράσταση της εικόνας σε πολύ μικρότερο μέγεθος, ενώ πρακτικά δεν χάνουμε μεγάλο μέρος της πληροφορίας. Συγκεκριμένα, με τις παραμέτρους που δόθηκαν στην εκφώνηση, επιτυγχάνεται αναπαράσταση της εικόνας από 390 superpixels. Δηλαδή πλέον μπορεί να αναπαρασταθεί από γράφο με τη χρήση ενός πίνακα μεγέθους 390x390.

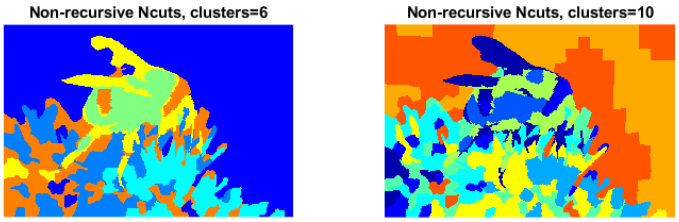


Εικόνα 9. *Χωρισμός της εικόνας σε superpixels*

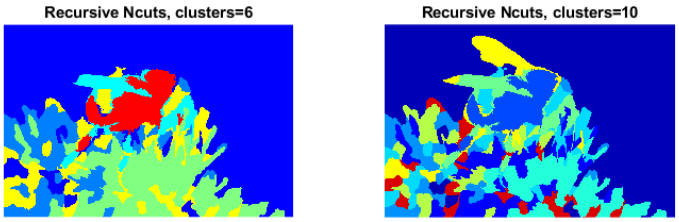
Οπότε πλέον κάθε pixel της αρχικής εικόνας ανήκει σε ένα από τα 390 superpixels που έχουν δημιουργηθεί. Με τη χρήση του superpixel descriptor που υλοποιήσαμε, κάθε pixel ενός superpixel, παίρνει το μέσο όρο των RGB τιμών των pixels που ανήκουν σε αυτό το superpixel. Άρα κατά προέκταση, πλέον κάθε superpixel έχει RGB τιμή τον μέσο όρο όλων των RGB τιμών των pixels που ανήκουν σε αυτό. Το αποτέλεσμα αυτής της διαδικασίας φαίνεται στην εικόνα 10.

Εικόνα 10. *Η εικόνα αφού χωριστεί σε superpixels και εφαρμοστεί ο superpixel descriptor*

Αφού έχουμε πλέον μια εικόνα διαχειρίσιμου μεγέθους, ακολουθείται η ίδια διαδικασία όπως παραπάνω, δηλ. δημιουργείται ο affinity πίνακας της εικόνας και πάνω σε αυτόν εφαρμόζονται οι αλγόριθμοι του αναδρομικού και μη-αναδρομικού ncuts. Οι αλγόριθμοι αυτοί δίνουν μια ταμπέλα cluster σε κάθε superpixel. Για να αναπαραστήσουμε αυτό στην αρχική μας εικόνα, αρκεί να δώσουμε την ίδια ταμπέλα cluster που έχει το superpixel σε όλα τα pixels που ανήκουν στο συγκεκριμένο superpixel.



Εικόνα 11. *Εφαρμογή του μη αναδρομικού ncuts αλγορίθμου*



Εικόνα 12. *Εφαρμογή του αναδρομικού ncuts αλγορίθμου*

Στις εικόνες 11, παρουσιάζονται τα αποτελέσματα και των 2 αλγορίθμων για να επιτευχθούν 6 και 10 clusters. Στον αναδρομικό ncuts αλγόριθμο για να δημιουργηθούν 6 clusters χρησιμοποιήθηκαν τα thresholds $t1 = 5, t2 = 0.98$, ενώ για δημιουργία 10 clusters $t1 = 12, t2 = 0.9953$.
Παρατηρούμε ότι, ενώ τα αποτελέσματα μεταξύ των δύο εκδόσεων είναι παρόμοια, ότι με τον μη αναδρομικό επιτυγχάνουμε καλύτερο διαχωρισμό της πληροφορίας.

Αυτό γίνεται φανερό κυρίως όταν δημιουργούνται 10 clusters. Στο μη αναδρομικό αλγόριθμο, όλο το φόντο της εικόνας είναι 1 cluster, ενώ στο μη αναδρομικό, χωρίζεται σε 2. Επιπλέον, υπάρχει καλύτερη διάκριση των στοιχείων της μέλισσας, όπως πχ μεταξύ των 2 φτερών. Αρά βλέπουμε ότι η αναδρομική εκδοχή είναι πιο robust στο να κάνει διάκριση των διακριτών στοιχείων της εικόνας.