Αναφορά 3ης εργασίας του μαθήματος «Ψηφιακή Επεξεργασία Εικόνας»

Χρυσολόγου Γεώργιος (ΑΕΜ: 10782)

Στα πλαίσια της εργασίας αυτής, κλήθηκα να υλοποιήσω ένα σύνολο συναρτήσεων σε κώδικα γλώσσας Python. Συγκεκριμένα, κατασκεύασα την συνάρτηση ‘’image_to_graph’’ για την αναπαράσταση εικόνων σαν γράφους, καθώς και τις συναρτήσεις ‘’ spectral_clustering’’ και ‘’ n_cuts’’, ‘’ calculate_n_cut_value’’, ‘’ n_cuts_recursive’’ για την κατάτμηση εικόνων σύμφωνα με τους αλγορίθμους Spectral Clustering και Normalized-cuts (μη αναδρομικό και αναδρομικό), αντίστοιχα.

Περιγραφή λειτουργίας συναρτήσεων

Τα δεδομένα που χρησιμοποιούνται έχουν αντληθεί από το δοσμένο αρχείο dip_hw_3.mat, το οποίο περιέχει έναν έτοιμο affinity πίνακα και 2 RGB εικόνες που αναπαρίστανται με τη μορφή τρισδιάστατων πινάκων.

image_to_graph(**img_array**: np.ndarray)

-> **affinity_mat**: np.ndarray

Η συνάρτηση image_to_graph μετατρέπει την εικόνα εισόδου σε έναν Affinity πίνακα. Δέχεται ως είσοδο τον πίνακα img_array, ο οποίος αναπαριστά την RGB εικόνα εισόδου με τη μορφή τρισδιάστατου πίνακα διαστάσεων N×M×3, όπου N το ύψος (πλήθος γραμμών), M το πλάτος (πλήθος στηλών) και 3 τα χρωματικά κανάλια (R, G, B). Κατά την κλήση της συνάρτησης, αρχικοποιούνται οι μεταβλητές N, M, C με τις αντίστοιχες διαστάσεις του πίνακα, και κατόπιν δημιουργείται ένας πίνακας affinity_mat διαστάσεων NM×NM, ο οποίος θα χρησιμοποιηθεί για την αποθήκευση των ζυγισμένων ακμών του γράφου. Η συνάρτηση διατρέχει με διπλή επανάληψη for όλα τα pixel της εικόνας, θεωρώντας κάθε pixel (I,J) ως κόμβο του γράφου. Για κάθε τέτοιο κόμβο, ανακτάται το RGB διάνυσμα τιμών pixel = img_array[I][J][:]. Ακολούθως, σε εσωτερικό ζεύγος επαναλήψεων for, το pixel (I,J) συγκρίνεται με όλα τα pixel της εικόνας (συμπεριλαμβανομένου του ίδιου): για κάθε άλλο pixel (i,j), υπολογίζεται η ευκλείδεια απόσταση μεταξύ των RGB διανυσμάτων τους μέσω της εντολής d = np.linalg.norm(pixel - img_array[i][j][:]). Η απόσταση αυτή χρησιμοποιείται για να καθοριστεί το βάρος της ακμής που συνδέει τα δύο pixel στον γράφο, το οποίο αποθηκεύεται στη θέση affinity_mat[I*M+J][i*M+j] ως $\frac{1}{e^d}$. Είναι σημαντικό να σημειωθεί ότι οι συντεταγμένες του affinity_mat επιλέχθηκαν με αυτόν τον τρόπο για την αντιστοίχιση των δισδιάστατων θέσεων των pixels του img_array σε μία διάσταση.

spectral_clustering(**affinity_mat**: np.ndarray, **k**: int)

-> **cluster_idx**: np.ndarray

Η συνάρτηση spectral_clustering πραγματοποιεί κατάτμηση της εικόνας εισόδου με βάση τον αλγόριθμο Spectral Clustering. Δέχεται ως εισόδους έναν πίνακα γειτνίασης affinity_mat διαστάσεων N×N καθώς και έναν ακέραιο αριθμό k, που δηλώνει το πλήθος των ομάδων (clusters) στις οποίες θα κατατμηθεί ο γράφος. Κατά την κλήση της συνάρτησης, αρχικοποιείται ένας πίνακας D ίδιων διαστάσεων με τον πίνακα affinity_mat. Για κάθε γραμμή του πίνακα affinity_mat, υπολογίζεται το άθροισμα όλων των συνδεδεμένων ακμών (δηλαδή το άθροισμα των στοιχείων της γραμμής), και η τιμή αυτή τοποθετείται στη διαγώνιο του πίνακα D, στη θέση D[i][i]. Κατόπιν, υπολογίζεται ο Λαπλασιανός πίνακας $L = D - affinity\_mat$. Στη συνέχεια, μέσω της συνάρτησης eigs, υπολογίζονται τα k ιδιοδιανύσματα του πίνακα L που αντιστοιχούν στις k μικρότερες ιδιοτιμές (παράμετρος which='SM', Smallest Magnitude).

Το αποτέλεσμα αποθηκεύεται στον πίνακα U, ο οποίος μετατρέπεται σε πραγματικό μέσω της συνάρτησης np.real, καθώς τα ιδιοδιανύσματα επιστρέφονται σε μιγαδική μορφή. Τέλος, χρησιμοποιείται οι συνάρτήσεις kmeans.fit(U) (όπου kmeans = KMeans(n_clusters=k, random_state=1) και cluster_idx = kmeans.labels, μέσω των οποίων ορίζεται σε ποιο από τα k clusters ανήκει κάθε pixel και επιστρέφεται ο πίνακας cluster_idx.

n_cuts(**affinity_mat**: np.ndarray, **k**: int)

-> **cluster_idx**: np.ndarray

Η συνάρτηση n_cuts πραγματοποιεί κατάτμηση της εικόνας εισόδου με βάση την μη αναδρομική εκδοχή του αλγορίθμου Normalized-Cuts. Εμφανίζει σχεδόν απόλυτη ομοιότητα με την συνάρτηση spectral_clustering που περιγράφηκε παραπάνω. Μοναδική διαφορά στον κώδικα της αποτελεί ο ορισμός της παραμέτρου M=D στην συνάρτηση eigs, ώστε να υπολογιστούν οι k μικρότερες ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα του προβλήματος $Lx = \lambda Dx$ (και όχι του προβλήματος $Lx = Dx$ για το οποίο γίνονται οι αντίστοιχοι υπολογισμοί στην συνάρτηση spectral_clustering).

calculate_n_cut_value(**affinity_mat**: np.ndarray, **cluster_idx**: np.ndarray)

-> **n_cut_value**: float

Η συνάρτηση calculate_n_cut_value υπολογίζει την τιμή της μετρικής Normalized Cut (Ncut) για ένα δοσμένο διαχωρισμό ενός γράφου σε δύο clusters. Δέχεται ως εισόδους τον πίνακα γειτνίασης affinity_mat διαστάσεων N×N και το διάνυσμα cluster_idx με μήκος N*N (όπου N το πλήθος των κόμβων ή pixels της εικόνας), το οποίο περιέχει την ετικέτα του cluster στο οποίο ανήκει κάθε κόμβος (ή pixel), με την υπόθεση ότι υπάρχουν μόνο δύο διαφορετικά clusters, A και B. Κατά την κλήση της συνάρτησης, ανακτώνται οι δύο μοναδικές ετικέτες από το cluster_idx μέσω της εντολής labelA, labelB = np.unique(cluster_idx). Στη συνέχεια, εντοπίζονται τα indices των κόμβων που ανήκουν στο κάθε cluster, τα οποία αποθηκεύονται στους πίνακες clusterA και clusterB αντίστοιχα. Κατόπιν, υπολογίζονται, μέσω διπλών for, οι τιμές assoc(A, A), assoc(B, B), assoc(A, V) και assoc(B, V). Τέλος, υπολογίζονται μέσω των δοσμένων σχέσεων οι τιμές Nassoc(A,B), Ncut(A,B) και η τελευταία επιστρέφεται.

n_cuts_recursive(**affinity_mat**: np.ndarray, **T1**: int, **T2**: float, parent_label: int)

-> **cluster_idx**: np.ndarray

Η συνάρτηση n_cuts_recursive υλοποιεί την αναδρομική μορφή του αλγορίθμου Normalized Cuts, με σκοπό την κατάτμηση εικόνας. Δέχεται ως είσοδο έναν πίνακα γειτνίασης affinity_mat διαστάσεων N×N (όπου N το πλήθος κόμβων ή pixels της εικόνας) και δύο κατώφλια T1 και T2 που καθορίζουν τα κριτήρια τερματισμού της αναδρομής (κατώφλι μεγέθους cluster και κατώφλι ποιότητας διαχωρισμού, αντίστοιχα). Σημαντικό είναι να σημειωθεί ότι στις παραπάνω εισόδους έχει προστεθεί μία integer μεταβλητή ‘’parent_label”, στην οποία αποθηκεύεται η τιμή της ετικέτας ενός cluster πρωτού κληθεί αναδρομικά η συνάρτηση και το cluster αυτό διαχωριστεί σε δύο μικρότερα clusters. Σε περίπτωση που ο έλεγχος επιδείξει ότι η τελευταία κατάτμηση αυτή δεν είναι ικανοποιητική, οι κόμβοι των δύο μικρότερων clusters συγκεντρώνονται πίσω στο αρχικό cluster, με την προηγούμενη ετικέτα parent_label.

Κατά την κλήση της συνάρτησης, αρχικά εφαρμόζεται η μέθοδος n_cuts για να διαχωριστεί το γράφημα σε δύο clusters. Οι δύο διακριτές ετικέτες που επιστρέφει η n_cuts αντικαθίστανται με νέες τυχαίες ακέραιες

ετικέτες new_labelA και new_labelB, ώστε να διασφαλιστεί μοναδικότητα των labels σε κάθε επίπεδο της αναδρομής.

Στη συνέχεια, εντοπίζονται οι κόμβοι που ανήκουν σε κάθε νέο cluster και υπολογίζεται η τιμή της μετρικής n_cut για τον διαχωρισμό που προέκυψε. Εάν έστω ένα από τα δύο clusters έχει μέγεθος μικρότερο από το κατώφλι T1 ή εάν η τιμή της μετρικής n_cut υπερβαίνει το κατώφλι T2, τότε η αναδρομή τερματίζεται για το συγκεκριμένο υποσύνολο και όλοι οι κόμβοι συγκεντρώνονται πίσω στο αρχικό cluster με την ετικέτα parent_label*.

Εφόσον το αποτέλεσμα του διαχωρισμού κριθεί αποδεκτό, ο πίνακας affinity_mat διασπάται σε δύο υποπίνακες sub_affinity_A και sub_affinity_B, που αντιστοιχούν στους κόμβους κάθε cluster. Η συνάρτηση καλεί τον εαυτό της αναδρομικά για καθένα από τα δύο υποσύνολα, περνώντας ως νέο parent_label τις αντίστοιχες new_labelA και new_labelB.

Κατά την επιστροφή από την αναδρομή, τα τελικά labels που υπολογίζονται για τους κόμβους των υπογράφων ανατίθενται στα αντίστοιχα σημεία του αρχικού cluster_idx. Τελικώς, επιστρέφεται ο πλήρης πίνακας cluster_idx, στον οποίο κάθε στοιχείο αντιστοιχεί στην ετικέτα cluster του αντίστοιχου κόμβου.

* **Η επαναφορά αυτή στο αρχικό cluster αποτελεί σχεδιαστική επιλογή που δεν συμφωνεί απόλυτα με τις υποδείξεις της εκφώνησης**. Με βάση την εκφώνηση, μετά τον διαχωρισμό ενός cluster σε δύο υπο-clusters, πραγματοποιείται έλεγχος (με χρήση των κατωφλίων T1,T2) μέσω του οποίου καθορίζεται εάν τα υπο-clusters θα διαχωριστούν εκ νέου. Για την εικόνα d2a, έπειτα από τον 1^{ο} διαχωρισμό (ο οποίος παρατηρήθηκε ότι οδηγεί στην απομόνωση του μπλε τμήματος από τα άλλα δύο), ο αλγόριθμος αποφασίζει μέσω του ελέγχου ότι τα δύο υπό-clusters θα διαχωριστούν εκ νέου, το οποίο αποδεικνύει εσφαλμένη συμπεριφορά της μεθόδου καθώς το cluster του μπλε τμήματος δεν θα πρέπει να υποστεί επιπλέον κατάτμηση. Με τον τροποποιημένο αλγόριθμο που προτείνεται στην παρούσα εργασία, έπειτα από κάθε κατάτμηση, τα δύο υπο-clusters διαχωρίζονται εκ νέου (μέσω επανάκλησης της n_cuts_recursive) και πραγματοποιείται έλεγχος που καθορίζει αν οι δύο τελευταίοι διαχωρισμοί ικανοποιούν τις συνθήκες που ορίζονται από τα κατώφλια. Σε περίπτωση που ο έλεγχος για κάποιον από τους δύο διαχωρισμούς αποδειχθεί αρνητικός, τα δύο νέα υπο-clusters που έχουν προκύψει επαναφέρονται στο αρχικό cluster-γονιό τους.

Παρουσίαση αποτελεσμάτων

Demo 1

```
Ετικέτες κόμβων για k=2: [1 1 1 1 0 0 0 0 0 0 0 0]
Ετικέτες κόμβων για k=3: [2 2 2 2 0 0 0 0 1 1 1 1]
Ετικέτες κόμβων για k=4: [2 2 3 3 0 0 0 0 1 1 1 1]
```

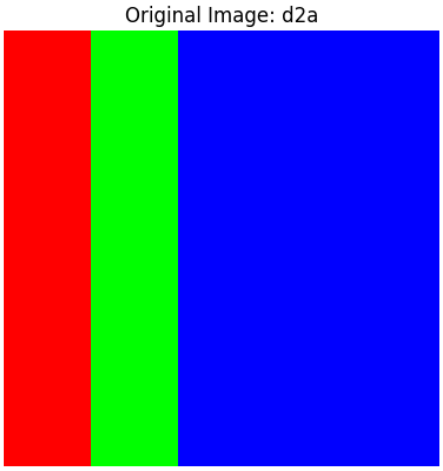
Στο Demo 1 εφαρμόστηκε ο αλγόριθμος spectral_clustering στον πίνακα γειτνίασης d1a του αρχείου dip_hw_3.mat για διαφορετικές τιμές του πλήθους των clusters k=2, 3, 4. Η έξοδος της συνάρτησης είναι το διάνυσμα cluster_idx, στο οποίο κάθε στοιχείο αντιστοιχεί στην ετικέτα ομάδας (cluster label) για τον αντίστοιχο κόμβο του γράφου.

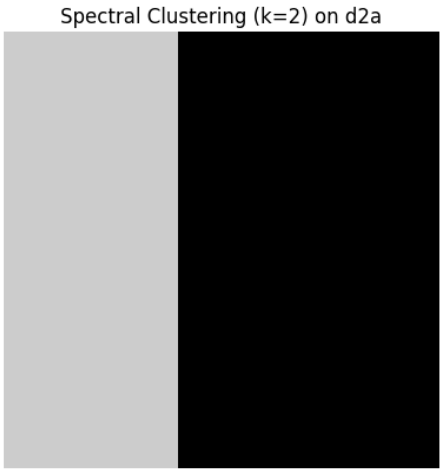
Για k=2, οι πρώτοι τρεις κόμβοι λαμβάνουν την ετικέτα 1, ενώ οι υπόλοιποι έξι την ετικέτα 0. Ο διαχωρισμός υποδηλώνει ύπαρξη μιας σαφούς διάκρισης μεταξύ δύο ομάδων κόμβων, πιθανόν με αρκετά ισχυρή εσωτερική συνοχή και μικρή μεταξύ τους συνδεσιμότητα.

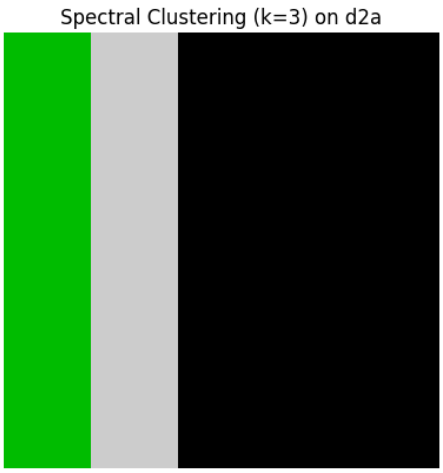
Για k=3, οι κόμβοι χωρίζονται στις ομάδες [2,2,2,0,0,0,1,1,1], γεγονός που υποδηλώνει ότι η ομάδα των έξι κόμβων για k=2 διαιρέθηκε σε δύο μικρότερες ομάδες (0 και 1), ενώ η αρχική ομάδα των τριών κόμβων παρέμεινε ίδια. Ο διαχωρισμός αυτός καθιστά αντιληπτό ότι η αύξηση του πλήθους των clusters οδηγεί στην ανάδειξη των διαφορών που υπάρχουν μεταξύ κόμβων που αρχικά ανήκουν στα ίδια clusters.

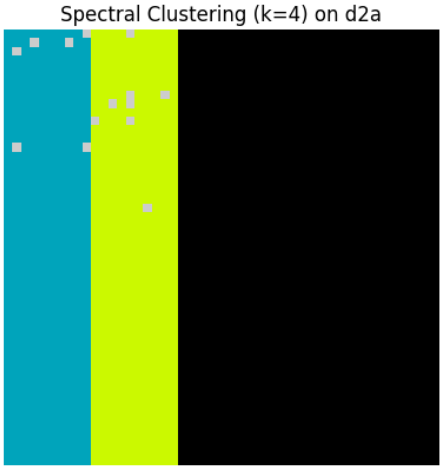
Για k=4, παρατηρείται περαιτέρω διάσπαση, με τους κόμβους να ομαδοποιούνται ως [2,2,3,0,0,0,1,1,1]. Σε αυτή την περίπτωση, εντοπίζεται και διάσπαση της αρχικής τριάδας. Το αποτέλεσμα αυτό ενισχύει το συμπέρασμα που διατυπώθηκε παραπάνω. Η αύξηση του πλήθους των clusters έχει ως συνέπεια τον διαχωρισμό των κόμβων σε ολοένα και πιο ολιγομελής ομάδες (clusters). Με την αύξηση της τιμής k, ακόμα και μικρές διαφορές μεταξύ των φωτεινοτήτων των pixels εντοπίζονται.

Demo 2









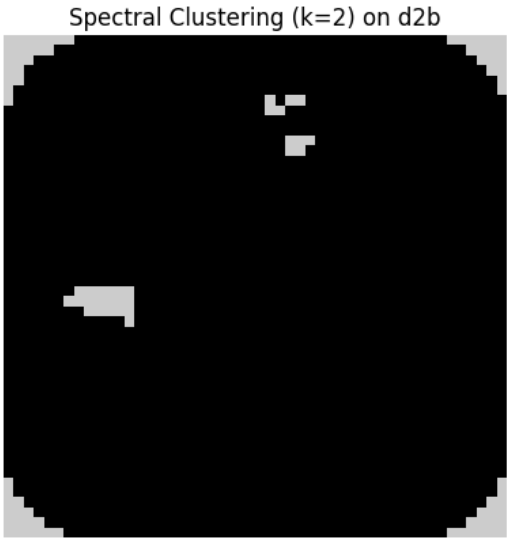
Παρατηρείται ότι η εικόνα d2a αποτελείται από 3 κατακόρυφα τμήματα διαφορετικών χρωμάτων. Συγκεκριμένα, το αριστερό τμήμα είναι κόκκινο (άρα οι RGB τιμές των pixel που το αποτελούν θα είναι [a, 0, 0], όπου a>0), το μεσαίο τμήμα είναι πράσινο (άρα οι RGB τιμές των pixel που το αποτελούν θα είναι [0, b, 0], όπου b>0) και το δεξιό τμήμα είναι μπλε (άρα οι RGB τιμές των pixel που το αποτελούν θα είναι [0, 0, c], όπου c>0). Εφαρμόζοντας την μέθοδο image_to_graph και κατόπιν την spectral_clustering, με όρισμα στην δεύτερη τον affinity πίνακα που προκύπτει από την πρώτη, αναμένουμε τον διαχωρισμό της μεθόδου στα τρία αυτά τμήματα με διαφορετικά χρώματα.

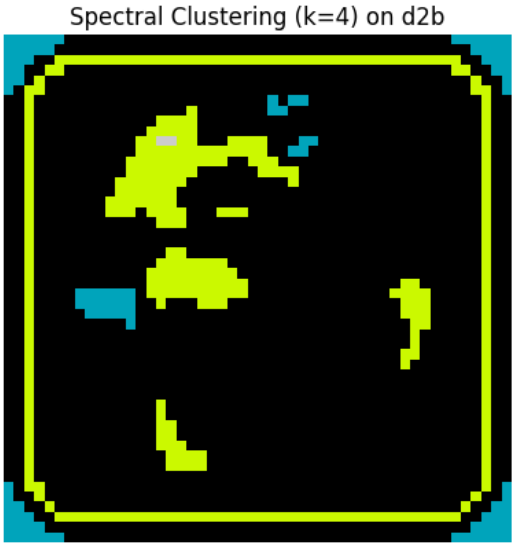
Παρατηρούμε ότι για k=2, η μέθοδος spectral_clustering διαχωρίζει την εικόνα σε δύο τμήματα. Το πρώτο αποτελείται από το κόκκινο και το πράσινο τμήμα της αρχικής εικόνας ενώ το δεύτερο από το μπλε τμήμα.

Για k=3, επιτυγχάνεται τέλειος διαχωρισμός της εικόνας, καθώς η μέθοδος διακρίνει τα τρία διαφορετικά τμήματα.

Για k=4, είναι εμφανές ότι η μέθοδος, αν και διαχωρίζει σωστά τις τρεις διαφορετικές χρωματικές περιοχές της αρχικής εικόνας, κατατάσσει ορισμένα pixels σε ένα άλλο cluster. Αυτό είναι συνέπεια της δημιουργίας τεσσάρων clusters, ενώ η εικόνα αποτελείται από μόνο τρία διαφορετικά τμήματα.

Οι παραπάνω παρατηρήσεις για τις διαφορετικές τιμές του πλήθους clusters αναδεικνύουν την σημαντικότητα της σωστής επιλογής της τιμής k. Για τιμή k μεγαλύτερη από το πλήθος των διαφορετικών τμημάτων της εικόνας, καθίσταται πιθανό το σφάλμα της μεθόδου Spectral Clustering στην κατάτμηση της εικόνας.





Η εικόνα d2b απεικονίζει τον χαρακτήρα “Super Mario” και, σε αντίθεση με την εικόνα d2a, περιλαμβάνει πληθώρα χρωμάτων και φωτεινοτήτων, καθιστώντας το πρόβλημα της κατάτμησης πιο απαιτητικό.. Με την εφαρμογή της μεθόδου spectral_clustering στον affinity matrix που προκύπτει από την image_to_graph, παράγονται εικόνες κατάτμησης με διαφορετικό αριθμό χρωματικών περιοχών, ανάλογα με την επιλεγμένη τιμή του k. Παρατηρείται ότι όσο το k αυξάνεται, τόσο περισσότερες λεπτομέρειες της

αρχικής εικόνας αναδεικνύονται, όπως το κόκκινο περίγραμμα και τα διακριτά χαρακτηριστικά του χαρακτήρα.

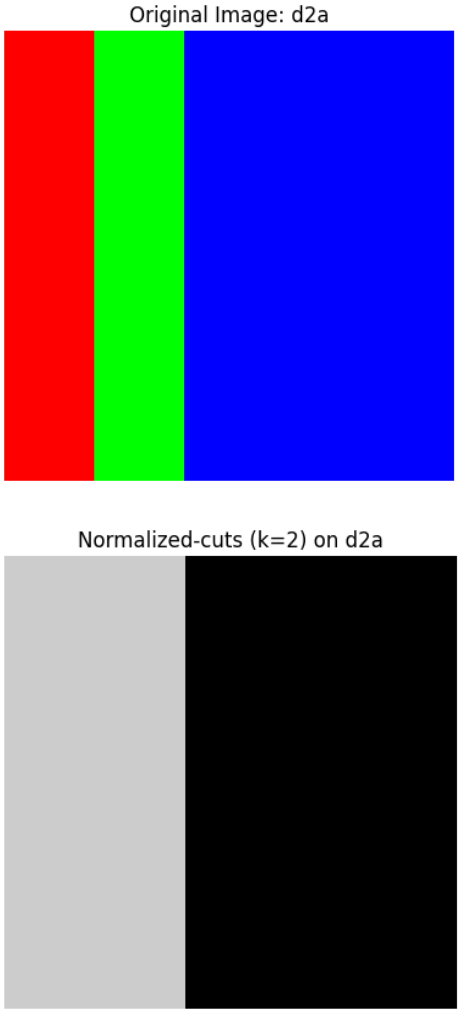
Συγκεκριμένα, για k=2 η εικόνα διαχωρίζεται σε δύο τμήματα. Ένα μικρό σύνολο pixels, που αντιστοιχούν κυρίως στις λευκές γωνίες του περιγράμματος καθώς και στα λευκά γάντια του χαρακτήρα, ομαδοποιείται ως ξεχωριστό cluster, ενώ όλα τα υπόλοιπα pixels κατατάσσονται στο άλλο cluster.

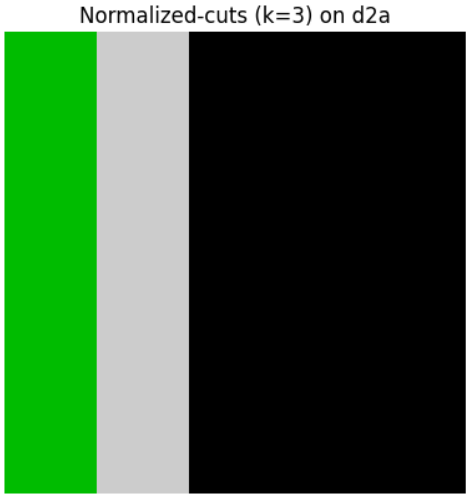
Για k=3, η μέθοδος επιτυγχάνει να διακρίνει τα άσπρα και τα κόκκινα pixel της εικόνας με ικανοποιητική ακρίβεια. Στο ίδιο cluster κατατάσσονται σχεδόν τα ίδια pixel άσπρου χρώματος με την περίπτωση k=2, ενώ μεγάλο μέρος των κόκκινων τμημάτων της ενδυμασίας του χαρακτήρα καθώς και του κόκκινου περιγράμματος αποτελούν το άλλο cluster. Είναι σημαντικό να σημειωθεί ότι, λόγω της παρόμοιας ευκλείδιας απόστασης των RGB διανυσμάτων των κόκκινων και τον καφέ (στα παπούτσια) pixels, η μέθοδος τα συγκεντρώνει στο ίδιο cluster. Το τρίτο cluster περιλαμβάνει τα υπόλοιπα pixels, κυρίως μπλε χρώματος.

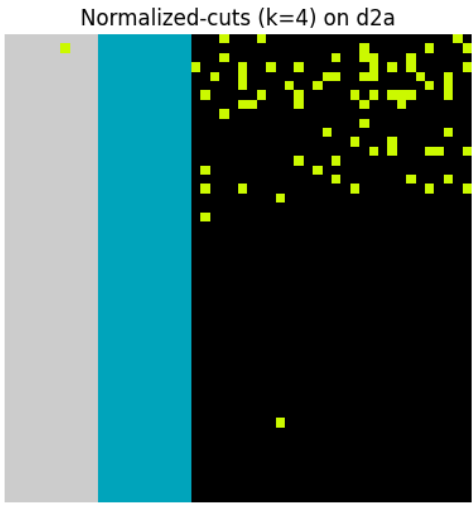
Για k=4, δεν παρατηρείται κάποια σημαντική αλλαγή σε σχέση με την εικόνα που προκύπτει για k=3.

Συμπεραίνεται, λοιπόν, η αδυναμία της μεθόδου να κατατμήσει ικανοποιητικά εικόνες με πλούσια λεπτομέρεια, όπως η d2b, για μικρές τιμές k. Αυτό συμβαίνει διότι η εικόνα αποτελείται από περισσότερα χρωματικά τμήματα σε σχέση με τα clusters πλήθους k στα οποία καλείται να την διαχωρίσει τα pixels η μέθοδος.

Demo 3a







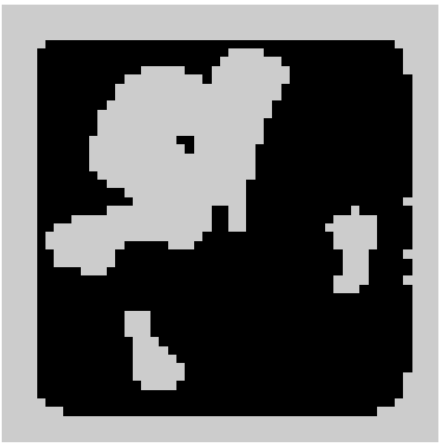
Τα αποτελέσματα της εφαρμογής της μεθόδου n_cuts στην εικόνα d2a για τις διάφορες τιμές της τιμής k εμφανίζουν σημαντική ομοιότητα με τα αντίστοιχα για την μέθοδο spectral_clustering.

Για k=2, η μέθοδος ομαδοποιεί τα μπλε pixel στο ένα cluster, ενώ τα κόκκινα και τα πράσινα pixels αποτελούν το άλλο cluster. Για k=3, σημειώνεται τέλεια κατάτμηση της εικόνας στις τρεις διαφορετικές χρωματικές περιοχές της. Για k=4, παρατηρείται το ίδιο φαινόμενο με την περίπτωση του αλγορίθμου Spectral Clustering για το ίδιο πλήθος clusters. Η μέθοδος, λόγω της ανάγκης διαχωρισμού μίας εικόνας τριών διακριτών χρωματικών τμημάτων σε τέσσερα clusters, κατατάσσει λανθασμένα ορισμένα pixels σε μία ξεχωριστή ομάδα.

Συμπερασματικά, η n_cuts, όπως και η spectral_clustering, αποδίδει ικανοποιητικά αποτελέσματα κατά την κατάτμηση απλών σε λεπτομέρεια εικόνων, όταν ο αριθμός k των clusters επιλέγεται κατάλληλα.

Original Image: d2b

Normalized-cuts (k=2) on d2b



Normalized-cuts (k=3) on d2b

Normalized-cuts (k=4) on d2b

Όπως και στην περίπτωση της εφαρμογής της μεθόδου spectral_clustering στην εικόνα d2b, η αύξηση της τιμής k επιτρέπει και στη μέθοδο n_cuts να αναδείξει περισσότερες λεπτομέρειες της εικόνας.

Για k=2, η εικόνα διαχωρίζεται σε δύο ομάδες. Στην πρώτη κατατάσσονται όλα τα pixels με γαλάζιες ή μπλε αποχρώσεις, δηλαδή το φόντο και η φόρμα του χαρακτήρα, ενώ στη δεύτερη τοποθετούνται τα υπόλοιπα.

Με την αύξηση των clusters σε k=3, το cluster που αποτελούταν από όλα τα pixels που δεν έχουν μπλε απόχρωση διαχωρίζεται περαιτέρω. Πιο συγκεκριμένα, τα κόκκινα pixels διακρίνονται από τα υπόλοιπα και ομαδοποιούνται σε τρίτο cluster. Το νέο αυτό cluster περιλαμβάνει τα pixels του εξωτερικού κόκκινου περιγράμματος της εικόνας, του καπέλου και των μανικιών του χαρακτήρα, καθώς και ορισμένα τμήματα του προσώπου του.

Τέλος, για k=4, η μέθοδος επιτυγχάνει να διαχωρίσει το αρχικό cluster των μπλε pixels με βάση την διαφορά στην φωτεινότητα τους. Είναι φανερό ότι τα pixels της σκούρας μπλε φόρμας του χαρακτήρα σε συνδυασμό με ορισμένα pixels στις εσωτερικές άκρες του περιγράμματος της εικόνας αποτελούν ένα ξεχωριστό clusters, διαφορετικό από αυτό των γαλάζιων pixels του φόντο.

Η σύγκριση των δύο μεθόδων κατάτμησης στην εικόνα d2b καταδεικνύει ότι η μέθοδος n_cuts είναι σημαντικά πιο αποτελεσματική σε σχέση με τη spectral_clustering.

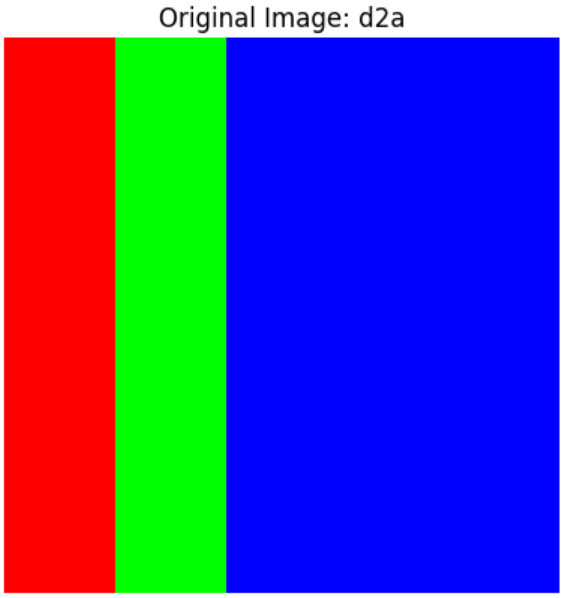
Παρατηρείται ότι για τις ίδιες τιμές k, η spectral_clustering αποτυγχάνει να διακρίνει με ακρίβεια τις χρωματικά διαφορετικές περιοχές της εικόνας. Αντιθέτως, η n_cuts καταφέρνει να διαχωρίσει σημαντικό κομμάτι των χαρακτηριστικών του χαρακτήρα, όπως τα γάντια, τα μανίκια, το καπέλο και η φόρμα.

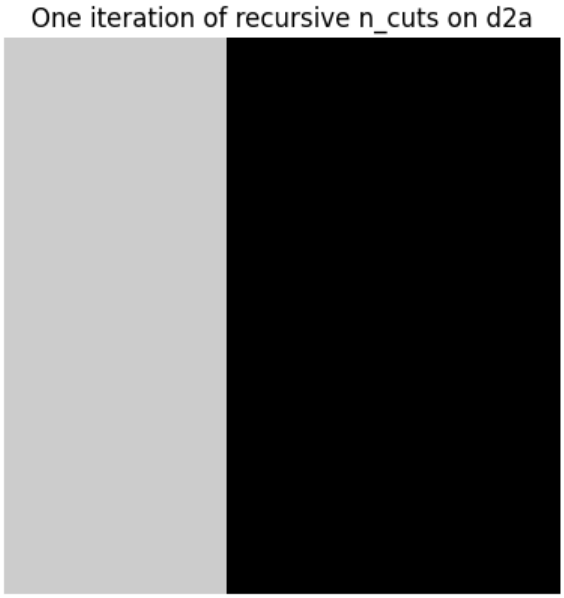
Ακόμη, στην περίπτωση k=4, η spectral_clustering πραγματοποιεί ομαδοποίηση του γαλάζιου φόντο με την μπλε φόρμα του χαρακτήρα, ενώ η n_cuts τα διαχωρίζει αποτελεσματικά, διακρίνοντας τη διαφορά στη φωτεινότητα.

Συμπεραίνεται, λοιπόν, ότι σε εικόνες με μεγάλη λεπτομέρεια και χρωματική ποικιλία, όπως η d2b, η μέθοδος n_cuts υπερτερεί έναντι της spectral_clustering, οδηγώντας σε πιο ικανοποιητική κατάτμηση.

Demo 3b

Η αναδρομική μέθοδος n_cuts για ένα βήμα πρόκειται για την μέθοδο n_cuts για k=2, καθώς στο πρώτο βήμα της αναδρομικής μεθόδου τα pixels της εικόνας χωρίζονται σε δύο clusters.

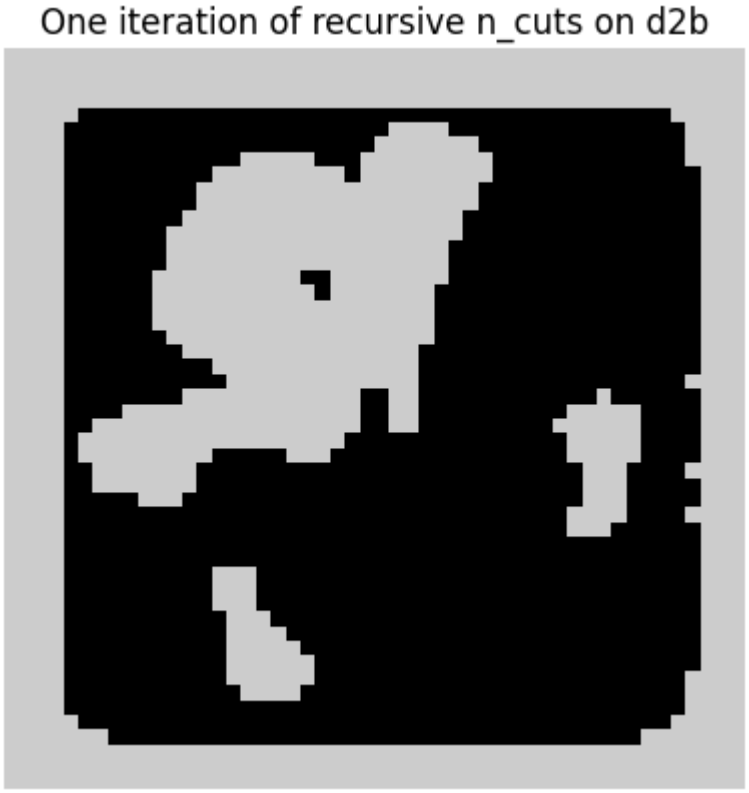





```
Τιμή μετρικής n_cuts για d2a: 0.5092379242331586
```

Παρατηρείται ότι η αναδρομική διαδικασία n_cuts για ένα βήμα οδηγεί στο ίδιο αποτέλεσμα με την μέθοδο spectral_clustering για k=2. Το μπλε τμήμα της εικόνας d2a αποτελεί το ένα cluster, ενώ το κόκκινο και το πράσινο τμήμα κατατάσσονται στο άλλο cluster.

Η τιμή της μετρικής ncuts κατά την κατάτμηση αυτή υπολογίζεται περίπου 0.5, υποδεικνύοντας ότι πρόκειται για ένα ποιοτικό διαχωρισμό.




```
Τιμή μετρικής n_cuts για d2b: 0.7852852891221864
```

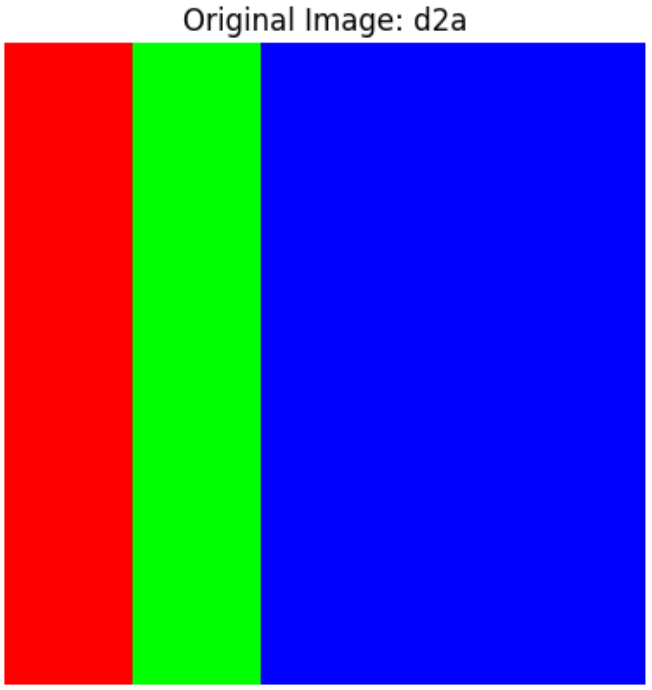
Παρατηρείται ότι η αναδρομική διαδικασία n_cuts για ένα βήμα οδηγεί στον διαχωρισμό των pixels της εικόνας σε ομάδες, όπως αναμενόταν. Συγκεκριμένα, στην πρώτη κατατάσσονται όλα τα pixels με γαλάζιες ή μπλε αποχρώσεις, δηλαδή το φόντο και η φόρμα του χαρακτήρα, ενώ στη δεύτερη τοποθετούνται τα υπόλοιπα.

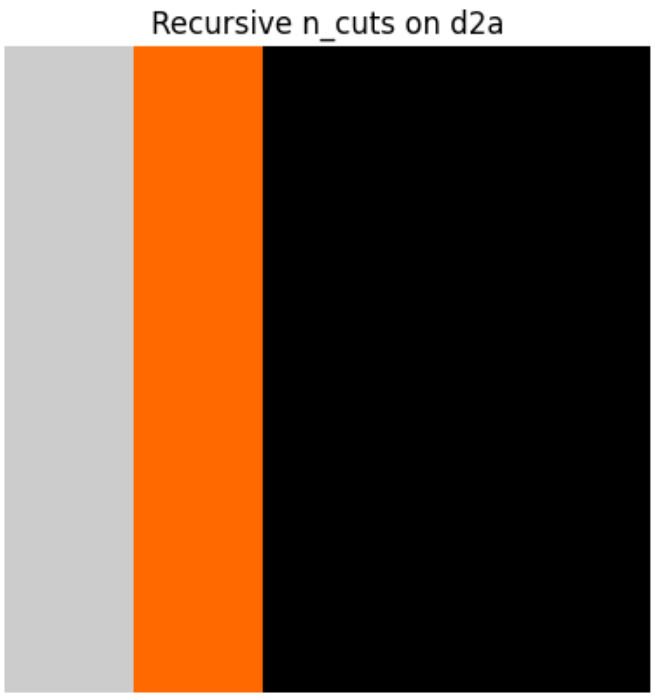
Η τιμή της μετρικής n_cuts κατά την κατάτμηση αυτή υπολογίζεται περίπου 0.785 . Σε σύγκριση με την αντίστοιχη τιμή για την κατάτμηση της εικόνας d2a παραπάνω, είναι μεγαλύτερη. Ως συμπέρασμα, ο διαχωρισμός αυτός θεωρείται υποδεέστερος ποιοτικά σε σχέση με αυτόν της d2a, γεγονός το οποίο οφείλεται στην ύπαρξη pixels αρκετά διαφορετικών χρωμάτων και φωτεινοτήτων εντός των δύο clusters που προέκυψαν στην περίπτωση της d2b (το ένα cluster περιλαμβάνει κόκκινα, καφέ και λευκά pixels).

Παρατηρώντας τα αποτελέσματα της μεθόδου n_cuts για ένα βήμα σε σχέση με αυτά της μεθόδου spectral_clustering, είναι εμφανές ότι η πρώτη οδηγεί σε σημαντικά πιο ακριβή διαχωρισμό των pixels. Η n_cuts ομαδοποιεί όλα τα pixels με μπλε απόχρωση, τα οποία είναι πολλά περισσότερα από το κλάσμα των λεύκων pixels που επιτυγχάνει να διακρίνει η spectral_clustering.

Demo 3c

Ως κατώφλια για την μέθοδο n_cuts_recursive, ορίστηκαν οι τιμές T1=5 και T2=0.95. Η διαφοροποίηση της τιμής T2 σε σχέση με αυτήν που υποδεικνύεται από την εκφώνηση πραγματοποιήθηκε ως αποτέλεσμα πειραμάτων με πληθώρα τιμών, μεταξύ αυτών και T2=0.2, τα οποία οδήγησαν στο συμπέρασμα ότι αποτελεί την τιμή που οδηγεί στα ικανοποιητικότερα αποτελέσματα αναφορικά με τις δύο εικόνες εισόδου.

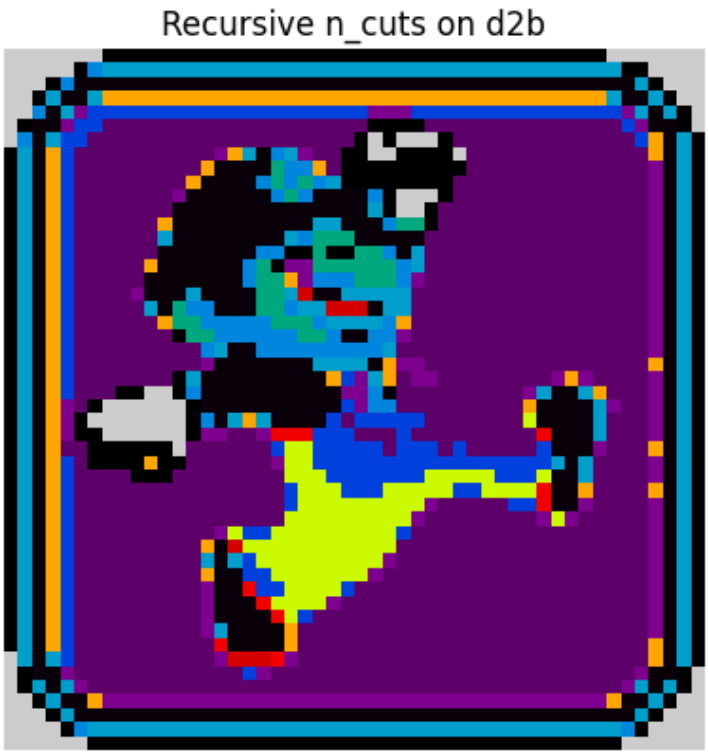




Παρατηρούμε ότι πραγματοποιείται τέλεια κατάτμηση της εικόνας d2a στα 3 τμήματα διαφορετικών χρωμάτων της.

Συγκρίνοντας το αποτέλεσμα αυτό με τα αντίστοιχα των μεθόδων spectral clustering και μη αναδρομικής n_cuts για k={2,3} , είναι εμφανές ότι υπάρχει απόλυτη ομοιότητα για την τιμή k=3. Η τιμή k=2, όπως αναλύθηκε και παραπάνω, οδηγεί σε αδυναμία των άλλων δύο μεθόδων να διαχωρίσουν την εικόνα σε πάνω από δύο τμήματα. Το φαινόμενο αυτό δεν εμφανίζεται για την αναδρομική μέθοδο n_cuts, καθώς ο

αλγόριθμος συνεχίζεται για οποιοδήποτε πλήθος τμημάτων, όσο ικανοποιούνται οι συνθήκες του ελέγχου διαχωρισμού.



Παρατηρώντας την εικόνα ‘’Recursive n_cuts on d2b”, παρατηρείται ότι η εικόνα έχει διαχωριστεί αναδρομικά σε μεγάλο πλήθος τμημάτων.

Σε σύγκριση με τις μεθόδους spectral clustering και μη αναδρομικής n_cuts για k={2,3}, σημειώνεται σημαντικά πιο ακριβής και λεπτομερής κατάτμηση, δεδομένης της ποικιλίας στα χρώματα και τις φωτεινότητες της εικόνας d2b. Η αδυναμία των δύο αυτών μεθόδων να διαχωρίσουν την εικόνα σε παραπάνω από k τμήματα δεν συναντάται και με την αναδρομική n_cuts, καθώς ο αλγόριθμος αυτός συνεχίζει τον επανειλημμένη κατάτμηση όσο οι συνθήκες ικανοποιούνται.

Λόγω της λεπτομέρειας που παρατηρείται στην εικόνα d2b, δεν είναι εύκολο να αντιληφθεί κανείς αν η αναδρομική μέθοδος n_cuts οδηγεί σε επιθυμητά αποτελέσματα καθώς στην εικόνα ‘’Recursive n_cuts on d2b” εμφανίζονται πολλά διαφορετικά τμήματα. Παράδειγμα που αναδεικνύει την δυσκολία αυτή αξιολόγησης της αποτελεσματικότητας της μεθόδου αποτελεί η εμφάνιση πολλών διαφορετικών χρωμάτων (άρα και τμημάτων) στο περίγραμμα της εικόνας που προκύπτει από την μέθοδο, ενώ στην d2b το περίγραμμα αποτελείται κυρίως από κόκκινα pixels.

Για τον λόγο αυτό, ο κώδικας του demo3c εκτυπώνει και την εικόνα που προκύπτει από την n_cuts_recursive με τα ‘’πραγματικά’’ της χρώματα. Πιο συγκεκριμένα, για κάθε cluster που προκύπτει τελικά, το χρώμα του στην εικόνα ‘’ ’Recursive n_cuts on d2b (true colors)” ορίζεται ως ο μέσος όρος των RGB φωτεινοτήτων των pixels που ανήκουν στο cluster. Ως αποτέλεσμα, καθίσταται ευκολότερη η εξέταση της πιθανότητας η μέθοδος να ορίζει pixels διαφορετικών χρωμάτων και φωτεινοτήτων στο ίδιο cluster, γεγονός που θα οδηγούσε σε σημαντικά διαφορετική κατατμημένη εικόνα σε σχέση με την αρχική.

Παρατηρούμε ότι η τρίτη εικόνα παρουσιάζει μεγάλη ομοιότητα με την αρχική. Αυτό υποδεικνύει ότι η μέθοδος λειτουργεί σωστά. Μοναδικό μειονέκτημα της μεθόδου που γίνεται εύκολα αντιληπτό αποτελεί η αδυναμία της να διαχωρίσει τα καφέ pixels των παπουτσιών από τα υπόλοιπα κόκκινα pixels.