

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Ψηφιακή Επεξεργασία Εικόνας

Εργασία 3

Κουκουλέτσου Κατερίνα 10218

Ιούνιος 2025

Εισαγωγή

Η παρούσα εργασία ασχολείται με το πρόβλημα της κατάτμησης εικόνας (image segmentation), το οποίο αποτελεί ένα από τα βασικά βήματα σε πολλές εφαρμογές υπολογιστικής όρασης. Στο πλαίσιο της εργασίας υλοποιείται η κατασκευή του *affinity matrix*, ο οποίος περιγράφει τις σχέσεις ομοιότητας μεταξύ των pixels μιας εικόνας. Το matrix αυτό αξιοποιείται στη συνέχεια από δύο φασματικές μεθόδους: το Spectral Clustering και το Normalized Cuts, τόσο στη μη αναδρομική όσο και στην αναδρομική του εκδοχή.

Κοινό χαρακτηριστικό και των δύο μεθόδων αποτελεί η ανάλυση του γράφου που προκύπτει από το *affinity matrix* μέσω των ιδιοδιανυσμάτων και ιδιοτιμών του Laplacian. Με αυτόν τον τρόπο, οι μέθοδοι επιχειρούν να διαχωρίσουν την εικόνα σε περιοχές (clusters) με τρόπο που να μεγιστοποιείται η ομοιογένεια εντός κάθε περιοχής και να ελαχιστοποιείται η ομοιότητα μεταξύ διαφορετικών περιοχών.

Η αξιολόγηση των αποτελεσμάτων επικεντρώνεται σε τρεις βασικούς άξονες: πρώτον, στην καθαρότητα των παραγόμενων περιοχών, εξετάζοντας κατά πόσο οι περιοχές είναι συμπαγείς, συνεκτικές και αντιστοιχούν σε οπτικά «φυσικά» σύνολα pixels. Δεύτερον, στην επίδραση των παραμέτρων k , T_1 και T_2 , καταγράφοντας και αναλύοντας φαινόμενα υπερβολικής (over-segmentation) ή ανεπαρκούς (under-segmentation) κατάτμησης. Τέλος, στη συγκριτική μελέτη των μεθόδων, όπου αξιολογείται η ικανότητά τους να διατηρούν τις φυσικές δομές της εικόνας, καθώς και η ευαισθησία τους στις τιμές των παραμέτρων.

Κατασκευή του Affinity Matrix

Στο πρώτο στάδιο, η εικόνα αναπαρίσταται ως μη-κατευθυντικός γράφος $G = (V, E)$, όπου κάθε pixel αντιστοιχεί σε έναν κόμβο (vertex) του γράφου. Οι ακμές (edges) μεταξύ όλων των κόμβων περιγράφονται μέσω ενός πίνακα affinity matrix $A \in R^{MN \times MN}$, με $M \times N$ να είναι οι διαστάσεις της εικόνας.

Κατά την εκφώνηση, για κάθε ζεύγος κόμβων (pixels) i και j , το αντίστοιχο βάρος της ακμής $A(i, j)$ υπολογίζεται ως εξής:

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

όπου $d(i, j)$ είναι η Ευκλείδεια απόσταση μεταξύ των χαρακτηριστικών φωτεινότητας των pixels i και j . Με την επιλογή αυτού του εκθετικού, διασφαλίζεται ότι pixels με παρόμοιες τιμές luminance θα συνδέονται με ισχυρές ακμές (υψηλά weights), ενώ pixels με μεγάλες διαφορές μειώνουν απότομα το βάρος κατά τον υπολογισμό.

Σημειώνεται επίσης ότι ο γράφος είναι fully connected, δηλαδή για κάθε ζεύγος κόμβων i, j υπάρχει ακμή με μη-μηδενικό βάρος:

$$\forall i, j, \quad A(i, j) > 0.$$

και ότι είναι συμμετρικός, $A(i, j) = A(j, i)$, αφού είναι μη-κατευθυντικός και εξαρτάται αποκλειστικά από την Ευκλείδεια απόσταση.

Spectral Clustering

Η μέθοδος Spectral Clustering αξιοποιεί τη φασματική ανάλυση του γραφήματος που προκύπτει από την εικόνα, ώστε να κατατμήσει τα pixels σε k συνεκτικές ομάδες. Αρχικά, δίνεται η εικόνα ως affinity πίνακας $W \in R^{n \times n}$, όπου $n = M \cdot N$ είναι ο αριθμός των pixels, όπως αναφέρθηκε και προηγουμένως.

Λαπλασιανός πίνακας Για την κατασκευή του Λαπλασιανού πίνακα, κατασκευάζεται αρχικά ο διαγώνιος πίνακας D ως:

$$D(i, i) = \sum_{j=1}^n W(i, j).$$

Στη συνέχεια, υπολογίζεται ο Λαπλασιανός:

$$L = D - W.$$

Ιδιοτιμές και ιδιοδιανύσματα Η μέθοδος βασίζεται στην εύρεση των k μικρότερων ιδιοτιμών του L και των αντίστοιχων ιδιοδιανυσμάτων. Συγκεκριμένα, λύνει το πρόβλημα:

$$Lx = \lambda x.$$

Τα k ιδιοδιανύσματα που αντιστοιχούν στις μικρότερες ιδιοτιμές (εκτός της μηδενικής, εφόσον υπάρχει) σχηματίζουν έναν πίνακα $U \in R^{n \times k}$, στον οποίο κάθε γραμμή $y_i \in R^k$ αναπαριστά το i -οστό pixel σε ένα νέο χαμηλότερης διάστασης χώρο.

Ομαδοποίηση με k-means Οι νέες αναπαραστάσεις y_i ομαδοποιούνται με τον αλγόριθμο k-means. Ο αλγόριθμος επιδιώκει να βρει k clusters C_1, C_2, \dots, C_k που ελαχιστοποιούν την απόσταση:

$$\sum_{i=1}^k \sum_{y_j \in C_i} \|y_j - \mu_i\|^2,$$

όπου μ_i είναι το κέντρο του i -οστού cluster.

Ο στόχος της μεθόδου είναι να κατασκευάσει clusters που να αντιστοιχούν σε περιοχές της εικόνας με υψηλή ομοιογένεια, εκμεταλλευόμενη τη δομή του γράφου και τη φασματική πληροφορία του Λαπλασιανού πίνακα.

Normalized Cuts

Σε αντίθεση με το Spectral Clustering, στο Normalized Cuts επιλύεται το γενικευμένο πρόβλημα ιδιοτιμών:

$$Lx = \lambda Dx.$$

Τα k ιδιοδιανύσματα που αντιστοιχούν στις k μικρότερες ιδιοτιμές (εκτός της μηδενικής) σχηματίζουν τον πίνακα $U \in R^{n \times k}$, όπου κάθε γραμμή $y_i \in R^k$ αντιπροσωπεύει το i -οστό pixel σε νέο διανυσματικό χώρο.

Ομαδοποίηση Τα y_i ομαδοποιούνται σε k clusters C_1, \dots, C_k μέσω του αλγορίθμου k-means. Στη μη-αναδρομική εκδοχή, η διαδικασία τερματίζεται σε αυτό το σημείο.

Μετρική Ncut Για το σενάριο διχοτόμησης (αναδρομική εκδοχή), εισάγεται η μετρική Normalized Cut:

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

με

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

και

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

Η ποσότητα $assoc(A, A)$ αναφέρεται στα συνολικά βάρη εντός του cluster A , ενώ $assoc(A, V)$ μετρά όλα τα βάρη από το A προς όλο το γράφο V .

Αναδρομική εκδοχή Στην αναδρομική εκδοχή, αρχικά πραγματοποιείται διχοτόμηση ($k = 2$) με τον ίδιο τρόπο. Έπειτα, κάθε cluster ελέγχεται με βάση δύο κριτήρια:

- Το πλήθος των κόμβων στο cluster είναι μικρότερο από ένα κατώφλι T_1 .
- Η τιμή της μετρικής $Ncut(A, B)$ είναι μεγαλύτερη από ένα κατώφλι T_2 .

Αν ισχύει κάποιο από τα παραπάνω, η διχοτόμηση σταματά. Αλλιώς, η διαδικασία συνεχίζεται αναδρομικά εντός κάθε υποσυνόλου.

Διαγράμματα και Σχολιασμός

Αποτελέσματα του Demo 1

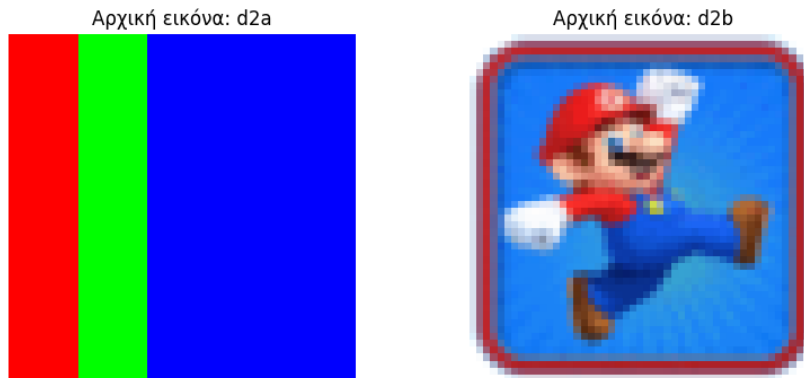
Παρουσιάζονται τα αποτελέσματα του Spectral Clustering για διάφορες τιμές του k , όπου ο affinity πίνακας έχει μέγεθος 12×12 .

Για $k = 2$, η μέθοδος διαχωρίζει καθαρά τα 12 σημεία σε δύο ομάδες, ξεχωρίζοντας τα πρώτα 4 σημεία που διαφοροποιούνται σημαντικά από τα υπόλοιπα 8. Με αύξηση του k σε 3, τα 8 αυτά σημεία διαχωρίζονται περαιτέρω σε δύο υποσύνολα των 4 σημείων, ενώ για $k = 4$ παρατηρείται επιπλέον διάσπαση, με μία από τις προηγούμενες ομάδες να χωρίζεται σε 3 και 1 σημεία αντίστοιχα. Η ύπαρξη ενός μοναδικού σημείου ως ξεχωριστού cluster μπορεί να υποδηλώνει θόρυβο ή αβεβαιότητα στα δεδομένα, καθώς επίσης και το ενδεχόμενο να ανήκει στην ομάδα στην οποία είχε αρχικά καταταχθεί. Ωστόσο, μπορεί επίσης να αποτελεί μία πραγματικά ξεχωριστή κατηγορία. Λόγω των περιορισμένων διαστάσεων του affinity πίνακα, η ταξινόμηση ενδέχεται να επηρεάζεται από τη δομή και την κατανομή των δεδομένων, καθιστώντας δύσκολη την καθοριστική ερμηνεία των αποτελεσμάτων.

k	Ετικέτες (labels)
2	[1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0]
3	[2, 2, 2, 2, 0, 0, 0, 0, 1, 1, 1, 1]
4	[2, 2, 2, 2, 3, 0, 0, 0, 1, 1, 1, 1]

Αποτελέσματα του Demo 2

Παρατίθενται οι αρχικές εικόνες και τα αποτελέσματα μετά την εφαρμογή του Spectral Clustering για $k = 2, 3$ και 4 .



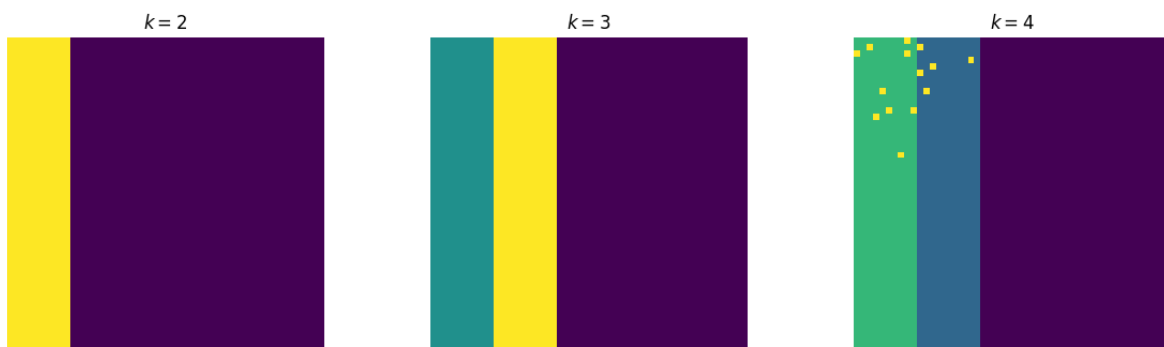
Στην εικόνα **d2a**, τα αποτελέσματα του αλγορίθμου spectral clustering είναι τα πλήρως αναμενόμενα. Η εικόνα αποτελείται από τρεις διακριτές οριζόντιες περιοχές (stripes) με διαφορετικά χρώματα: κόκκινο, πράσινο και μπλε. Συνεπώς, υπάρχουν ουσιαστικά τρεις φυσικές περιοχές που πρέπει να εντοπιστούν.

Θέτοντας $k = 2$, επιτρέπει κανείς στον αλγόριθμο να διακρίνει μόνο δύο clusters. Σε αυτή την περίπτωση, ο αλγόριθμος ομαδοποιεί τις δύο περιοχές που βρίσκονται πιο κοντά μεταξύ τους (πράσινη και μπλε) σε ένα κοινό cluster και ξεχωρίζει την κόκκινη περιοχή ως το δεύτερο cluster. Αυτό οφείλεται μάλλον στο γεγονός ότι το luminance των πράσινης και μπλε περιοχής είναι πιο κοντινό σε σύγκριση με το κόκκινο.

Με αύξηση του πλήθους των clusters σε $k = 3$, ο αλγόριθμος εντοπίζει με απόλυτη ακρίβεια τις τρεις διαφορετικές περιοχές, αποδίδοντας κάθε stripe σε ξεχωριστό cluster.

Τέλος, στην περίπτωση του $k = 4$, ενώ στην εικόνα υπάρχουν μόνο τρεις ουσιαστικές περιοχές, ο αλγόριθμος αναγκάζεται να δημιουργήσει έναν τέταρτο cluster. Σε αυτή την περίπτωση, αναγνωρίζει σωστά τις τρεις πραγματικές περιοχές και το τέταρτο cluster αποδίδεται σε μικρές περιοχές θορύβου (απομονωμένα pixels εντός των stripes), χωρίς πραγματική σημασιολογική διαφοροποίηση.

Αποτελέσματα spectral clustering για **d2a**

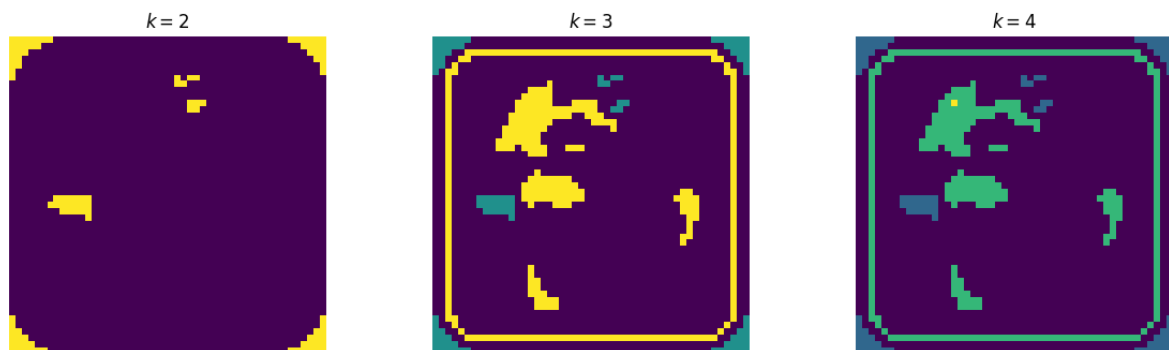


Στη δεύτερη εικόνα, ο αλγόριθμος spectral clustering ίσως οπτικά να μη φαίνεται να δίνει ικανοποιητικά αποτελέσματα, καθώς η φιγούρα «χάνεται» σε μεγάλο βαθμό στο φόντο. Παρατηρώντας όμως πιο προσεκτικά, διαπιστώνεται ότι το κομμάτι της φιγούρας που συγχωνεύεται με το φόντο έχει σχεδόν την ίδια απόχρωση του μπλε, γεγονός που καθιστά λογική τη συμπεριφορά του αλγορίθμου, ειδικά όταν το πλήθος των επιτρεπόμενων clusters είναι περιορισμένο (μέχρι 4).

Παρ' όλα αυτά, ο αλγόριθμος καταφέρνει να ξεχωρίσει τα παπούτσια της φιγούρας με καφέ απόχρωση, τα μανίκια και τα γάντια, το καπέλο και το λογότυπο στο καπέλο με κόκκινο και λευκό χρώμα. Αξίζει επίσης να σημειωθεί ότι η πρώτη κλάση που διαχωρίζεται είναι τα λευκά αντικείμενα, κάτι αναμενόμενο, καθώς στο luminance το λευκό έχει τη μέγιστη τιμή και επομένως η απόσταση στο affinity matrix μεγιστοποιείται. Σε μεγαλύτερες τιμές του k , αρχίζουν να διακρίνονται και οι κόκκινες και καφέ περιοχές.

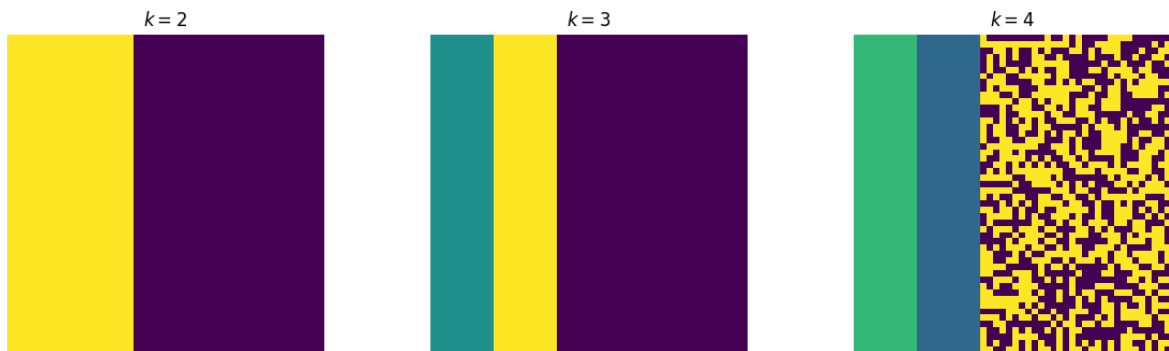
Συνολικά πάντως, το clustering δεν μπορεί να χαρακτηριστεί ιδιαίτερα ικανοποιητικό για αυτή την εικόνα.

Αποτελέσματα spectral clustering για **d2b**

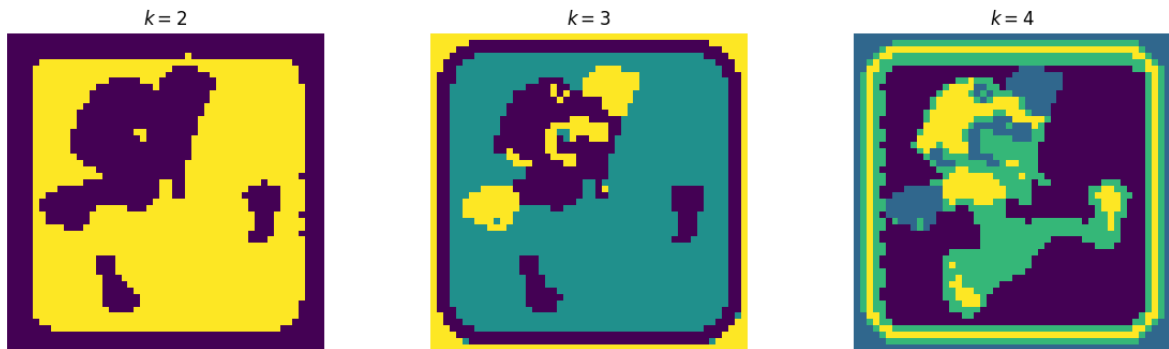


Αποτελέσματα του Demo 3a

Αποτελέσματα μη-αναδρομικής n -cuts για **d2a**



Αποτελέσματα μη-αναδρομικής n -cuts για **d2b**



Στην εικόνα με τα τρία ομοιογενή *stripes*, τα οποία διαφέρουν ξεκάθαρα ως προς τη φωτεινότητα (luminance), τόσο ο αλγόριθμος Spectral Clustering όσο και ο Normalized Cuts επιτυγχάνουν πολύ καλά αποτελέσματα όταν ο αριθμός των clusters οριστεί σωστά στο $k = 3$. Σε αυτήν την περίπτωση, κάθε ένα από τα τρία clusters αντιστοιχεί σε ένα καθαρά διαχωρισμένο *stripe*.

Ωστόσο, όταν επιλέγεται $k = 4$, δηλαδή ζητείται από τον αλγόριθμο να δημιουργήσει τέσσερα clusters, ενώ στην πραγματικότητα υπάρχουν μόνο τρεις φυσικές περιοχές, παρατηρείται ότι ο Normalized Cuts παράγει σημαντικό θόρυβο. Συγκεκριμένα, ένα από τα *stripes* εμφανίζεται διασπασμένο με διασκορπισμένα σημεία τα οποία ανήκουν σε ξεχωριστό cluster, παρά την ομοιογένεια της περιοχής αυτής και την έλλειψη ορατών διαφορών φωτεινότητας μεταξύ των pixels στην αρχική εικόνα.

Αυτή η συμπεριφορά του αλγορίθμου μπορεί ίσως να αποδοθεί στους παρακάτω λόγους:

- **Oversegmentation ($k = 4$ ενώ υπάρχουν μόνο 3 φυσικά groups):** Ο αλγόριθμος πρέπει να βρει τέσσερα clusters, οπότε αναγκαστικά «σπάει» κάποιο ήδη ομοιογενές region σε υποπεριοχές, γιατί δεν υπάρχει άλλος τρόπος να χωρίσει τα δεδομένα.
- **Ευαισθησία στις μικρές διαφορές φωτεινότητας:** Παρά την οπτική ομοιομορφία, ίως υπάρχουν πάλι μικρές διαφορές φωτεινότητας στην πράξη, λόγω παραγόντων όπως χβαντισμός της φωτεινότητας (quantization) ή σφάλματα στρογγυλοποίησης και floating point errors.

- **Χρήση πολλών ιδιοδιανυσμάτων για μεγαλύτερο k :** Όταν αυξάνετε ο αριθμός των clusters, ο αλγόριθμος λαμβάνει υπόψη περισσότερα ιδιοδιανύσματα του Laplacian. Έτσι αναγκαστικά επιλέγονται και ιδιοδιανύσματα που αποτυπώνουν μικρότερα τοπικά μοτίβα και που συχνά περιγράφουν θόρυβο ή μικρο-διαφορές. Η προσθήκη αυτών των «λεπτών» ιδιοδιανυσμάτων οδηγεί σε αυτό το fragmentation.

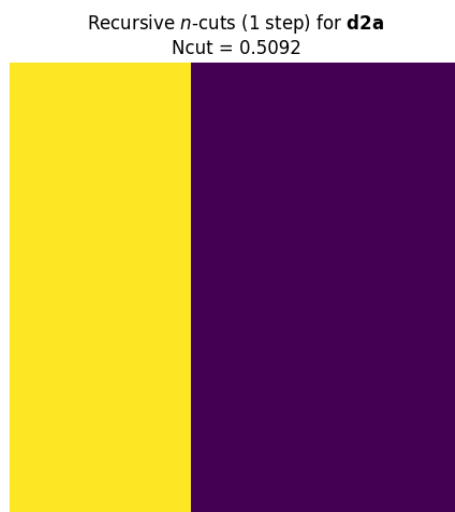
Επομένως, το συμπέρασμα είναι ότι μάλλον η παρουσία θορύβου και η διασπορά σε υπο-περιοχές όταν $k >$ πραγματικός αριθμός φυσικών περιοχών δεν αποτελεί αποτυχία του αλγορίθμου, αλλά αναγκαστική συμπεριφορά του αλγορίθμου λόγω της υποχρεωτικής υπερ-διαμέρισης και της ευαισθησίας του affinity matrix σε μικρο-διαφορές φωτεινότητας.

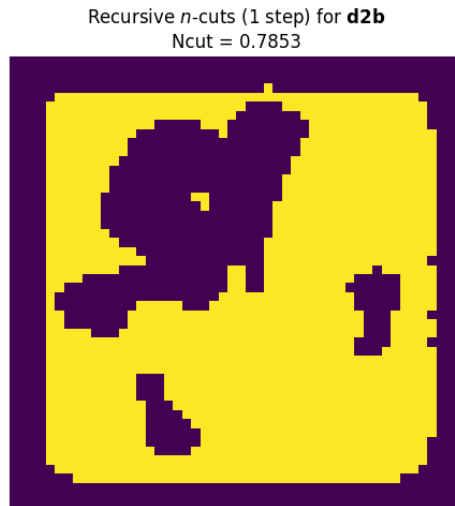
Παρόλ'αυτά, στην πιο περίπλοκη εικόνα d2b, φαίνεται ότι το Ncuts λειτουργεί πολύ καλύτερα σε σχέση με το Spectral Clustering αφού καταφέρνει να διαχωρίσει πολύ καλύτερα την φιγούρα από το background ενώ παράλληλα αρχίζουν να φαίνονται πολλές περισσότερες λεπτομέρειες. Φαίνεται πως ο αλγόριθμος είναι ικανός να πιάνει και τις πιο λεπτές διαφορές και να είναι πιο ακριβής. Αυτό αποτελεί πλεονέκτημα σε εικόνες με περισσότερες και πιο "λεπτές" λεπτομέρειες που απαιτούν προσεκτική ανάλυση, όμως μπορεί να έχει αρνητικές επιπτώσεις σε πολύ απλές εικόνες, όπως φάνηκε και στο προηγούμενο παράδειγμα.

Αποτελέσματα του Demo 3b

Στο συγκεκριμένο demo εφαρμόζεται η αναδρομική εκδοχή του αλγορίθμου Normalized Cuts (Ncuts), υλοποιώντας ακριβώς ένα βήμα διαχωρισμού του αρχικού γράφου για κάθε μία από τις δύο εικόνες εισόδου. Δηλαδή ο γράφος «σπάει» σε δύο υποσύνολα (clusters), πραγματοποιώντας έτσι μία μόνο επανάληψη.

Για να επιτευχθεί αυτό, πραγματοποιήθηκε ρύθμιση (tuning) των παραμέτρων $T1$ και $T2$ καθώς αυτές είναι που ελέγχουν πότε θα σταματήσει η αναδρομική διαδικασία. Πιο συγκεκριμένα, το $T1$ ορίζει το κατώτατο όριο για το μέγεθος των clusters που μπορούν να υποδιαιρεθούν. Αν ένα cluster έχει μέγεθος μικρότερο από $T1$, τότε δεν χωρίζεται περαιτέρω. Το $T2$ θέτει όριο στην τιμή του δείκτη Neut, κάτω από την οποία ο διαχωρισμός θεωρείται επαρκώς «καλός» και η περαιτέρω υποδιαίρεση δεν κρίνεται απαραίτητη. Μέσω μιας διαδικασίας trial and error, τελικά επιλέχθηκε $T1 = 500$ και $T2 = 1$ έτσι ώστε η διαδικασία να εκτελεστεί ακριβώς μία φορά.





Συγκρίνοντας με το Spectral Clustering για $k=2$, στην εικόνα με τις ρίγες, το Ncuts δεν ομαδοποιεί το μπλε με το πράσινο σε ένα cluster όπως έκανε το Spectral Clustering, αλλά αντίθετα ομαδοποιεί το πράσινο με το κόκκινο και το μπλε ξεχωρίζει μόνο του. Αυτό υποδηλώνει μάλλον ότι το Ncuts αξιολογεί διαφορετικά την ομοιότητα των περιοχών βασισμένο στον γράφο και τα κομμάτια στα οποία «σπάει» τον γράφο, πιθανώς λόγω της διαφορετικής διαχείρισης των συνόρων των clusters.

Στην εικόνα με τη φιγούρα, το Ncuts δίνει πιο λεπτομερή διαχωρισμό για $k=2$, διαχωρίζοντας καθαρότερα τη φιγούρα από το φόντο σε σχέση με το Spectral Clustering, γεγονός που δείχνει καλύτερη προσαρμογή στις δομές της εικόνας. Παρόλ'αυτά ούτε το NCuts καταφέρνει να ξεχωρίσει το μπλε παντελόνι της φιγούρας από το μπλε φόντο.

Οι τιμές του δείκτη Ncut που προέκυψαν είναι 0.5092 και 0.7853 για τις δύο εικόνες αντίστοιχα, που δείχνουν τη σχετική ποιότητα του διαχωρισμού.

Αποτελέσματα του Demo 3c



Στο πρώτο subplot κάθε γραμμής παρουσιάζονται τα αποτελέσματα της αναδρομικής (recursive) μεθόδου n-cuts. Στο δεύτερο και τρίτο subplot εμφανίζονται τα αποτελέσματα της μη αναδρομικής (non-recursive) n-cuts για $k = 2$ και $k = 3$ αντίστοιχα, ενώ το τέταρτο subplot παρουσιάζει τα αποτελέσματα του spectral clustering για $k = 2$, όπως ζητά η εκφώνηση.

Γενικά, παρατηρείται ότι για τις συγκεκριμένες τιμές παραμέτρων $T_1 = 5$ και $T_2 = 0.2$, η αναδρομική μέθοδος δεν λειτουργεί σωστά και δεν πραγματοποιείται κανένα «σπάσιμο» ούτε στην πρώτη ούτε στη δεύτερη εικόνα. Αυτό πιθανότατα οφείλεται στο γεγονός ότι οι τιμές των T_1 και T_2 είναι υπερβολικά αυστηρές. Συγκεκριμένα:

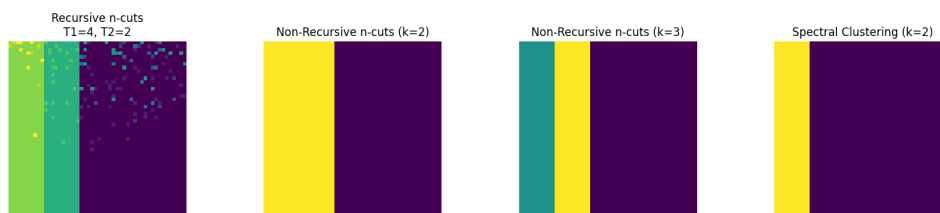
Η παράμετρος T_1 ορίζει το κατώτατο όριο μεγέθους για τα clusters, κάτω από το οποίο η διαδικασία διακοπής της αναδρομής ενεργοποιείται. Επομένως, αν η τιμή του T_1 είναι πολύ υψηλή, πολλά clusters θα θεωρούνται μικρά και η αναδρομική διάσπαση θα σταματά πρόωρα.

Αντίστοιχα, η παράμετρος T_2 αποτελεί το όριο αποδοχής για την τιμή του δείκτη N_{cut} . Μόνο όταν η τιμή N_{cut} ενός cluster είναι μικρότερη ή ίση του T_2 , η αναδρομή συνεχίζεται. Συνεπώς, αν το T_2 είναι ιδιαίτερα χαμηλό, είναι εξαιρετικά δύσκολο για έναν κόμβο να ικανοποιήσει αυτό το κριτήριο, με αποτέλεσμα η διάσπαση να διακόπτεται.

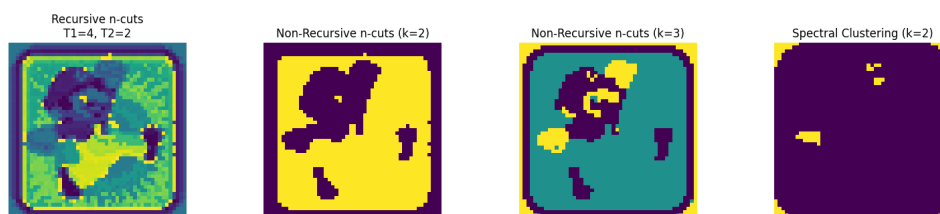
Συνδυάζοντας τα παραπάνω, προκύπτει ότι όταν η παράμετρος T_1 είναι αυξημένη και η T_2 πολύ μικρή, οι προϋποθέσεις για τη συνέχιση της αναδρομής δεν συναντώνται ταυτόχρονα. Αυτό οδηγεί σε έλλειψη κοινού «εδάφους» μεταξύ των δύο κριτηρίων, με συνέπεια να μην πραγματοποιείται καμία περαιτέρω διάσπαση του γράφου.

Παρατίθεται μια ακόμη εκτέλεση για διαφορετικές τιμές των παραμέτρων T_1 και T_2 , προκειμένου να φανεί λίγο καλύτερα η λειτουργία της αναδρομικής μεθόδου:

Σύγκριση clustering μεθόδων για **d2a**



Σύγκριση clustering μεθόδων για **d2b**



Με τιμές $T_1 = 4$ και $T_2 = 2$, η αναδρομική μέθοδος φαίνεται να εκτελεί αρκετές επαναλήψεις, παρέχοντας ένα segmentation στην πιο πολύπλοκη εικόνα που είναι σημαντικά πιο λεπτομερές σε σχέση με όλες τις προηγούμενες προσπάθειες στην εργασία. Η μέθοδος ανιχνεύει πολλές λεπτομέρειες, με αποτέλεσμα η εικόνα να φαίνεται σχεδόν ξεκάθαρα.

Παρόλα αυτά, στην εικόνα με τις ρίγες, παρατηρείται και πάλι η εμφάνιση θορύβου ως ξεχωριστό cluster, πιθανότατα για τους ίδιους λόγους που αναφέρθηκαν πιο πάνω. Συνεπώς, επιβεβαιώνεται πως σε περιπτώσεις απλών εικόνων, είναι προτιμότερο να χρησιμοποιείται ένας πιο απλός αλγόριθμος, καθώς η χρήση πιο σύνθετων μεθόδων μπορεί να προκαλέσει υπερβολική κατάτμηση (over-segmentation) και να αναδείξει θόρυβο ως ξεχωριστές περιοχές. Δηλαδή η αίσθηση ότι ένας περίπλοκος και σύνθετος αλγόριθμος θα λειτουργεί άριστα παντού και θα υπερέχει πάντα του απλού αλγορίθμου είναι λανθασμένη.

Σημείωση

Να σημειωθεί βέβαια και το εξής για την εικόνα της φιγούρας με $T_1=4$ και $T_2=2$. Επειδή είναι όλα πιο ευδιάκριτα δεν σημαίνει ότι αυτό είναι και το επιθυμητό αποτέλεσμα. Δηλαδή, κατά την υλοποίηση της εργασίας και την μελέτη των αποτελεσμάτων, προέκυψε ένας προβληματισμός σχετικά με το τι θεωρείται «ιδανικό» segmentation. Αν δηλαδή το ζητούμενο είναι να εντοπιστεί ολόκληρο το αντικείμενο ως μία ενιαία περιοχή ή αν πρέπει να διαχωριστούν περαιτέρω υποπεριοχές (π.χ. χαρακτηριστικά προσώπου, σώμα, ρούχα και άλλες λεπτομέρειες του background).

Έπειτα από κάποιο ψάξιμο, έγινε η υπόθεση ότι το «σωστό» αποτέλεσμα εξαρτάται σε μεγάλο βαθμό από το downstream task για το οποίο θα χρησιμοποιηθεί το segmentation. Για παράδειγμα, σε μια εφαρμογή αναγνώρισης αντικειμένων ίσως αρκεί ένα γενικό διαχωρισμένο περίγραμμα, ενώ σε εφαρμογές ανάλυσης σκηνής ή επεξεργασίας ιατρικών εικόνων μπορεί να είναι απαραίτητη μια πολύ πιο λεπτομερής κατάτμηση.

Καθώς όμως η παρούσα εργασία εστιάζει συγκεκριμένα στην υλοποίηση και ανάλυση των μεθόδων, αποφασίστηκε να παρουσιαστούν τα αποτελέσματα τόσο και για τα T_1 και T_2 που υποδείκνυε η εκφώνηση της εργασίας ούτως ή άλλως αλλά και για άλλες τιμές των T_1 και T_2 . Δηλαδή για πιο

λεπτομερείς εκδοχές της κατάτμησης. Με αυτόν τον τρόπο αναδεικνύονται οι δυνατότητες και η προσαρμοστικότητα των αλγορίθμων, και δίνεται η δυνατότητα να δει κανείς πώς μπορούν να τροποποιηθούν τα αποτελέσματα ανάλογα με τις ανάγκες κάθε εφαρμογής. Φαίνεται δηλαδή πως η Recursive δουλεύει επιτυχώς ακόμη και αν το αποτέλεσμα από άποψη αξιολόγησης της κατάτμησης δεν είναι το επιθυμητό γιατί έχει τόσο πολλή πληροφορία και τέτοια λεπτομέρεια.