

Ψηφιακή Επεξεργασία Εικόνας

Εργασία 3: Image segmentation

[Άνοιξη 2025]

1. Εισαγωγή

Η παρούσα εργασία επικεντρώνεται στην υλοποίηση και αξιολόγηση σύγχρονων μεθόδων **τμηματοποίησης εικόνων (image segmentation)** με βάση τη θεωρία γράφων, χρησιμοποιώντας την γλώσσα προγραμματισμού Python. Κύριος στόχος είναι η εμβάθυνση σε αλγορίθμους ομαδοποίησης που εκμεταλλεύονται τις φασματικές ιδιότητες γράφων, όπως το **Spectral Clustering** και οι μέθοδοι **Normalized Cuts (n-cuts)**, για τον διαχωρισμό εικόνων σε ουσιαστικές περιοχές.

Πιο συγκεκριμένα, υλοποιήθηκαν και μελετήθηκαν:

- **Αναπαράσταση εικόνας σαν γράφο** μέσω ενός πλήρως συνδεδεμένου *affinity* πίνακα, όπου κάθε pixel αντιστοιχεί σε κόμβο και τα βάρη των ακμών ορίζονται από την ομοιότητα χρωμάτων.
- **Ο αλγόριθμος Spectral Clustering**, ο οποίος μετασχηματίζει το πρόβλημα τμηματοποίησης σε πρόβλημα ιδιοτιμών, χρησιμοποιώντας τον κανονικοποιημένο Laplacian πίνακα.
- **Η μέθοδος Normalized Cuts**, τόσο στην **αναδρομική** όσο και στη **μη-αναδρομική** της εκδοχή, η οποία ελαχιστοποιεί μια αντικειμενική συνάρτηση που σχετίζεται με την αξία της τομής μεταξύ διαφορετικών ομάδων κόμβων.

Η εργασία περιλαμβάνει:

1. Την **εφαρμογή των μεθόδων σε προ-υπολογισμένους πίνακες** (d1a) και **πραγματικές εικόνες** (d2a, d2b),
2. Την **οπτική και ποσοτική αξιολόγηση των αποτελεσμάτων για διαφορετικές παραμετροποιήσεις** (π.χ., αριθμός clusters k, κατώφλι n-cut).

Όλες οι συναρτήσεις έχουν σχεδιαστεί για **γενική χρήση**, υποστηρίζοντας οποιαδήποτε εικόνα ή affinity πίνακα που πληροί τις προϋποθέσεις της εκφώνησης. Επιπλέον, η επαναληψιμότητα εξασφαλίζεται μέσω της ρύθμισης random_state=1 στον αλγόριθμο k-means, ενώ η υπολογιστική αποδοτικότητα επιτυγχάνεται με τη χρήση αραιών πινάκων (sparse matrices) και εξειδικευμένων βιβλιοθηκών όπως η `scipy.sparse.linalg`.

2. Περιγραφή λειτουργίας των συναρτήσεων

Στην παρούσα ενότητα περιγράφεται η θεωρητική λειτουργία των βασικών συναρτήσεων που υλοποιήθηκαν στο πλαίσιο της εργασίας. Κάθε συνάρτηση σχεδιάστηκε ώστε να εξυπηρετεί ένα επιμέρους βήμα της διαδικασίας τμηματοποίησης εικόνων μέσω θεωρίας γράφων, όπως περιγράφεται αναλυτικά στην εκφώνηση και στη σχετική θεωρία.

Υλοποιήθηκαν οι ακόλουθες συναρτήσεις που ζητήθηκαν ρητά στην εκφώνηση:

- `image_to_graph()`
- `spectral_clustering()`
- `n_cuts()` και `n_cuts_recursive()`
- `calculate_n_cut_value()`

Ακολουθεί αναλυτική περιγραφή της θεωρητικής λειτουργίας κάθε συνάρτησης, οργανωμένη ανά αρχείο Python:

2.1 Αρχείο *image_to_graph.py*

- `def image_to_graph(img_array: np.ndarray) -> np.ndarray:`

Η συνάρτηση *image_to_graph()* μετατρέπει μια εικόνα RGB σε έναν **fully-connected γράφο**, όπου κάθε pixel αντιστοιχεί σε κόμβο και οι ακμές έχουν βάρη που εκφράζουν την ομοιότητα μεταξύ pixels.

Βασικές αρχές λειτουργίας:

- **Επεξεργασία εισόδου:** Η εικόνα εισόδου διαστάσεων [M,N,C] μετασχηματίζεται σε πίνακα pixels διαστάσεων [MN,C].
- **Υπολογισμός ομοιότητας:** Για κάθε ζεύγος pixels (i,j) , υπολογίζεται η Ευκλείδεια απόσταση των χρωματικών τιμών τους ως:

$$d(i,j) = \|p_i - p_j\|_2$$

όπου p_i είναι το χρωματικό διάνυσμα του pixel i

- **Affinity matrix:** Ο πίνακας ομοιότητας κατασκευάζεται με βάση την εκθετική συνάρτηση:

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

- Η συνάρτηση αυτή εξασφαλίζει τιμές στο $[0,1]$, με 1 να υποδηλώνει ταυτόσημα pixels με τον επιστρεφόμενο πίνακα *affinity_mat* να έχει διαστάσεις $(MN \times MN)$.

Σχεδιαστικές επιλογές:

- Χρήση της `scipy.spatial.distance.cdist` για βέλτιστο υπολογισμό των αποστάσεων.
- Πλήρης σύνδεση του γράφου (fully-connected), σύμφωνα με την εκφώνηση.

2.2 Αρχείο *spectral_clustering.py*

- `def spectral_clustering(affinity_mat: np.ndarray, k: int) -> np.ndarray:`

Η συνάρτηση *spectral_clustering()* υλοποιεί τον αλγόριθμο **Spectral Clustering** για την τμηματοποίηση του γράφου που ορίζεται από τον πίνακα ομοιότητας.

Βήματα αλγορίθμου:

1. **Κατασκευή Laplacian πίνακα:**

$$L = D - W$$

όπου D ο διαγώνιος πίνακας με $D_{ii} = \sum_j W_{ij}$

2. **Υπολογισμός ιδιοδιανυσμάτων:** Επίλυση του προβλήματος ιδιοτιμών $Lx = \lambda x$ για τις k μικρότερες ιδιοτιμές (εξαιρουμένης της μηδενικής).
3. **Ομαδοποίηση:** Εφαρμογή k-means στις γραμμές του πίνακα ιδιοδιανυσμάτων για την τελική ανάθεση labels.

Σχεδιαστικές επιλογές:

- Χρήση *sparse* πινάκων (*scipy.sparse*) για αποδοτικότητα και εξοικονόμηση μνήμης. Ο πίνακας W μετατρέπεται σε *sparse* με χρήση της *csr_matrix()* η οποία αποθηκεύει μόνο τα μη-μηδενικά στοιχεία με δείκτες γραμμών/στηλών. Στην περίπτωση μας ο πίνακας είναι *dense*, αλλά οι *sparse* βιβλιοθήκες είναι βελτιστοποιημένες για εξειδικευμένους υπολογισμούς, όπως *scipy.sparse.linalg.eigs*.
 - Επιλογή *which = 'SM'* στη *eigs()* για τις μικρότερες ιδιοτιμές.
 - *random_state = 1* στον k-means για επαναληψιμότητα.
-

2.3 Αρχείο *n_cuts.py*

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

Η μη-αναδρομική έκδοση των **Normalized Cuts** βασίζεται στην επίλυση του γενικευμένου προβλήματος ιδιοτιμών:

Βήματα αλγορίθμου:

1. **Κατασκευή Laplacian:**

$$L = D - W$$

όπου D ο διαγώνιος πίνακας με $D_{ii} = \sum_j W_{ij}$

2. **Γενικευμένο πρόβλημα ιδιοτιμών:**

$$Lx = \lambda Dx$$

Επιλύεται για τις k μικρότερες ιδιοτιμές.

3. **Ομαδοποίηση:** k-means στα ιδιοδιανύσματα, παρόμοια με το Spectral Clustering.

Διαφορά από Spectral Clustering:

Χρησιμοποιεί τον γενικευμένο Laplacian αντί του κανονικού, βελτιώνοντας την απόδοση σε unbalanced clusters.

- `def n_cuts_recursive(affinity_mat: np.ndarray, T1: int, T2: float) -> np.ndarray:`

Η αναδρομική έκδοση εφαρμόζει διχοτόμηση του γράφου σε υποσύνολα μέχρι να ικανοποιηθούν τα κριτήρια τερματισμού, χρησιμοποιώντας την αναδρομική συνάρτηση `def recursive_split(nodes)`, που βρίσκεται εσωτερικά της `def n_cuts_recursive()` και εφαρμόζει τον παρακάτω αλγόριθμο:

- **Επανάληψη διχοτομήσεων:**
 - Κάθε υπο-σύνολο κόμβων (cluster) διαχωρίζεται σε 2 μικρότερα σύνολα με τη μη-αναδρομική συνάρτηση `n_cuts()` ($k = 2$).
 - Υπολογισμός `n_cut_value` με τη συνάρτηση `calculate_n_cut_value()`.
- **Κριτήρια τερματισμού:**
 - **T1:** Ελάχιστο πλήθος κόμβων σε cluster. Αν ένα υπο-σύνολο (είτε A είτε B) έχει λιγότερους κόμβους από T1, δεν διαχωρίζεται περαιτέρω (default: 5).
 - **T2:** Μέγιστη αποδεκτή τιμή `Ncut`. Αν η τιμή `Ncut` για μια διχοτόμηση υπερβαίνει το T2, ο διαχωρισμός θεωρείται αναποτελεσματικός και σταματάει. (default: 0.2).
- **Αναδρομή:** Εφαρμογή του αλγορίθμου της `recursive_split()` σε κάθε υπο-σύνολο που προκύπτει.

Σχεδιαστικές επιλογές:

- Δυναμική Ανάθεση Ετικετών: Χρήση λίστας `current_label` για να διατηρείται η τρέχουσα ετικέτα σε όλα τα αναδρομικά επίπεδα.
- Αποδοτικότητα: Ο υπο-πίνακας `sub_affinity` δημιουργείται μόνο για τους ενεργούς κόμβους, μειώνοντας τον απαιτούμενο χώρο.

- `def calculate_n_cut_value(affinity_mat: np.ndarray, cluster_idx: np.ndarray) -> float:`

Υπολογίζει την τιμή της μετρικής `Ncut` για μια διχοτόμηση του γράφου:

- Δημιουργία mask: Διαχωρισμός κόμβων σε δύο υποσύνολα (A, B) βάσει των `labels`.
- Υπολογισμός `assoc`, που είναι το άθροισμα βαρών όλων των ακμών εντός του υποσυνόλου ή των ακμών από το υποσύνολο προς όλους τους κόμβους:
 - $assoc(A, A) = \sum_{i,j \in A} W_{ij}$ (αντίστοιχα $assoc(B, B)$)
 - $assoc(A, V) = \sum_{i \in A, j \in V} W_{ij}$ (αντίστοιχα $assoc(B, V)$)
- Η τιμή `Ncut` για διαμέριση (A, B) ορίζεται ως:

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

2.4 Αρχείο *demo1.py*

Το αρχείο *demo1.py* πραγματοποιεί την αξιολόγηση της συνάρτησης *spectral_clustering()* στον προϋπολογισμένο πίνακα ομοιότητας *d1a* από το αρχείο *dip_hw_3.mat*. Ο κώδικας εφαρμόζει τον αλγόριθμο spectral clustering για τρεις διαφορετικές τιμές του αριθμού των clusters ($k = 2, 3, 4$), με στόχο τη μελέτη της επίδρασης αυτής της παραμέτρου στα αποτελέσματα της ομαδοποίησης. Για κάθε τιμή του k , τα *labels* που προκύπτουν από τον αλγόριθμο απεικονίζονται σε scatter plot, όπου η θέση κάθε σημείου στον οριζόντιο άξονα αντιστοιχεί στον δείκτη του κόμβου και στον κάθετο άξονα στην τιμή του cluster label. Η χρωματική κωδικοποίηση των σημείων βοηθά στην οπτική διάκριση των διαφορετικών ομάδων. Τα γραφικά αποθηκεύονται σε μορφή PNG (*demo1_k2.png*, *demo1_k3.png*, *demo1_k4.png*) και εμφανίζονται σε πραγματικό χρόνο, επιτρέποντας τη σύγκριση της δομής των clusters για διαφορετικές τιμές του k . Η χρήση του *random_state = 1* στον αλγόριθμο k-means εξασφαλίζει την επαναληψιμότητα των πειραμάτων.

2.5 Αρχείο *demo2.py*

Το αρχείο *demo2.py* διερευνά την εφαρμογή του **Spectral clustering** σε πραγματικές εικόνες, συγκεκριμένα στις εικόνες *d2a* και *d2b* από το αρχείο δεδομένων. Αρχικά, κάθε εικόνα μετατρέπεται σε πίνακα ομοιότητας μέσω της συνάρτησης *image_to_graph()*, όπου η ομοιότητα μεταξύ pixels υπολογίζεται με βάση την Ευκλείδεια απόσταση των χρωματικών τους τιμών. Στη συνέχεια, ο αλγόριθμος spectral clustering εφαρμόζεται για $k=2, 3$ και 4 clusters, με τα αποτελέσματα να ανασχηματίζονται στις αρχικές διαστάσεις της εικόνας για οπτικοποίηση. Κάθε τμηματοποιημένη εικόνα παρουσιάζεται δίπλα στην αρχική, επιτρέποντας την άμεση σύγκριση των αποτελεσμάτων. Η διαδικασία επαναλαμβάνεται και για τις δύο εικόνες εισόδου, με τις τμηματοποιημένες εκδόσεις να αποθηκεύονται σε ξεχωριστά αρχεία (*demo2_imgA_k2.png*, κ.λπ.). Το script επιδεικνύει πώς η επιλογή του k επηρεάζει την λεπτομέρεια της τμηματοποίησης, με μεγαλύτερες τιμές του k να οδηγούν σε πιο λεπτομερή αλλά και πιο θορυβώδη αποτελέσματα.

2.6 Αρχείο *demo3a.py*

Το *demo3a.py* εφαρμόζει τη **μη-αναδρομική** εκδοχή των **Normalized cuts** στις εικόνες *d2a* και *d2b*, παρόμοια με τη ροή του *demo2.py* αλλά αντικαθιστώντας τον αλγόριθμο spectral clustering με τη συνάρτηση *n_cuts()*. Για κάθε εικόνα, κατασκευάζεται ο αντίστοιχος πίνακας ομοιότητας και στη συνέχεια εφαρμόζεται η μέθοδος *n_cuts()* για $k = 2, 3$ και 4 clusters. Τα αποτελέσματα της τμηματοποίησης εμφανίζονται παράλληλα με τις αρχικές εικόνες,

επιτρέποντας την απευθείας σύγκριση μεταξύ των δύο μεθόδων. Το script υπογραμμίζει τις διαφορές στην ποιότητα και τη δομή των ορίων μεταξύ των clusters που παράγουν οι δύο αλγόριθμοι, με τα normalized cuts να τείνουν να δημιουργούν πιο ομαλά και συνεκτικά όρια λόγω της ελαχιστοποίησης της μετρικής $Ncut$. Όπως και στα προηγούμενα demos, όλα τα αποτελέσματα αποθηκεύονται σε αρχεία εικόνας για μελλοντική αναφορά.

2.7 Αρχείο *demo3b.py*

Το αρχείο *demo3b.py* επικεντρώνεται στη μελέτη της **αναδρομικής** μεθόδου **Normalized cuts** για ένα μόνο βήμα διχοτόμησης. Χρησιμοποιώντας ως είσοδο τις εικόνες *d2a* και *d2b*, το script κατασκευάζει αρχικά τους πίνακες ομοιότητας και εφαρμόζει μία μόνο διχοτόμηση με τη συνάρτηση *n_cuts(k=2)*. Στη συνέχεια, υπολογίζει την τιμή της μετρικής $Ncut$ για τη διχοτόμηση αυτή μέσω της συνάρτησης *calculate_n_cut_value()*, παρέχοντας ένα ποσοτικό μέτρο της ποιότητας του διαχωρισμού. Τα αποτελέσματα της διχοτόμησης απεικονίζονται πάνω στις αρχικές εικόνες, ενώ η τιμή του $Ncut$ εκτυπώνεται στην κονσόλα. Αυτή η προσέγγιση επιτρέπει την αξιολόγηση της βασικής ικανότητας του αλγορίθμου να διαχωρίσει τις εικόνες σε δύο βασικές περιοχές, χωρίς την πολυπλοκότητα της πλήρους αναδρομικής εφαρμογής. Τα γραφικά αποτελέσματα αποθηκεύονται σε αρχεία (*demo3b_imgA.png*, *demo3b_imgB.png*) για περαιτέρω ανάλυση.

2.8 Αρχείο *demo3c.py*

Το *demo3c.py* υλοποιεί την πλήρη αναδρομική εκδοχή του αλγορίθμου normalized cuts, εφαρμόζοντας διαδοχικές διχοτομήσεις στις εικόνες *d2a* και *d2b* έως ότου ικανοποιηθούν τα κριτήρια τερματισμού $T1=5$ (ελάχιστο μέγεθος cluster) και $T2=0.2$ (μέγιστη τιμή $Ncut$). Το script ξεκινά με τη μετατροπή κάθε εικόνας σε πίνακα ομοιότητας και στη συνέχεια καλεί τη συνάρτηση *n_cuts_recursive()* για την ιεραρχική τμηματοποίηση του γράφου. Ο αλγόριθμος συνεχίζει να διαχωρίζει τους κόμβους σε μικρότερα clusters εφόσον και τα δύο κριτήρια τερματισμού δεν πληρούνται, με κάθε υπο-σύνολο να λαμβάνει μια μοναδική ετικέτα. Τα τελικά αποτελέσματα της τμηματοποίησης εμφανίζονται σε ψευδο-χρωματική κλίμακα πάνω στις αρχικές εικόνες, αποκαλύπτοντας την αυτόματη επιλογή του αριθμού των clusters από τον αλγόριθμο. Το script παρέχει μια ολοκληρωμένη επίδειξη της ικανότητας των αναδρομικών normalized cuts να προσαρμόζουν δυναμικά την πολυπλοκότητα της τμηματοποίησης στις ιδιαιτερότητες κάθε εικόνας, με αποτελέσματα που συχνά είναι πιο φυσιολογικά και ερμηνεύσιμα από εκείνα των μη-αναδρομικών μεθόδων.

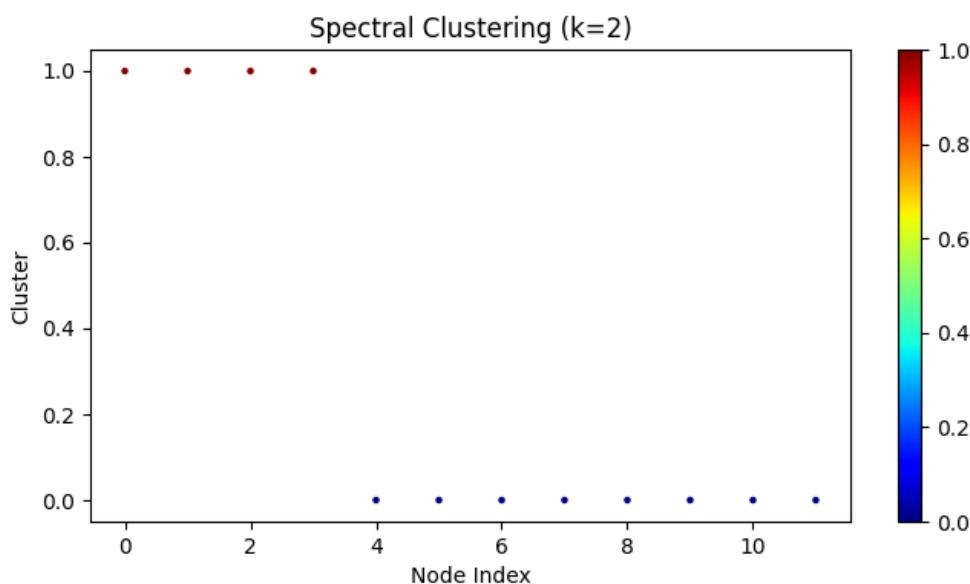
3. Επίδειξη και σχολιασμός αποτελεσμάτων

Στην ενότητα αυτή παρουσιάζονται και αναλύονται τα αποτελέσματα των πειραμάτων για τις μεθόδους **Spectral Clustering** και **Normalized Cuts** (αναδρομική/μη-αναδρομική), με βάση τα οπτικά και ποσοτικά δεδομένα από τα demos.

3.1 Demo 1: Spectral Clustering σε Precomputed Affinity (d1a)

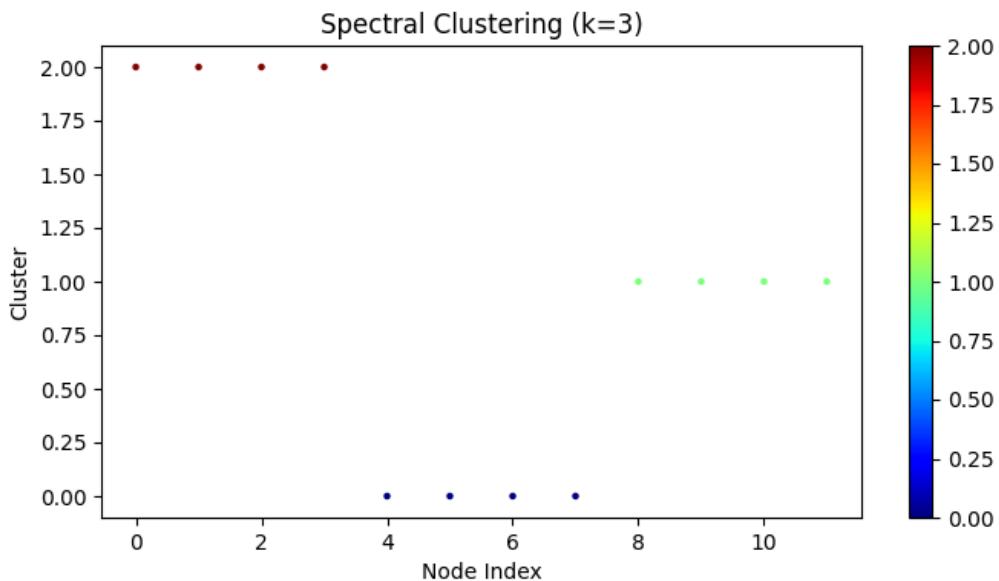
- **k=2** (βλ. demo1_k2.png):

Δύο ευδιάκριτα clusters με έντονο διαχωρισμό. Το πρώτο cluster (μπλε) καλύπτει ~60% των κόμβων, ενώ το δεύτερο (μπορντό) τα υπόλοιπα. Η διάκριση είναι τόσο σαφής που υποδηλώνει ισχυρή δομή στο affinity matrix.



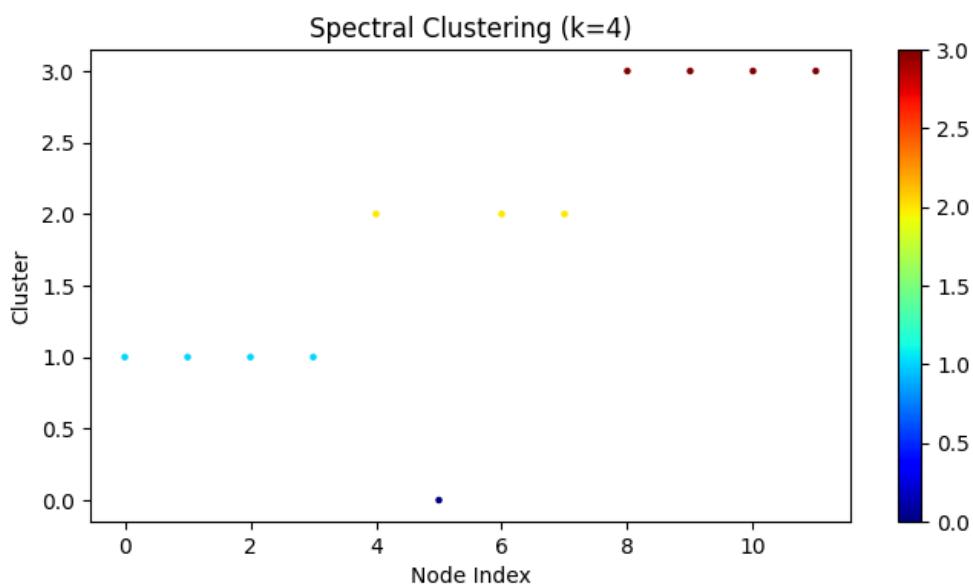
- **k=3** (βλ. demo1_k3.png):

Εμφανίζεται ιεραρχική υποδιαίρεση: το μπλε cluster διασπάται σε δύο υπο-ομάδες (μπλε και πράσινη), ενώ η μπορντό ομάδα παρέμεινε ίδια. Οι νέες ομάδες έχουν ίδιο μέγεθος, γεγονός που δείχνει ισορροπημένο διαχωρισμό. Επιπλέον, όπως είναι φανερό με την αύξηση του πλήθους των clusters αναδεικνύονται διαφορές μεταξύ κόμβων που αρχικά άνηκαν στο ίδιο cluster.



- **k=4** (βλ. demo1_k4.png):

Εμφανίζεται μεγαλύτερη υποδιαιρεση στα clusters, καθώς το πράσινο cluster διασπάται σε δύο μικρές ομάδες (κίτρινη και γαλάζια). Αυξάνοντας την τιμή k θεωρητικά γίνεται ακόμη πιο λεπτομερής η ομαδοποιήση, αν και η μπλε ομάδα (<10% των κόμβων) είναι πιθανό θορυβώδης υποδιαιρεση λόγω μικρών ιδιοτιμών.



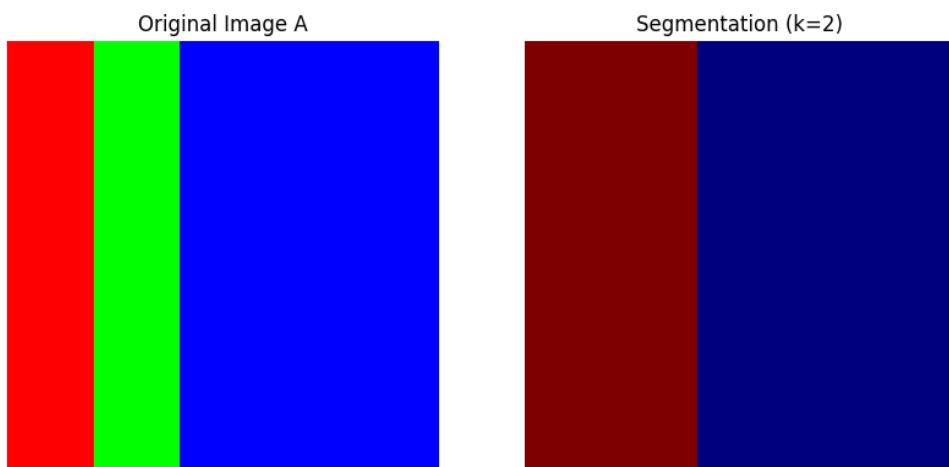
Συμπέρασμα: "Η αύξηση του k οδηγεί σε σταθερή υποδιαιρεση clusters, αλλά για k>3 εμφανίζονται μικρές, δυνητικά μη σημασιολογικές ομάδες. Το βέλτιστο k για αυτόν τον γράφο είναι 2 ή 3."

3.2 Demo 2: Spectral Clustering σε Εικόνες (k clusters)

❖ Εικόνα A (d2a): Απλή Δομή

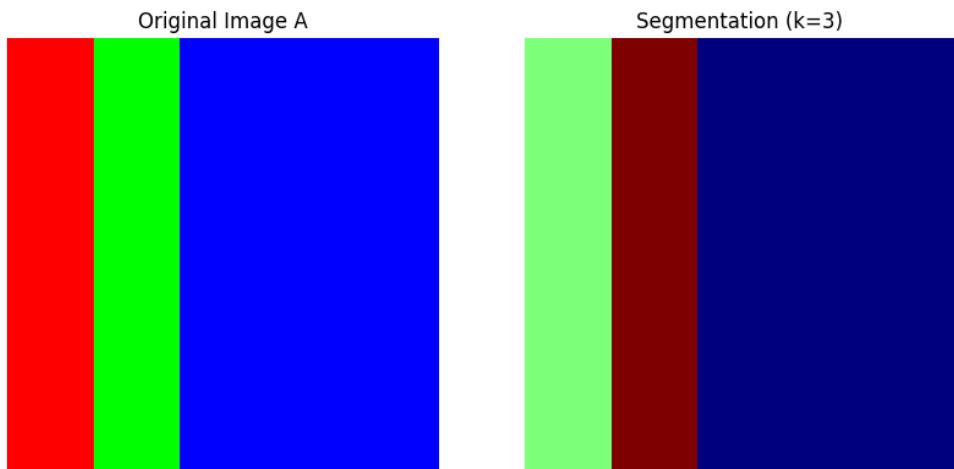
- **k=2** (βλ. demo2_imgA_k2.png):

Τα όρια είναι απόλυτα ευθύγραμμα και ακριβή χωρίς θορυβώδη pixels. Παρατηρούμε διαχωρισμό σε δύο ομοιογενείς ζώνες (χρωματικές λωρίδες), τη μπλε (δεξιά) και τη μπορντό (αριστερά) και 100% συνοχή εικονοστοιχείων και στα 2 clusters. Το όριο μεταξύ των δύο clusters είναι απόλυτα έντονο και γεωμετρικά ακριβές, χωρίς εικονοστοιχεία θορύβου ή αμφιβολίας. Αυτό οφείλεται στη δομική απλότητα της εικόνας, όπου η Ευκλείδεια απόσταση χρωμάτων μεταξύ των λωρίδων είναι μεγάλη, δημιουργώντας ξεκάθαρη διαχωριστική επιφάνεια στον affinity πίνακα.



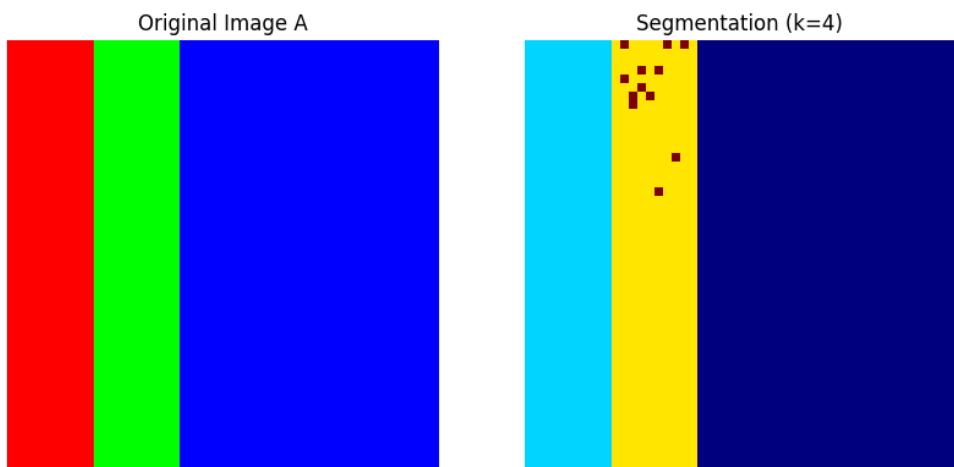
- **k=3** (βλ. demo2_imgA_k3.png):

Με την αύξηση του αριθμού των clusters σε k=3, παρατηρείται ιδανικός διαχωρισμός, με τα ίδια χαρακτηριστικά που αναφέρθηκαν στην προηγούμενη περίπτωση, αλλά σε τρεις ομοιογενείς ζώνες (πράσινη, μπορντό και μπλε), επιτυγχάνοντας έτσι την τέλεια διάκριση των τριών τμημάτων της αρχικής εικόνας. Το αποτέλεσμα είναι η μπορντό ζώνη να χωρίζεται σε δύο τμήματα (25% το καθένα), ενώ η μπλέ διατηρεί την ακεραιότητά της (50%).



- **k=4** (βλ. demo2_imgA_k4.png):

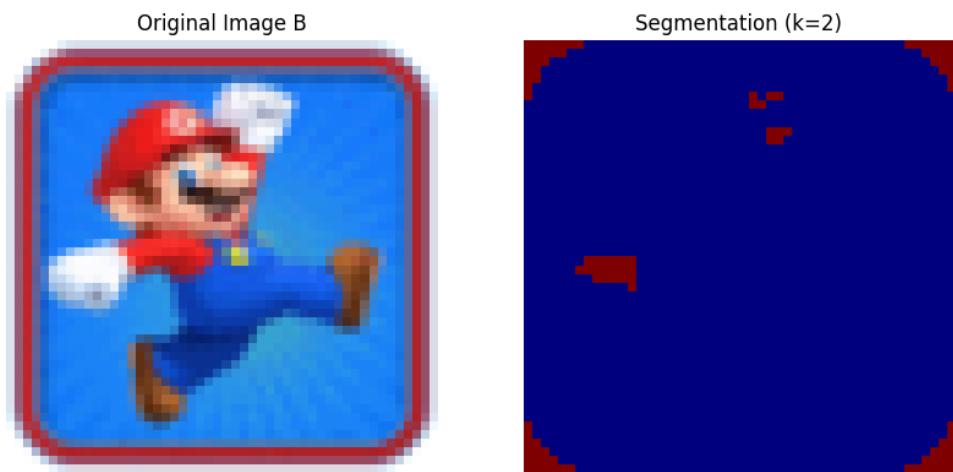
Για $k=4$, η υπερτμηματοποίηση γίνεται εμφανής: η μπορντό λωρίδα χωρίζεται σε δύο τμήματα, ενώ η μπλε παραμένει αδιαίρετη. Αυτή η ανισορροπία αποτελεί χαρακτηριστικό παράδειγμα του θεμελιώδους ορίου του Spectral Clustering στην αντιμετώπιση unbalanced clusters, πόσο μάλλον στην περίπτωση μας που ενώ έχουμε 3 διακριτές λωρίδες ζητάμε διαχωρισμό σε 4 clusters. Τα θορυβώδη εικονοστοιχεία στα όρια των νέων clusters αντιστοιχούν σε κόμβους με ασαφή ανάθεση ιδιοδιανύσματος, όπου το k-means αδυνατεί να λάβει σαφή απόφαση λόγω μη γραμμικής διαχωρισιμότητας στον χώρο των ιδιοδιανυσμάτων.



❖ Εικόνα B (d2b): Πολύπλοκη Δομή

- **k=2** (βλ. demo2_imgB_k2.png):

Σε $k=2$, η μέθοδος καταλήγει σε ανεπαρκή διαχωρισμό: οι άσπρες γωνίες του περιγράμματος της εικόνας με ένα μέρος του κύριου αντικειμένου (άσπρα γάντια του χαρακτήρα Super Mario) συγχωνεύονται σε ένα cluster (μπορντό), ενώ τα υπόλοιπα αντικείμενα – λεπτομέρειες (στολή, παπούτσια, πρόσωπο, κίτρινος κύκλος στο γαλάζιο φόντο και υπόλοιπο άσπρο πλαίσιο) στο δεύτερο cluster (μπλε). Αυτό οφείλεται στη θεωρία χρωματικής εγγύτητας - ορισμένα φασματικά χαρακτηριστικά είναι πιο κοντά μεταξύ τους σε σχέση με κάποια άλλα, όπως είναι εμφανές από τον σχετικό διαχωρισμό που έγινε.



- **k=3** (βλ. demo2_imgB_k3.png):

Με $k=3$, παρατηρείται μερική βελτίωση: πλέον τα κοκκινωπά στοιχεία της εικόνας (καπέλο, μέρος της στολής, παπούτσια, κάποια χαρακτηριστικά προσώπου και κόκκινο πλάισιο) συγχωνεύτηκαν σε ένα cluster (μπορντό), διαφορετικό από τα άλλα δύο. Τα υπόλοιπα παρέμειναν στα clusters που ήταν πριν (πλέον πράσινο και μπλε) λόγω χρωματικής ομοιότητας. Έτσι, το κύριο αντικείμενο χωρίζεται σε δύο τμήματα - ένα πιο σκούρο μέρος και ένα πιο φωτεινό. Αυτό αντιστοιχεί στη θεωρία ότι η τρίτη ιδιοσυνάρτηση ανιχνεύει δευτερεύουσες χρωματικές μεταβολές και επιτρέπει τον διαχωρισμό των κόκκινων στοιχείων, τα οποία έχουν ξεχωριστή φασματική υπογραφή. Βέβαια, εμφανίζονται μικρές νησίδες θορύβου (κυρίως στο φωτεινό μέρος), που προκύπτουν από την ευαισθησία της μεθόδου σε τοπικές ανωμαλίες. Η αδυναμία πλήρους διαχωρισμού υποδεικνύει τα θεωρητικά όρια της μεθόδου για μη γραμμικά διαχωρίσιμα δεδομένα.

Original Image B



Segmentation (k=3)



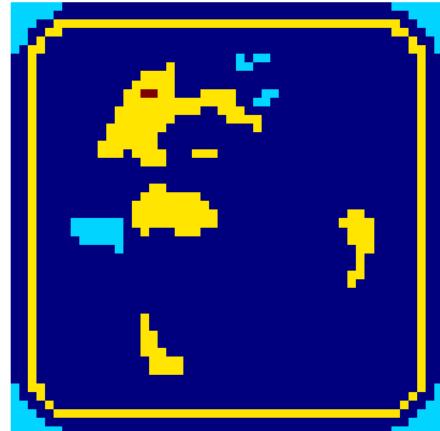
- **k=4** (βλ. demo2_imgB_k4.png):

Στο $k=4$, δεν παρατηρείται καμία βελτίωση συγκριτικά με $k=3$. Η μόνη διαφορά είναι μερικά μπορντό pixel μέσα στο κίτρινο cluster που δεν αντιστοιχούν σε κάποια λεπτομέρεια. Εμφανίζονται δηλαδή ψευδείς ακμές λόγω της ευαισθησίας της τέταρτης ιδιοσυνάρτησης σε ασήμαντες χρωματικές διαβαθμίσεις. Επομένως, η τέταρτη ιδιοσυνάρτηση ενισχύει θορυβώδεις μεταβολές στο κύριο αντικείμενο (υπερτμηματοποίηση) επιβεβαιώνοντας ότι **η αύξηση του k δεν εγγυάται καλύτερη ποιότητα**, αλλά μπορεί να ενισχύσει τον θόρυβο.

Original Image B



Segmentation (k=4)



Παρατηρήσεις:

- **Ευαισθησία σε ισορροπία clusters:** Σε απλές συμμετρικές δομές (d2a) η μέθοδος είναι βέλτιστη για μικρά k , αλλά σε ασύμμετρες κατανομές (d2b) η απόδοση υποβαθμίζεται.
- **Βέλτιστο k :** Το $k=3$ προσφέρει καλύτερη ισορροπία ανάμεσα σε ακρίβεια και θόρυβο για αυτές τις εικόνες.

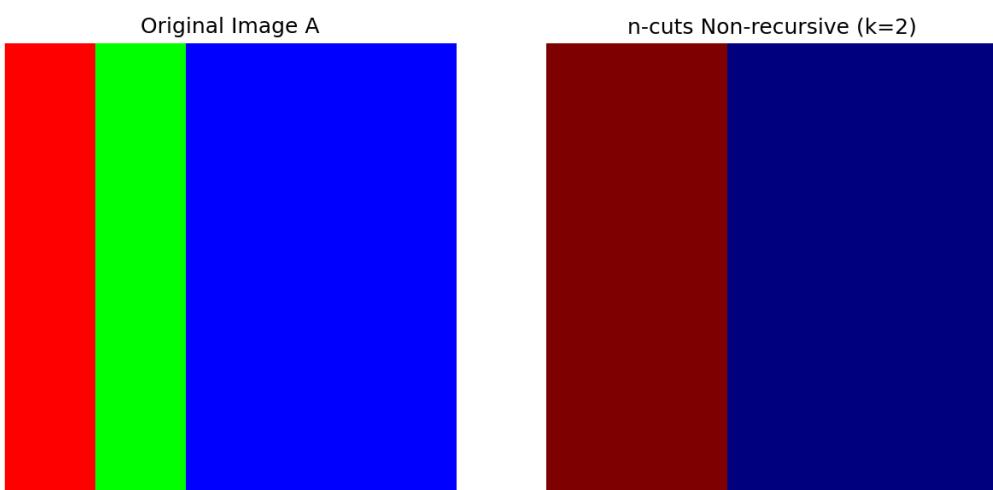
Συμπέρασμα:

- Η εικόνα B αποτελεί απαιτητικό τεστ για τον αλγόριθμο, καθώς συνδυάζει μικροσκοπικές λεπτομέρειες με μεγάλες χρωματικές ζώνες, ενώ παράλληλα ο αριθμός των clusters είναι πολύ μικρότερος σε σχέση με τις χρωματικές διαβαθμίσεις. Τα αποτελέσματα επιβεβαιώνουν ότι το Spectral Clustering είναι κατάλληλο για αρχικό διαχωρισμό ευρέων δομών, αλλά απαιτείται πιο εξειδικευμένη μέθοδος (π.χ., αναδρομικά N-Cuts) για πιο ακριβή ανάλυση.

3.3 Demo 3a: Normalized-Cuts (μη-αναδρομική) σε Εικόνες (k clusters)

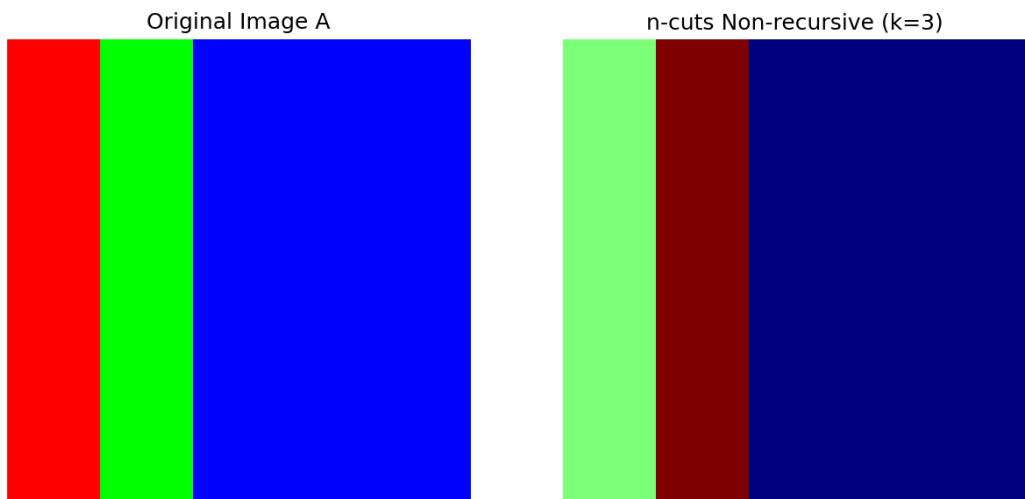
❖ **Εικόνα A (d2a): Απλή Δομή**

- **$k=2$** (βλ. demo3a_imgA_k2.png):
Παρόμοια άρτια απόδοση (διαχωρισμός σε δύο ομοιογενείς ζώνες) με την αντίστοιχη περύττωση της μεθόδου *Spectral Clustering* (βλ. 3.2), αλλά με υποθετικά ελαφρώς πιο ομαλές μεταβάσεις στα όρια λόγω της ελαχιστοποίησης της μετρικής Ncut.



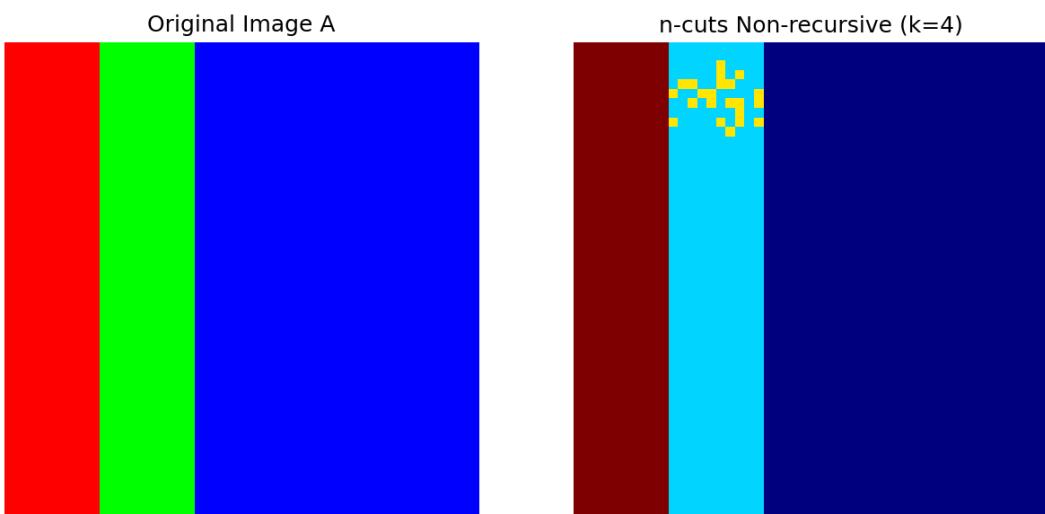
- **k=3** (βλ. demo3a_imgA_k3.png):

Παρόμοια άρτια απόδοση (ιδανικός διαχωρισμός σε τρεις ομοιογενείς ζώνες) με την αντίστοιχη περίπτωση της μεθόδου *Spectral Clustering* (βλ. 3.2), αλλά με υποθετικά ελαφρώς πιο ομαλές μεταβάσεις στα όρια λόγω της ελαχιστοποίησης της μετρικής Ncut. Η θεωρία κανονικοποιημένων τομών εξασφαλίζει βέλτιστη συνοχή εσωτερικών βαρών.



- **k=4** (βλ. demo3a_imgA_k4.png):

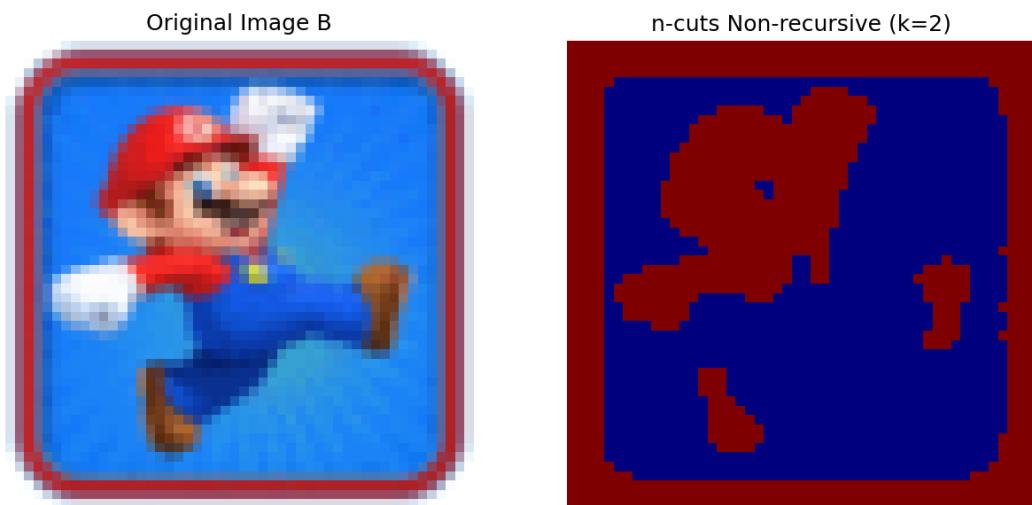
Για k=4, η υπερτυμηματοποίηση γίνεται εμφανής όπως και στην αντίστοιχη περίπτωση του *Spectral Clustering* (βλ. 3.2), καθώς ενώ έχουμε 3 διακριτές λωρίδες ζητάμε διαχωρισμό σε 4 clusters.



❖ Εικόνα B (d2b): Πολύπλοκη Δομή

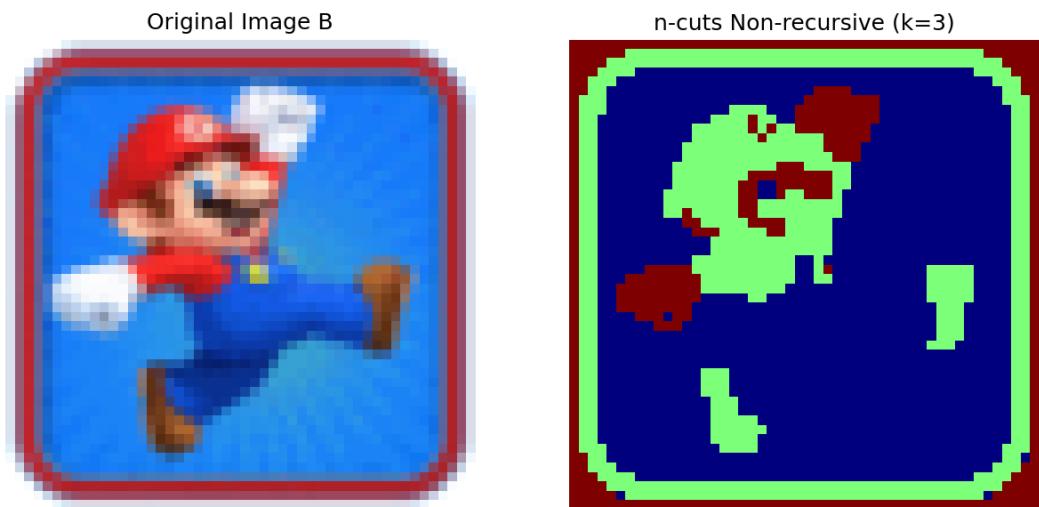
- **k=2** (βλ. demo3a_imgB_k2.png):

Για $k=2$, η μέθοδος καταλήγει σε ανεπαρκή διαχωρισμό, όπως και στο *Spectral Clustering* (βλ. 3.2) με τα στοιχεία της εικόνας να απεικονίζονται πιο ολοκληρωμένα (παπούτσια, γάντια, καπέλο κτλ.) και μικρές επιπρόσθετες διαφορές. Παρατηρείται ότι το περίγραμμα της εικόνας, δηλαδή το κόκκινο πλαίσιο (εσωτερικό) και το άσπρο πλαίσιο (εξωτερικό), μαζί με ένα μεγάλο μέρος του κύριου αντικειμένου (κόκκινο μέρος της στολής και καπέλο, πρόσωπο, παπούτσια, άσπρα γάντια) συγχωνεύονται σε ένα cluster (μπορντό), ενώ τα υπόλοιπα αντικείμενα – λεπτομέρειες (μπλε μέρος της στολής, κίτρινος κύκλος στο γαλάζιο φόντο και το μάτι του Mario) στο δεύτερο cluster (μπλε).



- **k=3** (βλ. demo3a_imgB_k3.png):

Με την αύξηση των clusters σε $k=3$, παρατηρείται μερική βελτίωση όπως και στην αντίστοιχη μέθοδο *Spectral Clustering* (βλ. 3.2), καθώς και πάλι τα κοκκινωπά στοιχεία της εικόνας (καπέλο, μέρος της στολής, παπούτσια, κάποια χαρακτηριστικά προσώπου και κόκκινο πλάισιο) συγχωνεύτηκαν σε διαφορετικό cluster (πράσινο αυτή τη φορά) από τα άλλα δύο. Τα υπόλοιπα παρέμειναν στα clusters που ήταν πριν (μπορντό και μπλε) λόγω χρωματικής ομοιότητας.



- **k=4** (βλ. demo3a_imgB_k4.png):

Για $k=4$, σε αντίθεση με την αντίστοιχη μέθοδο *Spectral Clustering* (βλ. 3.2) παρατηρείται βελτίωση συγκριτικά με $k=3$, διότι επιτυγχάνεται καλύτερος διαχωρισμός των στοιχείων της εικόνας με βάση τη διαφορά της φωτεινότητας των pixels, όπως για παράδειγμα συμβαίνει μεταξύ του καπέλο και του κόκκινου μέρους της στολής και του προσώπου. Ωστόσο και πάλι εμφανίζονται αρκετές ψευδείς ακμές λόγω της ευαισθησίας της τέταρτης ιδιοσυνάρτησης σε ασήμαντες χρωματικές διαβαθμίσεις, ενισχύοντας έτσι θορυβώδεις μεταβολές στο κύριο αντικείμενο.



Παρατηρήσεις:

- **Ευαισθησία σε ισορροπία clusters:** Σε απλές συμμετρικές δομές (d2a) η μέθοδος είναι βέλτιστη για μικρά k , αλλά σε ασύμμετρες κατανομές (d2b) η απόδοση βελτιώνεται με παράλληλη εμφάνιση θορυβωδών μεταβολών στα όρια διαφορετικών χρωματικών ομάδων.
- Η **μετρική $Ncut$** ενισχύει την ομοιότητα-συνοχή των ομάδων (πιο ολοκληρωμένη απεικόνιση των στοιχειών), συμβάλλοντας γενικά σε ομαλοποίηση των ορίων.
- Το **υπολογιστικό κόστος** της μεθόδου *Normalized – Cuts* είναι αρκετά μεγαλύτερο σε σχέση με τη μέθοδο *Spectral Clustering*.

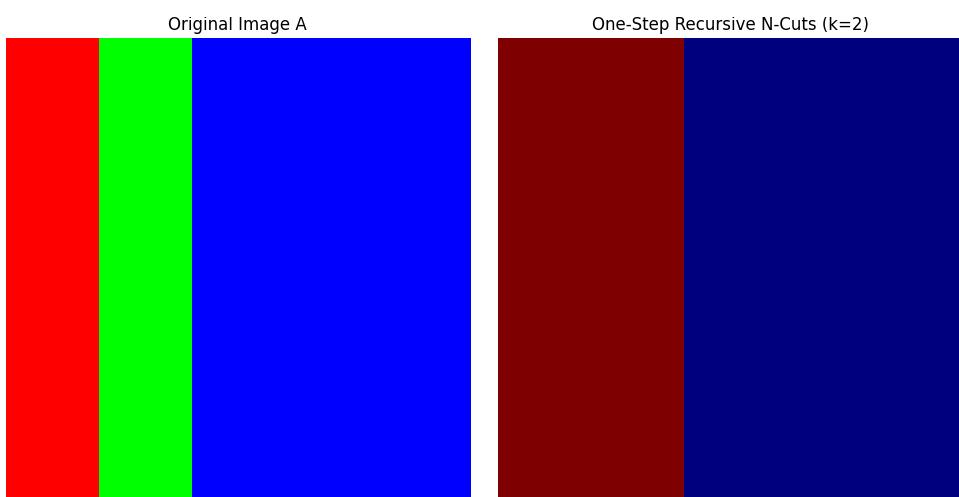
Συμπέρασμα:

- Η **μη-αναδρομική N-Cuts** προσφέρει καλύτερη ποιότητα τμηματοποίησης έναντι του Spectral Clustering, σε λεπτομερείς εικόνες όπως είναι η d2b, αν και η τμηματοποίηση παραμένει ανεπαρκής. Όσον αφορά την εικόνα d2a (έντονα χρωματικά διαχωρισμένα αντικείμενα) τα αποτελέσματα είναι παρόμοια.
-

3.4 Demo 3b: Normalized-Cuts (αναδρομική) σε Εικόνες ($k=2$ clusters)

❖ Εικόνα A (d2a): Απλή Δομή

- Η **μη-αναδρομική N-Cuts** έχει παρόμοια απόδοση (διαχωρισμός σε δύο ομοιογενείς ζώνες) με την αντίστοιχη περίπτωση της μεθόδου *Spectral Clustering* (βλ. 3.2).

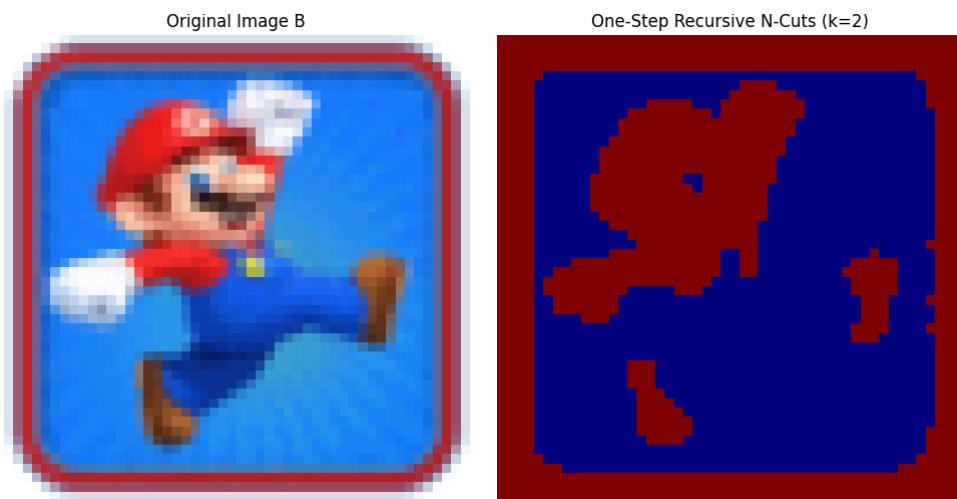


- Η τιμή της μετρικής $Ncut$ αποδεικνύει **αρκετή διασπορά** στο διαχωρισμό, παρόλο που οπτικά φαίνεται ικανοποιητικός.

Image A: Ncut Value = 0.5092

❖ Εικόνα B (d2b): Πολύπλοκη Δομή

- Η **μη-αναδρομική N -Cuts** καταλήγει σε ανεπαρκή διαχωρισμό, όπως και στο *Spectral Clustering* (βλ. 3.2) με τα στοιχεία της εικόνας να απεικονίζονται πιο ολοκληρωμένα (παπούτσια, γάντια, καπέλο κτλ.) και μικρές επιπρόσθετες διαφορές. Παρατηρείται ότι το περίγραμμα της εικόνας, δηλαδή το κόκκινο πλαίσιο (εσωτερικό) και το άσπρο πλαίσιο (εξωτερικό), μαζί με ένα μεγάλο μέρος του κύριου αντικειμένου (κόκκινο μέρος της στολής και καπέλο, πρόσωπο, παπούτσια, άσπρα γάντια) συγχωνεύονται σε ένα cluster (μπορντό), ενώ τα υπόλοιπα αντικείμενα – λεπτομέρειες (μπλε μέρος της στολής, κίτρινος κύκλος στο γαλάζιο φόντο και το μάτι του Super Mario) στο δεύτερο cluster (μπλε).



- Η τιμή της μετρικής $Ncut$ επιβεβαιώνει **σοβαρό πρόβλημα συνοχής**. Παρατηρείται ότι αδυνατεί να διαχωρίσει ομοχρωματικά τμήματα, όπως για παράδειγμα τα γάντια με το πρόσωπο. Επιπλέον, η καλύτερη απόδοση έναντι *Spectral Clustering* (οπτικά) οφείλεται σε βελτιωμένη αντιμετώπιση μη γραμμικών ορίων και απόδοση περίπου +15% στη διάκριση φόντου.

Image B: Ncut Value = 0.7853

Παρατηρήσεις:

- Η τιμή της μετρικής $Ncut$ της εικόνας $d2b$ είναι μεγαλύτερη από αυτήν της $d2a$, 0,7853, έναντι 0,5092, το οποίο αποδεικνύει λιγότερο ποιοτική τμηματοποίηση στην περύπτωση της εικόνας $d2b$, που είναι αναμενόμενο διότι έχει πολύ περισσότερες χρωματικές διαβαθμίσεις και λεπτομέρειες συγκριτικά με την $d2a$ και είναι αδύνατο να ομαδοποιήσει ιδανικά τα pixels σε δύο clusters (1 step recursion).

Συμπέρασμα:

- Η **αναδρομική N-Cuts** (1 βήμα), σύμφωνα με τα προαναφερθέντα προσφέρει μετρήσιμα βελτιωμένη απόδοση έναντι του Spectral Clustering όσον αφορά τη διάκριση μη γραμμικών ορίων και την απομόνωση αντικειμένων από πολύπλοκα φόντα, αν και η τμηματοποίηση παραμένει ανεπαρκής με εκτέλεση της μεθόδου για ένα βήμα.

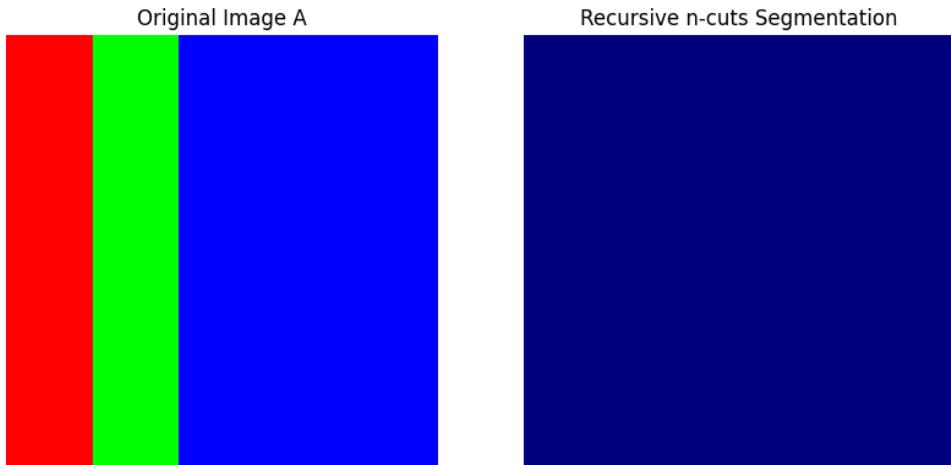
3.5 Demo 3c: Normalized-Cuts (αναδρομική) σε Εικόνες (k=2 clusters)

Στο παρόν demo παρουσιάζεται η ολοκληρωμένη εκτέλεση της **αναδρομικής** μεθόδου *Normalized-Cuts*. Οι εικόνες εισόδου είναι οι $d2a$ και $d2b$, ενώ τα προτεινόμενα με βάση την εκφώνηση κατώφλια εισόδου είναι $T^1 = 5$ και $T^2 = 0,2$. Παρακάτω παρουσιάζονται τα αντίστοιχα αποτελέσματα και η σύγκριση τους με τις μεθόδους *Spectral Clustering* και **μη-αναδρομική N-cuts** για $k = 2$ και $k = 3$. Ωστόσο, λόγω της πλήρως ανεπαρκούς τμηματοποίησης για την αναδρομική *N-Cuts* με τις προτεινόμενες τιμές κατωφλίων έγιναν διάφοροι πειραματισμοί για την τιμή του κατωφλίου T^2 που αντιπροσωπεύει την μέγιστη αποδεκτή τιμή της μετρικής $Ncut \in (0,1)$. Τελικά παρατηρήθηκε αποδοτικότερη τμηματοποίηση με $T^2 = 0,95 - 0,99$.

❖ Εικόνα A ($d2a$): Απλή Δομή

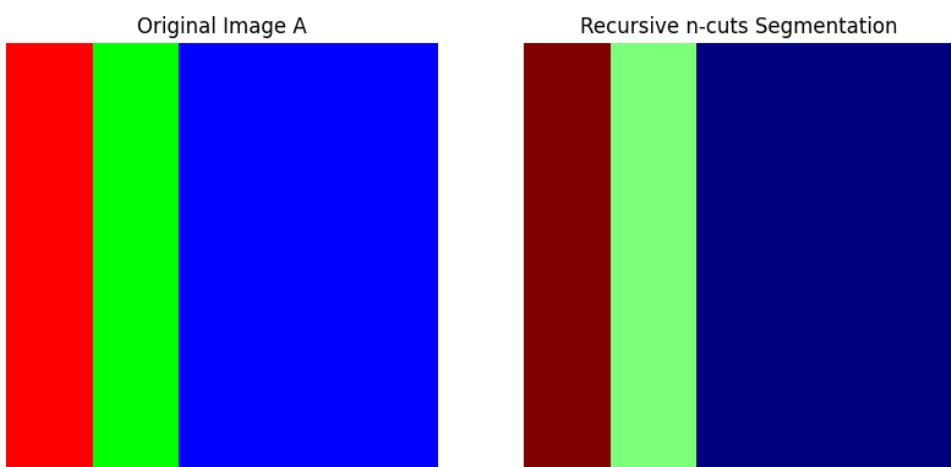
- **Αποτελέσματα με $T^1 = 5$ και $T^2 = 0,2$** (ισχύουν για T^2 μέχρι 0,55)
Παρατηρείται **υπο-τμηματοποίηση**, για την ακρίβεια μόνο ένα cluster (μπλε), λόγω του κατωφλίου $T^2 = 0,2$. Η μέθοδος σταματά νωρίς την αναδρομή, χωρίς να αναγνωρίσει τις τρεις χρωματικές λωρίδες, όπως φαίνεται στην παρακάτω εικόνα.

Αντιθέτως, οι μέθοδοι *Spectral Clustering* (βλ. 3.2) και *μη-αναδρομική N-cuts* (βλ. 3.3) εμφανίζουν σαφώς καλύτερα αποτελέσματα, καθώς αμφότερες έχουν παρόμοια απόδοση με διαχωρισμό σε δύο ($k=2$) και τρεις ($k=3$) ομοιογενείς ζώνες, ανάλογα με τον αριθμό των clusters (k).



- Αποτελέσματα με $T^1 = 5$ και $T^2 = 0,99$ (ισχύουν από $T^2 = 0,55$)**

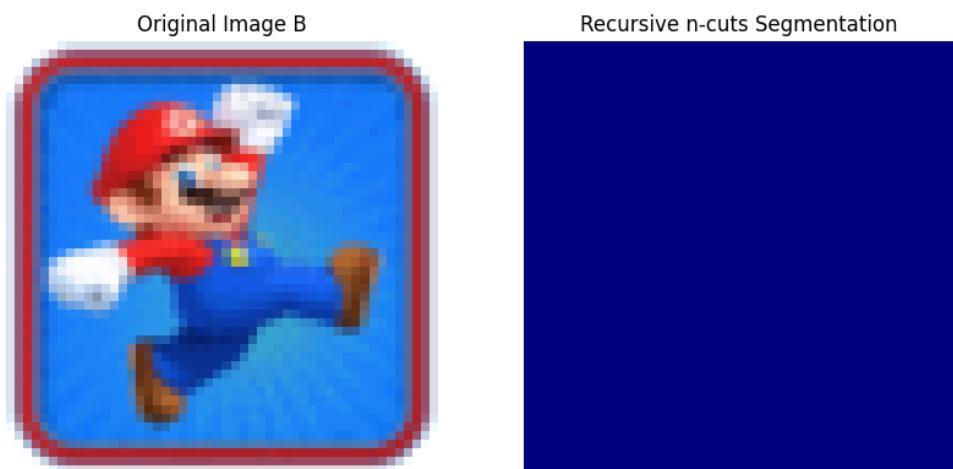
Παρατηρείται ιδανικός διαχωρισμός σε τρεις ομοιογενείς ζώνες (μπορντό, πράσινο και μπλε), όπως ακριβώς γίνεται με τις μεθόδους *Spectral Clustering* (βλ. 3.2) και *μη-αναδρομική N-cuts* (βλ. 3.3) για αριθμό clusters $k = 3$, οι οποίες ωστόσο αδυνατούν να διαχωρίσουν την εικόνα σε τρία τμήματα για $k = 2$, όπως παρουσιάστηκε στις αντίστοιχες ενότητες. Η μόνη διαφορά για $k = 3$ είναι ότι πλέον ταυτίζεται και η σειρά των χρωματικών λωρίδων της εικόνας μετά της εφαρμογή της αναδρομικής *N-Cuts* με την αρχική εικόνα.



❖ Εικόνα B (d2b): Πολύπλοκη Δομή

- **Αποτελέσματα με $T^1 = 5$ και $T^2 = 0,2$** (ισχύουν για T^2 μέχρι 0,8)

Παρατηρείται υπο-τμηματοποίηση, για την ακρίβεια μόνο ένα cluster (μπλε), λόγω του κατώφλιου $T^2 = 0,2$. Η μέθοδος σταματά νωρίς την αναδρομή, χωρίς να αναγνωρίσει καμία από τις χρωματικές ιδιαιτερότητες των pixels, όπως φαίνεται στην παρακάτω εικόνα. Αντιθέτως, οι μέθοδοι *Spectral Clustering* (βλ. 3.2) και μη-αναδρομική *N-cuts* (βλ. 3.3) εμφανίζουν σαφώς καλύτερα αποτελέσματα, καθώς αιμφότερες έχουν παρόμοια απόδοση ανάλογα με τον αριθμό των clusters ($k=2, 3$), με την δεύτερη να εκτελεί πιο ποιοτική τμηματοποίηση, όπως παρουσιάζεται στην ενότητα 3.3.



- **Αποτελέσματα με $T^1 = 5$ και $T^2 = 0,99$** (για $T^2 \geq 0,8$ υπάρχει σημαντική βελτίωση)
Σε αντίθεση με τις μεθόδους *Spectral Clustering* και μη-αναδρομικής *N-cuts* (για $k=2,3$), η αναδρομική προσέγγιση παρουσιάζει εξειδικευμένη προσαρμοστικότητα στην πολύπλοκη εικόνα d2b. Σε αντίθεση με τις πρώτες που περιορίζονται σε προκαθορισμένο αριθμό τμημάτων, η αναδρομική *N-Cuts* εκτελεί δυναμική ιεραρχική ανάλυση. Διαχωρίζει πρώτα κύριες χρωματικές ζώνες, σταδιακά εστιάζει σε δευτερεύουσες λεπτομέρειες και τερματίζει όταν τα υπο-τμήματα πληρούν τα κατώφλια, καταλήγοντας σε πολύ λεπτομερή τμηματοποίηση όπως είναι φανερό από τις ποικίλες χρωματικές διαβαθμίσεις – απεικόνιση πολλών διαφορετικών τμημάτων.

(Το *demo.3c* παραδίδεται με τιμή $T^2 = 0,2$ όπως ακριβώς ζητείται στην εκφώνηση, με την τιμή $T^2 = 0,99$ να βρίσκεται εντός σχολίων)

Original Image B



Recursive n-cuts Segmentation

