

# Ψηφιακή Επεξεργασία Εικόνας

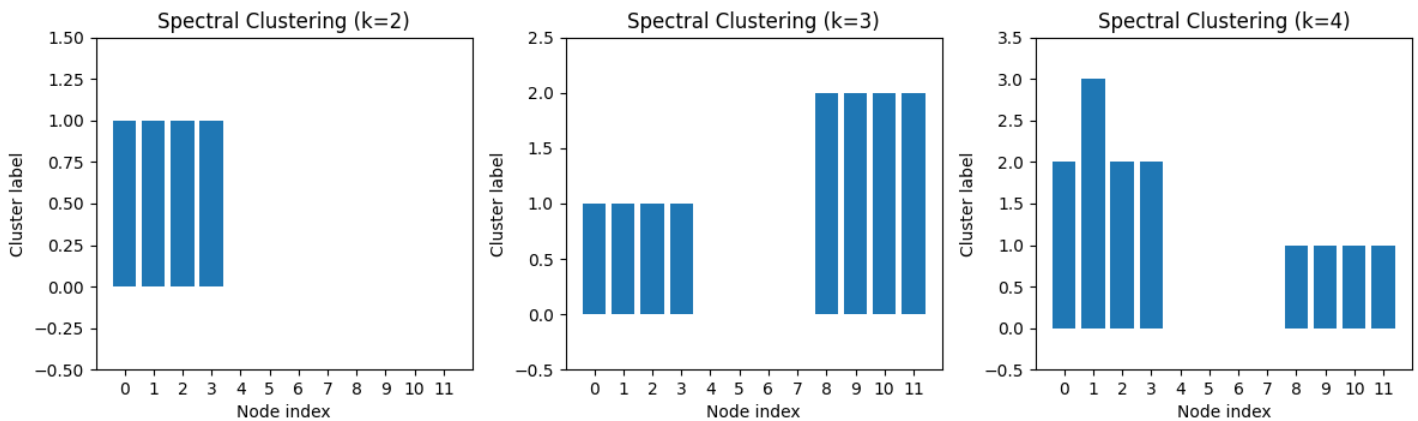
## Εργασία 3

### Image Segmentation

Στέλιος Ανδρεάδης    AEM: 10022

#### Demo 1

Δόθηκε affinity matrix (σε αρχείο MATLAB, μεταβλητή d1a), ο οποίος περιέχει 3 αρκετά συσχετισμένες ομάδες δεδομένων. Εφαρμόζοντας τη συνάρτηση `spectral_clustering()` στον συγκεκριμένο πίνακα, προέκυψαν τα εξής αποτελέσματα:

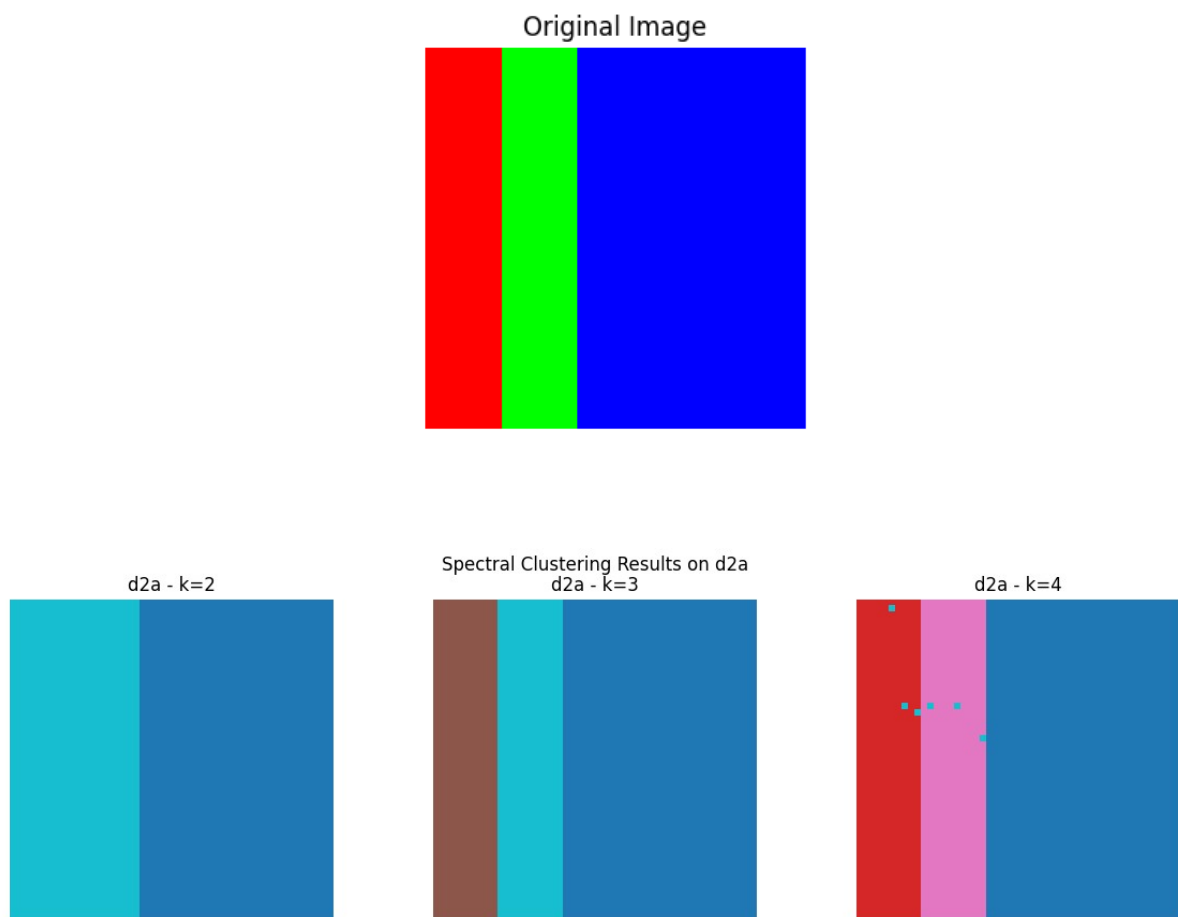


- Παρατηρούμε ότι για  $k=2$ , δύο από τις τρεις βασικές ομάδες συμπτήχθηκαν σε ένα cluster, με αποτέλεσμα τα πρώτα τέσσερα nodes να έχουν ετικέτα 1, ενώ τα υπόλοιπα ανήκουν στο συμπτηγμένο cluster με ετικέτα 0.
- Για  $k=3$ , η τμηματοποίηση είναι η φυσική και αναμενόμενη, με τη συνάρτηση να διαχωρίζει σωστά τις τρεις ομάδες στα αντίστοιχα clusters.
- Για  $k=4$ , η τμηματοποίηση γίνεται πιο λεπτομερής· ένα στοιχείο που προηγουμένως ανήκε στο cluster με ετικέτα 2 μεταφέρεται σε νέο cluster με ετικέτα 3. Αυτό το μοναδικό στοιχείο δεν προκαλεί έκπληξη, καθώς ο affinity matrix δεν παρέχει επιπλέον πληροφορίες για τη δημιουργία τεσσάρων φυσικών ομάδων.

## Demo 2

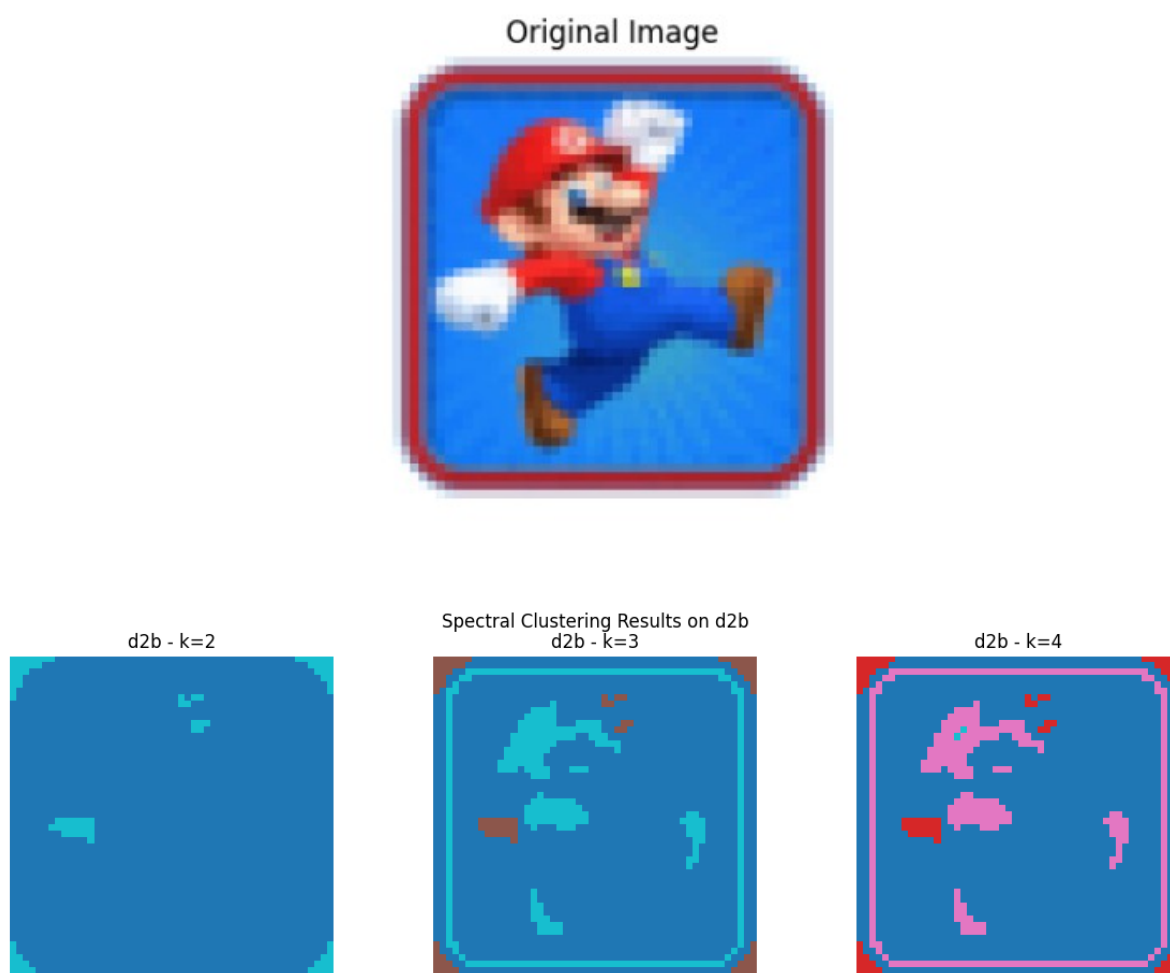
Δόθηκαν 2 πίνακες (MATLAB, d2a, d2b [50,50,3]RGB) οι οποίοι αντιπροσωπεύουν έγχρωμες RGB εικόνες. Με τη συνάρτηση `image_to_graph()` μετατρέπονται σε fully connected, μη κατευθυνόμενο γράφο και χρησιμοποιώντας πάλι τη συνάρτηση `spectral_clustering()` για τιμές  $k = 2, 3, 4$  παίρνουμε τα παρακάτω αποτελέσματα.

Για την **πρώτη εικόνα**:



Στην πρώτη εικόνα (d2a), η οποία εμφανίζει τρεις πλήρως διαχωρισμένες περιοχές, το spectral clustering αποδίδει ικανοποιητικά αποτελέσματα για  $k = 3$ , διαχωρίζοντας με ακρίβεια τις περιοχές. Για  $k = 2$ , δύο περιοχές συγχωνεύονται λανθασμένα, ενώ για  $k = 4$  εμφανίζεται ένα επιπλέον τεχνητό cluster που δεν αντιστοιχεί σε πραγματική δομή της εικόνας. (Αντιστοιχία με όσα είδαμε στο demo 1)

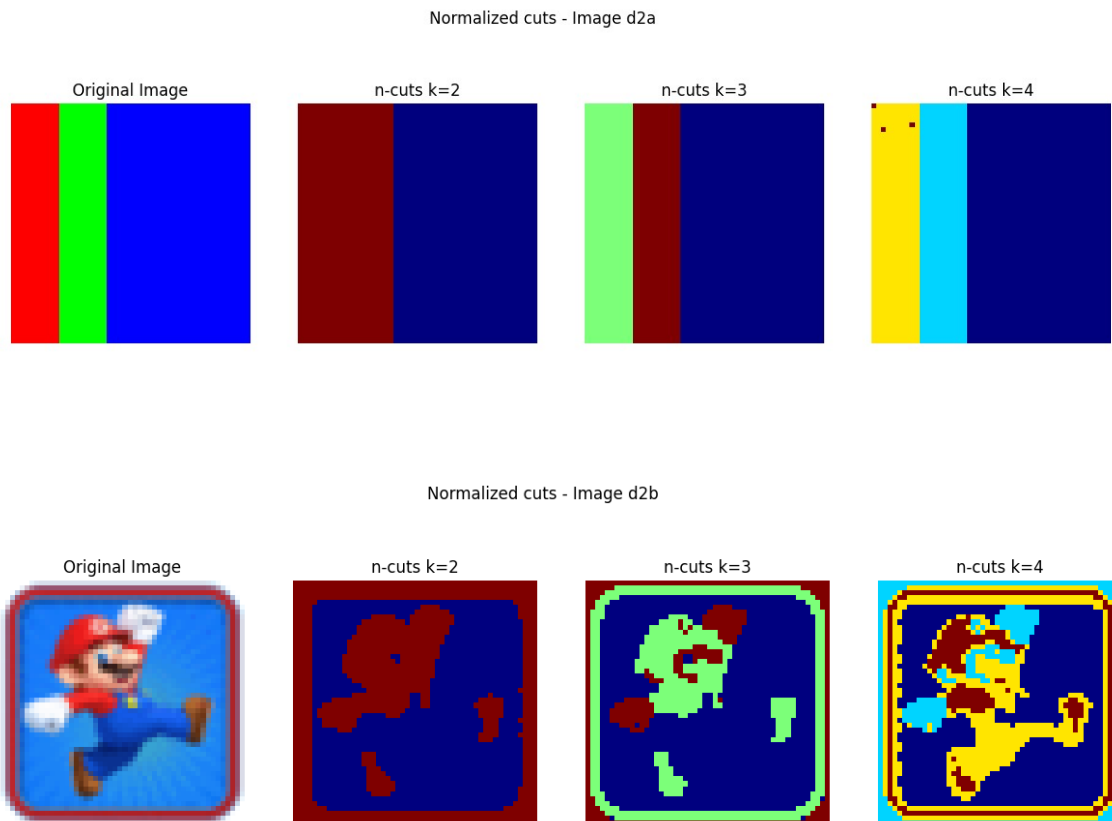
Για την **δεύτερη** εικόνα:



Στη δεύτερη εικόνα (d2b), η παρουσία λεπτομερειών οδηγούν σε καλύτερη απόδοση του αλγορίθμου για μεγαλύτερες τιμές του  $k$ . Η αύξηση των clusters επιτρέπει στον αλγόριθμο να αναγνωρίσει επιπρόσθετες διαφοροποιήσεις μέσα στην εικόνα, προσφέροντας πιο λεπτομερή διαχωρισμό.

## Demo 3a

Εφαρμόζουμε τη μη αναδρομική εκδοχή του αλγορίθμου n-cuts για ίδιες τιμές διαμέρισης, όπως στο spectral\_clustering( $k=2,3,4$ ).

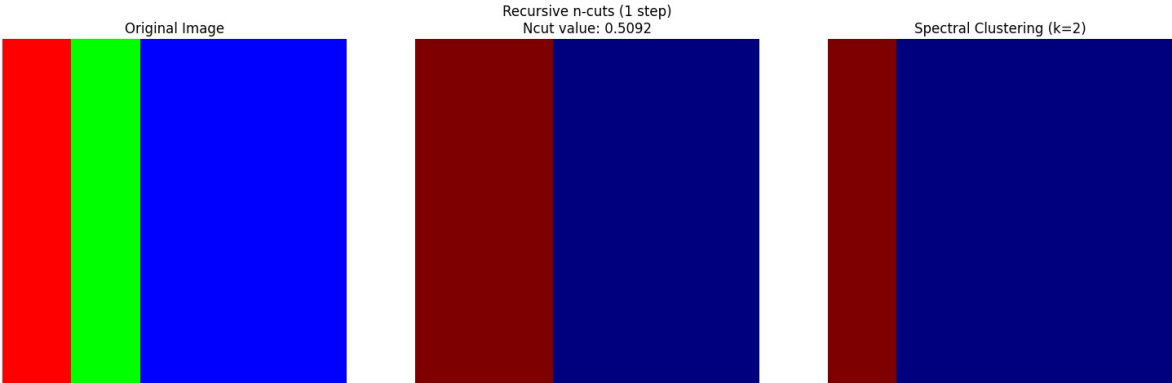


- Για την πρώτη εικόνα, τα αποτελέσματα του αλγορίθμου Normalized Cuts είναι παρόμοια με εκείνα του spectral clustering, καθώς και οι δύο μέθοδοι καταφέρνουν πολύ καλο αποτέλεσμα. Αυτό οφείλεται στην απλή δομή της εικόνας, η οποία έχει μεγάλες οριοθετημένες περιοχές.
- Αντίθετα, στη δεύτερη εικόνα παρατηρείται σημαντική διαφοροποίηση μεταξύ των δύο μεθόδων, με τη n-cuts να αποδίδει πολύ πιο ικανοποιητικά αποτελέσματα. Η εικόνα αυτή είναι σύνθετη με πολλές λεπτομέρειες, οι οποίες αγνοούνται ή συγχωνεύονται από το spectral clustering.  
Η μέθοδος n-cuts εκτός από την ελαχιστοποίηση της σύνδεσης μεταξύ διαφορετικών περιοχών, επιχειρεί να διατηρήσει σχετικά υψηλή συσχέτιση εντός κάθε cluster. Αυτή η προσέγγιση, επιτρέπει στην n-cuts να αναγνωρίζει καλύτερα τις τοπικές διαφοροποιήσεις, οδηγώντας σε πιο λεπτομερή αποτελέσματα.

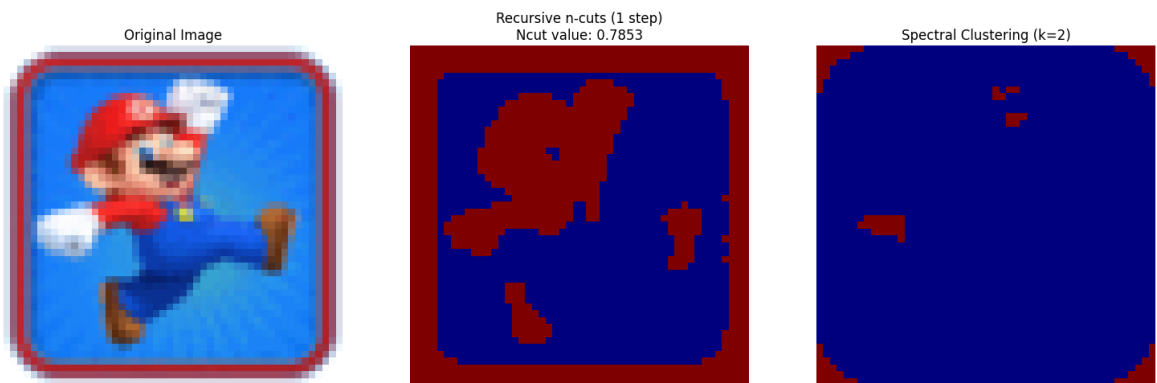
### Demo 3b

Εκτελούμε την αναδρομική μέθοδο n-cuts για ένα βήμα, δηλαδή παίρνουμε μη αναδρομική n-cuts για  $k=2$ .

Comparison on Image d2a



Comparison on Image d2b



Τα αποτελέσματα παραμένουν τα ίδια με τα προηγούμενα και έχουν ήδη σχολιαστεί. Σε αυτή τη φάση, ιδιαίτερο ενδιαφέρον παρουσιάζει η τιμή της Ncut value για τις δύο εικόνες.

Η χαμηλή τιμή στην πρώτη εικόνα επιβεβαιώνει και ποσοτικά την απλότητα της δομής της, καθώς υποδηλώνει ότι οι περιοχές που διαχωρίζονται είναι ασθενώς συνδεδεμένες μεταξύ τους και επομένως εύκολα ανιχνεύσιμες ως ξεχωριστά clusters.

Αντίθετα, η υψηλότερη τιμή της Ncut value στη δεύτερη εικόνα δείχνει την πολυπλοκότητα της υφής της και τη μεγαλύτερη διασύνδεση μεταξύ των περιοχών και υποδηλώνει ότι η εικόνα είναι πιο απαιτητική ως προς το clustering.

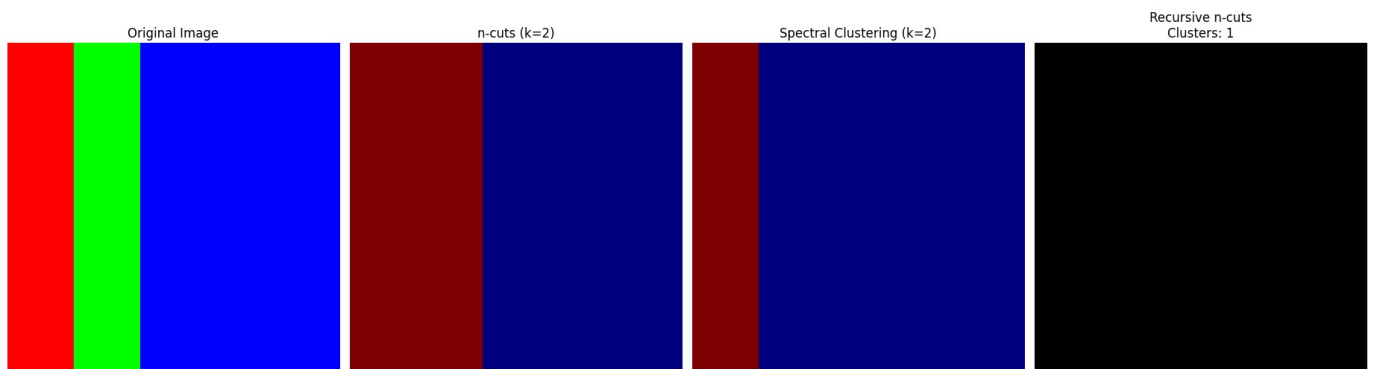
### Demo 3c

Σε αυτό το στάδιο καλούμαστε να υλοποιήσουμε τη μέθοδο αναδρομικού N-cuts.

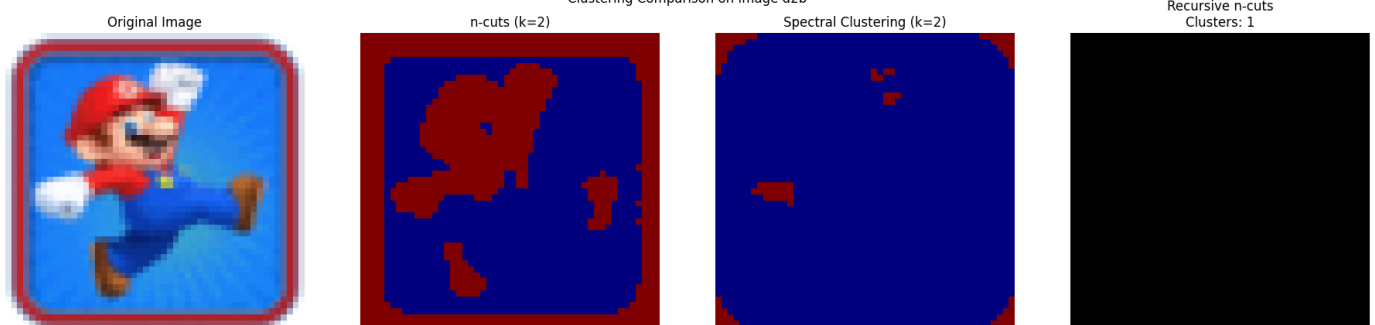
Για αρχικές τιμές των παραμέτρων  $T1=5$  και  $T2=0.2$ , παρατηρείται ότι η αναδρομή δεν προχωρά, καθώς και για τις δύο εικόνες η τιμή του Ncut στο πρώτο επίπεδο είναι ήδη μεγαλύτερη του  $T2$ , με αποτέλεσμα η διαδικασία να τερματίζεται αμέσως χωρίς κανέναν περαιτέρω διαχωρισμό.

Ενδεικτικά:

Clustering Comparison on Image d2a



Clustering Comparison on Image d2b



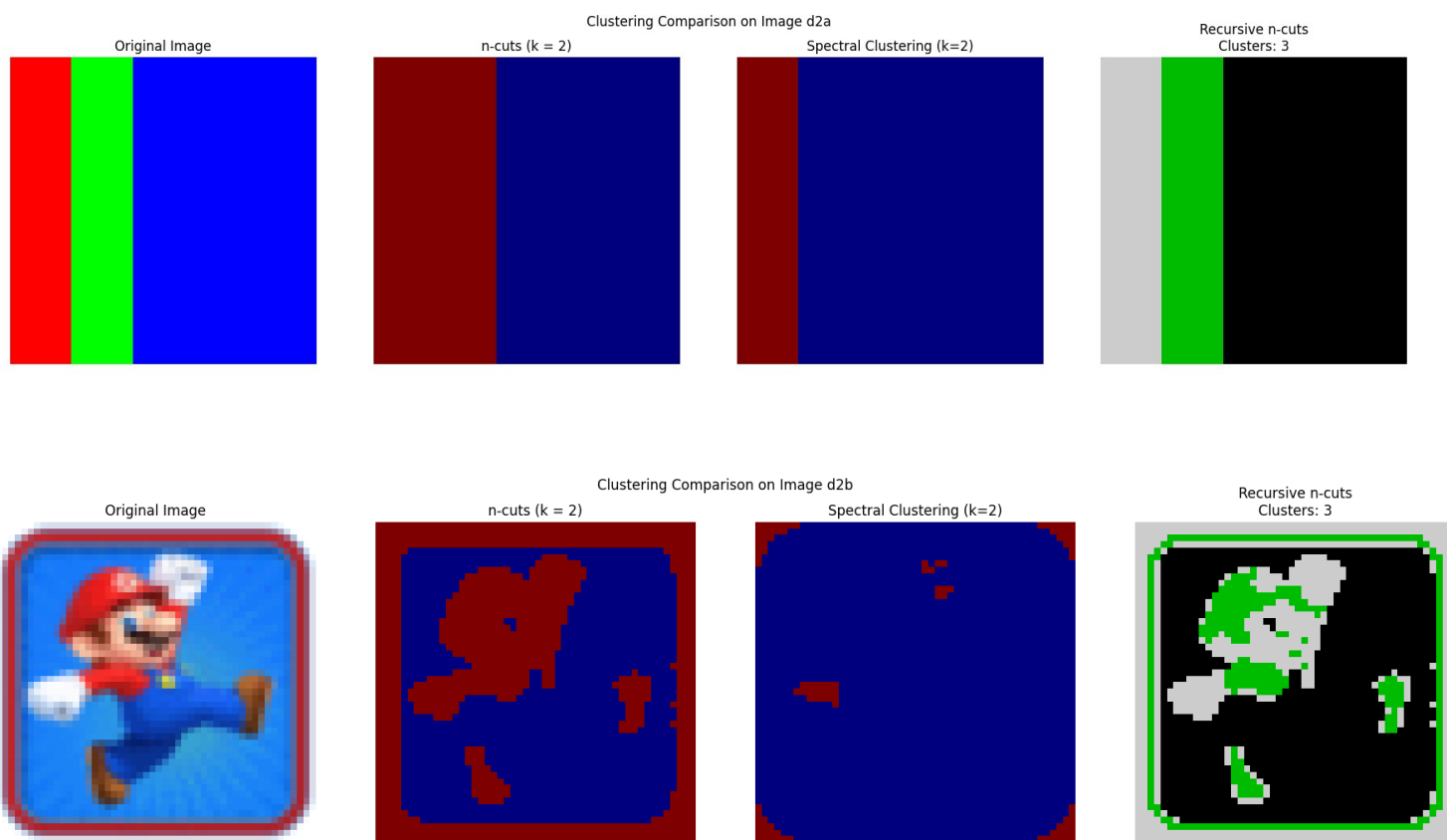
Συνεπώς, απαιτείται να δοκιμάσουμε εναλλακτικές τιμές για τα κατώφλια.

Η πρώτη εικόνα είναι σχετικά εύκολη στην επεξεργασία, καθώς εμφανίζει τρία "φυσικά" clusters, σαφώς διαχωρισμένα, μεγέθους 500, 500 και 1500 pixels αντίστοιχα.

Η δεύτερη εικόνα, αντίθετα, είναι πιο περίπλοκη, καθώς οι τιμές του Ncut στα επιμέρους βήματα βρίσκονται κυρίως στο διάστημα  $[0.7, 1.1]$ , κάτι που υποδηλώνει μεγαλύτερη δυσκολία στον διαχωρισμό περιοχών. Για αυτόν το λόγο, η επιλογή των κατάλληλων τιμών των παραμέτρων αποκτά ιδιαίτερη σημασία.

Πειραματίστηκα για διάφορες τιμές του  $T1 = \{20, 100, 500, 1000\}$  με σταθερά  $T2 = \{0.8, 0.9, 1, 1.1, 1.2\}$

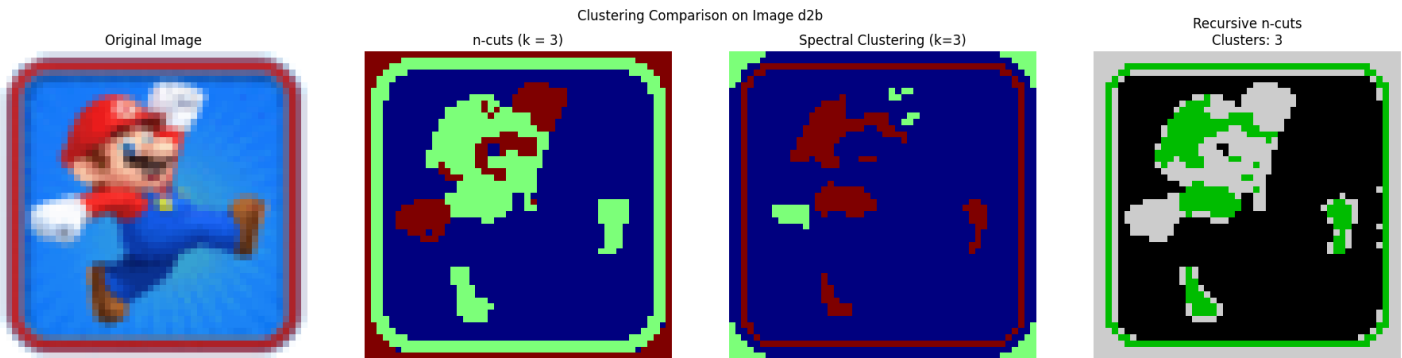
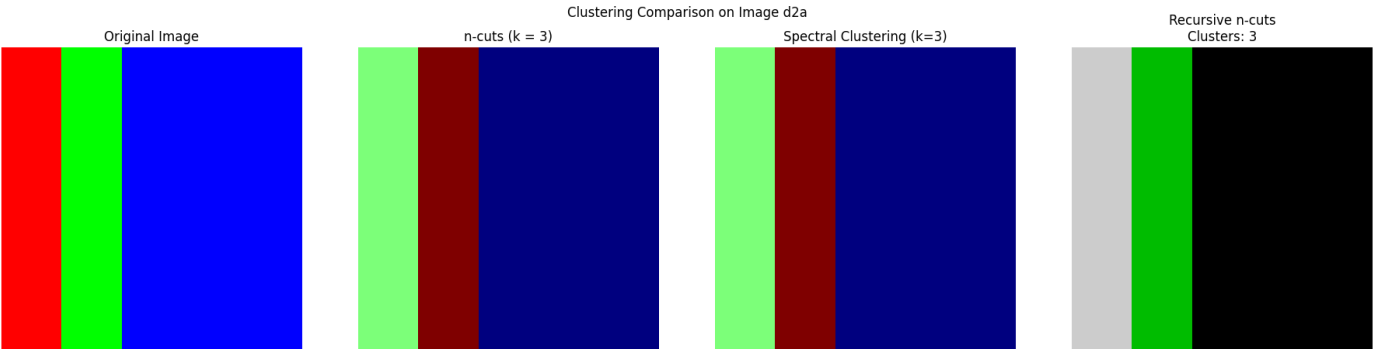
Παρακάτω φαίνονται κάποια από τα αποτελέσματα που παρουσιάζουν ενδιαφέρον.



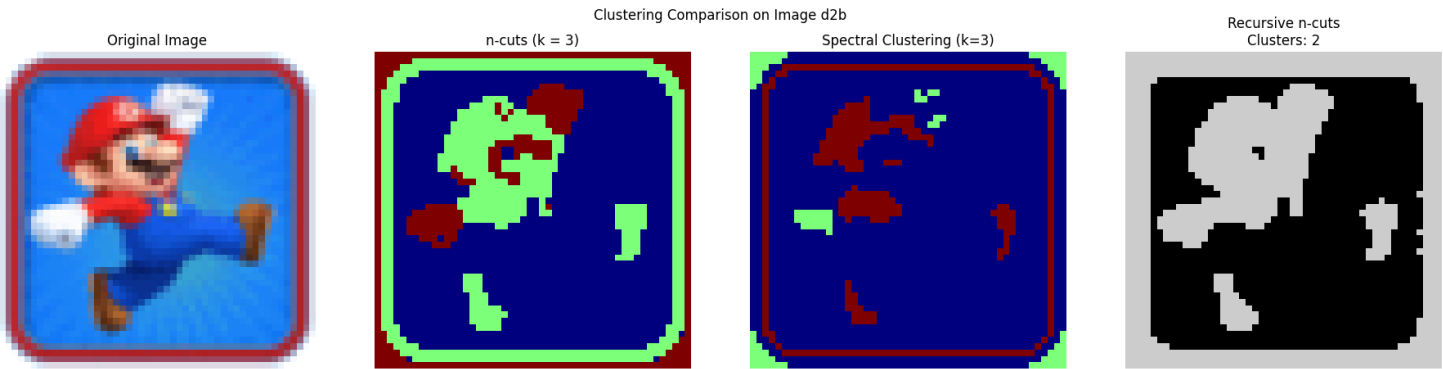
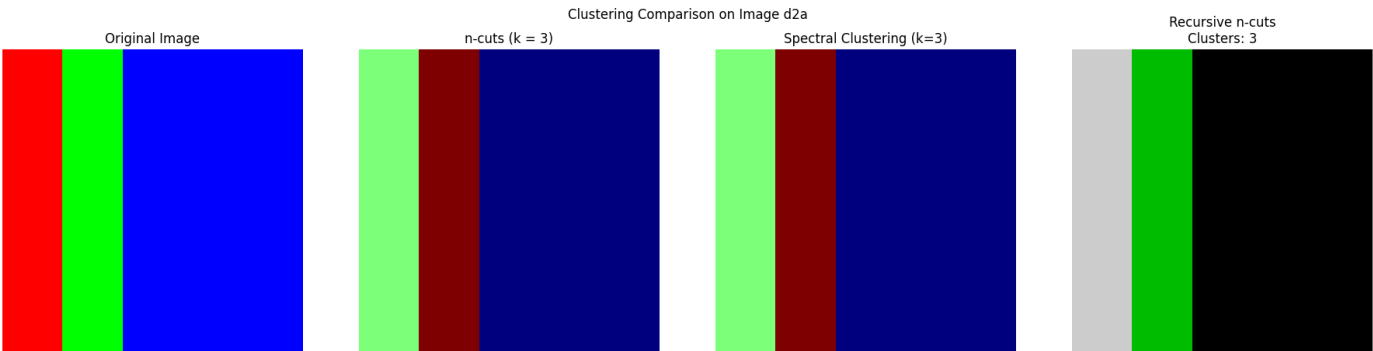
### Παρατηρήσεις

1. Στην παραπάνω περίπτωση χρησιμοποιήθηκαν  $k = 2$ ,  $T1 = 20$ ,  $T2 = 0.9$
2. Στη συνέχεια θα χρησιμοποιήσουμε τα αποτελέσματα non-recursive n-cut και spectral clustering για  $k = 3$ , καθώς παίρνουμε καλύτερα αποτελέσματα και παρουσιάζουν περισσότερο ενδιαφέρον.
3. Αφού παρουσιαστούν τα αποτελέσματα παρακάτω θα παρουσιάσουμε τα συμπεράσματά μας.

$k = 3, T1 = 100, T2 = 0.9$

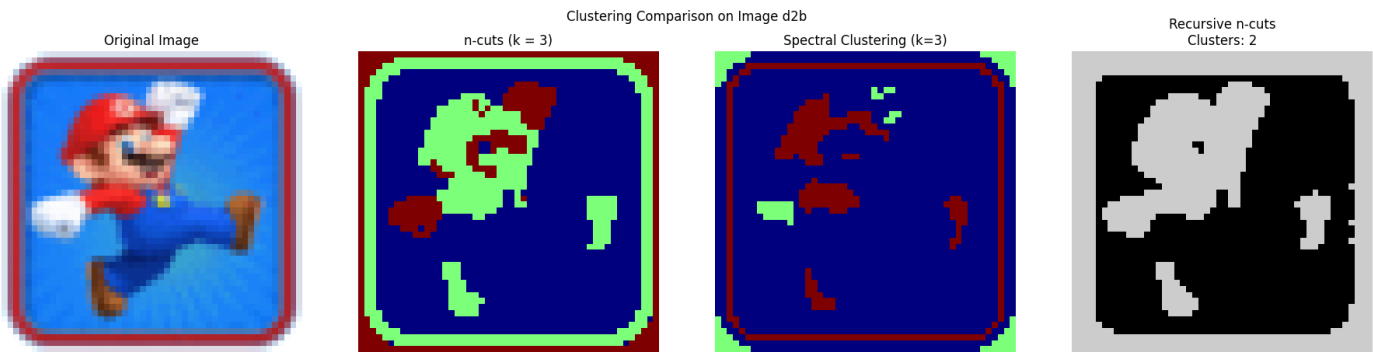
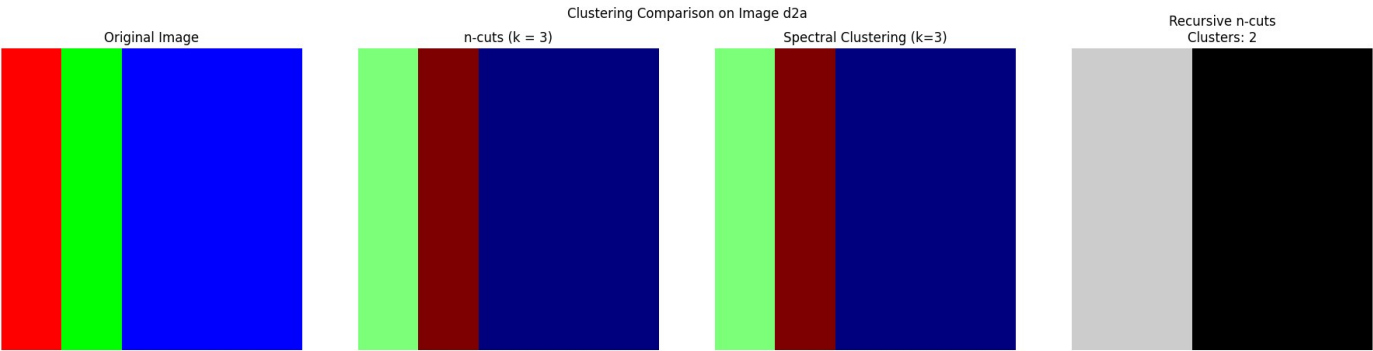


$k = 3, T1 = 500, T2 = 0.9$

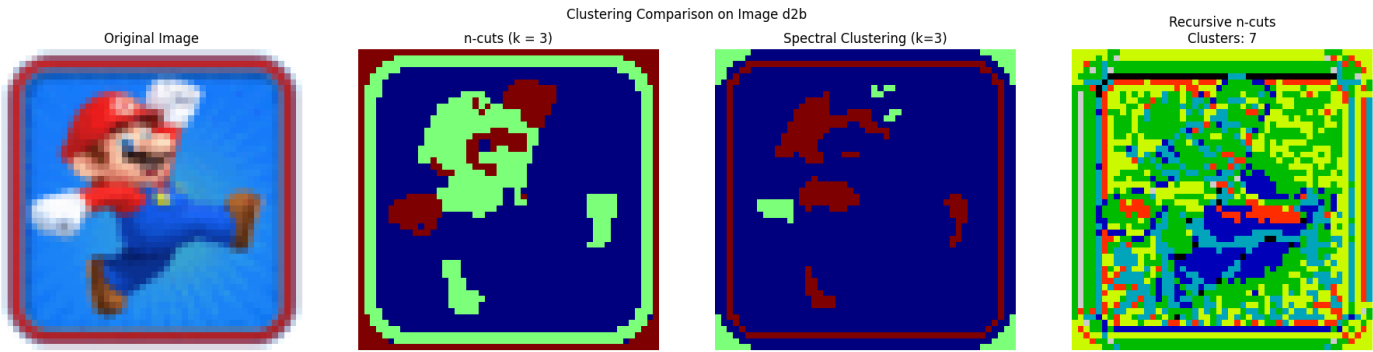
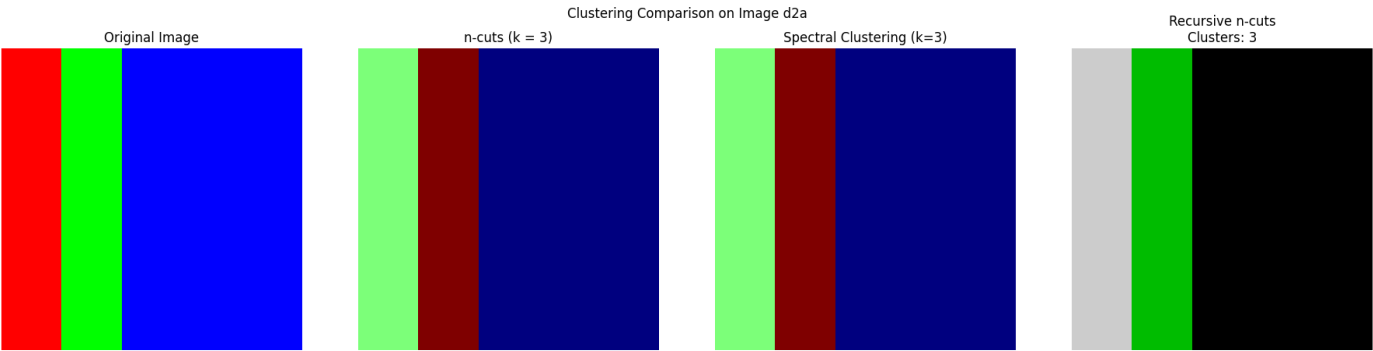




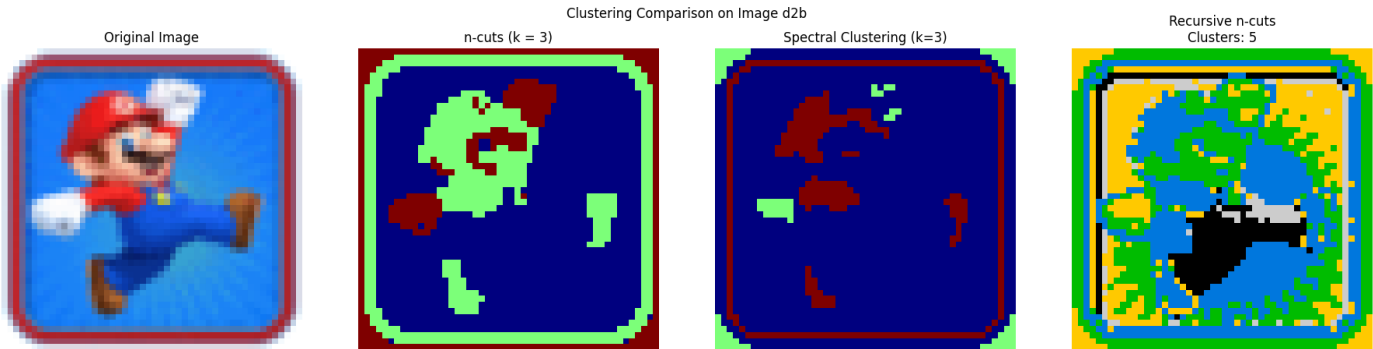
$k = 3, T_1 = 1000, T_2 = 0.9$



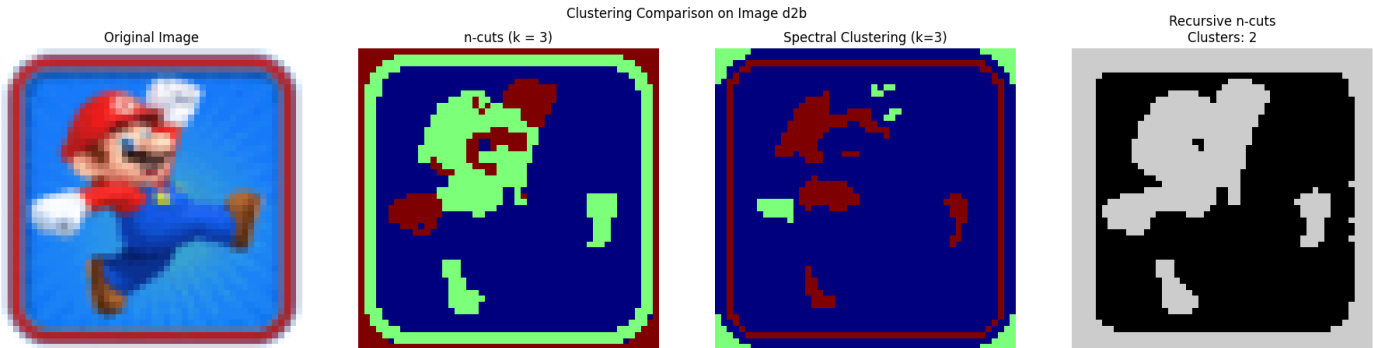
$k = 3, T_1 = 20, T_2 = 1$



$k = 3, T_1 = 100, T_2 = 1$

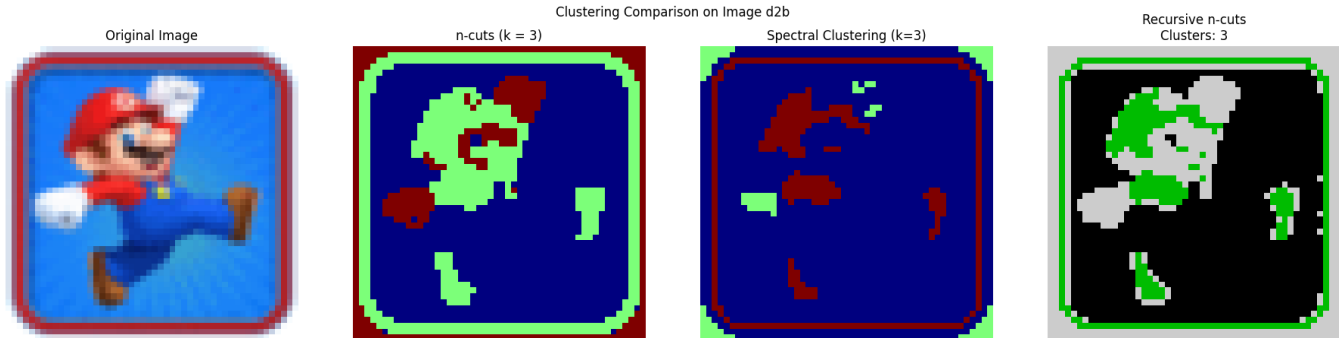
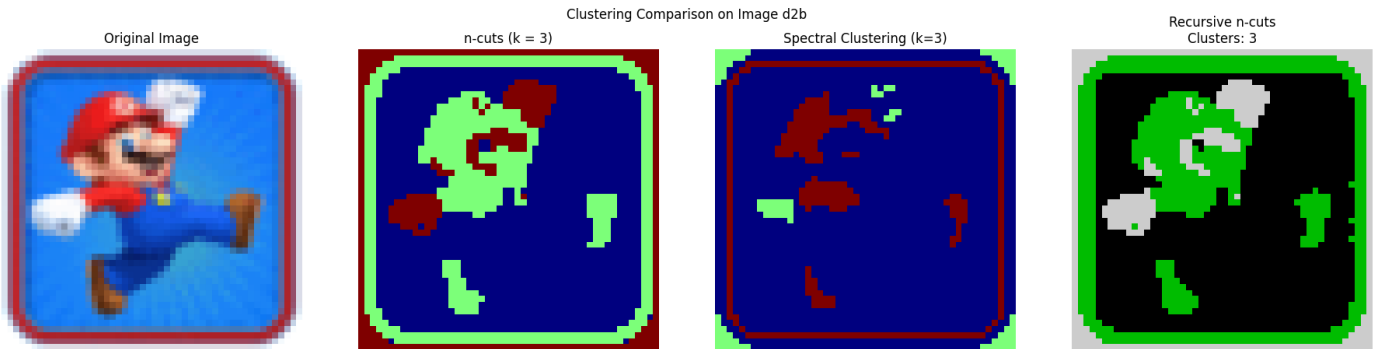


$k = 3, T_1 = 500, T_2 = 1$



$k = 3, T_1 = 20, T_2 = 0.8$

$(k = 3, T_1 = 100, T_2 = 0.9)$



## Συμπεράσματα

Για την **πρώτη εικόνα**:

- Για τιμή  $T2 = 0.9$  και  $k=3$ , παρατηρούμε ότι για τις περισσότερες τιμές του  $T1$  τα αποτελέσματα είναι παρόμοια και για τις τρεις μεθόδους. Οι λόγοι για αυτό έχουν ήδη αναλυθεί παραπάνω.
- Με την αναδρομική μέθοδο, παρατηρούμε ότι η ποιότητα του segmentation υποβαθμίζεται όταν  $T1 = 1000$  (δηλαδή όταν  $T1 > 500$ ), κάτι που ήταν αναμενόμενο λόγω του πλήθους των pixels που αντιστοιχούν σε κάθε χρωματική περιοχή.
- Για άλλες τιμές του  $T2 > 0.7$ , τα αποτελέσματα δεν διαφέρουν ουσιαστικά, οπότε δεν παρουσιάζονται περαιτέρω.

Για την **δεύτερη εικόνα**:

- Για  $T2 = 0.9$ :
  - Παίρνουμε σταθερά και αξιόπιστα αποτελέσματα για  $T1 = \{20, 100\}$ , ενώ για  $T1 = \{500, 1000\}$  η μέθοδος αποτυγχάνει να διακρίνει τις τοπικές λεπτομέρειες, με αποτέλεσμα απώλεια πληροφορίας.
- Για  $T2 = 1.0$ :
  - Όταν  $T1 = 20$ , παρατηρείται over-segmentation, δηλαδή διαχωρισμός περιοχών χωρίς πραγματική σημασία στην αρχική εικόνα.
  - Για  $T1 = 100$ , πετυχαίνουμε το καλύτερο segmentation της εργασίας, καθώς διατηρούνται πολλές λεπτομέρειες της αρχικής εικόνας με καλό διαχωρισμό των περιοχών.
  - Για υψηλότερες τιμές του  $T1$ , η μέθοδος εξακολουθεί να μην αναγνωρίζει τις τοπικές λεπτομέρειες, αναμενόμενο καθώς γενικότερα δεν υπάρχουν στην εικόνα πολλές μεγάλες ισχυρά συσχετισμένες περιοχές.
- Για  $T2=0.8$  και  $T1 = 100$ :
  - Η διαμέριση είναι παρόμοια με εκείνη για  $T2 = 0.9$ , με 3 τελικά clusters. Παρόλα αυτά, με το συγκεκριμένο παράδειγμα διαπιστώνουμε για άλλη μια φορά ότι με την αύξηση του  $T2$  η μέθοδος αποδίδει μεγαλύτερη σημασία ακόμη και σε μικρές τοπικές διαφοροποιήσεις.