



# ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών  
Υπολογιστών

## «Image Segmentation»

2<sup>η</sup> Εργασία στο μάθημα

«Ψηφιακή Επεξεργασία Εικόνας»

της φοιτήτριας

**Κελέση Ελπίδας, ΑΕΜ : 9410**

**Διδάσκοντες : Καθ. Α. Ντελόπουλος, Κ. Κυρίτσης, Α. Παπαδόπουλος**

**Υπεύθυνος εργασίας : Καθ. Α. Ντελόπουλος**

Θεσσαλονίκη, Εαρινό εξάμηνο 2020-2021

## Πίνακας περιεχομένων

<b>1 Εισαγωγή</b> .....	<b>Σελ 2</b>
<b>2. Spectral Clustering</b> .....	<b>Σελ 4</b>
2.1 Κώδικας MATLAB .....	Σελ 4
2.2 Αποτελέσματα .....	Σελ 4
<b>3. Εικόνες ως γράφοι</b> .....	<b>Σελ 5</b>
3.1 Κώδικας MATLAB .....	Σελ 5
3.2 Αποτελέσματα .....	Σελ 5
<b>4. Normalized Cuts</b> .....	<b>Σελ 6</b>
4.1 Κώδικας MATLAB .....	Σελ 6
4.2 Αποτελέσματα .....	Σελ 7
<b>5. Βιβλιογραφία</b> .....	<b>Σελ 9</b>

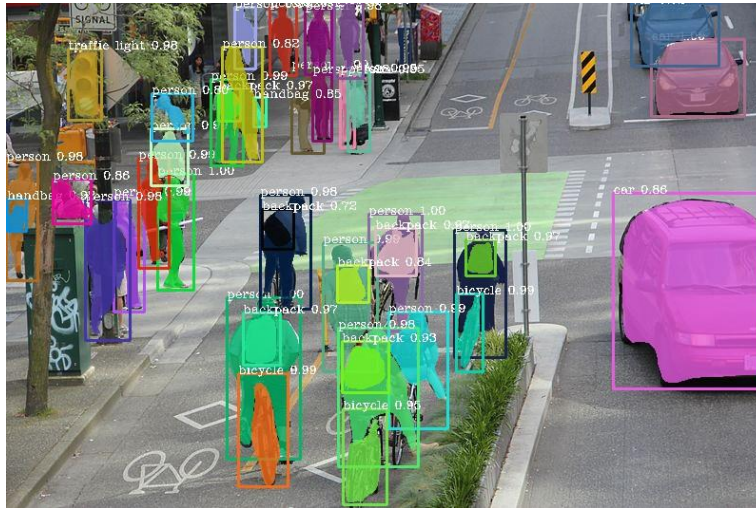
## 1. Εισαγωγή

Με τον όρο image segmentation, στην ψηφιακή επεξεργασία εικόνας και στην όραση του υπολογιστή, εννοούμε τη διαδικασία διαμέρισης μιας ψηφιακής εικόνας σε πολλαπλά τμήματα (σύνολα εικονοστοιχείων-pixels). Ο στόχος της τμηματοποίησης είναι να απλοποιήσει ή και να αλλάξει την αναπαράσταση μιας εικόνας σε κάτι που είναι πιο εύκολο να αναλυθεί. Η τμηματοποίηση εικόνας χρησιμοποιείται συνήθως για τον εντοπισμό αντικειμένων και ορίων (γραμμές, καμπύλες κ.λπ.) στις εικόνες. Πιο συγκεκριμένα, η τμηματοποίηση εικόνας είναι η διαδικασία εκχώρησης μιας ετικέτας σε κάθε εικονοστοιχείο σε μια εικόνα έτσι ώστε τα εικονοστοιχεία με την ίδια να έχουν ορισμένα χαρακτηριστικά.

Το αποτέλεσμα της τμηματοποίησης εικόνας είναι ένα σύνολο τμημάτων που καλύπτουν συλλογικά ολόκληρη την εικόνα ή ένα σύνολο περιγραμμάτων που εξάγονται από την εικόνα. Κάθε ένα από τα εικονοστοιχεία σε μια περιοχή είναι παρόμοιο σε σχέση με κάποια χαρακτηριστική ή υπολογισμένη ιδιότητα, όπως χρώμα, ένταση ή υφή. Οι παρακείμενες περιοχές έχουν σημαντικά διαφορετικό χρώμα σε σχέση με τα ίδια χαρακτηριστικά. Όταν εφαρμόζεται σε μια στοίβα εικόνων, τυπική στην ιατρική απεικόνιση, τα περιγράμματα που προκύπτουν μετά την τμηματοποίηση των εικόνων μπορούν να

χρησιμοποιηθούν για τη δημιουργία τρισδιάστατων ανακατασκευών με τη βοήθεια αλγορίθμων παρεμβολής.

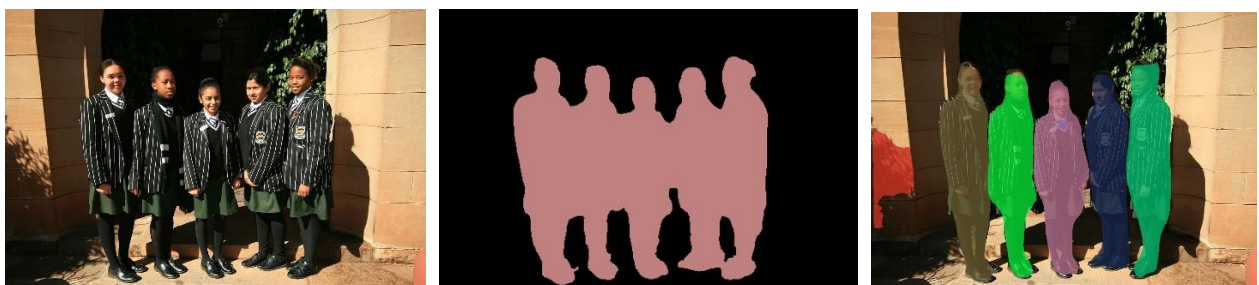
Οι εφαρμογές γενικότερα, του image segmentation είναι πολλαπλές και αυξάνονται καθημερινά. Ενδεικτικά κάποιες από αυτές είναι η ανάκτηση εικόνας βάση περιεχομένου, η μηχανική όραση, εικονική προσομοίωση εγχείρησης, ανίχνευση προσώπου, driverless αυτοκίνητα.



Εικόνα 1: Πηγή : <https://towardsdatascience.com/image-segmentation-with-six-lines-Of-code-acb870a462e8>

Υπάρχουν δύο βασικοί τύποι τμηματοποίησης εικόνας:

- ✓ Semantic Segmentation: Τα αντικείμενα που ταξινομούνται με τις ίδιες τιμές εικονοστοιχείων(pixel) σηματοδοτούνται με τα ίδια colormaps.
- ✓ Instance Segmentation: Διαφέρει από την semantic segmentation, επειδή διαφορετικές εμφανίσεις του ίδιου αντικειμένου τμηματοποιούνται με διαφορετικούς χρωματικούς χάρτες.



Real Image

Semantic Segmentation

Instance Segmentation

Εικόνα 2: Πηγή : <https://towardsdatascience.com/image-segmentation-with-six-lines-Of-code-acb870a462e8>

## 2. Spectral Clustering

### 2.1 Κώδικας MATLAB

Στο πρώτο ερώτημα υλοποιείται η πραγματοποίηση 3 πειραμάτων με βάση τη mySpectralClustering, η οποία έχει ως ορίσματα τον affinity πίνακα d1a και τον αριθμό k cluster, οποίος παίρνει τιμές k = 2,3,4. Οπότε στο demo 1 υλοποιήθηκε η εξής συνάρτηση:

**MySpectralClustering(anAffinityMat, k):** Η συνάρτηση αυτή υλοποιεί το spectral clustering της εικόνας που αντιστοιχεί στον τετράγωνο και συμμετρικό affinity πίνακα (MN x MN πίνακας που περιγράφει την εικόνα σας γράφο) για k clusters. Συγκεκριμένα, λύνει το πρόβλημα των ιδιοτιμών  $Lx = \lambda x$  όπου L είναι ο Λαπλασιανός πίνακας του affinity πίνακα. Στην συνάρτηση αρχικά υπολογίστηκε ο διαγώνιος πίνακας D, για τον οποίο ισχύει  $D(i,j) = \sum_j W(i,j)$ , όπου W είναι ο affinity πίνακας που περιγράφει τον γράφο. Στη συνέχεια υπολογίστηκε ο Laplace του πίνακα W και μετά μέσω της eig που έχει ορίσματα (L,k,'sm') λύσαμε την εξίσωση  $Lx = \lambda x$  και επιστρέψαμε τα πρώτα k eig vectors με τις μικρότερες ιδιοτιμές και τα ιδιοδιανύσματα που αντιστοιχούν σε αυτές, και τα τοποθετήσαμε στον clusterIdx (άρα έγινε διάνυσμα που περιέχει τον αριθμό k κάθε MN pixel της εικόνας).

### 2.2 Αποτελέσματα

Από το demo 1 προκύπτει ο εξής πίνακας :

pixels	k=2	k=3	k=4
1	2	1	1
2	2	1	1
3	2	1	1
4	2	1	1
5	1	2	2
6	1	2	3
7	1	2	3
8	1	2	3
9	2	3	4
10	2	3	4
11	2	3	4
12	2	3	4

Από τον affinity πίνακα μπορούμε να συμπεράνουμε ότι τα στοιχεία 1-4 και 9-12, καθώς και 5-8 έχουν κοντινές μεταξύ τους χρωματικές τιμές και άρα αυξημένη τιμή στον πίνακα. Αυτό αντικατοπτρίζεται και στις ετικέτες για k=3 για 1-4, 5-8, 9-12. Για k=4 βλέπουμε ότι η ομάδα στοιχείων 5-8 χωρίστηκε σε 2 υπο-clusters με ετικέτες 2 και 3.

### 3. Αναπαράσταση εικόνων σαν γράφους

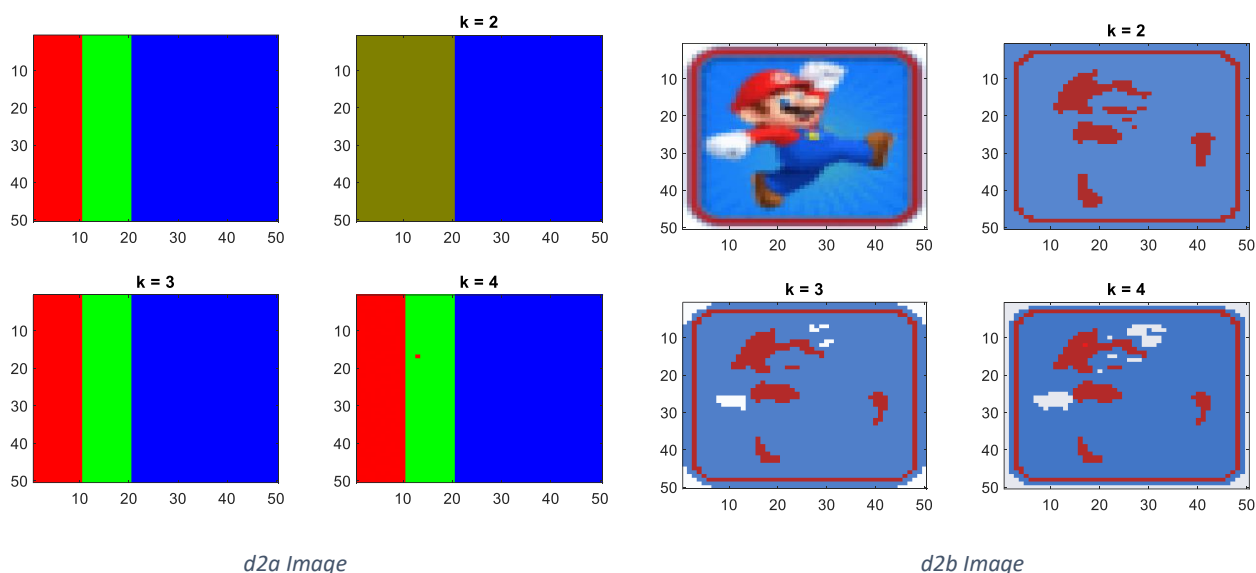
#### 3.1 Κώδικας MATLAB

Στο δεύτερο ερώτημα υλοποιείται η πραγματοποίηση 6 πειραμάτων με βάση την Image2Graph και δημιουργείται ένας affinity πίνακας των εικόνων d2a, d2b με την πραγματοποίηση του mySpecialClustering για  $k=2,3,4$ . Οπότε στο demo 2 υλοποιήθηκε η εξής συνάρτηση:

**Image2Graph (imIn):** Η συνάρτηση αυτή υπολογίζει τον affinity γράφο της εικόνας imIn ( $M \times N$  εικόνα εισόδου με  $N$  κανάλια) που περιγράφει μέσω βαρών πόσο όμοια είναι 2 pixels μεταξύ τους. Η ομοιότητά τους μετρείται ως η απόσταση της τιμής του χρώματος μεταξύ δύο pixel. Ουσιαστικά, η απόσταση ενός διάνυσματος  $x$  από το διάνυσμα  $y$ , εκφράζεται μέσω της σχέσης:  $x-y = \sqrt{x^2 + y^2 - 2xy}$ . Βέβαια στο πρόβλημα που μελετάμε το διάνυσμα  $x$  είναι ίδιο με το  $y$ . Οπότε στη συνάρτηση αρχικά υπολογίζουμε τις διαστάσεις και τα κανάλια της εικόνας και μετατρέπουμε την 2D εικόνα διαστάσεων  $N \times N \times 3$  σε διάνυσμα imVec διαστάσεων  $1 \times N \times N \times 3$ , ώστε να διατηρηθεί η σωστή σειρά των pixel μετά την αναδιαμόρφωση. Στη συνέχεια μέσω της  $v2v(\text{vec})$  υπολογίζεται μέσω απλών πράξεων η απόσταση  $x-y$  και εφαρμόζεται ο τύπος του affinity πίνακα με  $1/(e^{(x-y)})$ .

#### 3.2 Αποτελέσματα

Από το demo 2 προκύπτουν τα εξής :



Η εικόνα d2a βλέπουμε ότι αποτελείται από 3 χωρισμένες ομάδες χρωμάτων. Για  $k=2$  τα οι πρώτες 2 λωρίδες της εικόνας ομαδοποιήθηκαν σε ένα cluster, ενώ η τρίτη λωρίδα αποτέλεσε έναν ξεχωριστό

όπως ήταν αναμενόμενο από τον πίνακα που προέκυψε στο demo 1. Για  $k=3$ , όπως είναι λογικό κάθε λωρίδα αποτελεί έναν cluster και για  $k=4$ , όπως και στο demo 1, επειδή στο 5-8 εμφανίζονται δύο τιμές για το  $k$  (το οποίο προκύπτει επειδή το  $k$  είναι μεγαλύτερο από τον αριθμό των λωρίδων), η δεύτερη αλλοιώνεται ελάχιστα όπως βλέπουμε με μία πολύ μικρή κόκκινη κουκίδα, για να χωρέσει και το 4ο cluster.

Στην εικόνα d2b που δεν είναι τόσο ξεκάθαρος ο διαχωρισμός, παρατηρούμε ότι όσο μεγαλώνει το  $k$ , τόσο περισσότερες λεπτομέρειες στην εικόνα αποκτούν τον δικό τους cluster.

Για παράδειγμα, για  $k=2$  ο ένας cluster περιέχει τα στοιχεία με κόκκινο χρώμα (καπέλο μπλούζα) και ο άλλος τα υπόλοιπα μπλε κυρίως παντελόνι και background. Για  $k=3$ , παρατηρούμε ότι εμφανίζονται στοιχεία του λευκού (γάντια) ως ξεχωριστό cluster. Και τέλος για  $k=4$  ο λευκός cluster αποκτάει κάπως καλύτερη μορφή (προστίθενται κάτι από μάτια και λάμψεις του προσώπου).

## 4. Normalized Cuts

### 4.1 Κώδικας MATLAB

Στο τρίτο ερώτημα υλοποιείται η εφαρμογή της αναδρομικής και μη, μεθόδου n-cut για την κατάτμηση των εικόνων d2a, d2b καθώς και η υλοποίηση των συναρτήσεων myNCuts και calculateNCut και η σύγκριση των αποτελεσμάτων τους με τα αποτελέσματα από τη mySpecialClustering. Οπότε στο demo 3 υλοποιήθηκαν οι εξής συναρτήσεις:

**myNCuts(anAffinityMat ,k):** Αυτή η συνάρτηση υπολογίζει non-recursive normalized cuts αλγόριθμο για image clustering με δεδομένο τον affinity πίνακα για  $k$  clusters. Επομένως η συνάρτηση αυτή, υπολογίζει ότι υπολόγισε και η mySpectralClustering, με ίδια βήματα. Η μόνη διαφορά εντοπίζεται στο γεγονός ότι στην συγκεκριμένη περίπτωση οι ζητούμενες ιδιοτιμές προέκυψαν από την λύση του γενικευμένου προβλήματος ιδιοτιμών  $Lx = \lambda Dx$ .

**calculateNcut(anAffinityMat, clusterIdx):** Αυτή η συνάρτηση υπολογίζει τον αλγόριθμο normalized 2cuts για image clustering με  $k=2$ , προκειμένου να βαθμολογηθεί εάν το split ήταν καλό ή όχι και να αποφασίσει να σταματήσει ή όχι ανάλογα. Ουσιαστικά, η συνάρτηση αυτή πραγματοποιεί την διαδικασία υπολογισμού της τιμής nCut. Αρχικά, από τον affinity πίνακα υπολογίζονται τα αθροίσματα των γραμμών και στην συνέχεια το assocAV είναι το άθροισμα των αθροισμάτων που αντιστοιχούν στα στοιχεία της ομάδας A (ετικέτα = 1). Αντίστοιχα υπολογίζεται και η τιμή assocBV. Για τις τιμές assocAA και assocBB, υπολογίζεται το άθροισμα όλων των βαρών των υποπινάκων του affinity που αποτελούνται από τα στοιχεία που ανήκουν μόνο στην ομάδα A ή B αντίστοιχα. Για την τιμή nCut χρησιμοποιείται ο τύπος  $N_{assoc} = \text{assocAA}/\text{assocAV} + \text{assocBB}/\text{assocBV}$ .

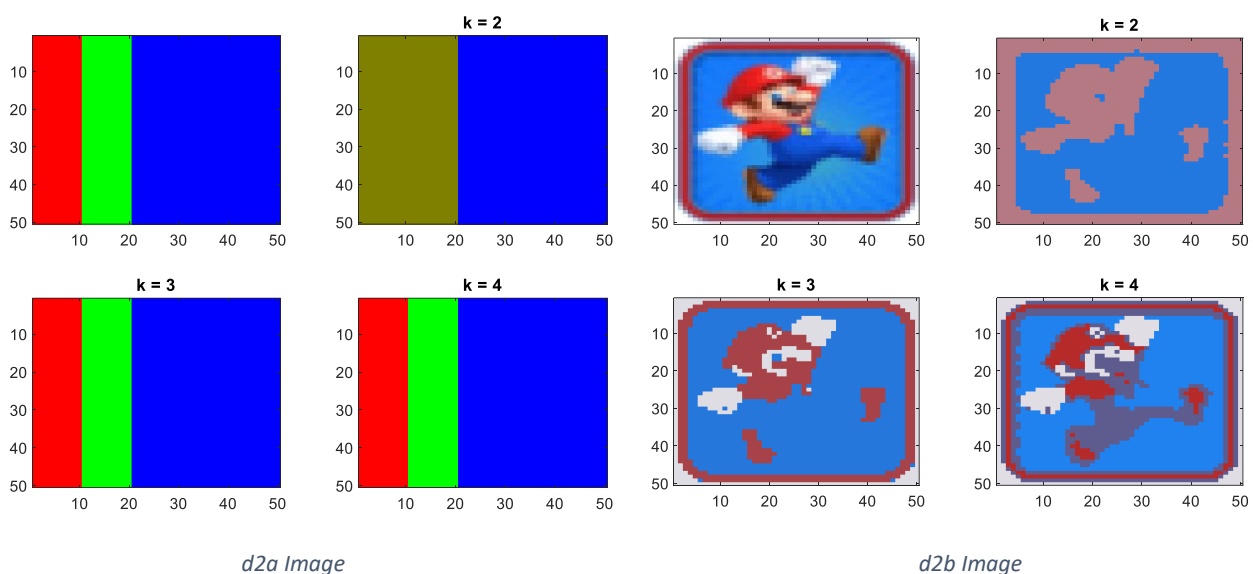
**iterNCuts(anAffinityMat, threshold):** Αυτή η συνάρτηση υπολογίζει τον recursive normalized cuts για image clustering, με threshold να είναι ένας  $1 \times 2$  πίνακας που καθορίζει τις συνθήκες για να τερματίσει το πρόγραμμα. Αρχικά, γίνεται διαχωρισμός της εικόνας με την myNCuts και υπολογίζεται η τιμή ncut. Στην συνέχεια, λαμβάνονται 2 υποπίνακες του affinity, ένας που αντιστοιχεί στα στοιχεία με ετικέτα 1 και ο άλλος στα στοιχεία με ετικέτα 2 και καλείται αναδρομικά για κάθε υποπίνακα η

συνάρτηση `iter`. Ως συνθήκη τερματισμού ελέγχονται το πλήθος των στοιχείων με ετικέτα 1 και 2, καθώς και η τιμή `ncut`. Σε κάθε βήμα της αναδρομής, οι ετικέτες που επιστρέφονται από την `iter` αποθηκεύονται στον πίνακα εξόδου `labs` στις θέσεις που αντιστοιχούν στα στοιχεία των υποπινάκων του `affinity`.

**result(imIn, labels):** Αυτή η συνάρτηση εξυπηρετεί τον σκοπό του `visualization` του `clustering` αποδίδοντας στα `pixel` του καθενός τον μέσο όρο των χρωμάτων των `pixel` του συν κάποια τυχαία ποσότητα χρώματος.

## 4.2 Αποτελέσματα

Από το demo 3α προκύπτουν τα εξής :

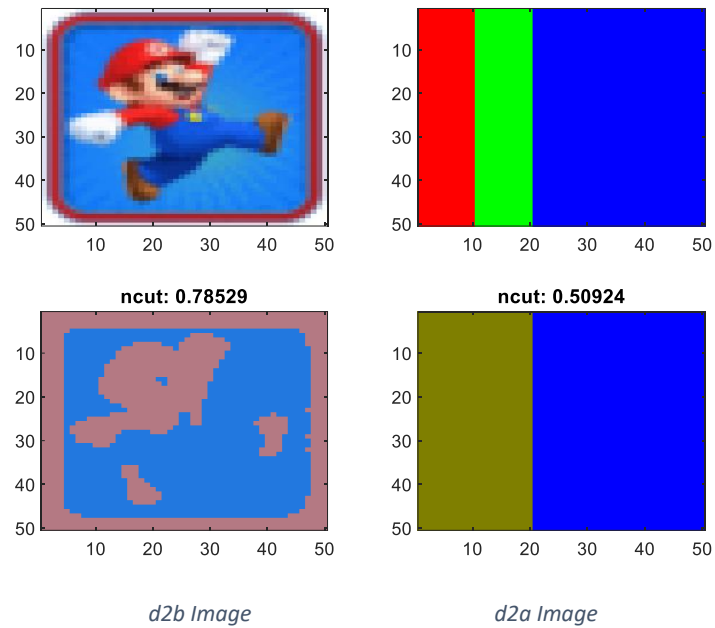


Σχετικά με την *d2a* εικόνα η μόνη διαφορά με την μέθοδο `spectral clustering` εντοπίζεται για  $k=4$ , όπου δεν υπάρχει καμία αλλοίωση στην 2<sup>η</sup> λωρίδα.

Ενώ για την *d2b* εικόνα προκύπτει ότι οι `clusters` και στις 3 περιπτώσεις έχουν διαφορές. Ειδικά για  $k=4$ , όπου σε σχέση με το `spectral clustering` τα `clusters` είναι καλύτερα διαμοιρασμένα και πιάνουν περισσότερα χαρακτηριστικά. Συγκεκριμένα για  $k=4$  ο αλγόριθμος κατάφερε να εντοπίσει όλη την εικόνα.

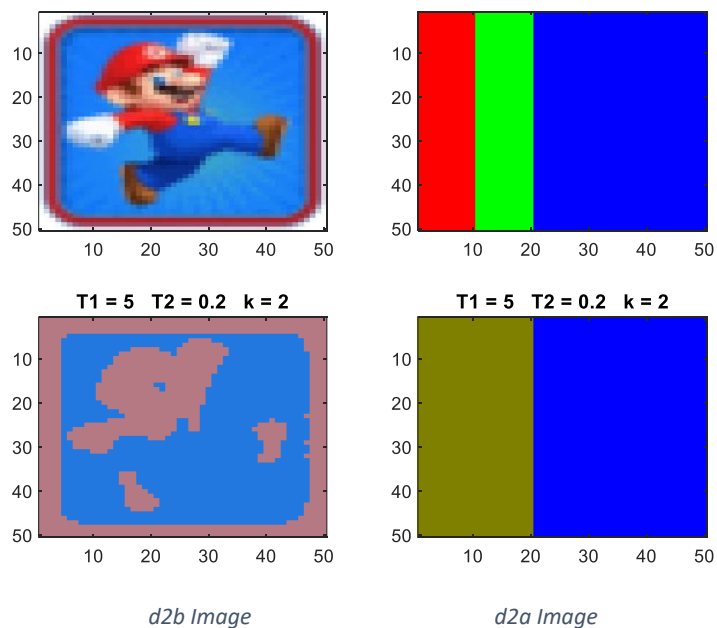
Επομένως η μέθοδος `n-cut` δίνει καλύτερα αποτελέσματα από την `spectral clustering`.

Από το demo 3b προκύπτουν τα εξής :



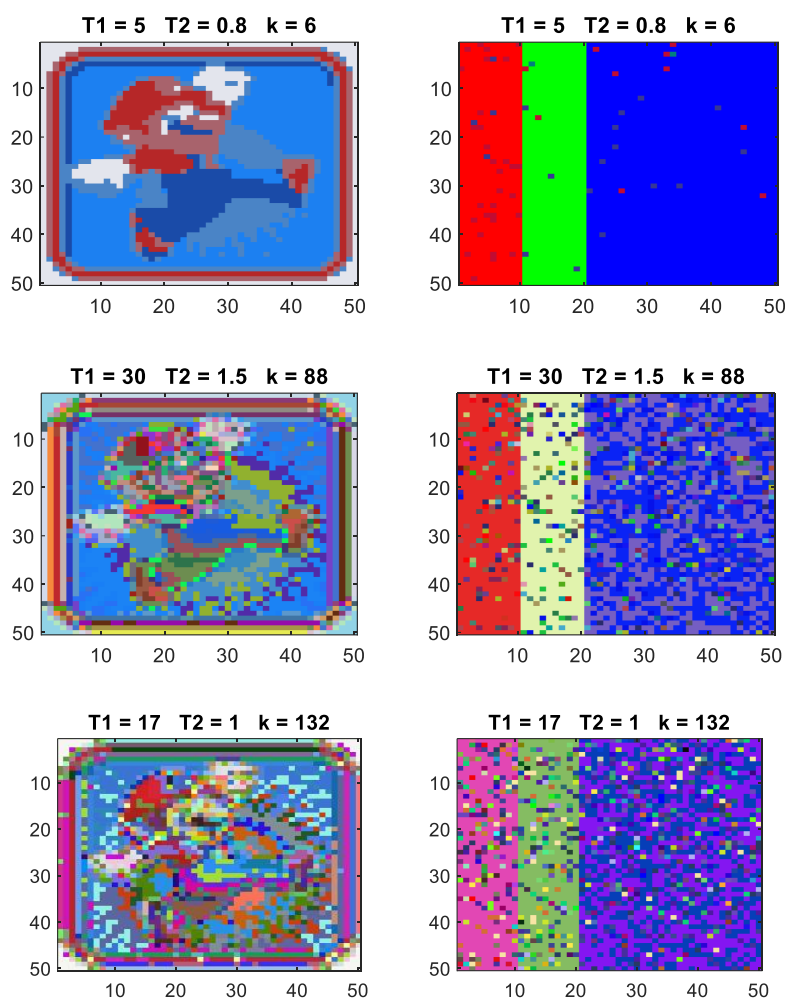
Παρατηρούμε ότι για την εικόνα d2b παρατηρούμε ότι οι cluster είναι διαφορετικοί σε σχέση με τους αντίστοιχους του spectral clustering ενώ για την εικόνα d2a δεν παρατηρούνται διαφορές.

Από το demo 3c προκύπτουν τα εξής :





Δοκιμάζοντας αναδρομικά έχουμε :



Η σύγκριση της απλής Ncuts με την αναδρομική της εκδοχή είναι πιο δύσκολη. Αυτό που μπορούμε να πούμε είναι ότι η απλή εκδοχή τείνει να δημιουργεί πιο ενωμένα κομμάτια ενώ η αναδρομική αντίθετα βρίσκει πιο πολλές διαφορές τμηματικά αλλά αδυνατεί να σχηματίσει μία τόσο ξεκάθαρη διαφοροποίηση τουλάχιστον ολικά – οπτικά.

## 5. Βιβλιογραφία

Papers :

Normalized Cuts and Image Segmentation : Jianbo Shi University of Pennsylvania, Jitendra Malik University of California, Berkeley

Ηλεκτρονικές πηγές :

[https://en.wikipedia.org/wiki/Image\\_segmentation](https://en.wikipedia.org/wiki/Image_segmentation)

<https://towardsdatascience.com/image-segmentation-with-six-lines-of-code-acb870a462e8>