



Universitat de les
Illes Balears



Treball Final de Grau

Comparació objectiva d'algorismes de segmentació d'imatges

MARGALIDA SALOM RAMIS

Tutors

Sebastià Massanet Massanet

Arnau Mir Torres

Escola Politècnica Superior
Universitat de les Illes Balears
Palma, 30 de juny de 2017

ÍNDIX

Índex	i
Índex de figures	iii
Resum	v
1 Introducció	1
2 Segmentació	5
2.1 Conceptes bàsics	5
2.2 Segmentació d'imatges	6
2.3 Tècniques de segmentació	8
2.3.1 Tècniques basades en l'espai de característiques	8
2.3.2 Tècniques basades en el domini de la imatge	9
2.3.3 Intel·ligència artificial	11
2.3.4 Tècniques de segmentació estadístiques	11
3 Mètodes de segmentació	13
3.1 Mètode basat en el model lingüístic de dues tuples	13
3.1.1 Conceptes i resultats preliminars	13
3.1.2 Algorisme	16
3.1.3 Modificacions de l'algorisme	24
3.2 Mètode de segmentació jeràrquica	24
3.2.1 Quantificació del color	25
3.2.2 Segmentació espacial	27
3.3 K-means	29
3.4 Fuzzy k-means	32
3.5 Recursive shortest spanning tree	34
4 Entorn experimental	37
4.1 Mesures objectives	37
4.1.1 Índex de Rand	37
4.1.2 Variació de la informació	38
4.2 Base d'imatges i ground truth	40
4.2.1 Imatge de consens	41
4.3 Resultats	42
4.4 Estudi estadístic: test de Wilcoxon	45

5 Conclusions	49
A Annexos	51
A.1 Model lingüístic de 2-tuples	51
A.2 K-means	61
A.3 Fuzzy k-means	62
A.4 Recursive shortest spanning tree	63
A.5 Avaluació de les segmentacions	65
Bibliografia	69

ÍNDIX DE FIGURES

2.1	Tipus de xarxes per digitalitzar imatges.	6
2.2	Funció de cost segons el nombre de regions.	8
3.1	Exemple d'associació de les 2-tuples a les dades d'un expert.	22
3.2	Resultats obtinguts per a cada color del mètode 2-tuples segons un expert.	24
3.3	Diferents mapes de classe del mètode de segmentació jeràrquica.	26
3.4	Exemple de segmentació dels mapes de classe segons l'algorisme de segmentació jeràrquica.	27
3.5	Resultats de cada passa de l'algorisme de segmentació jeràrquica.	29
3.6	Resultats de l'algorisme k-means segons el nombre de clústers.	31
3.7	Resultats de l'algorisme k-means variant el nombre d'iteracions realitzades.	32
3.8	Resultats de l'algorisme fuzzy k-means variant el nombre d'iteracions.	34
3.9	Diferència entre les tècniques del mètode RSST i k-means.	36
3.10	Resultats obtinguts segons el nombre de regions en el mètode RSST.	36
4.1	Segmentacions fetes per humans d'una imatge.	41
4.2	Comparació dels resultats obtinguts pels diferents algorismes de segmentació.	43
4.3	Comparació del model dels resultats obtinguts per l'algorisme de 2-tuples lingüístiques original i el de 2-tuples modificat.	45
4.4	Comparació de les segmentacions per diferents escales del mètode de segmentació jeràrquic.	45
4.5	Mètodes que obtenen millors i pitjors resultats per algunes imatges.	46

RESUM

El processament d'imatges és un tema de gran interès i que està present en molts àmbits de la societat. Per tal de ressaltar les característiques d'una imatge o simplificar la seva estructura per facilitar-ne la comprensió, s'han desenvolupat diferents tècniques i mètodes automàtics no supervisats. Entre els diferents mètodes que existeixen per processar una imatge, en aquest treball s'analitzarà el procés de segmentació d'imatges.

Segmentar és l'acte que consisteix en dividir una imatge en diferents subconjunts o regions disjunctes de tal forma que la seva unió forma tota la imatge. Per dur a terme aquest procés existeixen una gran diversitat de tècniques, que es poden dividir en les basades en l'espai de característiques, les basades en el domini de la imatge, les que utilitzen intel·ligència artificial i les tècniques de segmentació estadístiques. En aquest treball, una vegada formalitzat el concepte de segmentació d'una imatge, es proposa una classificació de les tècniques més emprades en segmentació.

Per posar de manifest algunes d'aquestes tècniques, s'analitzen en detall i s'implementen els mètodes de segmentació d'imatges basats en el model lingüístic de dues tuples, el de segmentació jeràrquica, k-means, fuzzy k-means i recursive shortest spanning tree (RSST).

En el capítol 4 s'han aplicat els diferents algorismes a una base d'imatges per fer una comparació visual i una quantitativa dels diferents mètodes, utilitzant l'índex de Rand i el mètode de variació de la informació. Per fer la comparació quantitativa, ha estat necessari tenir una imatge ideal amb la que comparar la segmentació obtinguda amb els mètodes anteriorment estudiats. Per això s'ha adoptat una imatge de consens d'entre les segmentacions ideals que es poden trobar a la base d'imatges considerada. Finalment utilitzant el test estadístic de Wilcoxon, es determina si els resultats obtinguts són estadísticament significatius o no.

Finalment no s'ha arribat a una unanimitat respecte quin és el millor i el pitjor mètode de segmentació, donat que les dues mesures quantitatives considerades no coincideixen en les conclusions obtingudes. Segons l'estudi realitzat, els millors serien el de segmentació jeràrquica considerant escales grans mentre que els pitjors serien el mètode de segmentació jeràrquica considerant escales petites i el fuzzy k-means. Així doncs, en resum, no existeix un mètode de segmentació que sigui òptim per totes les imatges i segons tots els criteris de rendiment possibles. Això fa que el camp de la segmentació d'imatges sigui, encara avui en dia, un camp obert amb molta investigació per realitzar.

INTRODUCCIÓ

El processament d'imatges és des de fa temps un tema de gran interès i que està present en molts àmbits de la societat. Podríem dir que és un conjunt de tècniques que s'apliquen a una imatge amb la intenció de ressaltar-ne les seves característiques, millorar-ne la qualitat o facilitar la recerca d'informació. Per exemple, les càmeres fotogràfiques tenen mètodes que processen la imatge en el mateix temps que feim una fotografia, per així poder obtenir uns resultats òptims.

El processament d'imatges té grans aplicacions en àmbits com la medicina o imatges espacials, entre altres. En el cas d'imatges mèdiques, la interpretació per part dels experts és molt subjectiva fins al punt de poder variar en un 20% dels casos la decisió final. Per això tractar aquesta imatge permet ajudar-los a diagnosticar tumors o altres patologies, ja que s'obté una informació molt més precisa i una observació més fàcil. Aquest fet és aplicable en la detecció de tumors o de situacions anormals a través de les ressonàncies magnètiques [1].

També hi ha algunes aplicacions curioses, com la recol·lecta de pomes segmentant les branques de pomes a partir d'imatges amb petites diferències de nivell de gris [2]. La intenció és crear un robot que detecti les branques dels arbres fruitals i sigui capaç d'evitar altres objectes, per així poder fer una recol·lecta dels fruits satisfactòria.

Similar a aquesta aplicació, tenim la d'identificar el gra de cafè idoni per la producció o per detectar-hi una possible plaga [3]. Segons el color del gra de cafè, es pot determinar el seu estat de maduració i si la imatge presenta zones negres, que poden fer referència als forats que deixa la plaga sobre el gra de cafè.

Els mètodes de processament d'imatges es poden classificar segons si volem analitzar la imatge o millorar-la. Si el que volem és analitzar les característiques de la imatge podem utilitzar un mètode de segmentació, de detecció de contorns o de reconeixement de formes, entre d'altres. En canvi, si el que volem és millorar-la, podem utilitzar mètodes com els de reducció del renou o de canvi de contrast i brillantor, entre d'altres.

En aquesta memòria, ens centrarem en analitzar les imatges utilitzant mètodes de segmentació. Segmentar és l'acte que consisteix en dividir una imatge en diferents subconjunts o regions disjunts (que anomenarem regions o clústers) de tal forma que

la unió d'ells forma tota la imatge. Per dur a terme aquest procés existeixen una gran diversitat de tècniques, que les dividirem entre les basades en l'espai de característiques, com agrupacions o histogrames de llindar, les basades en el domini de la imatge, com tècniques d'unió i divisió o basades en contorns, les tècniques de segmentació estadístiques, com el model de Markov o de Dirichlet, i les basades en intel·ligència artificial. En aquest treball, s'analitzaran algorismes de segmentació que tenen present algunes d'aquestes tècniques, explicant en detall els seus passos i les tècniques que s'hi empren. Concretament, s'estudiaran el mètode de segmentació basat en el model lingüístic de dues tuples [4], de segmentació jeràrquica [5], k-means [6], fuzzy k-means [7] i recursive shortest spanning tree (RSST) [8].

El mètode de dues tuples lingüístiques està basat en els histogrames de llindar. L'objectiu és convertir el problema de segmentar una imatge en un problema de presa de decisions. Inicialment tendrem un conjunt d'experts on cada un d'ells assigna un grau de pertinença de cada píxel a cada una de les regions basant-se en etiquetes lingüístiques de color. Cada píxel serà representat per la mitjana de les etiquetes lingüístiques i utilitzant un mètode de presa de decisions basat en dues tuples, es determinarà el grau de pertinença global de cadascun a cada regió. Per obtenir la imatge segmentada, es considerarà tant el valor de pertinença associat a cada píxel com als seus veïns.

El mètode de segmentació jeràrquica està basat en agrupacions. Es basa principalment en dos passos: quantificació del color i segmentació espacial. En el primer pas, es quantifiquen els colors de la imatge en diferents classes representatives que es poden utilitzar per diferenciar regions d'una imatge. Després es substituirà el valor de cada píxel per l'etiqueta de color corresponent a la classe en què pertanyi, construint així un mapa de classes de la imatge. En el segon pas, s'utilitzarà un criteri de segmentació basat en el mapa de classes. Aquest criteri s'aplicarà a una finestra del mapa de classes, on els majors o menors valors són candidats a pertànyer a les fronteres o interiors de la imatge. Posteriorment s'utilitzarà un mètode de creixement de regions per obtenir les segmentacions per diferents escales.

El k-means és també un mètode basat en agrupacions i en tècniques de creixement de regions. L'objectiu és fer una partició de la imatge en k subconjunts diferents, on cada píxel pertany al clúster que té mitjana de color més pròxima a ell. Inicialment es trien un grup de k píxels que seran els centres dels clústers. Els altres píxels s'aniran assignant al centre més pròxim i posteriorment s'actualitzaran els centres. Aquest procés es repetirà fins que una funció objectiu convergeixi.

Molt similar al k-means tenim el fuzzy k-means. En aquest cas és un mètode d'agrupament borrós, on cada píxel de la imatge pertany a cada regió amb un cert grau de pertinença. Igual que amb el k-means, aquest procés d'assignació s'anirà repetint fins que una funció objectiu convergeixi o sigui menor que un cert llindar.

El mètode RSST està basat també en agrupacions. En aquest cas, transformem la imatge en un graf on els píxels són els nodes i la relació entre ells, les arestes amb pesos. El mètode consisteix en unir els dos nodes que estan connectats per l'aresta que té pes mínim i actualitzar el graf. Aquest procés s'anirà repetint fins que s'arribi a un nombre de regions (nodes) fixat inicialment o segons algun criteri establert prèviament.

Després d'analitzar els mètodes anteriors, ens adonem que la comparació visual és insuficient per determinar quin és el millor mètode. Per això farem una avaluació quantitativa per tal de poder determinar quin és millor d'una forma objectiva. Per dur-ho a terme, es consideraran el mètode de l'índex de Rand [9] i el de variació de la

informació [10]. L'índex de Rand el que fa és mesurar la similitud entre els clústers de la imatge segmentada i la segmentació ideal, mentre que el mètode de variació de la informació mesura la informació que es guanya i la que es perd quan canviem un píxel d'una regió a una altra. Per comprovar si els resultats obtinguts són estadísticament significatius o no, utilitzarem el test estadístic de Wilcoxon [11].

Per poder calcular el valor dels índexs, necessitarem un conjunt d'imatges amb la seva *ground truth*, que és una segmentació ideal de cada una d'aquestes. Per això utilitzarem una base d'imatges de la Universitat de Berkeley [12] on, per a cada imatge, tenim l'original i diferents segmentacions fetes per humans (que considerem segmentacions ideals).

Per tal de desenvolupar aquest estudi, en el capítol 2 definirem el concepte de segmentació d'una imatge i es descriuran les diferents tècniques de segmentació així com també alguns conceptes previs que necessitarem. En el capítol 3 veurem 5 mètodes de segmentació, on posarem de manifest algunes de les tècniques. Després d'explicar i analitzar en detall els mètodes considerats i amb l'objectiu de realitzar una comparació objectiva entre ells, en el capítol 4 s'introduiran dues mesures objectives de rendiment. Emprant una base d'imatges, s'aplicaran els mètodes a totes les imatges trobant-ne els valors de les mesures. Així, podrem determinar quin és el millor mètode de segmentació d'entre els considerats i amb un test estadístic determinarem si els resultats són o no fiables. Finalment en el capítol 5 presentarem les conclusions obtingudes del treball i possibles projectes futurs que podrien ser interessants.

Al llarg d'aquesta memòria, s'han duit a terme determinades aportacions. La primera és una classificació de tècniques de segmentació, ja que la gran diversitat de tècniques fa que hi hagi moltes classificacions diferents, algunes d'elles obsoletes. Per altra part, s'han implementat i fet modificacions als algorismes, com afegir el mètode del colze per determinar el nombre de clústers òptim. En el cas del mètode de dues tuples només estava dissenyat per fer segmentacions en 6 regions i s'ha modificat l'algorisme utilitzant una tècnica d'unió de regions per així poder obtenir resultats amb menys de 6 regions finals. En els mètodes k-means i fuzzy k-means s'ha implementat un mètode que, utilitzant els algorismes ja implementats en MATLAB, ens realitzi un gran nombre de segmentacions d'una mateixa imatge, per així poder obtenir uns resultats més fiables. Finalment, el mètode de RSST ha estat implementat directament a partir de l'explicació de l'article. Després d'analitzar els mètodes de segmentació, s'ha obtingut una imatge de consens a partir de diferents segmentacions ideals i, d'acord a dos índexs coneguts, s'ha fet una comparació, que no es pot trobar a la literatura, de les diferents segmentacions obtingudes pels algorismes analitzats.

SEGMENTACIÓ

En aquest capítol definirem formalment el concepte de segmentació, veurem alguns dels conceptes bàsics que necessitarem al llarg de la memòria i una possible classificació de les diferents tècniques de segmentació.

2.1 Conceptes bàsics

Podem definir una imatge com una funció de dues dimensions $f(x, y)$, on x i y són les coordenades espaials i $f(x, y)$ la intensitat de la imatge en aquest punt [13]. Una imatge serà una funció $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ en el cas d'imatges en nivells de gris o $f: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ en el cas d'imatges en color. Entre els diferents espais de color que podem trobar (RGB, YIQ, HSV o HSL), s'utilitzarà el sistema RGB. En aquest espai de color, cada imatge està formada per tres imatges monocromàtiques individuals, anomenades component vermella, verda i blava (Red, Green and Blue).

Una imatge és una funció continua que depèn de les coordenades (x, y) i de l'amplitud d'intensitats. Per poder treballar amb una imatge, el primer que haurem de fer serà digitalitzar-ne les seves components. Per digitalitzar les coordenades espaials farem un mostreig (*sampling*) que consisteix en elegir la quantitat de punts que utilitzarem per representar la imatge. Aquest procés depèn del tipus de xarxa elegida, ja que es converteix la intensitat de dins cada polígon a un únic valor d'intensitat. Per exemple, si tenim una imatge en nivell de gris, es tenen xarxes com les de la figura 2.1 entre d'altres. Per altra part, també és necessari digitalitzar l'amplitud. Per això es realitza una quantificació que consisteix en elegir la quantitat de valors que s'utilitzaran per aproximar la intensitat de cada polígon de la xarxa. Després de dur a terme aquest procés, obtindrem una imatge digital, és a dir, una matriu de valors on cada nombre fa referència a la intensitat de la imatge en cada punt. A cada un d'aquests punts (x, y) l'anomenarem píxel.

Un altre dels conceptes que s'utilitza és el d'histograma d'una imatge. L'histograma d'una imatge digital amb nivell de gris entre 0 i 255 és una funció discreta $h(r_k) = n_k$

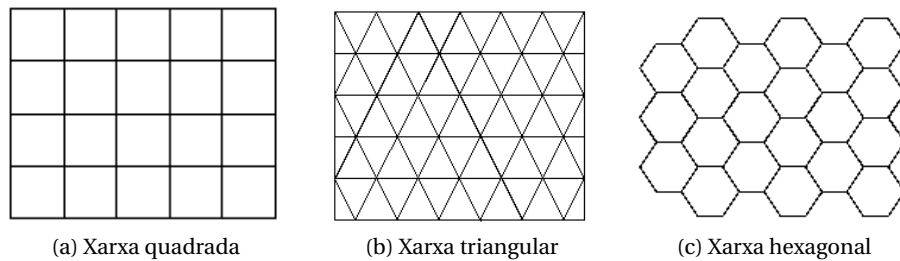


Figura 2.1: Tipus de xarxes.

on r_k és el k -èssim nivell de gris i n_k la quantitat de píxels de la imatge que tenen un nivell de gris r_k . En el cas d'imatges en l'espai de color RGB, tenim tres histogrames, un corresponent a cada un dels colors. L'histograma d'una imatge aporta molta informació, per exemple, podem saber si el contrast d'una imatge digital és o no l'adequat.

2.2 Segmentació d'imatges

La segmentació d'imatges és un problema fonamental en l'actualitat. Podríem dir que és un procés que consisteix en dividir una imatge en diferents subconjunts o regions disjunctes (que anomenarem regions o clústers) de tal forma que la unió forma tota la imatge. L'any 1985 Haralick va establir una serie de normes que s'havien de satisfer a l'hora de segmentar una imatge [14]:

1. Les regions han de ser uniformes i homogènies respecte algunes característiques.
2. Els interiors han de ser simples i sense gaire detalls.
3. Les regions adjacents han de ser diferents respecte de les característiques que les fan uniformes.
4. Les fronteres de cada regió han de ser simples i no presentar talls.

En moltes ocasions, no es requereix que es satisfaci la propietat 4. A partir d'aquestes condicions, podem establir la següent definició formal.

Definició 2.2.1 (Segmentació d'una imatge [14]). *Sigui R el domini de definició d'una imatge i siguin R_i amb $1 \leq i \leq n$ les n regions no buides de R . Aleshores s'ha de satisfer que:*

1. $\bigcup_{i=1}^n R_i = R$,
2. per a cada i, j amb $i \neq j$, $R_i \cap R_j = \emptyset$,
3. per a cada $i = 1, \dots, n$ s'ha de satisfer que $P(R_i) = \text{CERT}$,
4. per a cada $i \neq j$ s'ha de satisfer $P(R_i \cup R_j) = \text{FALS}$,

on $P(R_i)$ és un propietat uniforme per a tots els elements del conjunt R_i . En algunes ocasions, s'exigeix també la següent propietat:

5. Per a tot $1 \leq i \leq n$, R_i ha de ser una component connexa.

Aquesta propietat pot ser per exemple el color dels píxels, la textura o en general, la proximitat en l'espai de característiques.

Aquesta definició és la que caracteritza la segmentació dura (*crisp*). Per altra part, també veurem alguns mètodes que utilitzen la segmentació suau o borrosa (*soft*).

A diferència de la segmentació forta, en la suau no podem dir si un píxel pertany o no a una regió, sinó que pertany a totes les regions amb un cert grau de pertinença [15]. En aquest tipus de segmentació, és imprescindible la noció de funció de pertinença. Aquesta funció el que fa és assignar un valor dins $[0, 1]$ a cada píxel respecte a cada regió, on 1 indica el grau màxim de pertinença i 0 el grau mínim. Aquest tipus de segmentació és més natural que la forta, ja que en la vida real no solem tenir particions exactes. Un exemple on s'utilitza aquesta tècnica és el mètode de *fuzzy k-means* (FCM) [15] que veurem més endavant.

Després de segmentar les imatges, necessitarem saber com de bona és la segmentació realitzada, per això ens serà de gran utilitat el següent concepte.

Definició 2.2.2 (SSC). *Sigui I una imatge de N píxels amb vectors de característiques associats $x_1, \dots, x_N \in \mathbb{R}^n$. Consideram una segmentació de I en K regions i amb centres $c_1, \dots, c_k \in \mathbb{R}^n$. Aleshores es defineix la suma de quadrats dins de cada regió com*

$$SSC(x_1, \dots, x_N; K) = \sum_{j=1}^K \sum_{i=1}^N \|x_i - c_j\|^2.$$

Molts algorismes de segmentació requereixen determinar el nombre de regions de la imatge a priori. Determinar aquest nombre no és una tasca senzilla i de fet, no existeix una tècnica òptima per dur-ho a terme. Amb poques regions el que obtindrem serà una imatge molt simplificada amb regions molt heterogènies, mentre que considerar moltes regions pot dificultar etapes posteriors a la segmentació a causa de la complexitat de la imatge resultant. Per determinar un nombre de regions que ens proporcioni una segmentació de qualitat amb regions homogènies però a la vegada el nombre de regions no sigui molt gran, utilitzarem el mètode del colze [8].

Donada una imatge I , definim:

1. $Seg_i(I)$ la segmentació de I amb menor SSC quan es considera un nombre i de regions, amb $i \in \{1, \dots, K\}$.
2. C la funció $C: \{1, \dots, K\} \rightarrow \mathbb{R}^+$ on $C(i) = SSC(Seg_i(I))$.

El que volem és determinar el nombre de regions per aconseguir una segmentació òptima. Per això construïm la recta que uneix $(1, C(1))$ amb $(K, C(K))$ i volem cercar i tal que el punt $(i, C(i))$ tenguí distància màxima a n'aquesta recta. Aquest nombre de regions serà el considerat com a idoni per aconseguir la segmentació òptima ja que és el que presenta una variació màxima en el valor de C entre ell i el nombre de regions anterior. En la figura 2.2., es pot observar aquest fet i la raó per la que s'anomena mètode del colze, ja que el nombre de regions que es cerca és el corresponent al "colze" de la funció.

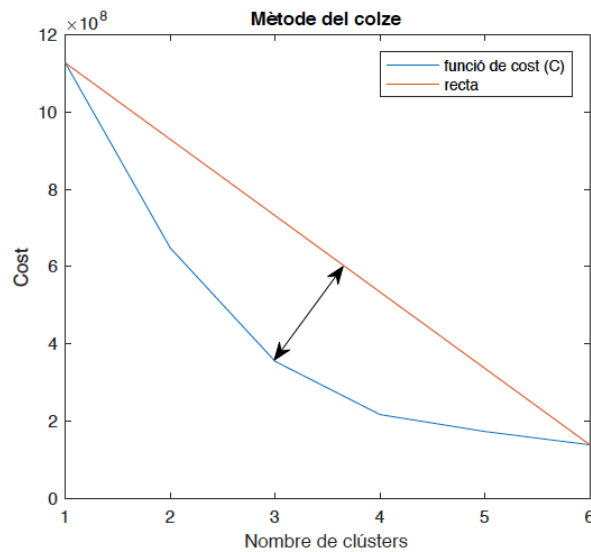


Figura 2.2: Funció de cost segons el nombre de regions.

Formalment, siguin

$$r : y = mx + n; P_i = (i, C(i))$$

on r és la recta que va de $(1, C(1))$ a $(K, C(K))$, m és el pendent $\left(m = \frac{C(K) - C(1)}{K - 1}\right)$, $n = C(1) - m$. Donat un punt $(i, C(i))$ amb $1 \leq i \leq K$, sabem que la distància del punt a la recta ve donada per

$$d(P_i, r) = \frac{|mi - C(i) + C(1) - m|}{\sqrt{m^2 + 1}}.$$

Així, el nombre de clústers òptim vendrà donat per

$$\text{Nombre clústers òptim} = \arg \max_{i=2, \dots, K-1} d(P_i, r).$$

2.3 Tècniques de segmentació

A dia d'avui no existeix un algorisme de segmentació òptim, en el sentit que funcioni per tots els tipus d'imatges. Per això, s'han proposat algorismes de segmentació basats en teories i tècniques d'allò més diverses. Cada una d'elles té la seva importància i per cada tipus d'imatge és més adequada una tècnica de segmentació o una altra. A continuació, es donarà una classificació de les metodologies més emprades per realitzar la segmentació d'una imatge. Aquesta classificació és novedosa, introduint algunes modificacions en les classificacions que es poden trobar en [16, 17, 18].

2.3.1 Tècniques basades en l'espai de característiques

Aquestes tècniques es caracteritzen per assignar un vector de característiques a cada un dels píxels i posteriorment, a partir d'aquest espai de característiques, es realitza la segmentació de la imatge.

Existeixen dues estratègies principals en aquest tipus de tècnica de segmentació. La primera es basa en suposar que tots els píxels d'un objecte de la imatge tenen un color semblant. Per tant, si es considera com espai de característiques l'espai de color, els píxels d'un mateix objecte s'assignaran a una mateixa regió. Per altra banda, una segona estratègia considera histogrames generats a partir de la imatge per realitzar la segmentació. Aquesta tècnica té com a hipòtesi que els màxims dins l'histograma poden ser identificats amb els objectes de la imatge, que hauran de ser assignats a regions diferents en la segmentació.

Així entre els algorismes que utilitzen tècniques basades en l'espai de característiques, tenim els que utilitzen agrupacions i els que utilitzen histogrames de llindars.

1. Agrupacions.

El *clustering* o agrupacions es una tècnica de segmentació no supervisada que genera particions sense tenir un coneixement previ. Es basa en dividir una imatge en diferents regions, on cada una d'elles està formada per un conjunt de píxels semblants segons l'espai de color.

Entre els mètodes que utilitzen aquesta tècnica, els més coneguts són el k-means [6], que és un mètode d'agrupació dura, o el fuzzy k-means [7], que és un mètode d'agrupació suau. També es pot destacar el proposat per Yung i Lai [19], on es realitza la segmentació segons la força gravitacional que actua entre diferents partícules.

2. Histogrames de llindar.

Aquesta és una de les tècniques més utilitzades per segmentar imatges en nivell de gris. Permet identificar fàcilment els objectes d'una imatge i el fons, ja que aquests es corresponen, respectivament, amb els turons i les valls de l'histograma.

En el cas d'imatges de color és una mica més complicat, ja que tenim tres histogrames (un per a cada component de l'espai de color) i el que es fa és combinar els turons i les valls dels tres histogrames. Un problema habitual d'aquesta tècnica és el renou ja que, degut a aquest, els histogrames sovint presenten irregularitats, donant lloc a màxims erronis. Per evitar aquests errors, generalment s'adopten mesures per suavitzar les imatges abans de calcular-ne els histogrames.

En aquest apartat, l'algorisme més conegut que utilitza aquesta tècnica és l'algorisme de watershed [20], que es basa en trobar les divisòries del relleu topogràfic de la imatge segons la brillantor de cada píxel.

2.3.2 Tècniques basades en el domini de la imatge

Una de les característiques que esperam tengui una segmentació és que les regions siguin homogènies respecte a alguna característica. Però, aquest fet no garanteix que totes les regions siguin connexes, cosa que també sovint es desitja d'una segmentació. De fet, les tècniques d'agrupacions i d'histogrames de llindar no tenen en compte que els punts d'un mateix objecte solen ser espacialment propers.

Per altra part, si es dividissin les regions únicament segons la relació espacial dels seus píxels, el resultat final serien regions completament connectades però que no garanteixen l'homogeneïtat respecte alguna característica.

2. SEGMENTACIÓ

Algunes de les tècniques que es poden utilitzar per satisfer una certa homogeneïtat segons alguna característica i tal que les regions siguin connexes són les d'unió i divisió, de creixement de regions, basades en detecció de contorns o en models deformables entre d'altres.

1. Tècniques d'unió i divisió.

Inicialment tenim una partició no homogènia de la imatge i es van separant les regions o píxels amb la intenció d'obtenir una partició homogènia respecte alguna característica. En cas de tenir més regions de les desitjades, s'aplica una tècnica d'unió que s'encarregarà d'unir regions veïnes si satisfan una certa condició d'homogeneïtat.

Entre els mètodes que utilitzen aquesta tècnica destaquem el descrit per Barni a l'article [21] que descriu un mètode d'unió i divisió utilitzant experts borrosos. Un altre mètode on s'utilitza la tècnica d'unió, és en la versió modificada del mètode de dues tuples lingüístiques que veurem més endavant (veure capítol 3). En aquest mètode, tenim més regions de les desitjades i apliquem un procés d'unió per tal de disminuir el nombre de regions resultant.

2. Tècniques de creixement de regions.

Donada una imatge, es seleccionen alguns píxels (manualment si es tenen alguns coneixements previs o de forma automàtica) que seran les llavors de cada regió. Les diferents regions que formen cada un d'aquests píxels s'aniran ampliant amb els seus píxels veïns si satisfan una certa condició d'homogeneïtat. El procés finalitzarà quan tots els píxels estiguin en alguna de les regions.

Després de dur a terme aquest procés, és possible que hi hagi regions molt petites o regions veïnes amb característiques molt similars. En aquest cas, es pot utilitzar la tècnica d'unió descrita anteriorment i així obtenir regions més grans.

La tècnica de creixement de regions es considera *clustering* seqüencial, ja que els resultats obtinguts depenen de l'ordre en què es processa cada píxel. En aquest cas, les regions obtingudes són homogènies i connexes. El problema d'aquesta tècnica és elegir les llavors correctament i un criteri d'homogeneïtat adequat.

Entre els mètodes que utilitzen aquestes tècniques, destacarem el mètode de segmentació jeràrquica que analitzarem en detall més endavant (veure capítol 3) i el de Xiaofeng Ren [22].

3. Tècniques basades en detecció de contorns.

Sovint es poden obtenir segmentacions detectant els contorns dels objectes d'interès d'una imatge. Si la imatge està en nivell de gris, es poden detectar els contorns utilitzant el gradient o laplaciana, que són funcions escalars. Però, si són a color, el gradient fa referència a unir en una única mesura la variació dels tres canals de color o calcular el gradient de cada un dels tres canals de color i combinar-los segons algun criteri.

Entre els mètodes que utilitzen aquesta tècnica, cal destacar el mètode proposat per Ma i Manjunath [23], que utilitza un model per identificar la direcció de canvi de color i textura d'una imatge. També cal destacar el descrit per Xiaofeng Ren i Liefeng Bo [24].

4. Models deformables.

L'objectiu d'aquesta tècnica és adaptar una corba que coincideixi el màxim possible amb el contorn dels objectes d'interès d'una imatge. Inicialment es comença amb una frontera inicial i es representa una corba sobre ella. Iterativament, aquesta es modificarà aplicant operacions de contracció i expansió. Aquestes operacions tenen per objectiu minimitzar una funció d'energia associada que serà òptima quan la corba encaixi perfectament amb la frontera d'un objecte de la imatge.

Entre les subtasques en què es basa aquesta tècnica trobem la detecció de paràmetres que regulen l'algorisme, com per exemple pesos de les funcions de cost, o la inicialització de la corba per tal de facilitar la convergència cap a la frontera.

S'han proposat diversos algorismes de segmentació emprant aquesta tècnica (veure l'article recopilatori [25]). Entre ells es troben el model *Snake* de Tanipanonid i Covavisaruch [26], utilitzat per fer resonàncies magnètiques o el de Karungaru [27], utilitzat per detecció de cares.

2.3.3 Intel·ligència artificial

La intel·ligència artificial és el procés que consisteix en fer que una màquina es comporti com si tingués una intel·ligència semblant a la dels humans. En aquest àmbit destacarem el machine learning i les xarxes neuronals.

El machine learning és una tècnica que consisteix en aprendre a reconèixer patrons i relacions entre les dades per tal de poder prendre una decisió precisa. Habitualment aquestes tècniques s'utilitzen per diagnosticar i segmentar patologies mèdiques com els tumors [28].

Les xarxes neuronals són estructures formades per una gran extensió de processos connectats entre ells amb funcions simples. El disseny de xarxes neuronals el que intenta és simular el comportament humà. Tot i la gran complexitat en la seva implementació, ofereixen una gran velocitat computacional i per tant, la possibilitat d'aplicacions en temps real, a més de poca sensibilitat a les distorsions.

En el cas de segmentació d'imatges, les xarxes neuronals permeten considerar la informació espacial. A més, es pot realitzar una fase d'aprenentatge on la xarxa s'entrenarà per reconèixer patrons.

Entre els mètodes més coneguts que utilitzen les xarxes neuronals trobem l'algorisme de Hopfield [29] o el d'Okii [30].

2.3.4 Tècniques de segmentació estadístiques

Els mètodes que utilitzen tècniques estadístiques el que fan és aproximar algunes característiques de les imatges (com el color o la textura, entre d'altres) utilitzant una funció probabilística. Degut a l'aproximació, en general els mètodes que utilitzen aquestes tècniques són molt sensibles al renou. Entre els mètodes que utilitzen tècniques estadístiques, destacarem el model de Markov i el de Dirichlet.

1. El camp aleatori de Markov o *Markov Random Field* (MRF) [31] és una tècnica que consisteix en modelar els píxels de la imatge segons el seu context i la relació

2. SEGMENTACIÓ

entre les característiques, com poden ser el color o les textures. Aquest fet és possible gràcies a les distribucions condicionals de Markov.

2. *Dirichlet process mixture model* [32] és una tècnica que incorpora informació espacial entre els píxels veïns basada en la distribució de Dirichlet. A més, aquest model requereix d'una funció de versemblança amb diversos paràmetres que es necessiten estimar.

MÈTODES DE SEGMENTACIÓ

En aquest apartat s'analitzen en detall cinc mètodes de segmentació d'imatges. S'explicarà cadascuna de les passes en què consta cada mètode i es mostraran els seus efectes quan s'apliquen a una imatge determinada.

3.1 Mètode basat en el model lingüístic de dues tuples

El mètode basat en el model lingüístic de dues tuples [4] és un mètode que es situa dins les tècniques basades en l'espai de característiques on únicament es té en compte el color dels píxels, és a dir, no es considera ni la textura ni la seva posició espacial. Aquest és un mètode basat en tècniques de segmentació borrosa i l'objectiu és convertir el problema de segmentar una imatge en un problema de presa de decisions.

El mètode considera diferents experts "ficticis" caracteritzats per visions diferents de similaritat de dos elements i per graus d'ignorància diferents. Cada expert assigna a cada píxel un grau de pertinença a cada una de les regions de la imatge. Per representar les regions s'utilitzaran etiquetes lingüístiques i el model de presa de decisions basat en 2-tuples lingüístiques on, gràcies al valor de pertinença associat a cada píxel i als seus veïns, s'obtindrà la imatge segmentada. Aquest mètode sempre realitza una segmentació en sis regions diferents, per això el que es farà serà ajuntar de totes les formes possibles aquestes sis regions, permetent així una segmentació amb menys de sis regions. Per saber quin és el nombre de clústers òptim, aplicarem el mètode del colze.

Abans de començar a explicar el mètode, veurem alguns conceptes i resultats que després ens seran necessaris per tal de poder entendre totes les passes.

3.1.1 Conceptes i resultats preliminars

La funció d'equivalència restringida és un concepte molt utilitzat per comparar imatges i el que fa és intentar donar una mesura que indiqui com d'igual o de semblants són dos

elements x i y . Per donar una definició formal, necessitarem primer algunes definicions i resultats previs.

Definició 3.1.1. *Una negació borrosa és una funció $N : [0, 1] \rightarrow [0, 1]$ que satisfà $N(0) = 1$ i $N(1) = 0$ i que és decreixent. A més, si N és involutiva, és a dir, $N(N(x)) = x$ per a tot $x \in [0, 1]$, s'anomena una negació forta.*

Definició 3.1.2. *Una funció $\phi : [0, 1] \rightarrow [0, 1]$ s'anomena un automorfisme en $[0, 1]$ si és continua i estrictament creixent i satisfà $\phi(0) = 0$ i $\phi(1) = 1$.*

Per construir els conjunts borrosos associats a una imatge, utilitzarem el concepte de funció d'equivalència restringida.

Definició 3.1.3 ([33]). *Una funció $REF : [0, 1]^2 \rightarrow [0, 1]$ s'anomena funció d'equivalència restringida associada amb una negació forta N si satisfà les següents propietats:*

- $REF(x, y) = REF(y, x)$ per a tot $x, y \in [0, 1]$.
- $REF(x, y) = 1$ si i només si $x = y$.
- $REF(x, y) = 0$ si i només si $x = 1$ i $y = 0$ o $x = 0$ i $y = 1$.
- $REF(x, y) = REF(N(x), N(y))$ per tot $x, y \in [0, 1]$, amb N una negació forta.
- Per a tots $x, y, z \in [0, 1]$, si $x \leq y \leq z$, aleshores $REF(x, y) \geq REF(x, z)$ i $REF(y, z) \geq REF(x, z)$.

El resultat següent dona un mètode de construcció de funcions d'equivalència restringida.

Proposició 3.1.1. *Siguin dos automorfismes ϕ_1 i ϕ_2 de l'interval unitat, aleshores*

$$REF(x, y) = \phi_1^{-1}(1 - |\phi_2(x) - \phi_2(y)|) \text{ amb } N(x) = \phi_2^{-1}(1 - \phi_2(x)),$$

és una funció d'equivalència restringida amb negació forta associada N .

Demostració. Vegem que efectivament és satisfan les propietats de les funcions d'equivalència restringida.

- Clarament $REF(x, y) = REF(y, x)$ ja que $|\phi_2(x) - \phi_2(y)| = |\phi_2(y) - \phi_2(x)|$.
- $REF(x, y) = 1$ si i només si $|\phi_2(x) - \phi_2(y)| = 0$, si i només si, $\phi_2(x) - \phi_2(y) = \phi_2(y) - \phi_2(x) = 0$ i això passa si i només si $x = y$.
- Si $REF(x, y) = 0$, aleshores $|\phi_2(x) - \phi_2(y)| = 1$ és a dir, pot passar que $\phi_2(x) - \phi_2(y) = 1$ i per tant $x = 1$ i $y = 0$ o que $\phi_2(y) - \phi_2(x) = 1$ i per tant $y = 1$ o $x = 0$. Per altra part, si $x = 1$ i $y = 0$, $REF(1, 0) = \phi_1^{-1}(1 - |\phi_2(1) - \phi_2(0)|) = \phi_1^{-1}(1 - 1) = \phi_1^{-1}(0) = 0$. De forma anàloga, es pot provar per $x = 0$ i $y = 1$.
- $REF(N(x), N(y)) = \phi_1^{-1}(1 - |\phi_2(N(x)) - \phi_2(N(y))|)$ amb $N(x) = \phi_2^{-1}(1 - \phi_2(x))$. Substituint, tenim que $REF(N(x), N(y)) = \phi_1^{-1}(1 - |1 - \phi_2(x) - 1 + \phi_2(y)|) = \phi_1^{-1}(1 - |\phi_2(x) - \phi_2(y)|) = REF(x, y)$.

- Si $x \leq y \leq z$, aleshores $\phi_2(x) \leq \phi_2(y) \leq \phi_2(z)$, a més $\phi_2(y) - \phi_2(x) \leq \phi_2(z) - \phi_2(x)$. Per tant, $1 - (\phi_2(y) - \phi_2(x)) \geq 1 - (\phi_2(z) - \phi_2(x))$ i com ϕ_1 és un automorfisme, $REF(x, y) \geq REF(x, z)$. Anàlogament, tenim que $REF(y, z) \geq REF(x, z)$.

Vegem a més que $N(x) = \phi_2^{-1}(1 - \phi_2(x))$ és una negació borrosa.

- $N(0) = \phi_2^{-1}(1 - \phi_2(0)) = \phi_2^{-1}(1) = 1$.
- $N(1) = \phi_2^{-1}(1 - \phi_2(1)) = \phi_2^{-1}(1 - 1) = \phi_2^{-1}(0) = 0$.
- Vegem que és decreixent. Per això hem de veure que $x_1 \leq x_2 \Rightarrow N(x_1) \geq N(x_2)$. Si $x_1 \leq x_2 \Rightarrow$ per ser ϕ_2 automorfisme $\phi_2(x_1) \leq \phi_2(x_2) \Rightarrow 1 - \phi_2(x_1) \geq 1 - \phi_2(x_2) \Rightarrow \phi_2^{-1}(1 - \phi_2(x_1)) \geq \phi_2^{-1}(1 - \phi_2(x_2)) \Rightarrow N(x_1) \geq N(x_2)$.
- Vegem que $N(N(x)) = x$.
 $N(N(x)) = \phi_2^{-1}(1 - \phi_2(\phi_2^{-1}(1 - \phi_2(x)))) = \phi_2^{-1}(1 - (1 - \phi_2(x))) = \phi_2^{-1}(\phi_2(x)) = x$.

□

Per intentar modelar la falta de precisió dels experts que es pot produir quan es determinen els graus de pertinença d'un píxel a una regió, utilitzarem la funció d'ignorància.

Definició 3.1.4. Una funció $IG: [0, 1]^2 \rightarrow [0, 1]$ s'anomena funció d'ignorància si satisfà les propietats següents:

- $IG(x, y) = IG(y, x)$ per a tots $x, y \in [0, 1]$.
- $IG(x, y) = 0$ si i només si $x = 1$ o $y = 1$.
- Si $x = 0.5$ i $y = 0.5$, aleshores $IG(x, y) = 1$.
- IG és decreixent a la regió $[0.5, 1]^2$.
- IG és creixent a la regió $[0, 0.5]^2$.

A més, també necessitarem alguns conceptes de lògica borrosa.

Definició 3.1.5. Un conjunt borrós A definit sobre un univers finit no buit $X = \{x_1, \dots, x_n\}$ ve donat per

$$A = \{(x_i, \mu_A(x_i)) \mid x_i \in X\}$$

on $\mu_A: X \rightarrow [0, 1]$ és la funció de pertinença del conjunt A .

Denotem per $FS_S(X)$ el conjunt de conjunts borrosos definits sobre l'univers X .

La funció de pertinença d'un conjunt borrós ens serà útil per assignar un valor a cada píxel respecte cada una de les regions, mentre que la funció de pertinença a un conjunt borrós interval-valorat (*IVFS*) el que fa és assignar a cada valor d'un univers X un subinterval de $[0, 1]$.

Definició 3.1.6. Sigui $L([0, 1]) = \{[M_L, M_U] \mid (M_L, M_U) \in [0, 1]^2 \text{ i } M_L \leq M_U\}$ el conjunt de subinterval·s tancats de $[0, 1]$. Un conjunt borrós interval·valorat \tilde{A} de l'univers X està definit per

$$\tilde{A} = \{(x_i, M_{\tilde{A}}(x_i) = [M_{\tilde{A}L}(x_i), M_{\tilde{A}U}(x_i)]) \mid x_i \in X\}$$

on la funció $M_{\tilde{A}} : X \rightarrow L([0, 1])$ és la funció de pertinença del conjunt \tilde{A} .

Denotarem per $IVFS_S(X)$ el conjunt de conjunts borrosos interval·valorats de X . Utilitzarem la següent definició d'entropia d'un conjunt borrós interval·valorat.

Definició 3.1.7. Una funció $\varepsilon_F : IVFS_S(X) \rightarrow \mathbb{R}^+$ s'anomena entropia de $IVFS_S(X)$ si ε_F satisfà les següents propietats:

1. $\varepsilon_F(\tilde{A}) = 0$ si i només si $\tilde{A} \in FS_S(X)$.
2. $\varepsilon_F(\tilde{A}) = 1$ si i només si $M_{\tilde{A}L}(x_i) = 0$ i $M_{\tilde{A}U}(x_i) = 1$ per a tot $x_i \in X$.
3. $\varepsilon_F(\tilde{A}) = \varepsilon_F(\tilde{A}_N)$ per a tot $\tilde{A} \in IVFS_S(X)$, on $\tilde{A}_N = \{(x_i, M_{\tilde{A}_N}(x_i) = [1 - M_{\tilde{A}U}(x_i), 1 - M_{\tilde{A}L}(x_i)]) \mid x_i \in X\}$.
4. Si $\tilde{A} \preceq \tilde{B}$ i.e. $M_{\tilde{A}L}(x_i) \leq M_{\tilde{B}L}(x_i)$ i $M_{\tilde{B}U}(x_i) \leq M_{\tilde{A}U}(x_i)$ per a tot $x_i \in X$, aleshores $\varepsilon_F(\tilde{A}) \geq \varepsilon_F(\tilde{B})$.

Com ja hem dit, el que feim és convertir el problema en un problema de presa de decisions. Per resoldre aquest problema de presa de decisions, és necessari tenir una estructura que doni suport i un significat a les preferències donades pels experts. Suposem que tenim un problema amb n alternatives $X = \{x_1, \dots, x_n\}$, utilitzarem com a estructura el vector d'utilitats. El vector d'utilitats conté el valor de preferència respecte cada una de les alternatives considerades. En el nostre cas, el vector d'utilitats per X és el conjunt $\{y_i \mid y_i \in [0, 1], i = 1, \dots, n\}$, on y_i representa el valor donat per l'expert a l'alternativa x_i .

3.1.2 Algorisme

Suposem que tenim una imatge formada per diferents objectes, on cada un d'ells està format per un color diferent. Utilitzant tècniques de lògica borrosa, intentarem separar cada un dels objectes de la imatge. El pseudocodi de l'algorisme que utilitzarem per dur a terme aquest mètode es pot veure a l'algorisme 1. Anem a descriure en detall cada un dels passos.

Transformació de l'espai de color.

En aquest apartat el que feim és transformar la imatge original a un nou espai de color. Així el problema es redueix a separar un objecte del fons per a cada un dels nous canals de color.

Sigui I la imatge original, de $n \times m$ píxels, en l'espai de color RGB. Cada píxel té assignat una terna (q_R, q_G, q_B) on cada component representa la intensitat de color de cada píxel en aquell canal de color. Per convertir aquesta imatge en un nou espai de color, normalitzem la intensitat de cada component, per obtenir així components dins $[0, 1]$ i poder aplicar els operadors introduïts. Després representem les diferències entre cada

Algorisme 1: Mètode basat en el model lingüístic de dues tuples

Entrada: Imatge que es vol segmentar considerada en l'espai de color RGB.

Sortida: Imatge segmentada.

- 1 *Transformar la imatge de RGB a un nou espai de color* $\{GR, BG, BR\}$;
 - 2 *Seleccionar K experts* $\{E_1, \dots, E_K\}$;
 - 3 **per** E_1 fins E_K **fer**
 - 4 *Obtenir els llindars* (t_{GR}, t_{BG}, t_{BR}) *corresponents a cada component* $\{GR, BG, BR\}$;
 - 5 *Construir les etiquetes lingüístiques per a cada objecte utilitzant* (t_{GR}, t_{BG}, t_{BR}) ;
 - 6 *Construir un vector d'utilitats per a cada píxel considerant el grau de pertinença de l'etiqueta lingüística definida al pas anterior;*
 - 7 *Presa de decisions;*
 - 8 *Per a cada píxel afegir el vector d'utilitats donat per* $\{E_1, \dots, E_K\}$;
 - 9 *Afegir les preferències de cada píxel i dels seus veïns;*
 - 10 *Classificar cada píxel;*
-

una de les components, donades per $f(q_i, q_j) = \frac{q_i - q_j + 1}{2}$ amb $i, j \in \{R, G, B\}$. Així el nou espai de color vendrà definit pels tres graus de llibertat $\{f(q_G, q_R), f(q_B, q_G), f(q_B, q_R)\}$. El denotarem per $\{GR, BG, BR\}$, on cada píxel tindrà una intensitat de color definida per (q_{GR}, q_{BG}, q_{BR}) . Notem que $f(q_i, q_j) \geq 0$ per a tot $i, j \in \{R, G, B\}$, ja que $q_i, q_j \leq 1$.

Selecció dels experts.

Donada una imatge formada per diferents objectes, suposem que cada un d'ells està format per un color diferent. A continuació, el que es farà és seleccionar un conjunt d'experts que, per a cada píxel, assignin un valor de pertinença a cada píxel respecte a cada objecte de la imatge.

Un dels problemes més grans és seleccionar correctament la funció que soluciona aquest problema. Nosaltres ens basarem en l'article [34] on defineix que un expert ve definit per una funció d'ignorància (IG) i una funció d'equivalència restringida.

1. Funcions d'ignorància:

$$\begin{aligned}
 \text{a) } IG_1(x, y) &= \begin{cases} \frac{(1-x) \cdot (1-y)}{0.2} & \text{si } (1-x) \cdot (1-y) \leq 0.2, \\ 1 & \text{si } 0.2 \leq (1-x) \cdot (1-y) \leq 0.8, \\ \frac{0.2}{1((1-x) \cdot (1-y)) - 3} & 0.8 \leq (1-x) \cdot (1-y) \leq 1. \end{cases} \\
 \text{b) } IG_2(x, y) &= \begin{cases} 4(1-x) \cdot (1-y) & \text{si } (1-x) \cdot (1-y) \leq 0.25, \\ \frac{1}{4(1-x) \cdot (1-y)} & \text{altrament.} \end{cases} \\
 \text{c) } IG_3(x, y) &= \begin{cases} 2\sqrt{(1-x) \cdot (1-y)} & \text{si } (1-x) \cdot (1-y) \leq 0.25, \\ \frac{1}{2\sqrt{(1-x) \cdot (1-y)}} & \text{altrament.} \end{cases}
 \end{aligned}$$

2. Funcions d'equivalència restringida:

a) $REF_1(x, y) = 1 - |x - y|$.

- b) $REF_2(x, y) = 1 - |x^{\frac{1}{2}} - y^{\frac{1}{2}}|$.
c) $REF_3(x, y) = (1 - |x - y|)^2$.
d) $REF_4(x, y) = \left(1 - |x^{\frac{1}{2}} - y^{\frac{1}{2}}|\right)^2$.

Així, els nostres experts seran totes les combinacions possibles entre les funcions d'ignorància i les funcions d'equivalència restringida. En total tendrem dotze experts diferents que denotarem per E_{ij} on $1 \leq i \leq 3$ és l'índex de IG i $1 \leq j \leq 4$ l'índex de REF .

Llindar de l'espai de color $\{GR, BG, BR\}$.

Com ja hem dit al primer pas, cada píxel es representa amb una terna (q_{GR}, q_{BG}, q_{BR}) . En aquest pas, aplicant l'algorisme de binarització presentat a [34] per cada una de les components de $\{GR, BG, BR\}$ obtenim una terna de llindars (t_{GR}, t_{BG}, t_{BR}) . Aquest mètode el que fa és seleccionar el llindar com el que té funció d'ignorància mínima entre la funció de pertinença de cada píxel del fons i de l'objecte de la imatge. Els passos són els següents:

1. Construir L conjunts borrosos associats a la imatge I .

Denotarem per I_{Bt} els que representen el fons de la imatge i I_{Ot} els que representen els objectes, on t és el nivell d'intensitat al que estan associats amb $t \in [0, L-1]$ on L és el nombre de nivells de color utilitzats.

Les corresponents funcions de pertinença associades es defineixen amb les funcions d'equivalència restringida, on els valors més grans de la funció de pertinença associada a un píxel p , $\mu_{I_{Bt}}(p)$ i $\mu_{I_{Ot}}(p)$ estan associats a major valors entre un nivell de color del píxel p i la mitjana d'intensitat del fons (o objecte).

Per a cada $t \in [0, L-1]$, denotarem per $m_B(t)$ i $m_O(t)$ la mitjana de la intensitat de color dels píxels del fons i de l'objecte respectivament, és a dir,

$$m_B(t) = \frac{\sum_{q=0}^t qh(q)}{\sum_{q=0}^t h(q)}, \quad m_O(t) = \frac{\sum_{q=t+1}^{L-1} qh(q)}{\sum_{q=t+1}^{L-1} h(q)}$$

on $h(q)$ és el nombre de píxels que tenen intensitat q .

Seguint la notació anterior, les funcions de pertinença de I_{Bt} i I_{Ot} vénen donades per

$$\mu_{I_{Bt}} = REF\left(\frac{q}{L-1}, \frac{m_B(t)}{L-1}\right), \quad \mu_{I_{Ot}} = REF\left(\frac{q}{L-1}, \frac{m_O(t)}{L-1}\right)$$

on $I_{Bt} = \{(q, \mu_{I_{Bt}}(q)) | q = 0, \dots, L-1\}$, $I_{Ot} = \{(q, \mu_{I_{Ot}}(q)) | q = 0, \dots, L-1\}$.

2. Per a cada $t \in [0, L-1]$ es construeix un interval de conjunts borrosos a partir de les funcions d'ignorància.

A partir de les funcions de pertinença d'un objecte i del fons obtingudes en el pas anterior, es vol construir un conjunt borrós interval-valorat amb l'objectiu que la longitud dels intervals representi la incertesa de l'expert en l'elecció de la funció de pertinença. La construcció d'aquests intervals ve donada per la proposició següent.

Proposició 3.1.2. *Siguin I_B i I_O els conjunts borrosos associats al fons i objectes de la imatge construïts per un expert respectivament. Sigui IG una funció d'ignorància. Definim $\phi: FS_S(X) \times FS_S(X) \rightarrow IVFS_S(X)$, on $\phi(I_B, I_O) = \{(x, [M_L(x), M_U(x)]) \mid x \in X\}$ tal que $[M_L(x), M_U(x)] = [IG(0.5, 0.5) - IG(\mu_{I_B}(x), \mu_{I_O}(x)), IG(0.5, 0.5)]$. A més es satisfà*

$$W([M_L(x), M_U(x)]) = IG(\mu_{I_B}(x), \mu_{I_O}(x))$$

on W és la llargària de l'interval.

Demostració. Sabem que la llargària de l'interval ve donada per

$$W([M_L(x), M_U(x)]) = |M_U(x) - M_L(x)|.$$

Per definició, $M_U(x) = IG(0.5, 0.5)$ i $M_L(x) = IG(0.5, 0.5) - IG(\mu_{I_B}(x) - \mu_{I_O}(x))$. Substituint, tenim que $|M_U(x) - M_L(x)| = |IG(0.5, 0.5) - IG(0.5, 0.5) + IG(\mu_{I_B}(x) - \mu_{I_O}(x))| = IG(\mu_{I_B}(x) - \mu_{I_O}(x))$. \square

- Es pren com a millor llinard t el valor corresponent al conjunt borrós interval·valorat construït en el pas anterior amb menor valor d'entropia.

S'utilitzarà com entropia la funció definida en [35] i donada per

$$\varepsilon_F(\tilde{A}) = \mathcal{M}_{i=1}^N W([M_{\tilde{A}L}(x_i), M_{\tilde{A}U}(x_i)])$$

amb $x_i \in X$, on \mathcal{M} és l'operador d'agregació corresponent a la mitjana aritmètica.

Notem que si $\mu_{I_{B_i}}(q) \rightarrow 1$, aleshores $REF\left(\frac{q}{L-1}, \frac{m_B(t)}{L-1}\right) \rightarrow 1$, i.e. $q \approx m_B(t)$. Això vol dir que els píxels amb intensitat q són els que tenen una intensitat més propera a la intensitat mitjana del fons. Aquest fet ens indica que el píxel en qüestió pertany al fons de la imatge.

Notem que es té una terna de llinards per a cada expert, que ve caracteritzat per una funció d'ignorància i una funció d'equivalència restringida.

Construcció de les etiquetes lingüístiques que representen cada objecte.

En aquest pas el que feim és introduir un mètode de construcció d'etiquetes lingüístiques per a cada píxel de la imatge. Recordem que l'algorisme es basa només en el color i que per tant, cada regió correspondrà a una etiqueta lingüística que denotarà el color. Així, l'objectiu serà segmentar en 6 clústers que representen els colors primaris i secundaris: cian, magenta, groc, vermell, verd i blau. També cal parar esment a la zona de transició entre el fons i els objectes de cada component, per això tenim la taula de transició 3.1.

Component	Objecte	Fons	Transició
GR	Verd	Vermell	Groc
BG	Blau	Verd	Cian
BR	Blau	Vermell	Magenta

Taula 3.1: Transició de color entre un objecte i el fons.

Per construir les etiquetes lingüístiques ens basarem en els següents valors:

3. MÈTODES DE SEGMENTACIÓ

1. Els llindars (t_{GR}, t_{BG}, t_{BR}) calculats en el tercer pas.
2. La intensitat associada al màxim valor de l'histograma en l'espai AB amb $A, B \in \{R, G, B\}$ a l'esquerra de cada llindar (q_{AB}^{esq}).
3. La intensitat associada amb el màxim valor de l'histograma en l'espai AB , a la dreta del llindar (q_{AB}^{dreta}).

En el cas de tenir més d'un màxim, prendrem sempre el més proper al llindar.

Anem ara a calcular el grau de pertinença d'un píxel a cada etiqueta lingüística. Notem que podem tenir un objecte d'un determinat color definit per components diferents. Per exemple, un objecte blau pot venir definit per la component BG o per la BR . Així, per a cada píxel p amb intensitats de color (q_{GR}, q_{BG}, q_{BR}) el grau de pertinença a cada etiqueta lingüística vendrà definit per:

- $\mu_{\text{vermell}}(q_{GR}, q_{BG}, q_{BR}) = \min\{\mu_R(q_{GR}), \mu_R(q_{BR})\}$,
- $\mu_{\text{groc}}(q_{GR}, q_{BG}, q_{BR}) = \mu_{GR}(q_{GR})$,
- $\mu_{\text{verd}}(q_{GR}, q_{BG}, q_{BR}) = \min\{\mu_G(q_{GR}), \mu_G(q_{BG})\}$,
- $\mu_{\text{cian}}(q_{GR}, q_{BG}, q_{BR}) = \mu_{BG}(q_{BG})$,
- $\mu_{\text{blau}}(q_{GR}, q_{BG}, q_{BR}) = \min\{\mu_B(q_{BG}), \mu_B(q_{BR})\}$,
- $\mu_{\text{magenta}}(q_{GR}, q_{BG}, q_{BR}) = \mu_{BR}(q_{BR})$,

on $\mu_A(q_{AB})$ amb q_{AB} la intensitat en la component AB de l'espai de color (amb llindar de color t_{AB}) d'un píxel qualsevol ve definida de diferent forma segons si pertany al fons, a la zona de transició o a l'objecte de la imatge. La μ del fons ve definida per

$$\mu_B(q_{AB}) = \begin{cases} 1 & \text{si } 0 \leq q_{AB} \leq q_{AB}^{\text{esq}}, \\ \frac{q_{AB}^{\text{esq}} - q_{AB}}{t_{AB} - q_{AB}^{\text{esq}}} + 1 & \text{si } q_{AB}^{\text{esq}} \leq q_{AB} \leq t_{AB}, \\ 0 & \text{si } q_{AB} \geq t_{AB}. \end{cases}$$

La μ de la zona de transició ve definida per

$$\mu_{AB}(q_{AB}) = \begin{cases} 0 & \text{si } 0 \leq q_{AB} \leq q_{AB}^{\text{esq}}, \\ \frac{q_{AB} - q_{AB}^{\text{esq}}}{t_{AB} - q_{AB}^{\text{esq}}} & \text{si } q_{AB}^{\text{esq}} \leq q_{AB} \leq t_{AB}, \\ \frac{t_{AB} - q_{AB}}{q_{AB}^{\text{dreta}} - t_{AB}} + 1 & \text{si } t_{AB} \leq q_{AB} \leq q_{AB}^{\text{dreta}}, \\ 0 & \text{si } q_{AB} \geq q_{AB}^{\text{dreta}}. \end{cases}$$

La μ de l'objecte ve definida per

$$\mu_A(q_{AB}) = \begin{cases} 0 & \text{si } 0 \leq q_{AB} \leq t_{AB}, \\ \frac{q_{AB} - t_{AB}}{q_{AB}^{\text{dreta}} - t_{AB}} & \text{si } t_{AB} \leq q_{AB} \leq q_{AB}^{\text{dreta}}, \\ 0 & \text{si } q_{AB} \geq q_{AB}^{\text{dreta}}. \end{cases}$$

Notem que per a cada expert, tenim graus de pertinença per a cada un dels sis colors diferents.

Construcció del vector d'utilitats associat a cada expert.

A partir dels graus de pertinença obtinguts a la segona passa de l'algorisme i les etiquetes lingüístiques de la passa anterior, el següent pas és construir, per a cada píxel de la imatge, un vector d'utilitats associat a cada expert. El vector d'utilitats de cada píxel ens indica el grau de pertinença respecte a cada color de la imatge. Per calcular aquest vector utilitzem el model lingüístic basat en dues tuples [35, 36, 37]. En aquest model, la informació lingüística ve donada per una 2-tupla (s_i, α_i) on $s_i \in S$, amb S el conjunt d'etiquetes lingüístiques, i $\alpha_i \in [-0.5, 0.5]$. Baix aquesta nomenclatura, tenim que s_i representa l'etiqueta lingüística de la informació i α_i és un valor numèric que expressa la translació simbòlica, és a dir, el valor de la translació entre el resultat original β a l'index i que té etiqueta més propera en el conjunt lingüístic s_i .

El model basat en 2-tuples va suposar una revolució ja que no es perdia informació quan es duia a terme una operació simbòlica. Considerem, a mode d'exemple, una escala lingüística $S = \{\text{dolent}, \text{regular}, \text{bo}\}$ dotada d'un ordre total $\text{dolent} < \text{regular} < \text{bo}$. Suposem que dos experts valoren respectivament una inversió com "regular" i "bona" i volem obtenir la mitjana d'aquestes valoracions. Identificant S amb l'escala $\{0, 1, 2\}$ podem calcular la mitjana com $\beta = \frac{1+2}{2} = 1.5$ però ara no es sap què significa 1.5. En els primers models lingüístics s'arrodonia a l'etiqueta de S més propera "bo" però així es perdia informació. El model de 2-tuples evita aquesta pèrdua introduint el concepte de translació simbòlica α_i . En l'exemple anterior, les valoracions inicials serien (regular, 0) i (bo, 0) obtenint com a valor mitjà (bo, -0.5). Vegem a continuació les definicions formals.

Definició 3.1.8 ([37]). *Sigui $S = \{s_0, \dots, s_g\}$ un conjunt d'etiquetes lingüístiques i $\beta \in [0, g]$ un valor que representa el resultat d'una operació d'agregació simbòlica, aleshores la 2-tupla que expressa la informació equivalent a β s'obté a partir de la funció $\nabla : [0, g] \rightarrow S \times [-0.5, 0.5]$ amb*

$$\nabla(\beta) = (s_i, \alpha) \text{ amb } \begin{cases} s_i & i = \text{arrodonir}(\beta), \\ \alpha = \beta - i & \alpha \in [-0.5, 0.5], \end{cases}$$

on arrodonir és l'operació habitual d'arrodonir, s_i és l'etiqueta de l'index més proper a β i α és el valor de la translació simbòlica.

El següent resultat és evident.

Proposició 3.1.3 ([37]). *Sigui $S = \{s_0, \dots, s_g\}$ un conjunt lingüístic i (s_i, α) una 2-tupla. Sempre existeix una funció ∇^{-1} tal que donada una 2-tupla (s_i, α) retorna el seu nombre equivalent $\beta \in [0, g]$. Concretament, $\nabla^{-1} : S \times [-0.5, 0.5] \rightarrow [0, g]$ donada per*

$$\nabla^{-1}(s_i, \alpha) = i + \alpha = \beta.$$

Posteriorment, el que es farà serà transformar el grau de pertinença de cada objecte o color amb un model basat en etiquetes de preferències lingüístiques.

Suposem ara que tenim $S = \{s_0, \dots, s_{14}\}$ un conjunt d'etiquetes lingüístiques amb un ordre total establert entre elles. A la figura 3.1(a) tenim la funció de pertinença que representa el color magenta (μ_{BR}). A partir de les figures, podem veure que $\mu_{BR}(90) = 1$ (figura 3.1(a)) i per la figura 3.1(b) tenim que l'etiqueta que té associat el valor 1

correspon exactament amb l'etiqueta s_{14} ja que, mirant horitzontalment, aquests valors estan completament alineats. Per $\mu_{BR}(65)$, podem veure que el valor del conjunt borrós que el caracteritza és 0.3 (veure figura 3.1(a)) però, a la figura 3.1(b) no hi ha cap etiqueta que estigui a la mateixa altura al valor 0.3. En aquest cas, el valor 0.3 es troba entre les etiquetes s_4 i s_5 , essent més proper a s_4 . Ara la diferència d'altura entre el valor 0.3 i s_4 és aproximadament 0.2. Per això, la 2-tupla associada és $(s_4, 0.2)$.

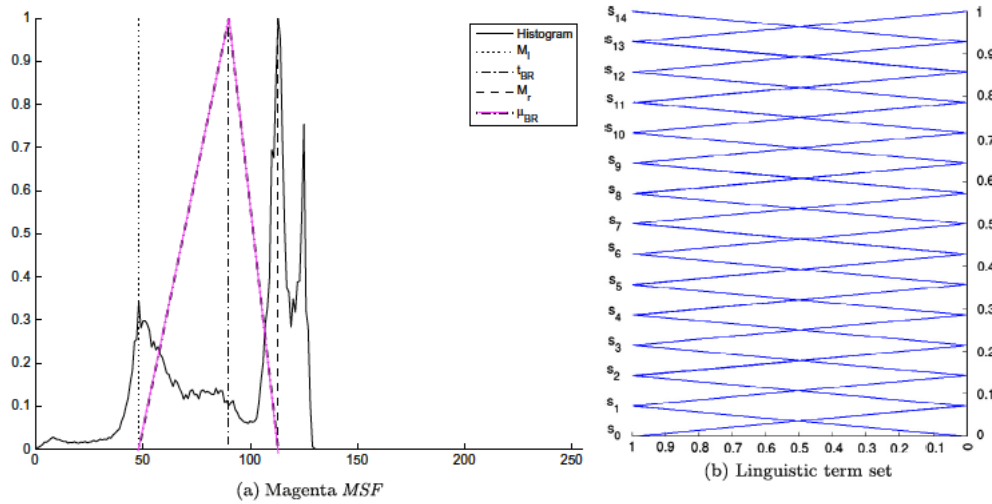


Figura 3.1: Representació de les preferències del color magenta segons un expert. En la figura (a) trobem l'histograma de color, així com els llinars i el conjunt borrós caracteritzat per μ_{BR} (el grau de pertinença al color magenta), mentre que en la figura (b) tenim el conjunt d'etiquetes lingüístiques en l'eix vertical esquerre, els corresponents valors numèrics en $[0, 1]$ en l'eix vertical dret i els distintes conjunts borrosos que caracteritzen cadascuna de les etiquetes lingüístiques en el centre. Aquesta figura s'ha extret de [4].

Acabada aquesta passa per a cada píxel de la imatge, construïm un vector d'utilitats associat a cada expert que conté les preferències lingüístiques en forma de 2-tuples a cadascun dels 6 colors considerats.

Procés de presa de decisions.

En aquesta darrera passa, s'obindrà una imatge segmentada segons un model de presa de decisions basat en la representació lingüística de dues tuples. Ara per a cada píxel agregarem els vectors d'utilitats donats pels experts. Després incorporarem a cada un dels píxels la informació donada pel vector d'utilitats dels veïns del píxel i finalment, classificarem cada píxel obtenint així una imatge segmentada.

Per calcular l'agregació dels vectors d'utilitats donats pels experts per un píxel concret, utilitzem l'operador LOWA de dues tuples (ϕ_{2t}) [38].

Definició 3.1.9. Siguin $\{a_i, a_j\}$ dues etiquetes de $S = \{a_1, \dots, a_T\}$. Sigui w tal que $0 \leq w \leq 1$. L'operador de LOWA ϕ_{2t} ve definit per

$$\phi_2(a_i, a_j; w) = a_k$$

on $k = \min \{T, \min\{i, j\} + \text{arrodonir}(w \cdot (\max\{i, j\} - \min\{i, j\}))\}$.

Recursivament podem definir

$$\phi_{n+1}(a_1, \dots, a_{n+1}; W) = \phi_2(a_{(1)}, \phi_n(a_{(2)}, \dots, a_{(n+1)}; W'); w_1)$$

on $W = \{w_i\}_{i=1, \dots, n}$ és el vector de pesos, $W' = \{w'_i\}_{i=1, \dots, n-1}$ amb $w'_i = \frac{w_{i+1}}{\sum_{i=2}^n w_i} i a_{(1)}, \dots, a_{(n+1)}$ són les etiquetes ordenades en ordre decreixent.

L'operador LOWA genera una mitjana ordenant primer els elements que es volen agregar assignant pesos segons l'ordenació. Resta definir com es generen els pesos. Una de les formes habituals és emprant un quantificador lingüístic per agregar la informació [39].

Un quantificador proporcional no decreixent és una funció $\mathcal{P} : [0, 1] \rightarrow [0, 1]$ amb $\mathcal{P}(0) = 0$, i tal que existeix $r \in [0, 1]$ amb $\mathcal{P}(r) = 1$. Existeixen diversos quantificadors. En aquest cas s'ha emprat l'anomenat quantificador lingüístic "com a mínim la meitat" que ve donat per

$$\mathcal{P}(r) = \begin{cases} 0 & \text{si } r < 0, \\ \frac{r}{0.5} & \text{si } 0 \leq r \leq 0.5, \\ 1 & \text{si } r > 0.5, \end{cases}$$

amb $r \in [0, 1]$. Donat el quantificador \mathcal{P} , el vector de pesos es pot calcular mitjançant

$$w_i = \mathcal{P}\left(\frac{i}{n}\right) - \mathcal{P}\left(\frac{i-1}{n}\right).$$

Quan s'ha agregat per cada píxel la informació de tots els experts, cada píxel té associats sis valors de pertinença, un per a cada color. Després, per intentar obtenir una millor segmentació, definim una dimensió de finestra que ens definirà quins són els veïns de cada píxel i s'agrega el vector de cada píxel amb els vectors dels seus veïns utilitzant ϕ_{2t} . En haver fet aquestes unions, cada píxel té associat un vector que té en compte l'opinió de tots els experts i que es veu influenciat pels seus veïns.

Finalment, cada píxel és classificat amb el color corresponent al màxim valor de pertinença dels sis possibles, segons l'ordre lexicogràfic.

Definició 3.1.10 (Ordre lexicogràfic). *Siguin (s_i, α_i) i (s_j, α_j) dues 2-tuples, cada una representant la informació com segueix:*

- Si $s_i < s_j$, (s_i, α_i) és menor que (s_j, α_j) .
- Si $s_i = s_j$:
 1. Si $\alpha_i = \alpha_j$ aleshores (s_i, α_i) , (s_j, α_j) representen la mateixa informació.
 2. Si $\alpha_i < \alpha_j$ aleshores (s_i, α_i) és menor que (s_j, α_j) .

Si el valor màxim de pertinença correspon a més d'un color, es pot assignar a qualsevol d'ells.

Donada la gran quantitat d'experts i de colors es fa difícil mostrar tots els resultats. Per això podem veure, per exemple, a la figura 3.2 quines són les etiquetes de color de cada píxel fixant un expert.

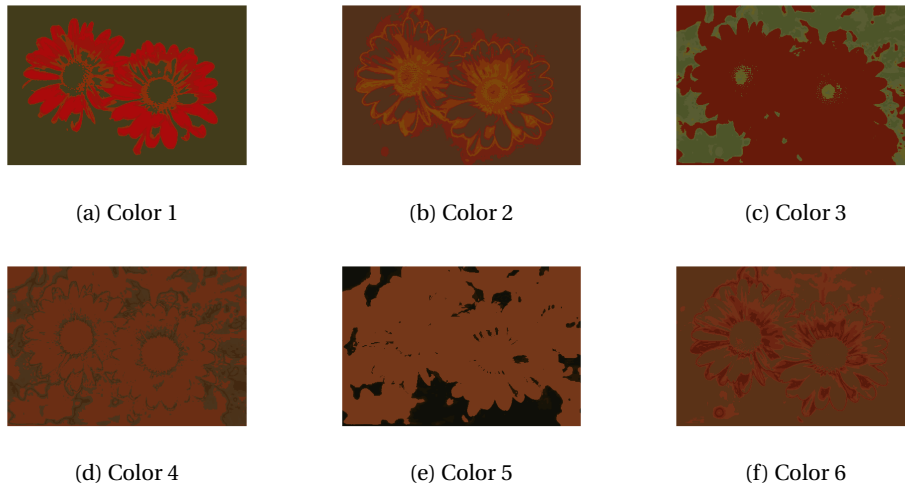


Figura 3.2: Resultats de cada color segons el primer expert.

3.1.3 Modificacions de l'algorisme

Tal i com acabam de veure, aquest algorisme sempre ens retorna una segmentació amb sis clústers, però aquest no sempre és el nombre òptim per a totes les segmentacions. Per això hem modificat l'algorisme anterior per tal de poder realitzar segmentacions amb menys de sis clústers.

Per fer-ho hem calculat totes les possibles formes d'unir els sis clústers que ens dona la segmentació. Per exemple, si tenim 3 clústers (enumerats de l'1 al 3) les possibles formes d'unir-los serien: per tenir tres clústers, [1, 2, 3], que és el resultat original; per tenir-ne dos [1-2,3], [1,2-3], [1-3,2] i per tenir-ne un [1-2-3]. Això és el que coneixem com particions sense repetició i per 6 clústers el nombre de possibles combinacions és

$$\left\{ \begin{matrix} 6 \\ 6 \end{matrix} \right\} + \left\{ \begin{matrix} 6 \\ 5 \end{matrix} \right\} + \left\{ \begin{matrix} 6 \\ 4 \end{matrix} \right\} + \left\{ \begin{matrix} 6 \\ 3 \end{matrix} \right\} + \left\{ \begin{matrix} 6 \\ 2 \end{matrix} \right\} + \left\{ \begin{matrix} 6 \\ 1 \end{matrix} \right\} = 203,$$

on $\left\{ \begin{matrix} m \\ n \end{matrix} \right\}$ és el nombre d'Stirling amb m els clústers (distingibles) i n la quantitat que en volem (indistingibles). Dit d'altra forma, són totes les possibles formes d'agrupar 6 clústers amb 6, 6 clústers amb 5, etc..

En total tendrem 203 segmentacions i, per cada una d'elles, calcularem el valor de SSC. Després prendrem el mínim per a cada nombre de clústers, i per tant tendrem 6 mínims, i posteriorment aplicam el mètode del colze amb funció de cost SSC per determinar el nombre de clústers òptim. Així, la segmentació que hagi tengut SSC mínim amb aquest nombre de clústers serà la segmentació de la imatge en qüestió. El codi de la implementació es pot trobar a l'annex A.1.

3.2 Mètode de segmentació jeràrquica

El mètode de segmentació jeràrquica [5] està basat en el creixement de regions tot i que també s'utilitzen tècniques pròpies dels mètodes basats en el llinard.

L'objectiu és identificar les regions d'interès evitant segmentar les textures de la imatge, ja que això és un procés molt difícil i costós. Així, el que es farà serà segmentar les imatges utilitzant regions d'un color homogeni.

Aquest mètode consta de dos passos diferenciats: quantificació del color i segmentació espacial. En el primer pas, es quantifiquen els colors de la imatge en diferents classes representatives que s'utilitzen per etiquetar les regions de la mateixa. Després es canvia el color de cada píxel pel color de l'etiqueta de la seva classe, construint així un mapa de classes de la imatge. Aquest procés es repetirà a cada una de les regions d'aquest mapa de classes obtenint-se, en cada una d'elles, a una J -imatge. Aquesta J -imatge és a la vegada un mapa de classes per a cada regió, on els valors majors i menors són els candidats a contorns i textures de la imatge respectivament. Aquests valors són anomenats J -valors i es calculen a partir d'una regió del mapa de classes. La segona passa es centra en la segmentació espacial, proposant un criteri de bona segmentació utilitzant les J -imatges. Finalment, utilitzant un mètode de creixement de regions, es segmentarà la imatge en base a les J -imatges obtingudes.

Els passos que duu a terme aquest algorisme de segmentació es poden veure en el pseudocodi de l'algorisme 2. Vegem en detall en què consisteix cada una de les passes.

Algorisme 2: Segmentació jeràrquica

Entrada: Imatge que es vol segmentar, llistat de quantificació del color, nombre màxim d'escala desitjades (màxim 4), llistat d'unió de regions.

Sortida : Imatge segmentada.

- 1 *Quantificació del color:*
 - 2 └─ Construir el mapa de classes ;
 - 3 *Segmentació espacial:*
 - 4 └─ **per cada escala fer**
 - 5 └─ **per cada regió fer**
 - 6 └─ Calcular J -valors;
 - 7 └─ Creixement de regions;
 - 8 └─ Determinar les valls;
 - 9 └─ Creixement de valls;
 - 10 └─ Segmentar regions;
 - 11 Unir les regions;
-

3.2.1 Quantificació del color

L'objectiu és identificar alguns colors significatius de la imatge per tal de diferenciar els diferents objectes que s'hi poden trobar. Per això s'utilitza l'algorisme de quantificació de color *Peer Group Filtering and Perceptual Color Image Quantization* [40] basat en la percepció dels humans, ja que aquesta és més sensible a canvis en les textures d'una imatge. Aquest mètode indica el conjunt de píxels de la imatge que tenen el mateix color, que els anomenarem classe de color. El que es fa és assignar les etiquetes segons la quantificació del color, de tal forma que cada classe serà un conjunt de píxels que

han estat etiquetats amb el mateix color. Després es canvia el color de cada píxel pel color corresponent a l'etiqueta de la seva classe, obtenint una nova imatge anomenada mapa de classe. Així es calcularan ara els J -valors.

Cada regió del mapa de classes està formada per un conjunt de píxels d'una mateixa classe de color. Sigui doncs X el conjunt dels N punts del mapa de classe. Es considera $x = (x_1, x_2)$ amb $x \in X$ les coordenades espaials i $m = \frac{1}{N} \sum_{x \in X} x$ la mitjana. Suposem que hem classificat X en K classes, tenint diferents conjunts que anomenarem C_i amb $1 \leq i \leq K$, i denotem per m_i la mitjana dels N_i punts de la classe C_i . Siguin

$$S_T = \sum_{x \in X} \|x - m\|^2,$$

i

$$S_W = \sum_{i=1}^K S_i = \sum_{i=1}^K \sum_{x \in C_i} \|x - m_i\|^2.$$

Llavors es defineix com a J -valor de la imatge al valor J donat per

$$J = \frac{S_T - S_W}{S_W}.$$

Per exemple, en la figura 3.3 tenim tres mapes de classes, on en el primer, una bona segmentació estaria formada per 3 regions, en el segon ja és uniforme per tant, no es necessiten regions i en el tercer en tendríem 2.

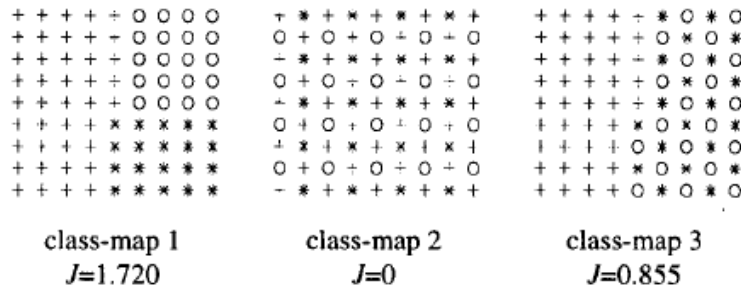


Figura 3.3: Exemple de diferents mapes de classe i els corresponents J -valors. Els símbols +, o i * defineixen tres classes diferents. Imatge extreta de l'article [5].

Es pot observar que si tenim una imatge amb diferents regions homogènies, les classes de color estaran més separades entre elles per tant, el valor de J serà major. Per contra, si les classes de color estan distribuïdes uniformement al llarg de la imatge, J tendirà a ser petit.

A continuació, es calcularà un valor de J per a cada una de les regions del mapa de classes que es considera com una imatge en la que es pot executar novament l'algorisme de quantificació de color abans esmentat. Sigui J_i amb $1 \leq i \leq K$ el valor local de J calculat per a cada una de les regions, N_i amb $1 \leq i \leq K$ el nombre de píxels de la regió i -èssima i N el nombre total de píxels del mapa de classe. Denotarem per \bar{J} la mitjana de tots els J -valors, que ve doanda per

$$\bar{J} = \frac{1}{N} \sum_{i=1}^K N_i J_i.$$

L'objectiu és ara minimitzar el valor de \bar{J} , ja que si tenim una bona segmentació, els valors de J_i són petits i per tant, també de \bar{J} . Notem que el valor mínim de \bar{J} és 0, ja que $J_i \geq 0$ i $N_i \geq 0$. Notem també que el J definit anteriorment és un cas particular de \bar{J} que seria quan tenim només una regió en el mapa de classes. Així, recuperant l'exemple anterior els resultats dels J -valors de cada regió i el valor \bar{J} es poden trobar a la figura 3.4.

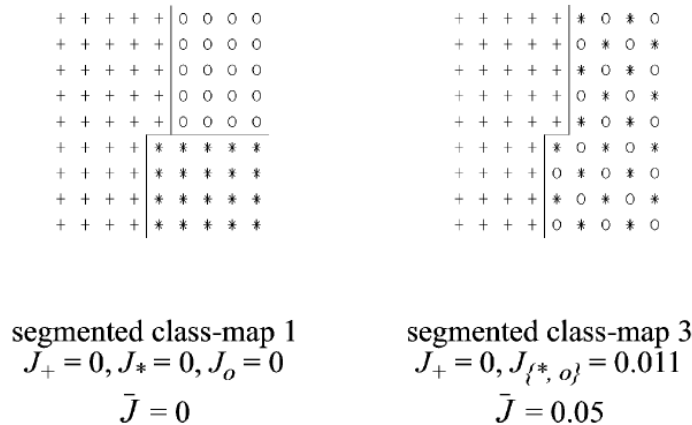


Figura 3.4: J -valors de cada regió del mapa de classes (on el subíndex defineix la classe de la regió) i valor de \bar{J} .

3.2.2 Segmentació espacial

Com ja sabem, hi ha moltes formes diferents de segmentar una imatge i un dels problemes més difícils és trobar la segmentació corresponent al mínim global \bar{J} . Per això el que es farà és trobar la imatge que, per cada regió i , tenguí com a valor associat del mapa de classe el J_i local calculat anteriorment. La imatge que en cada regió satisfà aquesta característica és la que anomenem J -imatge. Així, els procés que es durà a terme a continuació es realitzarà per a cada una de les regions J_i del mapa de classes. Aquesta J -imatge, es pot entendre com un mapa en 3 dimensions, que conté valls i muntanyes, on les valls representen els interiors d'una regió i les muntanyes les fronteres entre regions. Això és degut a que com més gran és el valor de J , més probable és que ens trobem propers a una frontera entre regions.

Per calcular el valor local de J es considerarà una finestra. La dimensió de les finestres determinarà la dimensió de les regions de la imatge que es poden detectar. Les finestres petites són útils per localitzar la intensitat o color de les arestes, mentre que finestres grans són útils per detectar contorns entre les textures.

Les característiques de les J -imatges permeten dur a terme un procés de creixement de regions segons els detalls que es volen obtenir en la segmentació. Per determinar aquesta simplicitat, s'utilitzarà un paràmetre d'entrada en el que es determina el nombre d'escala de la segmentació. Com més petita sigui l'escala, més petites seran les regions i per tant, més regions diferents es tenen. Per contra, com més gran sigui, menys regions es tenen i per tant, menys detalls es poden apreciar en la segmentació.

3. MÈTODES DE SEGMENTACIÓ

Si el nombre d'escala és, per exemple, 4 l'algorisme comença fent la segmentació amb escala 4, després 3, etc..

El procés de creixement de regions consta principalment de tres passos: determinar les valls, creixement d'aquestes i unió de regions.

1. Per determinar les valls s'elegeix inicialment un conjunt de píxels com a representants de les àrees que després es faran créixer. Aquestes àrees seran les que tenen un valor local de J mínim. En general, trobar el millor conjunt de valls no és un problema fàcil. Per això s'utilitza el següent algorisme:
 - a) Calcular la mitjana i la desviació estàndard dels valors J_i resultants de calcular el J sobre cada regió, que denotarem per μ_J i σ_J respectivament.
 - b) Es defineix el llindar com $T_J = \mu_J + a\sigma_J$ on a és el valor d'entre $\{-0.6, -0.4, -0.2, 0, 0.2, 0.4\}$ que dona el major nombre de valls. Els píxels amb valor local de J menor que T_J són punts candidats a valls. Aquests es connecten segons la 4-connectivitat i s'obtenen les regions candidates a valls.
 - c) Si una regió candidata a vall té dimensió major que el mínim de píxels per regió segons l'escala corresponent definida a la taula 3.2, serà definida com una vall.

Escala	Dimensió finestra	dimensió de la regió	mínim píxels per regió
1	9 × 9	64 × 64	32
2	17 × 17	128 × 128	128
3	33 × 33	256 × 256	512
4	65 × 65	512 × 512	2048

Taula 3.2: Dimensió de les finestres i regions i mínim de píxels que ha de tenir una regió per ser considerada regió.

2. Ara les noves regions han de sorgir a partir de les valls que s'han determinat anteriorment. Aquest és un procés molt lent si s'ha de fer per cada píxel individualment. Per això es segueix el següent algorisme:
 - a) En primer lloc, els píxels interiors a una vall que no formen part d'ella d'acord al procés anterior s'assignen a aquesta.
 - b) A continuació, es calcula la mitjana dels J -valors de les regions no assignades a cap vall. Aquells píxels amb J -valor per davall d'aquesta mitjana formaran regions de creixement i, si són adjacents a una i només una vall, se'ls hi assigna aquesta.
 - c) Calcular el valor local de J pels píxels restants a l'escala més petita posterior per localitzar amb més precisió les fronteres (o muntanyes) i repetir el pas anterior.
 - d) Creixement dels píxels restants. Els píxels no classificats en cap dels contorns, es guarden a una pila. Paral·lelament, el píxel amb menor valor local de J és assignat a la vall adjacent i s'actualitza la pila fins que tots els píxels estiguin classificats.

3. Finalment, com que el procés anterior tendeix a segmentar massa la imatge, s'uneixen les regions segons la semblança dels seus colors. Per això, es calcula l'histograma de color de cada regió i la distància entre dos histogrames i i j segons

$$D(i, j) = \|P_i - P_j\|,$$

on P és el vector de color de l'histograma. S'utilitza la distància Euclídea ja que hi ha poca correlació entre els dos histogrames de color.

Per unir les regions [41] el que es farà és guardar a una taula les distàncies entre els histogrames de color de regions veïnes. Després s'uneixen les regions que tenen aquesta distància mínima i es calcula l'histograma de color de la nova regió, actualitzant també la taula de distàncies. Aquest procés es repeteix fins que s'arriba al llindar màxim de les distàncies, que és l'altre paràmetre que introdueix l'usuari.

Si apliquem el mètode de segmentació jeràrquica a una imatge, podem veure a la figura 3.5 quins són els resultats obtinguts quan finalitzem cada un dels subprocessos.

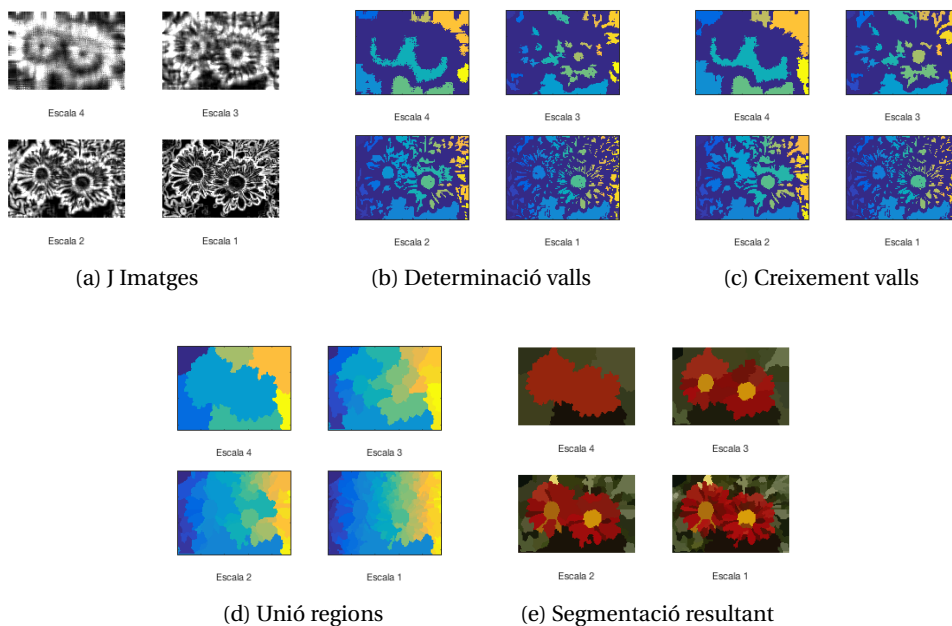


Figura 3.5: Passes de l'algorisme JSEG.

3.3 K-means

El k-means [6] és un mètode d'agrupament que té com objectiu fer una partició d'una imatge en K conjunts diferents que anomenem clústers. Els píxels d'un clúster tenen com a particularitat que pertanyen al clúster que té un valor mitjà dels colors més proper a ell. És a dir, si tenim X el conjunt dels N píxels de la imatge i C el conjunt de K

centroides, l'objectiu del mètode és minimitzar la funció objectiu

$$J = \sum_{i=1}^N \min_{c_j \in C} \|x_i - c_j\|^2 \quad (3.1)$$

on $x_i \in X$ amb $i \in \{1, \dots, N\}$ és el píxel i -èssim i $c_j \in C$ amb $j \in \{1, \dots, K\}$ el centre del clúster j -èssim.

Inicialment s'elegeix el nombre de clústers que es volen obtenir en la segmentació de la imatge i es trien els centroides inicials de cada clúster segons el criteri de l'algorisme K-means++ [6].

1. Elegir un píxel aleatòriament entre tots els de la imatge que serà el primer centre (c_1).
2. Per $2 \leq j \leq K$:

Sigui c_j el resultat de l'experiment aleatori amb variable aleatòria $X =$ píxel de la imatge, on $X = \{x_1, \dots, x_N\} \setminus \{c_1, \dots, c_{j-1}\}$ amb x_i amb $1 \leq i \leq N$ els píxels de la imatge i c_j amb $1 \leq j \leq K$ els píxels que ja s'han elegit com centroides, amb funció de densitat

$$P(X = x_i) = \frac{d^2(x_i, c_{x_i})}{\sum_{x \in X} d^2(x, c_x)}$$

on c_{x_i} és el centre més proper a x_i i d la distància Euclídea.

Quan ja es tenen els centroides, es classifica cada píxel per tal que pertanyi al clúster amb un centroe més proper a ell. Posteriorment s'actualitzen els centroides de cada clúster i es repeteix el procés fins que no hi hagi canvis en l'assignació dels píxels als clústers, fins que la diferència del valor de la funció objectiu 3.1 entre una iteració i la següent sigui menor que un cert llinar o fins que s'hagi arribat a un màxim d'iteracions.

Aquest mètode no garanteix que l'SSC (veure definició 2.2.2) sigui mínim, és a dir, no garanteix que la suma de les distàncies entre cada píxel i el centroe del seu clúster sigui un mínim global. Per tant, no podem assegurar que la imatge segmentada sigui l'òptima segons aquest criteri. Aquest fet és degut a la inicialització del procés, ja que començar amb uns píxels o amb altres pot canviar totalment els resultats finals. Per això el que es farà serà repetir el mètode de k-means un nombre suficientment gran de vegades, per així disminuir el caire d'aleatorietat de les condicions inicials, i quedar-nos amb la segmentació que tenguí un menor valor de SSC.

Un dels paràmetres que s'ha d'introduir quan s'inicialitza el procés és el nombre de clústers en què es vol segmentar la imatge. Per a cada imatge, depenent de les seves característiques, serà més adient utilitzar un nombre més gran o més petit. Per determinar aquest nombre de clústers, utilitzarem el mètode del colze, on la funció de cost és la funció SSC definida anteriorment.

Les passes que es duran a terme en aquest algorisme, es poden veure en el pseudocodi de l'algorisme 3.

La funció k-means està ja implementada en MATLAB, per això l'utilitzarem com a part de l'algorisme. L'elecció automàtica del nombre de clústers i el nombre d'execucions del k-means per reduir l'aleatorietat s'han implementat des de zero. El codi de la implementació es pot trobar a l'annex A.2.

Algorisme 3: K-means

Entrada: Imatge en RGB, nombre màxim de clústers, nombre màxim d'execucions.

Sortida : Imatge segmentada.

```

1 per i=1 : màxim clústers fer
2   per j=1: màxim d'execucions fer
3     Funció k-means:
4       Inicialitzem els centres i els clústers;
5       mentre la distància entre els píxels i centroides no convergeixi o
        no s'hagi arribat al nombre màxim d'iteracions fer
6         Reassignem els píxels als clústers ;
7         Actualitzem centroides;
8       si Segmentada té SSC mínim per 'i' clústers aleshores
9         Solució(i) = Segmentada;
10 Mètode del colze aplicat al vector solució;

```



(a) 2 clústers



(b) 3 clústers



(c) 4 clústers



(d) 5 clústers



(e) 6 clústers

Figura 3.6: Resultats de l'algorisme variant el nombre de clústers.

En la figura 3.6 es poden veure les diferències que podem obtenir segons el nombre de clústers que tenguim la nostra imatge.

A més del nombre de clústers i de la situació inicial dels centres, un altre fet que hem de tenir en compte a l'hora d'analitzar els resultats és el nombre d'iteracions que ha fet l'algorisme. Durant la realització de les segmentacions, s'ha utilitzat com nombre màxim d'iteracions el que ve per defecte, que és 100. Si fixem el nombre de clústers en 3, en la figura 3.7 es pot veure com va convergint l'algorisme a mesura que es realitzen més iteracions.



(a) 1 Iteració ($SSC = 5.1575 \cdot 10^8$). (b) 3 Iter. ($SSC = 3.7893 \cdot 10^8$). (c) 5 Iter. ($SSC = 3.6188 \cdot 10^8$).



(d) 10 Iter. ($SSC = 3.5518 \cdot 10^8$). (e) 25 Iter. ($SSC = 3.5479 \cdot 10^8$). (f) 100 Iter. ($SSC = 3.5479 \cdot 10^8$).

Figura 3.7: Resultats de l'algorisme variant el nombre màxim d'iteracions per 3 clústers.

3.4 Fuzzy k-means

Tal i com acabem de veure, amb el mètode k-means tenim que un píxel pertany només a la regió que té centre més proper a ell és a dir, és un algorisme d'agrupació forta. En canvi, el fuzzy k-means [7] és un mètode d'agrupament borrós. Això provoca que un píxel pertanyi a cada una de les regions amb un cert grau de pertinença.

Aquest mètode es basa en minimitzar la funció objectiu

$$J_m = \sum_{i=1}^N \sum_{j=1}^K \mu_{ij}^m \|x_i - c_j\|^2, \quad (3.2)$$

on N és el nombre total de píxels, K el nombre total de clústers, $m \in \mathbb{R}^+$ el grau de superposició (que prendrem per defecte $m = 2$) i que determina el grau de borrositat de l'agrupament, x_i el píxel corresponent a la posició i -èsima, c_j el centre del clúster j i μ_{ij} el grau de pertinença amb $0 \leq \mu_{ij} \leq 1$ i $\sum_{j=1}^K \mu_{ij} = 1 \forall 1 \leq i \leq N$ del píxel x_i al clúster j . Notem que quan $m = 0$, no tenim superposició entre les regions, per tant, recuperariem així el mètode k-means vist anteriorment.

L'algorisme en pseudocodi que hem seguit per aquest mètode és pot veure a l'algorisme 4.

Igual que amb el mètode k-means, aquest també està implementat en MATLAB i és peça clau en l'algorisme implementat. Anàlogament al k-means, s'ha afegit el mètode del colze i l'execució múltiple del fuzzy k-means per reduir l'aleatorietat. El codi complet en MATLAB es pot trobar a l'annex A.3. L'algorisme que segueix el fuzzy k-means implementat en MATLAB és el següent [42]:

1. Inicialitzar aleatòriament la pertinença de cada píxel als clústers (μ_{ij}). Un mètode habitual d'inicialització és

$$\mu_{ij} = \frac{e^{-\|x_i - c_j\|^2}}{\sum_{t=1}^K e^{-\|x_i - c_t\|^2}}.$$

Algorisme 4: Fuzzy k-means

Entrada: Imatge en RGB, nombre màxim de regions, nombre màxim d'iteracions.

Sortida: Imatge segmentada.

```

1 per i=1 : màxim regions fer
2   per j=1: màxim iteracions fer
3     Funció Fuzzy k-means:
4       Inicialització;
5       mentre funció objectiu no convergent fer
6         Calcular centres;
7         Actualitzar valor pertinença;
8         Calcular funció objectiu;
9     si Segmentada té error mínim per 'i' regions aleshores
10      Solució(i) = Segmentada;
11 Mètode del colze;

```

2. Mentre la funció objectiu no convergeixi:

a) Calcular els centres dels clústers

$$c_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m}.$$

b) Actualitzar el valor de pertinença

$$\mu_{ij} = \frac{1}{\sum_{k=1}^K \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{1}{m-1}}}.$$

c) Calcular la funció objectiu J_m (3.2).

L'algorisme aturarà quan la funció objectiu es vegi modificada menys d'un llinard establert inicialment entre una iteració i la següent o quan s'arribi a un nombre màxim d'iteracions fixat prèviament. Tant el nombre d'iteracions màximes com el llinard, poden ser elegits inicialment en un vector d'opcions que passarem a la funció ja implementada en MATLAB. Per les segmentacions, hem elegit com a nombre d'iteracions màximes 25 i com a llinard 0.001, ja que per a més iteracions la disminució del valor de la funció objectiu és molt petita i fer més iteracions suposa un gran cost computacional.

Aquest és també un mètode aleatori, ja que amb valors inicials diferents podem obtenir resultats diferents. Per això realitzarem cada segmentació 10 vegades per intentar minimitzar l'error que es pugui produir. Igual que amb els mètodes anteriors, realitzarem la segmentació variant el nombre de clústers entre 1 i 6 i després aplicarem el mètode del colze per obtenir la segmentació òptima.

En la seqüència d'imatges 3.8, es pot veure quins resultats obtindríem per una segmentació de 6 regions segons el nombre d'iteracions realitzades.

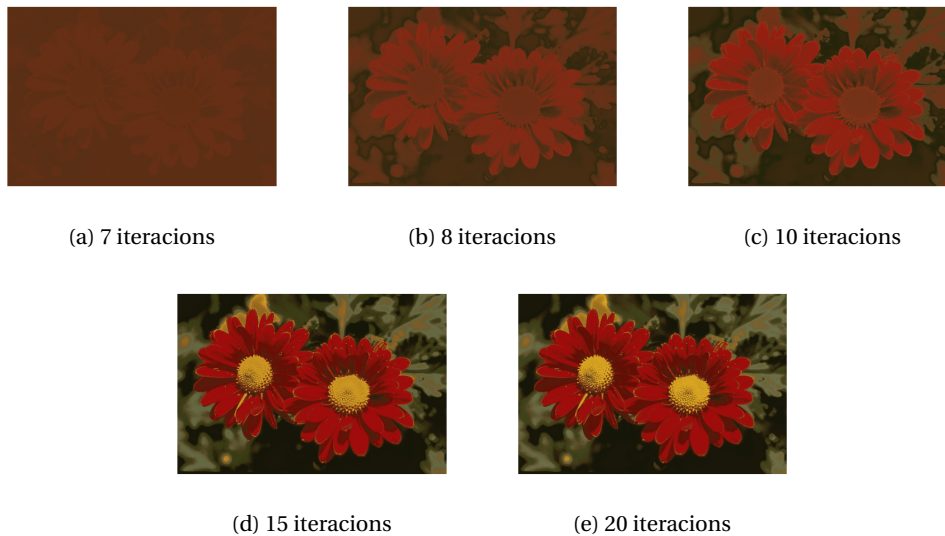


Figura 3.8: Resultats obtinguts per 6 regions segons el nombre d'iteracions.

3.5 Recursive shortest spanning tree

En aquest mètode es considera la imatge com un graf no dirigit amb pesos on cada píxel o regió representa els nodes del graf i les arestes amb pesos la informació de la segmentació. El resultat de la segmentació serà l'obtingut en el moment de tallar el graf segons un criteri de finalització que veurem més endavant. L'algorisme d'aquest mètode es pot resumir amb els passos que es mostren en l'algorisme 5. Anem a veure cada un dels passos amb més detall.

Algorisme 5: RSST

Entrada: Imatge que es vol segmentar.

Sortida: Imatge segmentada.

- 1 *Convertir la imatge amb un graf;*
 - 2 **mentre** *nombre de nodes > 1 fer*
 - 3 *Llevar l'aresta que té pes mínim;*
 - 4 *Actualitzar el graf;*
 - 5 *Actualitzar els pesos;*
 - 6 **si** *Menor que 25 aleshores*
 - 7 *Guardar cost acumulat;*
 - 8 *Guardar segmentació;*
 - 9 *Mètode del colze;*
-

Convertir la imatge amb un graf.

Inicialment es considera que cada píxel de la imatge representa una regió diferent així, quan s'inicialitza l'algorisme, el graf tindrà tants nodes com píxels tenguí la imatge. La

connexió entre dos píxels seran les arestes i es considera que dos píxels estan connectats si es troben en connectivitat 8 amb la imatge [43]. Els pesos inicials de les arestes vénen donats per

$$C(x_i, x_j) = \|c_i - c_j\|_2^2 \cdot \frac{1}{n_{img}} \cdot \frac{1}{2},$$

on x_i, x_j són els píxels que estan connectats amb vectors en l'espai de color c_i i c_j respectivament i n_{img} és la dimensió de la imatge original.

Llevar l'aresta que té pes mínim.

El vector C calculat anteriorment, conté els pesos de totes les arestes. El que es fa és mirar quins són els dos nodes que estan connectats per l'aresta de menor cost i s'ajunten formant un nou node i eliminant els dos anteriors. D'aquesta forma, el que es fa és ajuntar en cada iteració els dos nodes més semblants.

Després s'actualitza el graf que tindrà un node menys. A més, també s'hauran de modificar les arestes ja que si un altre node estava connectat amb algun dels dos nodes que s'han juntat, ara estarà connectat amb el nou node. Posteriorment el que es farà serà actualitzar el centre de la nova regió amb la mitjana dels vectors en l'espai de color dels píxels que formen part de la regió. Finalment es recalculen els pesos de les arestes que s'han modificat en base a la següent fórmula:

$$C(R_i, R_j) = \|\bar{c}_i - \bar{c}_j\|_2^2 \cdot \frac{1}{n_{img}} \cdot \frac{n_i n_j}{n_i + n_j},$$

on R_i, R_j són els nodes que estan connectats, \bar{c}_i i \bar{c}_j són els seus centres i n_i i n_j són les dimensions respectives.

A més, cada vegada que s'ajunten 2 nodes, calcularem quin ha estat el cost acumulat d'ajuntar-los, és a dir,

$$C_{acum}(t) = \begin{cases} \sum_{n=t}^{N_I-1} C_{unir}(n) & \text{si } 1 \leq t < N_I, \\ 0 & \text{altrament,} \end{cases}$$

on $C_{unir}(n)$ és el cost de reduir el nombre de nodes de $n+1$ a n és a dir, el pes de l'aresta amb pes mínim de cada iteració, i N_I el nombre de regions inicial.

En les primeres iteracions de l'algorisme, el cost acumulat es veurà modificat molt lentament. Per això considerarem $N_I = 25$, ja que per les imatges que tractarem i amb la finalitat en què ho feim no té sentit considerar una segmentació de més de 25 clústers.

Determinació de la segmentació òptima

Arribats en aquest pas tenim la funció C_{acum} amb els costos acumulats. Per saber quin és el nombre de regions adequat per obtenir una bona segmentació aplicarem el mètode del colze a aquesta funció i així sabrem quina és la segmentació òptima de totes les que calcula l'algorisme i quants clústers té.

El codi de la implementació es pot trobar a l'annex A.4.

Aquest mètode, a diferència dels que havíem vist fins ara, té les regions connexes. Com podem veure a la figura 3.9, la segmentació amb el mètode k-means pareix que té més regions que l'obtinguda amb l'algorisme RSST. Aquest fet és degut precisament a que l'algorisme RSST utilitza la component espacial dels píxels, obtenint un resultat

3. MÈTODES DE SEGMENTACIÓ

amb regions connexes (en aquest exemple, en la imatge 3.9 b) en tenim 6). En canvi, amb el k-means, la segmentació es realitza a partir del vector de color de cada un dels píxels de la imatge, independentment de la posició espacial d'aquests. Per això, tot i parèixer que tenim més regions, n'hi ha només 3.

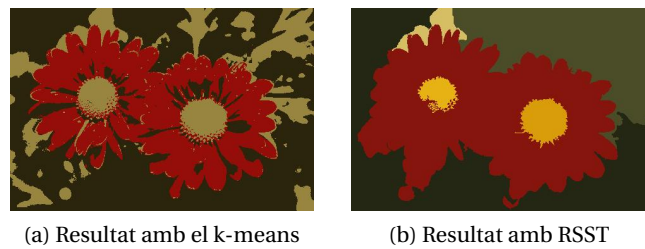


Figura 3.9: Comparació de les regions de diferents segmentacions.

Si apliquem aquest mètode de segmentació a la imatge de la flor ja utilitzada anteriorment, podem veure a la figura 3.10 com es van unint les regions (inicialment formades per píxels) per formar-ne de més grans i obtenir una imatge més simple. Aquesta imatge té una dimensió de 321×481 és a dir, a l'inici del procés tenim 154401 regions diferents, fet que suposa un gran cost computacional ja que en cada iteració ajuntem només 2 píxels.

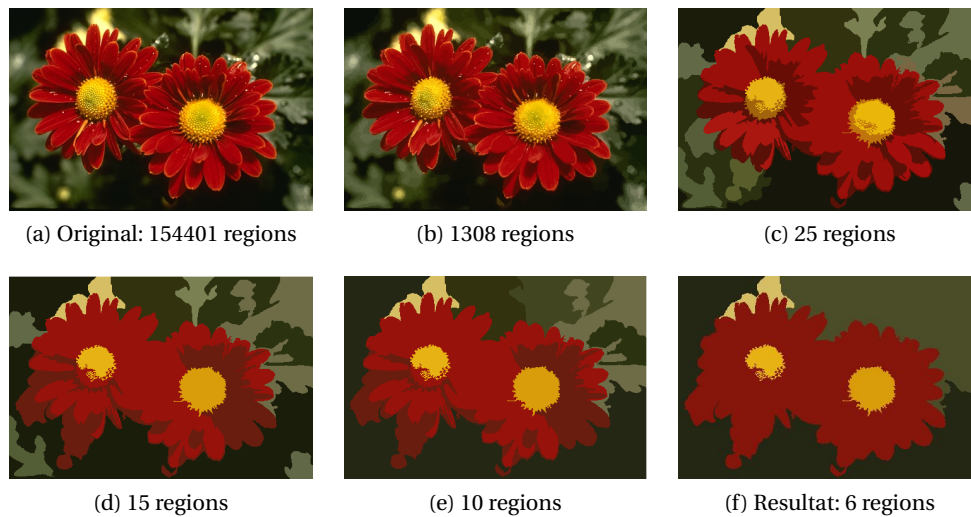


Figura 3.10: Resultats obtinguts segons el nombre de regions.

ENTORN EXPERIMENTAL

En aquest capítol es farà una comparació dels resultats obtinguts a l'hora de segmentar diferents imatges utilitzant cada un dels mètodes amb l'objectiu de determinar quin dels mètodes considerats obté millors resultats. Donat que la comparació visual és insuficient, es realitzarà també una comparació quantitativa utilitzant dos mètodes d'avaluació quantitativa de segmentacions, l'índex de Rand i el mètode de variació de la informació. El que fan aquests mètodes és comparar dues segmentacions d'una mateixa imatge, on una d'elles serà la "ideal" i l'altra la sortida de cada un dels algorismes estudiats. Entendrem per imatge "ideal" la segmentació feta pels experts.

En el conjunt d'imatges que utilitzem [12], cada imatge està segmentada per 4 o 5 humans diferents, per això també haurem de definir una imatge de consens d'entre aquestes per tal de poder després comparar-la amb les segmentacions obtingudes amb els mètodes explicats.

4.1 Mesures objectives

Per fer una comparació objectiva entre la segmentació ideal, també anomenada *ground truth*, i les segmentacions realitzades pels algorismes, utilitzarem les mesures de l'índex de Rand i de variació de la informació.

4.1.1 Índex de Rand

L'índex de Rand [9] es basa en mesurar la similitud entre els diferents clústers en que segmentem una imatge. Suposem que tenim dues segmentacions d'una mateixa imatge Q que denotam per $C = \{C_1, \dots, C_k\}$ i $C' = \{C'_1, \dots, C'_{k'}\}$ amb k i k' el nombre total de regions de cada segmentació i n_i i $n_{i'}$ el nombre de píxels de cada clúster C_i i $C'_{i'}$ amb $1 \leq i \leq k$ i $1 \leq i' \leq k'$. A l'algorisme 6, podem trobar com es calcula aquest índex.

Per entendre les passes explicades a l'algorisme 6, anem a introduir el concepte de taula de contingència.

Algorisme 6: Índex de Rand**Entrada:** Dues segmentacions d'una mateixa imatge (C i C').**Sortida:** Índex de Rand

- 1 Taula de contingència de C i C' ;
- 2 N = nombre total de píxels;
- 3 $n_{ii'}$ = píxels que es troben simultàneament en el clúster i de la segmentació C i en el clúster i' de la segmentació C' ;
- 4 $\text{Rand Índex} = \frac{\binom{N}{2} - \left[\frac{1}{2} \left(\sum_{i=1}^k \left(\sum_{i'=1}^{k'} n_{ii'} \right)^2 + \sum_{i'=1}^{k'} \left(\sum_{i=1}^k n_{ii'} \right)^2 \right) - \sum_{i=1}^k \sum_{i'=1}^{k'} n_{ii'}^2 \right]}{\binom{N}{2}}$;

Definició 4.1.1. Una taula de contingència de dues segmentacions C i C' és una matriu de dimensió $k \times k'$ tal que el valor de la posició (i, i') amb $1 \leq i \leq k$ i $1 \leq i' \leq k'$ es correspon amb la quantitat de píxels que hi ha a la intersecció entre els clústers C_i i $C'_{i'}$, de cada una de les dues particions corresponents, és a dir, $n_{ii'} = |C_i \cap C'_{i'}|$, on $|A|$ denota el cardinal del conjunt A .

Notem que el que fa aquest índex és quantificar el nombre de píxels que hem identificat correctament. Així, l'índex de Rand pren valors entre 0 i 1, valent 0 quan dues segmentacions són totalment diferents (una té un únic clúster que és tota la imatge i en l'altra cada píxel és un clúster diferent) i val 1 quan les dues segmentacions són la mateixa. Vegem-ho.

- Suposem que tenim dues segmentacions totalment diferents, és a dir $C = \{C_1\}$ segmentada amb una única regió i $C' = \{C'_1, \dots, C'_N\}$ on cada píxel és una regió de la imatge. Notem que en aquest cas, $n_{1i} = 1$ per a qualssevol $1 \leq i \leq N$. Substituint a l'equació de l'índex de Rand, i tenint en compte que $\binom{N}{2} = \frac{N(N-1)}{2}$

$$\frac{\binom{N}{2} - \left[\frac{1}{2} (N^2 + N) - N \right]}{\binom{N}{2}} = 1 - \frac{\frac{N^2 + N - 2N}{2}}{\binom{N}{2}} = 1 - 1 = 0.$$

- Suposem que tenim dues segmentacions totalment iguals, és a dir, $C = C' = \{C_1, \dots, C_k\}$ i suposem per simplicitat que $n_i = |C_i \cap C'_i|$. En aquest cas, notem que $n_{ii'} = n_i$ si $i = i'$ i 0 altrament. Substituint a l'equació de l'índex de Rand,

$$\frac{\binom{N}{2} - \left[\frac{1}{2} \left(\sum_{i=1}^k n_i^2 + \sum_{i=1}^k n_i^2 \right) - \sum_{i=1}^k n_i^2 \right]}{\binom{N}{2}} = \frac{\binom{N}{2} - 0}{\binom{N}{2}} = 1.$$

Així, entendrem que com més proper a 1 sigui l'índex resultant d'aplicar aquest mètode, més semblants seran les dues segmentacions.

4.1.2 Variació de la informació

El mètode de variació de la informació [44] el que fa és mesurar la informació que es guanya i la que es perd quan canviem un píxel d'un clúster a un altre. Per calcular aquest índex necessitarem primer alguns conceptes previs.

Considerem que tenim una imatge I , sobre la qual tenim dues segmentacions $C = \{C_1, \dots, C_k\}$ i $C' = \{C'_1, \dots, C'_{k'}\}$ amb n_i i $n_{i'}$ el nombre de píxels de cada clúster C_i i $C'_{i'}$, amb $1 \leq i \leq k$ i $1 \leq i' \leq k'$ respectivament. En l'algorisme 7 podem trobar el pseudocodi que es duu a terme per calcular aquest índex. A continuació s'introdueixen alguns conceptes emprats en aquest algorisme.

Algorisme 7: Variació de la informació

Entrada: Dues segmentacions d'una mateixa imatge (C i C')

Sortida : Índex del mètode de variació de la informació

- 1 $E1 = \text{Entropia de } C$;
 - 2 $E2 = \text{Entropia de } C'$;
 - 3 $I = \text{Informació mútua entre } C \text{ i } C'$;
 - 4 $\text{índex} = E1 + E2 - 2I$;
-

Definició 4.1.2. Sigui C una segmentació $C = \{C_1, \dots, C_k\}$ amb variable aleatòria associada $X: I \rightarrow \{1, \dots, k\}$ amb probabilitat que un píxel pertanyi a un clúster C_i donada per $P(i) = \frac{n_i}{N}$ amb $1 \leq i \leq k$. Definim l'entropia de la variable aleatòria X com

$$H(C) = - \sum_{i=1}^k P(i) \log P(i).$$

Notem que l'entropia sempre serà positiva, ja que hem suposat les regions no buides i $P(i) \leq 1$ amb $1 \leq i \leq k$ per ser una probabilitat i, per tant, $\log P(i) \leq 0$. A més, $H(C) = 0$ quan $\log P(i) = 0$ per a tot i , donat que $P(i) > 0$ per a tot i i cada un dels sumands de $\sum_{i=1}^k P(i) \log P(i) \leq 0$ aleshores necessàriament $P(i) = 1$ que és el cas en què hi hagi un únic clúster.

La informació mútua entre dues segmentacions és la informació que una segmentació té de l'altra. Siguin $P(i)$, $1 \leq i \leq k$ i $P(i')$, $1 \leq i' \leq k'$ les probabilitats associades a les segmentacions C , C' . Sigui $P(i, i')$ la probabilitat que un punt pertanyi a C_i i a $C'_{i'}$, aleshores

$$P(i, i') = \frac{|C_i \cap C'_{i'}|}{N}$$

on N és el nombre de píxels total de la imatge.

Definició 4.1.3. Definim $I(C, C')$ la informació mútua entre dues segmentacions C , C' com la informació mútua entre les variables aleatòries associades

$$I(C, C') = \sum_{i=1}^k \sum_{i'=1}^{k'} P(i, i') \log \frac{P(i, i')}{P(i)P(i')}.$$

El mètode de variació de la informació utilitza la següent fórmula per comparar dues segmentacions C i C' :

$$VI(C, C') = H(C) + H(C') - 2I(C, C'), \quad (4.1)$$

que és equivalent a

$$VI(C, C') = [H(C) - I(C, C')] + [H(C') - I(C, C')]. \quad (4.2)$$

L'índex del mètode de variació de la informació satisfà $0 \leq VI(C, C') \leq \log N$, essent 0 en el millor dels casos i $\log N$ en el pitjor. Vegem-ho.

- $0 \leq VI(C, C')$. En el millor dels casos, tendrem dues segmentacions iguals, és a dir, $C = C' = \{C_1, \dots, C_k\}$. En aquest cas, tendrem que

$$P(i, i') = \frac{|C_i \cap C'_{i'}|}{N} = \frac{n_i}{N} \text{ on } n_i \text{ és el nombre de clústers de la regió } i\text{-èsima,}$$

$$P(i) = P(i') = \frac{n_i}{N}; \quad H(C) = H(C') = - \sum_{i=1}^k \frac{n_i}{N} \log \frac{n_i}{N},$$

$$I(C, C') = \sum_{i=1}^k \frac{n_i}{N} \log \frac{n_i/N}{n_i^2/N^2} = - \sum_{i=1}^k \frac{n_i}{N} \log \frac{n_i}{N},$$

Finalment, substituïnt a l'equació de VI

$$VI(C, C') = -2 \sum_{i=1}^k \frac{n_i}{N} \log \frac{n_i}{N} + 2 \sum_{i=1}^k \frac{n_i}{N} \log \frac{n_i}{N} = 0.$$

- $VI(C, C') \leq \log N$. En el pitjor dels casos, les dues segmentacions seran totalment diferents, és a dir, $C = \{C_1, \dots, C_N\}$ és la segmentació on cada píxel és un clúster diferent i $C' = \{C'_1\}$ la segmentació formada per un únic clúster. En aquest cas,

$$P(i, i') = \frac{|C_i \cap C'_{i'}|}{N} = \frac{1}{N} \quad 1 \leq i \leq N, \quad i' = 1,$$

$$P(i) = \frac{1}{N}; \quad P(i') = 1,$$

$$H(C) = - \sum_{i=1}^N \frac{1}{N} \log \frac{1}{N} = - \log \frac{1}{N} = \log N; \quad H(C') = - \sum_{i=1}^1 1 \cdot \log(1) = 0,$$

$$I(C, C') = \sum_{i=1}^N \sum_{i'=1}^1 \frac{1}{N} \log \frac{1/N}{1/N} = 0.$$

Substituïnt a l'equació de VI ,

$$VI(C, C') = \log N.$$

Així, amb aquest mètode, i a diferència de l'índex de Rand, entendrem que com més petit sigui l'índex resultant, més semblants seran les dues segmentacions.

4.2 Base d'imatges i ground truth

Per mostrar l'efectivitat dels algorismes considerats, hem utilitzat com a base les imatges de la Berkeley Segmentation Dataset [12]¹. En aquest conjunt d'imatges tenim dos grans grups disponibles, un format per 300 imatges (BSDS300) i l'altre per 500 imatge

¹Aquesta base es pot descarregar a <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

(BSDS500), cada un d'ells està dividit en un conjunt d'imatges d'entrenament, un de test i un de validació, totes elles en l'espai de color RGB i una resolució suficient per tal de no tenir distorsions. A més, per a cada una de les imatges podem trobar segmentacions diferents que han fet 5 humans i una imatge binària amb les fronteres en què s'han dividit els objectes d'interès de cada una de les imatges (veure figura 4.1).

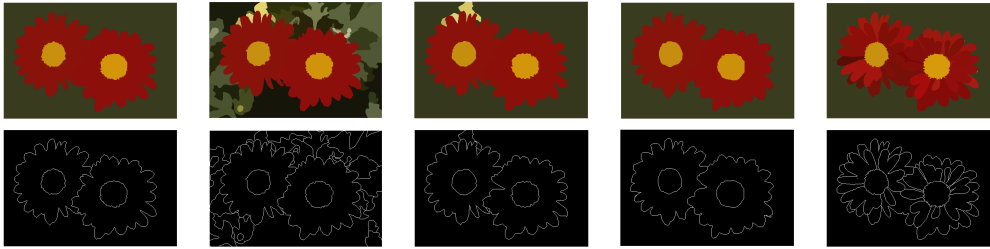


Figura 4.1: Resultats de les segmentacions dels humans.

Així, si el que volem és comparar la segmentació obtinguda a partir dels algorismes estudiats respecte a la imatge “ideal”, el primer que haurem de fer serà elegir, per a cada imatge, amb quina de les 5 segmentacions fetes pels humans volem comparar la nostra segmentació. Per això el primer que haurem de fer serà definir una imatge de consens a partir d'aquestes segmentacions realitzades pels experts.

4.2.1 Imatge de consens

Per tal d'obtenir la imatge de consens [45], utilitzarem el pseudocodi descrit a l'algorisme 8.

Algorisme 8: Imatge de consens

Entrada: O_i = contorns de les 4 o 5 imatges de la ground truth.

Sortida : Segmentació de consens corresponent a una de les O_i d'entrada.

- 1 Generam n imatges de consens C_j amb $0 \leq j \leq n$ on $n = 4$ o 5 depenent del nombre d'imatges d'entrada,

$$C_j = \{\text{imatge amb píxels consensuats en almenys } j \text{ imatges}\}$$

- 2 $V_{j,i} = D(C_j, O_i)$ on D és una mesura de discrepància;
 - 3 Obtenim un valor de discrepància G_j . La imatge de consens és la segmentació corresponent a O_i que té valor de discrepància G_j mínim;
-

Els píxels consensuats en almenys j imatges són els que formen part dels contorns en j o més imatges. Notem així que cada una de les imatges C_j és binària, amb 1 en la posició dels píxels que formen part dels contorns d'almenys j de les imatges de contorn d'entrada i 0 altrament.

Com a mesura de discrepància utilitzarem la mesura de Baddeley [46], ja que és una mesura d'error utilitzada en imatges binàries i que es basa en la distància de Hausdorff.

Aquesta mesura és

$$\Delta_w^p(A, B) = \left[\frac{1}{N} \sum_{x \in X} |w_c(d(x, A)) - w_c(d(x, B))|^p \right]^{\frac{1}{p}},$$

on $d(x, A)$ és la distància més curta de $x \in X$ a $A \subseteq X$, c i p són variables d'inicialització, que per defecte prendrem com a valors inicials $c = 5$ i $p = 2$, w és la funció $w(t) = \min(t, c)$ que és contínua, definida en l'interval $[0, \infty]$, creixent i còncava i $1 \leq p < \infty$. Aquesta mesura dóna valors no negatius i com menor és el valor, més semblants són A i B .

El valor de discrepància G_j ve donat pel procés d'unió del mínim. El que fa aquest procés és elegir com nivell òptim de consens el j , amb $0 \leq j \leq n$ que minimitza l'equació

$$G_j = \frac{1}{N} \sum_{i=1}^N V_{j,i}.$$

Finalment, per obtenir la imatge de consens, ho farem a partir de la imatge que hagi tengut un valor de discrepància G_j mínim. Aquesta imatge està associada a una mesura de discrepància $V_{j,i}$ que, a la vegada, està associada a una O_i , una imatge de la ground truth donada per un expert concret.

4.3 Resultats

Per avaluar els algorismes, hem utilitzat el conjunt d'entrenament del conjunt d'imatges BSDS500, on trobem un total de 200 imatges. El fet de tenir un nombre suficientment gran d'imatges, ens proporcionarà una certa fiabilitat en els resultats obtinguts.

Per solventar el problema de triar el nombre de clústers en què volem segmentar una imatge, en els mètodes del model lingüístic de 2-tuples, k-means i fuzzy k-means, s'han calculat segmentacions variant els clústers entre 1 i 6 i aplicant posteriorment el mètode del colze. A diferència dels mètodes anteriors, en el cas del mètode RSST, s'ha aplicat el mètode del colze a les segmentacions entre 1 i 25 clústers. Triar els nombres de clústers 6 i 25 és una elecció motivada pels resultats experimentals i pel nombre de colors considerats en el mètode de les 2-tuples per una banda i pel paràmetre original del RSST, per altra banda. Un altre dels inconvenients és l'aleatorietat en les condicions inicials, fet que hem solucionat, com ja s'ha comentat en el capítol 3 en els algorismes corresponents, fent cada una de les segmentacions un nombre suficientment gran de vegades.

Les segmentacions que hem obtingut per algunes de les imatges amb cada un dels diferents mètodes es poden veure a la figura 4.2.

Observant les imatges, podem veure que el mètode de dues tuples dóna una segmentació bastant deficient en la imatge 176035. Aquest fet és degut a que, tal i com hem explicat al mètode, suposem que objectes diferents tenen colors diferents, i en aquest cas, tenim que colors diferents estan classificats en la mateixa regió. Per altra part, és bastant bo a l'hora de segmentar imatges d'objectes que sobresurten sobre un fons d'un altre color, com podem veure a la imatge 12003. A més, en aquesta mateixa imatge, així com també en la 135069, es pot apreciar com la segmentació amb la modificació del mètode de dues tuples ens proporciona un resultat de més qualitat i amb regions

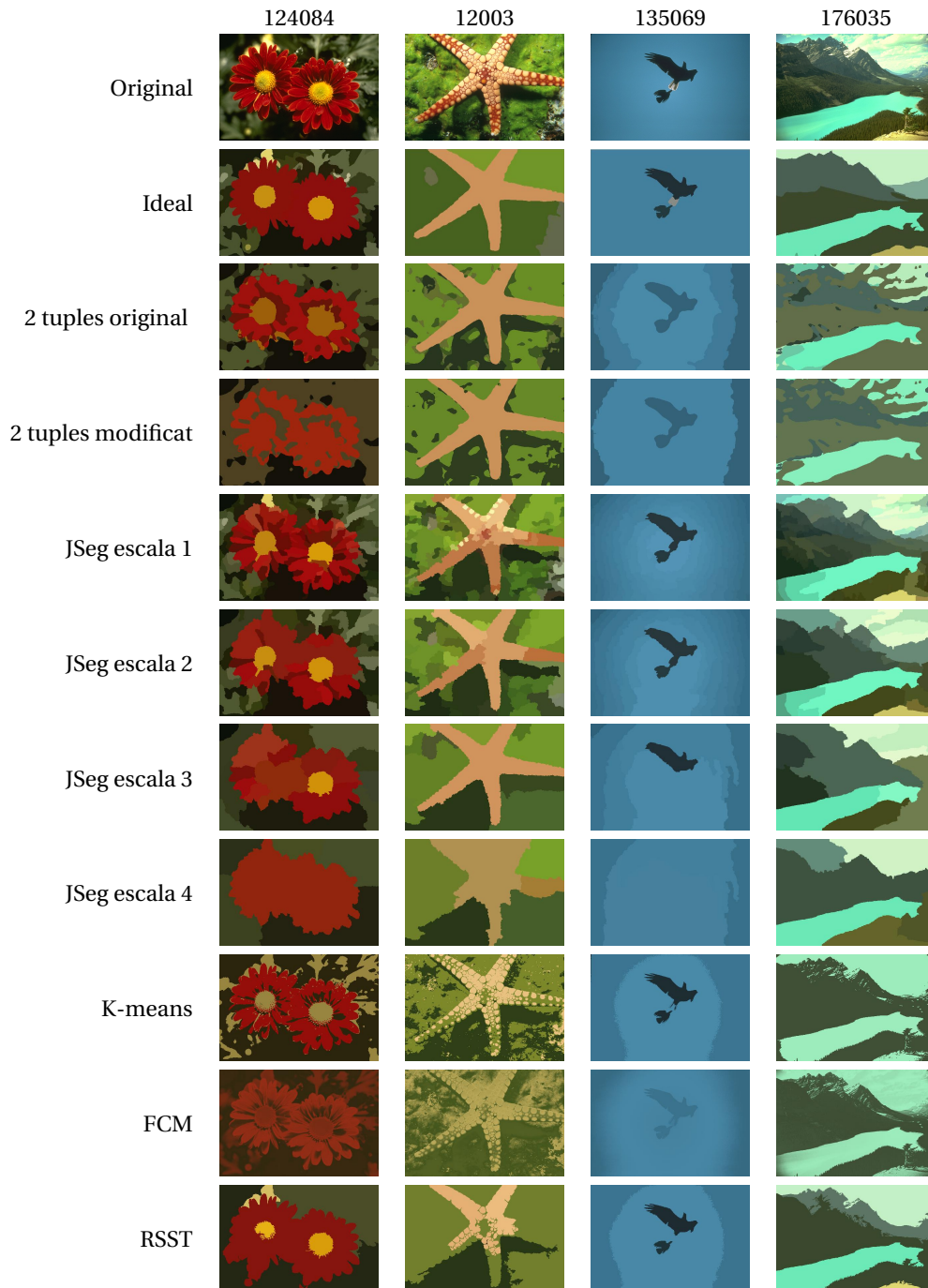


Figura 4.2: Resultats de les segmentacions.

més simples, ja que no en tots els casos són necessàries 6 regions per segmentar una imatge. Per contra, en la imatge 124084 el mètode de 2-tuples original ens proporciona uns millors resultats que el modificat.

Per altra part, el mètode de segmentació jeràrquica té alguns problemes de lluminositat, com podem veure a les segmentacions amb escales 3 i 4 de la imatge 135069. Tot i

4. ENTORN EXPERIMENTAL

Imatge	RI				Voi				
	mitjana	desviació	# guanyadors	# perdedors	mitjana	desviació	# guanyadors	# perdedors	
2 tuples original	0.6734	0.1198	14	0	1.9434	0.5449	6	4	
2 tuples modificat	0.5690	0.1097	14	61	1.7533	0.5913	25	3	
JSEG	Escala 1	0.7117	0.1998	24	28	2.8942	0.7484	1	123
	Escala 2	0.7223	0.1916	33	4	2.1674	0.5417	11	2
	Escala 3	0.7420	0.1714	57	1	1.6461	0.3822	36	0
	Escala 4	0.7268	0.1448	37	2	1.4073	0.3835	88	0
K-means	0.5878	0.1099	16	31	1.7346	0.6279	22	0	
FCM	0.5748	0.1110	5	33	1.7275	0.6243	11	0	
RSST	0.5685	0.0844	0	40	2.5219	0.5642	0	68	

Taula 4.1: Resultats numèrics de les segmentacions i nombre d'imatges que són les millors i les pitjors comparant-les amb tots els altres mètodes.

que visualment no hi ha una frontera clara en la imatge original, aquesta es fa present en la segmentació resultant.

Respecte als mètodes de k-means i fuzzy k-means, les regions resultants de les segmentacions no són connexes, fet que és degut a que aquests mètodes utilitzen únicament la component de color per realitzar la segmentació.

El mètode de RSST defineix els contorns de la imatge amb alta precisió, com podem veure a la segmentació de la imatge 176035. Tot i això, podem veure que té també alguns problemes de lluminositat, els quals es poden observar a les segmentacions de les imatges 12003 i 135069. Un altre dels problemes que té aquest mètode és l'alt cost computacional comparat amb els altres algorismes ja que, recordem, en cada iteració el que fa és ajuntar dos píxels.

Emerò, aquestes observacions són de caràcter purament subjectiu, vegem quins han estat els resultats numèrics d'acord a les mesures objectives introduïdes en la secció 4.1.

Com podem veure en la taula 4.1, no és possible determinar amb exactitud quin és el millor mètode, donat que tenim resultats diferents amb cada una de les dues mesures. Segons l'índex de Rand, el millor mètode és l'escala 3 del mètode de segmentació jeràrquica, mentre que el pitjor és el fuzzy k-means. En canvi, amb el mètode de Variació de la Informació el millor mètode és l'escala 4 del mètode de segmentació jeràrquica i el pitjor l'escala 1 d'aquest mateix mètode. També cal destacar que, tot i no ser el pitjor numèricament, el mètode RSST no és el millor per cap imatge en cap de les dues mesures.

També podem observar que, segons el mètode de variació de la informació, el mètode de dues tuples modificat és millor que l'original. Però amb el mètode de l'índex de Rand, passa just el contrari. Com podem veure a la figura 4.3, les segmentacions per les dues primeres imatges (56028 i 35070) són millors amb el mètode de dues tuples modificat, mentre que per les dues darreres (323016 i 97017) els resultats són molt millors amb el mètode de 2-tuples. Recordem que el de 2-tuples modificat el que fa és, a partir de la segmentació obtinguda amb el de 2-tuples, anar unint alguns dels seus clústers i aplicar després el mètode del colze. Per tant, en alguns casos on siguin necessaris menys de 6 clústers podem obtenir millors resultats però, no és d'estranyar que hi hagi segmentacions que no són tan òptimes.

Referent al mètode de segmentació jeràrquica, segons els resultats quantitius, les millors escales són la 3 i 4 mentre que les pitjors són la 1 i 2. En la figura 4.4 podem veure que com més petita és l'escala, més detalls tenim de la imatge i com més gran,

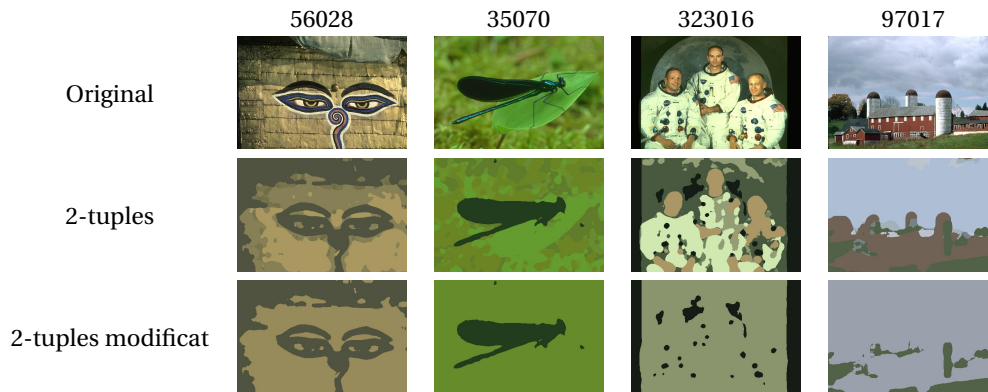


Figura 4.3: Comparació del model dels resultats obtinguts per l'algorisme de 2-tuples lingüístiques original i el de 2-tuples modificat.

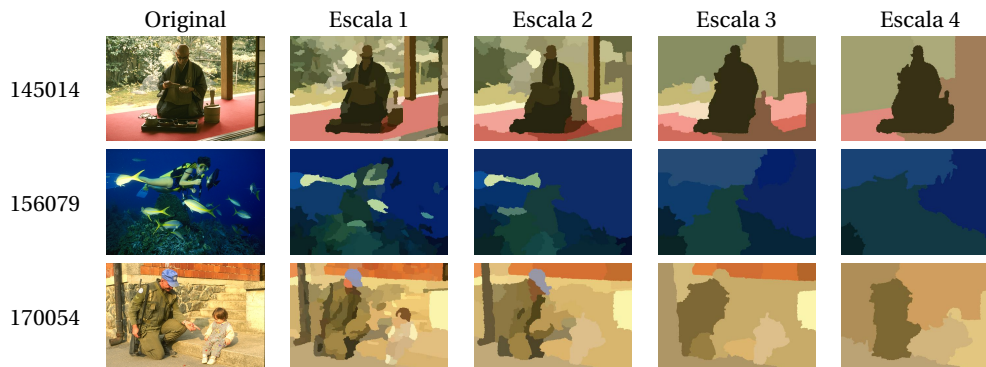


Figura 4.4: Comparació de les segmentacions per diferents escales del mètode de segmentació jeràrquic.

més simplicitat. En alguns casos, el fet de tenir massa detalls ens dificulta l'anàlisi de la imatge, ja que una segmentació hauria d'obtenir una representació simple de la imatge sense massa detalls. Tot i això, en algunes de les escales la imatge més simple (d'escala 4) ens fa perdre objectes importants de la imatge (com es pot veure en la segmentació de la imatge 156079). Però en altres com més gran és l'escala millor és la segmentació obtinguda (com es pot veure en la imatge 145014).

Vegem en la figura 4.5 quins han estat els millors i els pitjors resultats obtinguts pels mètodes considerats per algunes de les imatges segmentades.

4.4 Estudi estadístic: test de Wilcoxon

Encara que la taula 4.1. ens dóna una primera idea quantitativa del rendiment dels algorismes considerats, obtenir una mitjana superior en una mostra no implica en tot cas que un algorisme sigui millor que un altre des del punt de vista estadístic. Per això es realitzarà un contrast utilitzant un test estadístic per contrastar la superioritat estadística d'un mètode sobre un altre. En aquest cas, s'utilitzarà el test de Wilcoxon [11]. Aquest test duu a terme un contrast d'hipòtesis d'igualtat de medianes poblacionals per

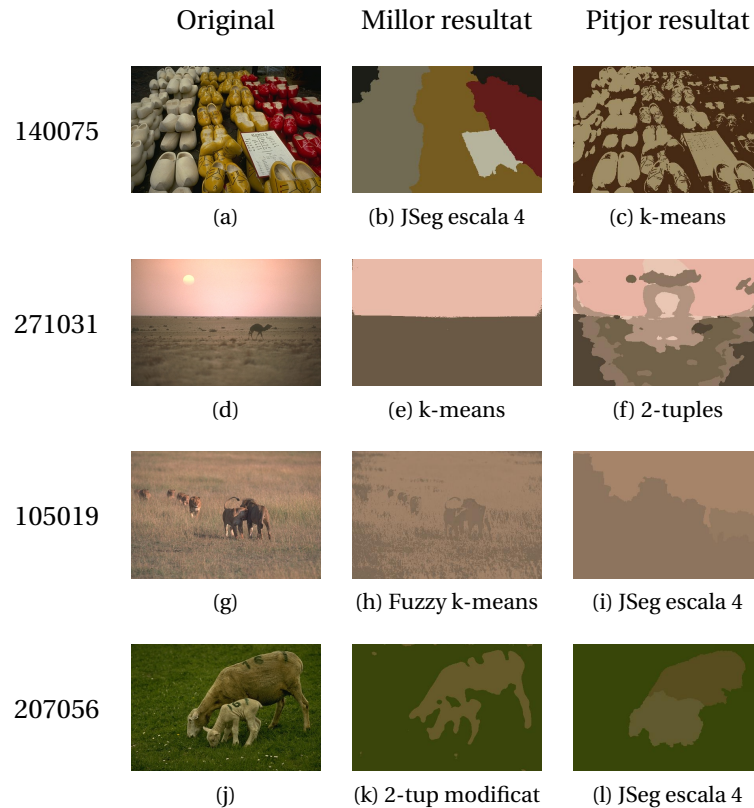


Figura 4.5: Millors i pitjors resultats d'algunes imatges.

determinar si existeixen diferències estadístiques significatives entre els dos mètodes. Aquest test és d'especial utilitat quan no podem suposar normalitat en les nostres dades (com és el nostre cas).

Per poder aplicar el test de Wilcoxon, s'han de satisfer les següents hipòtesis:

1. Les dades estan aparellades i provenen de la mateixa població.
2. Cada parell s'elegeix de forma aleatòria i independent.
3. Les dades es mesuren en una escala ordinal.

Siguin x_i, y_i els valors obtinguts pels dos algorismes, amb $1 \leq i \leq N$ on N és el nombre total d'imatges que té cada mostra. Siguin H_0 i H_1 la hipòtesi nul·la i alternativa. El contrast vendrà donat per:

$$\begin{cases} H_0 : & \text{la mediana de la diferència entre cada parell és zero,} \\ H_1 : & \text{la mediana de la diferència no és zero.} \end{cases}$$

Les passes que duu a terme aquest mètode són les següents:

1. Per a cada un dels i amb $1 \leq i \leq N$, calculem $z_i = x_i - y_i$.
2. Excloem els parells x_i, y_i que tenen $z_i = 0$ i definim N_r com la dimensió de la mostra reduïda.

3. Ordenam de menor a major els valors absoluts de z_i , amb $1 \leq i \leq N_r$ i assignem a cada un d'ells el seu rang R_i ².
4. Calculem el valor de l'estadístic W com

$$W = \left\| \sum_{i=1}^{N_r} \text{sign}(z_i) \cdot R_i \right\|,$$

on sign és la funció signe, donada per

$$\text{sign}(x) = \begin{cases} -1 & \text{si } x < 0, \\ 0 & \text{si } x = 0, \\ 1 & \text{si } x > 0. \end{cases}$$

5. Quan N_r creix, la distribució mostral de W convergeix a una distribució Normal $(0, 1)$. Per tant,

- Per $N_r \geq 10$, calculem

$$\sigma_W = \sqrt{\frac{N_r(N_r + 1)(2N_r + 1)}{6}}, \quad z = \frac{W - 0.5}{\sigma_W}$$

i si $z > z_{\text{crític}}$ rebutgem la hipòtesi H_0 .

- Per $N_r \leq 10$, comparem W amb una taula de referència [47] i si $W \geq W_{\text{crítica}, N_r}$, rebutgem la hipòtesi H_0 .

Les mesures obtingudes per cada mètode estan aparellades, donat que són els resultats de dos algorismes diferents però d'una mateixa imatge. Per dur a terme aquest estudi, utilitzarem la funció `wilcox.test` de R, fixant el nivell de significació en $\alpha = 0.05$ per tots els tests.

Per fer un rànquing dels algorismes, s'han utilitzat els criteris de la taula 4.2. Seguint aquests criteris, s'han obtingut els resultats que es troben a les taules 4.3, 4.4. Recordem que en el cas de l'índex de Rand, una millor mediana vol dir un valor més gran, mentre que en el cas del mètode de variació de la informació, una millor mediana fa referència a un valor més petit.

Com podem veure a partir dels resultats obtinguts, no tenim unanimitat respecte quin és el millor i pitjor mètode. Segons l'índex de Rand, l'algorisme estadísticament millor és el d'escala 3 del mètode de segmentació jeràrquica, mentre que el pitjor és el fuzzy k-means. Per contra, amb el mètode de variació de la informació, l'algorisme estadísticament millor és el d'escala 4 del mètode de segmentació jeràrquica i el pitjor el d'escala 1 d'aquest mateix mètode.

²Donada una llista d'elements z_1, \dots, z_n on els z_i estan ordenats de menor a major, definim el rang R_i de l'element z_i com $R_i = i$ si no existeix un z_j amb $j \neq i$ tal que $z_i = z_j$ i en el cas que existeixin s z 's iguals a z_i , llavors $R_j = \frac{i+(i+1)+\dots+(1+s)}{s+1}$ amb $i \leq j \leq s+i+1$.

4. ENTORN EXPERIMENTAL

✓✓	La mediana poblacional de l'algorisme de la fila és millor que la de l'algorisme de la columna.
✓	No es pot descartar la igualtat de medianes poblacionals, però la mediana mostral de l'algorisme de la fila és millor que la de l'algorisme de la columna.
✗	No es pot descartar la igualtat de medianes poblacionals, però la mediana mostral de l'algorisme de files és pitjor que la de l'algorisme de la columna.
✗✗	La mediana poblacional de l'algorisme de la fila és pitjor que la de l'algorisme de la columna.

Taula 4.2: Criteri de classificació dels algorismes.

	2-tuples	2-tup modificat	JSeg 1	JSeg 2	JSeg 3	JSeg 4	KM	FKM	RSST
2-tuples	-	✓✓	✗✗	✗✗	✗✗	✗✗	✓✓	✓✓	✓✓
2-tup modificat	✗✗	-	✗✗	✗✗	✗✗	✗✗	✗	✓	✗
JSeg 1	✓✓	✓✓	-	✗✗	✗✗	✗✗	✓✓	✓✓	✓✓
JSeg 2	✓✓	✓✓	✓✓	-	✗✗	✓	✓✓	✓✓	✓✓
JSeg 3	✓✓	✓✓	✓✓	✓✓	-	✓✓	✓✓	✓✓	✓✓
JSeg 4	✓✓	✓✓	✓✓	✗	✗✗	-	✓✓	✓✓	✓✓
KM	✗✗	✓	✗✗	✗✗	✗✗	✗✗	-	✓	✓
FKM	✗✗	✗	✗✗	✗✗	✗✗	✗✗	✗	-	✗
RSST	✗✗	✓	✗✗	✗✗	✗✗	✗✗	✗	✓	-

Taula 4.3: Test de Wilcoxon segons l'índex de Rand.

	2-tuples	2-tup modificat	JSeg 1	JSeg 2	JSeg 3	JSeg 4	KM	FKM	RSST
2-tuples	-	✗✗	✓✓	✓✓	✗✗	✗✗	✗✗	✗✗	✓✓
2-tup modificat	✓✓	-	✓✓	✓✓	✗	✗✗	✓	✗	✓✓
JSeg 1	✗✗	✗✗	-	✗✗	✗✗	✗✗	✗✗	✗✗	✗✗
JSeg 2	✗✗	✗✗	✓✓	-	✗✗	✗✗	✗✗	✗✗	✓✓
JSeg 3	✓✓	✓	✓✓	✓✓	-	✗✗	✓✓	✓✓	✓✓
JSeg 4	✓✓	✓✓	✓✓	✓✓	✓✓	-	✓✓	✓✓	✓✓
KM	✓✓	✗	✓✓	✓✓	✗✗	✗✗	-	✗	✓✓
FKM	✓✓	✓	✓✓	✓✓	✗✗	✗✗	✓	-	✓✓
RSST	✗✗	✗✗	✓✓	✗✗	✗✗	✗✗	✗✗	✗✗	-

Taula 4.4: Test de Wilcoxon segons el mètode de variació de la informació.

CONCLUSIONS

En aquest treball, s'ha duit a terme una introducció al tema de la segmentació d'imatges, introduint en primer lloc la definició de segmentació i una classificació de les tècniques més emprades. Una vegada establerts els fonaments del treball, s'han explicat i implementat cinc algorismes de segmentació i s'ha realitzat una comparació visual i quantitativa dels resultats obtinguts.

Concretament, en el capítol 2 es poden trobar alguns conceptes bàsics del tractament d'imatges digitals, així com també una definició formal de segmentació. A més, hi ha una recopilació de tècniques de segmentació d'imatges, amb una breu explicació de les seves característiques i alguns exemples de mètodes que utilitzen cadascuna d'aquestes tècniques. Aquesta classificació, encara que està basada en algunes existents, és novedosa i s'ha introduït per primera vegada en aquest treball.

En el capítol 3 s'han analitzat en detall i implementat cinc mètodes. Concretament, el mètode basat en el model lingüístic de dues tuples, el de segmentació jeràrquica, el k-means, el fuzzy k-means i el recursive shortest spanning tree. Als mètodes que utilitzen funcions ja implementades, se n'han fet algunes modificacions. Per una banda, en el mètode basat en les dues tuples, la modificació es realitza per permetre segmentacions de menys de 6 regions. Per altra banda, en els mètodes del k-means i el fuzzy k-means, les modificacions tenen l'objectiu de reduir l'aleatorietat i determinar de forma automàtica el nombre de clústers.

En el capítol 4 s'han aplicat els diferents algorismes a la base d'imatges de Berkeley per fer una comparació visual i una quantitativa dels diferents mètodes, utilitzant l'índex de Rand i el mètode de variació de la informació. Per fer la comparació quantitativa, ha estat necessari tenir una segmentació ideal amb la que comparar la segmentació obtinguda amb els mètodes anteriorment estudiats. Per això s'ha adoptat una imatge de consens d'entre les segmentacions ideals que es poden trobar a la base d'imatges considerada. Finalment, per determinar si els resultats obtinguts són estadísticament significatius o no, s'ha utilitzat el test estadístic de Wilcoxon.

Com s'ha pogut observar amb els resultats obtinguts, no tenim un mètode de segmentació que doni bons resultats per a totes les imatges, sinó que, per a cada imatge,

5. CONCLUSIONS

hi ha mètodes més adequats que d'altres. Aquesta mateixa idea es pot transmetre dels valors obtinguts pels índexs que emprats a la comparació quantitativa. S'ha vist que l'índex de Rand i el mètode de variació de la informació, tot i ser mètodes quantitius i no subjectius, donen resultats diferents. Per tant, no tenim un mètode de segmentació millor en tots els aspectes, igual que tampoc tenim un únic criteri per determinar quin és el millor mètode de segmentació. De totes formes, si ens haguéssim de decantar per alguns dels mètodes, el mètode de segmentació jeràrquica d'escala 3 i escala 4 serien els més indicats ja que són els que obtenen resultats superiors des del punt de vista estadístic a la resta segons l'índex de Rand i el de variació de la informació, respectivament.



ANNEXOS

En aquest apartat hi trobem els algorismes que s'han implementat dels diferents mètodes analitzats.

A.1 Model lingüístic de 2-tuples

```
function [Resultat, NClust] = Segment2_tuples(pathdesti, Foto, maxclusters, NImg)
    %'Calculant segmentació amb el mètode 2-tuples'

    CarpetaArrel = fileparts( mfilename('fullpath') );
    DirOriginal = cd([CarpetaArrel, filesep, '2Tuples']);

    %Elegim els experts segons els exemples de la referència 11 de l'article
    f_gu={'ex20' 'ex31' 'ex32'};
    f_ref={'ix1-ix2' 'rx1-ix2' 'ix1-rx2' 'rx1-rx2'}; %MSF
    f_m={'am'}; %Mitjana aritmètica
    vec=15;

    [x,y,z]=size(Foto);
    PertVerm1=zeros(x,y);
    PertGroc0=zeros(x,y);
    PertVerd1=zeros(x,y);
    PertVerd2=zeros(x,y);
    PertCia0=zeros(x,y);
    PertBlau1=zeros(x,y);
    PertBlau2=zeros(x,y);
    PertMag0=zeros(x,y);
    PertVerm2=zeros(x,y);
    Pertenencies=zeros(x,y,6); %Pertenències = probabilitat que un píxel pertanyi a una component
    PertenenciaExperts=zeros(x,y,6,6*2);
    PertenenciesExpEtiq=zeros(x,y,6,6*2);
    PertenenciesExpTrans=zeros(x,y,6,6*2);
    expert=0;

    %Elegim experts i assignem grau de pertinença de cada píxel a cada expert
```

```

for i_gu=1:3,
    for i_ref=1:4,
        for i_m=1,
            expert= i_ref + (i_gu-1)*4;
            [PertVerm1,PertGroc0,PertVerd1]=OverlapColor_vMSR(Foto, char(f_ref(i_ref)),char(
                f_gu(i_gu)),char(f_m(i_m)),'R','G');
            [PertVerd2,PertCia0,PertBlau1]=OverlapColor_vMSR(Foto, char(f_ref(i_ref)),char(
                f_gu(i_gu)),char(f_m(i_m)),'G','B');
            [PertVerm2,PertMag0,PertBlau2]=OverlapColor_vMSR(Foto, char(f_ref(i_ref)),char(
                f_gu(i_gu)),char(f_m(i_m)),'R','B');
            [Pertencencies]=Calcular_PTotal(PertVerm1,PertGroc0,PertVerd1,PertVerd2,PertCia0,
                PertBlau1,PertBlau2,PertMag0,PertVerm2, 6);
            PertenciaExperts(:,i_ref,i_m)=Pertencencies(:,i_ref,i_m);
        end
    end
end

%Construir vector de preferències conjunta
PertenciaColors=zeros(x,y,6*2,6);

%Calculem la matriu transposta
for i=1:6,
    for j=1:6*2,
        PertenciaColors(:,j,i)=PertenciaExperts(:,i,j);
    end
end

[PertencenciesColEtiq,PertencenciesColTrans]=DuesTuples(PertenciaColors);

%Transposició colors–experts realitzada amb els índexos

%Combinació convexa per vector de preferències col.lectiu
PertencenciesColorEtiq=zeros(x,y,6);
PertencenciesColorTrans=zeros(x,y,6);
for color=1:6,
    [PertencenciesColorEtiq(:,,color),PertencenciesColorTrans(:,,color)]=
        CombinacioConvexaPreferencia(6, PertencenciesColEtiq(:,,color),
            PertencenciesColTrans(:,,color));
end

% Incluir informació local a cada píxel
[PertencenciesColorEtiq,PertencenciesColorTrans]=CombinacioConvexaVeins(PertencenciesColorEtiq,
    PertencenciesColorTrans,vec);

%Func IndMax2Tuplas
PertencenciesColor=DuesTuplesInv(PertencenciesColorEtiq,PertencenciesColorTrans);

%Assignem cada valor al que té major grau de pertinença
ColorPixels=zeros(x,y);
for i = 1:x
    for j = 1:y
        ColorPixels(i,j)=find(PertencenciesColor(i,j,:)==max(reshape(PertencenciesColor(i,j,:),1,6,1),
            ,1));
    end
end

[Segmentades, etiquetes, Comb] = UnirRegions(ColorPixels, maxclusters);

```

```

% Calculem SSC
SSC = zeros(length(Comb),1);
for i=1 : length(Comb)
    Aux = resaltar_colormitja(length(unique(etiquetes(i,:))), pathdesti, Foto, NImg ,
        Segmentades(:,i));
    a = double(Foto) - double(Aux);
    for j = 1 : length(a)
        SSC(i) = SSC(i) + norm(a(j)).^2;
    end
end

%Comparem totes les que tenen el mateix nombre de clústers
a = zeros(length(etiquetes),1);
for j = 1 : length(etiquetes)
    a(j) = length(unique(etiquetes(j,:)));
end

optim = zeros(maxclusters,1); %Imatge de n clústers amb millor segmentació
min = Inf(1,maxclusters);
for j=1: length(etiquetes)
    if SSC(j)<min(a(j))
        min(a(j)) = SSC(j);
        optim(a(j)) = j;
    end
end

cd(DirOriginal);
NClust = elbow(min,maxclusters);

CarpetaArrel = fileparts( mfilename('fullpath') );
DirOriginal = cd([CarpetaArrel, filesep, '2Tuples']);

Resultat = Segmentades(:,optim(NClust));
Aux = resaltar_colormitja(NClust, pathdesti, Foto, NImg , Resultat);

cd(DirOriginal);
end

```

```

function [PertC1,PertC12,PertC2]=OverlapColor_vMSR(Foto, F_ref,F_Gu,F_M,C1,C2)

%Diferència entre C1 i C2
OrigDif=PreprocesImatge(Foto,C1,C2);

%Convertim imatge a dimensió 1*(nm)
[m,n]=size(OrigDif);
Serie_O=reshape(OrigDif,1,m*n);

%Cream histograma
aux=[0:255];
HistProcessat=hist(Serie_O,aux);

M_Objecte=Crear_mitja_Objecte(HistProcessat);
M_Fons=Crear_mitja_Fons(HistProcessat);

P_Objecte=Calcular_pertenencia(F_ref,M_Objecte);
P_Fons=Calcular_pertenencia(F_ref,M_Fons);

```

A. ANNEXOS

```
%Files són threshold, columnes són intensitats

Overlap=Crear_Overlap(F_Gu,P_Fons,P_Objecte);

DesconeixementTotal=CalcularDT(F_M,Overlap);

ThresholdVector=find(DesconeixementTotal==min(DesconeixementTotal));

ThresholdVector=ThresholdVector-1;

[x,y]=size(ThresholdVector);
if y>1
    Threshold=ThresholdVector(1,floor(y/2));
else
    Threshold=ThresholdVector(1,1);
end

%Reconstruïm la imatge
[x,y]=size(OrigDif);

Segmentada=zeros(x,y,2);
Segaux1=zeros(x,y);
Segaux2=zeros(x,y);

PertC1=zeros(x,y);
PertC12=zeros(x,y);
PertC2=zeros(x,y);
[PertC1,PertC12,PertC2]=CalcularPP(OrigDif,Threshold);
end
```

```
function [Pertencia]=Calcular_pertenencia(F_ref,mitja);

Pertencia=zeros(256,256);
Pertencia=double(Pertencia);

aux=zeros(1,256);
for i=0:255
    aux=i/255;
    meddist=mitja./255;
    Pertencia(i+1,:)=REF(F_ref,aux,meddist);
end
end
```

```
function [pesos]=calcular_pesos_minmeitat(n)

pesos=[1:n]';
pesos=double(pesos);
pesos=pesos./n;

doblepesos=2*pesos;
%Importa l'ordre
pesos(pesos>=0.5)=1;
pesos(pesos<0.5)=doblepesos(pesos<0.5);

pesos_1=[0:(n-1)]';
pesos_1=double(pesos_1);
pesos_1=pesos_1./n;
```

```

doblepesos_1=2*pesos_1;
%L'ordre importa
pesos_1(pesos_1>=0.5)=1;
pesos_1(pesos_1<0.5)=doblepesos_1(pesos_1<0.5);

pesos=pesos - pesos_1;
end

```

```

function [Pertencencies]=Calcular_PTotal(PertVerm1,PertGroc0,PertVerd1,PertVerd2,PertCia0,
    PertBlau1,PertBlau2,PertMag0,PertVerm2, nclust)
[x,y]=size(PertVerm1);
Pertencencies=zeros(x,y,nclust);

Pertencencies(:,,1)=min(PertVerm1,PertVerm2);
Pertencencies(:,,2)=PertGroc0;
Pertencencies(:,,3)=min(PertVerd1,PertVerd2);
Pertencencies(:,,4)=PertCia0;
Pertencencies(:,,5)=min(PertBlau1,PertBlau2);
Pertencencies(:,,6)=PertMag0;
end

```

```

function [DTporT]=CalcularDT(F_M,Overlap);

Overlap=Overlap';
%Mitjana
switch F_M
    case 'am'
        DTporT=mean(Overlap);
    otherwise
        DTporT=NaN;
end
end

```

```

function [VeinsColorEtiq,VeinsColorTrans]=CombinacioConvexaVeins(MatriuColorsEtiq,
    MatriuColorsTrans,vec)

[p,q,r]=size(MatriuColorsEtiq);
VeinsColor=zeros(p,q,r);
VeinsColor=double(VeinsColor);
rad=(vec-1)/2;

MatriuColors=DuesTuplesInv(MatriuColorsEtiq,MatriuColorsTrans);

pesos=zeros(vec*vec,1);
pesos=double(pesos);
pesos=calcular_pesos_minmeitat(vec*vec);

MatriuColorsAux=zeros(p+(2*rad),q+(2*rad),r);
MatriuColorsAux=MatriuColorsAux+0.5;

MatriuColorsAux(1+rad:p+rad,1+rad:q+rad,1:r)=MatriuColors(:,,:);

for k = 1:r
    for i = 1:p
        for j = 1:q

```

```

[nm,nn]=size(MatriuColorsAux((i:i+2*rad),(j:j+2*rad),k));
FilaOriginal=reshape(MatriuColorsAux((i:i+2*rad),(j:j+2*rad),k),1,nn*nm);

VeinsColor(i,j,k)=(-sort(-FilaOriginal))*pesos;
    end
end
end

[VeinsColorEtiq,VeinsColorTrans]=DuesTuples(VeinsColor);
end

```

```

function [PreferenciaColorCCE2T,PreferenciaColorCCE]=CombinacioConvexaPreferencia(nclust,
    PreferenciaColorEtiq,PreferenciaColorTrans)

% p y q son els indexos dels pixels
%r= 3*4*1 --> Num d'experts
[p,q,r]=size(PreferenciaColorEtiq);
PreferenciaColor=zeros(p,q,r);

PreferenciaColor=DuesTuplesInv(PreferenciaColorEtiq,PreferenciaColorTrans);
PreferenciaColorCCE=zeros(p,q,1);

pesos=zeros(nclust*2,1);
pesos=double(pesos);
pesos=calcular_pesos_minmeitat(nclust*2);

for i = 1:p
    for j = 1:q
        auxil=reshape((-sort(-PreferenciaColor(i,j,:))),1,nclust*2,1);
        PreferenciaColorCCE(i,j,1)=auxil*pesos;
    end
end

[PreferenciaColorCCE2T,PreferenciaColorCCE]=DuesTuples(PreferenciaColorCCE);
end

```

```

function [mMatriu2TEtiq,mMatriu2TTrans]=DuesTuples(mMatriuColors)

[i,j,k,l]=size(mMatriuColors);

% Quatre etiquetes linguistiques: (1) no pertany, (2) una mica, (3) bastant, (4) pertany
mMatriu2TEtiq((mMatriuColors<0.17))=1;
mMatriu2TEtiq((mMatriuColors>=0.17 & mMatriuColors<0.5))=2;
mMatriu2TEtiq((mMatriuColors>=0.5 & mMatriuColors<0.85))=3;
mMatriu2TEtiq((mMatriuColors>=0.85))=4;

mMatriu2TEtiq=reshape(mMatriu2TEtiq,i,j,k,l);

mMatriu2TTrans((mMatriuColors<0.17))=mMatriuColors(mMatriuColors<0.17);
mMatriu2TTrans((mMatriuColors>=0.17 & mMatriuColors<0.5))=mMatriuColors(mMatriuColors
    >=0.17 & mMatriuColors<0.5)-0.33;
mMatriu2TTrans((mMatriuColors>=0.5 & mMatriuColors<0.85))=mMatriuColors(mMatriuColors
    >=0.5 & mMatriuColors<0.85)-0.66;
mMatriu2TTrans((mMatriuColors>=0.85))=mMatriuColors(mMatriuColors>=0.85)-1;

mMatriu2TTrans=reshape(mMatriu2TTrans,i,j,k,l);
end

```



```

function [M_Fons]=Crear_mitja_Fons(HistProcessat);

M_Fons=zeros(1,256);
M_Fons=double(M_Fons);

aux=[0:255];

num=aux.*HistProcessat;

for i=1:256
    if (sum(HistProcessat(1:i))==0)
        M_Fons(i)=0;
    else
        M_Fons(i)=sum(num(1:i))/sum(HistProcessat(1:i));
    end
end
end

```

```

function [M_Objecte]=Crear_mitja_Objecte(HistProcessat);

M_Objecte=zeros(1,256);
M_Objecte=double(M_Objecte);

aux=[0:255];

num=aux.*HistProcessat;

for i=1:255
    if (sum(HistProcessat(i+1:256))==0)
        M_Objecte(i)=0;
    else
        M_Objecte(i)=sum(num(i+1:256))/sum(HistProcessat(i+1:256));
    end
end
M_Objecte(256)=M_Objecte(255);
end

```

```

function [Overlap]=Crear_Overlap(F_Gu,P_Fons,P_Objecte);

Overlap=zeros(256,256);
Overlap=double(Overlap);
switch F_Gu
    %Exemple 2.0
    case 'ex20'
        for i=1:256
            for j=1:256
                aux=(1-P_Fons(i,j))*(1-P_Objecte(i,j));
                if aux <= 0.20
                    Overlap(i,j)= aux /0.2;
                end
                if (aux <= 0.8) & (aux > 0.20)
                    Overlap(i,j)= 1;
                end
                if (aux <= 1) & (aux > 0.8)
                    Overlap(i,j)= (4*aux)-3;
                end
            end
        end
    end
end

```

```

        end
    end
    %Exemple 3.1
    case 'ex31'
        for i=1:256
            for j=1:256
                aux=(1-P_Fons(i,j))*(1-P_Objecte(i,j));
                if aux <= 0.25
                    Overlap(i,j)= 4*aux;
                end
                if aux > 0.25
                    Overlap(i,j)= 1/(4*aux);
                end
            end
        end
    end
    %Exemple 3.2
    case 'ex32'
        for i=1:256
            for j=1:256
                aux=(1-P_Fons(i,j))*(1-P_Objecte(i,j));
                if aux <= 0.25
                    Overlap(i,j)= 2*sqrt(aux);
                end
                if aux > 0.25
                    Overlap(i,j)= 1/(2*sqrt(aux));
                end
            end
        end
    end
    otherwise
        Overlap=NaN;
    end
end
end

```

```

function [Segmentades, etiquetes, Comb] = UnirRegions( ColorPixels, maxclusters )
[m,n] = size(ColorPixels);

Comb = Combinacions(maxclusters); %Totes les possibles combinacions amb maxclusters
Segmentades = zeros(m,n,length(Comb));
etiquetes = zeros(length(Comb),maxclusters);

for i=1 : length(Comb)
    for j = 1: length(Comb{i,1})
        for k = 1 : length(Comb{i,1}{1,j})
            etiquetes(i,Comb{i,1}{1,j}(k)) = j;
        end
    end

    end
    for k = 1 : m
        for z = 1:n
            Segmentades(k,z,i) = etiquetes(i,ColorPixels(k,z));
        end
    end
end
end
end

```

```

function Aux = resaltar_colorsmitja(nclust, pathdesti, Foto, NImg, Segmentada);

```

```

Ori=zeros(size(Foto));
Ori=Foto;
Ori1=Ori(:,,1);
Ori2=Ori(:,,2);
Ori3=Ori(:,,3);
Aux=Foto;
Aux1=Aux(:,,1);
Aux2=Aux(:,,2);
Aux3=Aux(:,,3);

for i=1:nclust
    Aux1(Segmentada==i)=mean(mean(Ori1(Segmentada==i)));
    Aux2(Segmentada==i)=mean(mean(Ori2(Segmentada==i)));
    Aux3(Segmentada==i)=mean(mean(Ori3(Segmentada==i)));
end
Aux(:,,1)=Aux1;
Aux(:,,2)=Aux2;
Aux(:,,3)=Aux3;
Aux= uint8(Aux);

minsal=[pathdesti NImg]; %Guardar imatge
imwrite(Aux, minsal);
end

```

```

function [Fila_pert]=REF(F_ref,X,Y);
%Funcio MSF
switch F_ref
    case 'ix1-ix2'
        Fila_pert=1.-abs(X-Y);
    case 'rx1-ix2'
        Fila_pert=(1.-abs(X-Y)).^2;
    case 'ix1-rx2'
        Fila_pert=1.-abs(sqrt(X)-sqrt(Y));
    case 'rx1-rx2'
        Fila_pert=(1.-abs(sqrt(X)-sqrt(Y))).^2;
    otherwise
        Fila_pert=NaN;
end
end

```

```

function [OrigDif]=PreprocesImatge(Foto,C1,C2)

switch C1
    case 'R'
        OrigC1=Foto(:,,1);
    case 'G'
        OrigC1=Foto(:,,2);
    case 'B'
        OrigC1=Foto(:,,3);
    otherwise
        disp('Error_en_la_component_1')
end

switch C2
    case 'R'
        OrigC2=Foto(:,,1);
    case 'G'

```

```

    OrigC2=Foto(:,2);
case 'B'
    OrigC2=Foto(:,3);
otherwise
    disp('Error_en_la_component_2')
end

OrigC1=double(OrigC1);
OrigC2=double(OrigC2);

Aux=(OrigC2-OrigC1+255)/2;
OrigDif=floor((OrigC2-OrigC1+255)/2);
end

```

```

function [mMatriuColors]=DuesTuplesInv(mMatriu2TEtiq,mMatriu2TTrans)

[i,j,k,l]=size(mMatriu2TEtiq);

%Càlcul dels inversos dels pas a dues tuples
mMatriuColors((mMatriu2TEtiq==1))=mMatriu2TTrans(mMatriu2TEtiq==1);
mMatriuColors((mMatriu2TEtiq==2))=mMatriu2TTrans(mMatriu2TEtiq==2)+0.33;
mMatriuColors((mMatriu2TEtiq==3))=mMatriu2TTrans(mMatriu2TEtiq==3)+0.66;
mMatriuColors((mMatriu2TEtiq==4))=mMatriu2TTrans(mMatriu2TEtiq==4)+1;

mMatriuColors=reshape(mMatriuColors,i,j,k,l);
end

```

```

function [PertC1,PertC12,PertC2]=CalcularPP(OrigDif,Threshold)

Thresmes1=Threshold+1;
[m,n]=size(OrigDif);
HistInd=0:255;
[histOrigDif]=hist(reshape(OrigDif,1,m*n),HistInd);
if Thresmes1 == 1
    Thresmes1=2;
    Threshold=1;
elseif Thresmes1 == 256
    Thresmes1=255;
    Threshold=254;
end
M1=find(histOrigDif(1:Thresmes1-1)==max(histOrigDif(1:Thresmes1-1)));
Max1=M1(1)-1;
M2=find(histOrigDif(Thresmes1+1:256)==max(histOrigDif(Thresmes1+1:256)));
Max2=Thresmes1+M2(end)-1;

PertC1=zeros(m,n);
PertC12=zeros(m,n);
PertC2=zeros(m,n);

PertC1(OrigDif<=Max1)=1;
PertC2(OrigDif>=Max2)=1;

%% Noves equacions de rectes per C12: Aux1 ascendent i Aux2 descendent
Aux1=OrigDif./(Threshold-Max1)-(Max1/(Threshold-Max1));
Aux2=-OrigDif./(Max2-Threshold)+(Max2/(Max2-Threshold));

PertC12(OrigDif>Max1 & OrigDif<Threshold)=Aux1(OrigDif>Max1 & OrigDif<Threshold);

```

```

PertC12(OrigDif>=Threshold & OrigDif<Max2)=Aux2(OrigDif>=Threshold & OrigDif<Max2);
Aux3=1-Aux1;
Aux4=1-Aux2;
PertC1(OrigDif>Max1 & OrigDif<Threshold)=Aux3(OrigDif>Max1 & OrigDif<Threshold);
PertC2(OrigDif>=Threshold & OrigDif<Max2)=Aux4(OrigDif>=Threshold & OrigDif<Max2);
end

```

```

function [NClust] = elbow( Min, maxclusters )
    %Construïm la recta que va de min(1) a min(maxclusters)
    pendent = (Min(maxclusters)-Min(1))/(maxclusters-1);

    dist = zeros(1,maxclusters-2); %Distància d'un punt (i,y(i)) a la recta
    for i=2 :(maxclusters-1)
        pendent = (Min(maxclusters)-Min(1))/(maxclusters-1);
        dist(i-1) = abs(pendent * i - Min(i) + Min(1)-pendent)/(sqrt(pendent^2 +1));
    end

    NClust = find(dist == max(dist)) +1;
end

```

A.2 K-means

```

function [SegmentadaKM,NClust] = KMeans(Foto, pathdesti, maxclusters, maxiter, NImg)
    'Calculant_segmentacio_KMEANS'

    [x,y,z] = size(Foto);
    Foto = double(Foto);

    C1=[Foto(:,,1)]; %Agafem només un color (C1=Red, C2=Green,C3=Blue)
    C2=[Foto(:,,2)];
    C3=[Foto(:,,3)];

    C1=reshape(C1,[],1); %Posam les matrius com una llista
    C2=reshape(C2,[],1);
    C3=reshape(C3,[],1);

    IM=[C1 C2 C3]; %Cream una matriu nm * 3

    min = Inf * ones(1,maxclusters);
    C=zeros(maxclusters,x,y,3);
    etiquetes = zeros(maxclusters,x,y);
    for t=1:maxclusters
        for u=1:maxiter
            [idx,centre,sumd] = kmeans(IM,t);
            if min(t) > sum(sumd)
                min(t) = sum(sumd);
                idx=reshape(idx,x,y);
                etiquetes(t,,:) = idx;
                for i=1:x
                    for j=1:y
                        c=idx(i,j);
                        C(t,i,j,:)=[centre(c,1),centre(c,2),centre(c,3)];
                    end
                end
            end
        end
    end
end

```

```

    end
end

NClust = elbow(min, maxclusters);

SegmentadaKM = reshape(etiquetes(NClust, :, :), x, y);
ImatgeKM = reshape(C(NClust, :, :), x, y, z); %reshape ja que sinó és 4-D
minsal=[pathdesti NImg]; %Guardar imatge
imwrite(uint8(ImatgeKM), minsal);
end

```

A.3 Fuzzy k-means

```

function [SegmentadaF, NClust] = FUZZY( Foto, pathdesti, maxclusters, maxiter, NImg )

'Calculant_Segmentacio_Fuzzy'
[x, y, z] = size(Foto);
Foto = double(Foto);

C1=[Foto(:, :, 1)]; %Agafem només un color (C1=Red, C2=Green, C3=Blue)
C2=[Foto(:, :, 2)];
C3=[Foto(:, :, 3)];

C1=reshape(C1, [], 1); %Posem les matrius com una llista
C2=reshape(C2, [], 1);
C3=reshape(C3, [], 1);

IM=[C1 C2 C3]; %Creem una matriu nm * 3

min = Inf * ones(1, maxclusters);
C=zeros(maxclusters, x, y, 3);
etiquetes = zeros(maxclusters, x, y);
Z = zeros(maxclusters, x * y);
for t=1:maxclusters
    for u=1:maxiter
        options = [NaN 25 0.001 0]; %Vector necessari perquè no mostri funcions objectiu
        [centre, U, objFun]=fcm(IM, t, options);
        if min(t) > objFun(end)
            min(t) = objFun(end);
            for i=1:x
                for j=1:y
                    C(t, i, j, :) = [sum(U(:, i+j*x-x)) * centre(:, 1)), sum(U(:, i+j*x-x) * centre(:, 2)), sum(
                        U(:, i+j*x-x) * centre(:, 3))];
                end
            end
            %Necessari per poder calcular després l'índex de Rand
            for i=1:x*y
                for j= 1 : t
                    if U(j, i) == max(U(:, i))
                        ma = j;
                    end
                end
                Z(t, i) = centre(ma, 3);
            end
            etiquetes(t, :, :) = reshape(Z(t, :, :), x, y);
        end
    end
end

```

```

        end
    end
end

NClust = elbow(min,maxclusters);

SegmentadaF = reshape(etiquetes(NClust,:),x,y);
ImatgeF = reshape(C(NClust,:), x,y,z);

minal=[pathdesti NImg];
imwrite(uint8(ImatgeF), minal);
end

```

A.4 Recursive shortest spanning tree

```

function [Resultat,NReg] = RSST( I,NImg, DirDesti )
[m,n,z] = size(I);
T = 25; %T_cum de s'article

CarpetaArrel = fileparts( mfilename('fullpath') );
DirOriginal = cd([CarpetaArrel, filesep, 'Image_Graphs']);

%Creem el graf amb connectivitat 8
G = imageGraph([n m],8); %El graf agafa els índexos n i m al revés
cd(DirOriginal);

%Inicialitzem variables
NumReg=G.Nodes(:,3);
L=length(unique(NumReg));
DimReg = ones(height(G.Nodes),1); %Dimensió de cada regió, inicialment cada píxel en forma una
diferent
Pesos = G.Edges(:,2); %Pesos de les arestes
EdgesNodes = G.Edges(:,1); %Nodes connectats per arestes
NodesCoord = [G.Nodes(:,1), G.Nodes(:,2)]; %Coodenades dels nodes
Cost = zeros(m*n,1); %Cost de tenir 'i' nodes
Segmentada = zeros(m,n,3,T); %Tendrem T segmentacions diferents
Etiquetes = zeros(m,n,T);

%Inicialitzam la variable Centres amb el color de cada píxel
Centres = zeros(height(G.Nodes),3);
for i=1 : height(G.Nodes)
    Centres(i,:) = I(NodesCoord(i,1),NodesCoord(i,2),:);
end

%Actualitzem els pesos de les arestes
vector = linspace(1,length(Pesos),length(Pesos));
Pesos = CalcularPes(EdgesNodes,vector, Centres, DimReg,m,n, Pesos);
clear vector;

while L > 1
    %Ajuntem l'aresta que té pes mínim i recalculam
    Min= find(Pesos == min(Pesos));
    nodes = EdgesNodes(Min(1),:);
    NouNode = min(nodes); %Etiqueta que tendrà la nova regió

```

```

Regio1 = find(NumReg == nodes(1)); %Nodes que tenim a la regió
Regio2 = find(NumReg == nodes(2));

%Canviem la regió dels nodes (en els que sigui necessari)
NumReg(Regio1,1) = NouNode;
NumReg(Regio2,1) = NouNode;

%Actualitzem el tamany de les regions
ai = DimReg(Regio1(1));
aj = DimReg(Regio2(1));

Modificar = NumReg == nodes(1); %Funció 'find' més ràpida
DimReg(Modificar) = ai+aj;

Modificar = NumReg == nodes(2);
DimReg(Modificar) = ai+aj;

%Trobem la posició de l'aresta que hem de llevar:
posicio = TrobarAresta(EdgesNodes,nodes(1), nodes(2));
if L <= T
    Cost(L-1) = (Pesos(posicio) + Cost(L)); %Cost de tenir 'n' nodes a tenir-ne 'n-1'
end
Pesos(posicio) = []; %Llevam el pes de l'aresta

%Llevem l'aresta
EdgesNodes(posicio,:)=[];

%Actualitzem el centre de la nova regió
NouCentre = ai/(ai+aj) * Centres(Regio1(1,:),) + aj/(ai+aj)*Centres(Regio2(1,:),);

Centres(Regio1,:) = repmat(NouCentre,length(Regio1),1);
Centres(Regio2,:) = repmat(NouCentre,length(Regio2),1);

%Canviar índexs i recalculam pesos de les que canviem
% (Les arestes van del node menor al major)
EdgesNodes(EdgesNodes(:,1) == nodes(1),1) = NouNode;
EdgesNodes(EdgesNodes(:,1) == nodes(2),1) = NouNode;
EdgesNodes(EdgesNodes(:,2) == nodes(2),2) = NouNode;

%Eliminem els llaços
Pesos(EdgesNodes(:,1) == EdgesNodes(:,2)) = [];
EdgesNodes(EdgesNodes(:,1) == EdgesNodes(:,2)) = [];

%Mirem si hi ha cap aresta invertida i la girem.
Canviar = find(EdgesNodes(:,2) < EdgesNodes(:,1));
EdgesNodes(Canviar,[1, 2]) = EdgesNodes(Canviar,[2,1]);

[EdgesNodes, idx]=unique(EdgesNodes, 'rows', 'first'); %Llevem les arestes repetides
Pesos = Pesos(idx);

if nodes(1) == NouNode
    n1 = [find(EdgesNodes(:,1) == NouNode); find(EdgesNodes(:,2) == NouNode)];
elseif nodes(2)==NouNode
    n1 = [find(EdgesNodes(:,1) == NouNode)];
end

Pesos = CalcularPes(EdgesNodes, n1, Centres, DimReg,m,n, Pesos);

```



```

L = L-1; %Tenim una regió menys

if L<=T
    Re = Centres(:,1);
    Gr = Centres(:,2);
    Bl = Centres(:,3);

    Re = reshape(Re,n,m)';
    Gr = reshape(Gr,n,m)';
    Bl = reshape(Bl,n,m)';

    Novalmg = cat(3,Re,Gr,Bl);
    Segmentada(:,:,L) = Novalmg;
    Imatge = reshape(NumReg,m,n);
    Etiquetes(:,:,L) = Imatge;
end
end

T = min(m*n,T);
Cost = Cost(1:T);
NReg = elbow(Cost,T);

Resultat = Etiquetes(:,:,NReg);
minal=[DirDesti NImg];
imwrite(uint8(Segmentada(:,:,NReg)),minal);
end

```

```

function posicio = TrobarAresta( EdgesNodes, Node1, Node2 )
    posicio = 0;

    A = ismember(EdgesNodes,[Node 1, Node2], 'rows');
    posicio = find(A==1);

    if posicio == 0 %No existeix aresta de Node1 a Node2, però potser si viceversa (però les arestes són
        no dirigides)
        posicio = TrobarAresta(EdgesNodes,Node2, Node1);
    end
end

```

```

function Pesos = CalcularPes( EdgesNodes,vector, Centres, DimReg,m,n, Pesos )

for i=1 : length(vector)
    nodi = EdgesNodes(vector(i),1); %Node 'i' i node 'j'
    nodj = EdgesNodes(vector(i),2);
    ai = DimReg(nodi); %Tamany regió node 'i' i 'j'
    aj = DimReg(nodj);
    Pesos(vector(i)) = norm(double(Centres(nodi,:)-Centres(nodj,:)))^2*(1/(m*n)) * (ai*aj)/(ai+aj);
end
end

```

A.5 Avaluació de les segmentacions

```

function RI=RandIndex(Segmentada, ImgConsens)
% Segmentada = etiquetes de la imatge segmentada
% ImgConsens = Imatge de consens de la ground truth

[x,y] = size(Segmentada);

Segmentada = reshape(Segmentada(:,,:), x*y, 1);
Ground = reshape(ImgConsens(:,,:), x*y, 1);

C = crosstab(Segmentada,Ground); %Taula de contingencia

%Apliquem algorisme del calcul de l'index de Rand
n=sum(sum(C));
nis=sum(sum(C,2).^2);
njs=sum(sum(C,1).^2);

t1=nchoosek(n,2); %possibles combinacions
t2=sum(sum(C.^2));
t3=.5*(nis+njs);

A=t1+t2-t3; %nombre píxels classificats correctament
RI=A/t1; %index de Rand (1971)
end

```

```

function dvi = varinfo(S1,S2)
% Mètode de variació de la informació.
% DVI = VARINFO(L1,L2) torna la variació de la informació en dos vectors
% L1 i L2 són les imatges convertides a tamany Nx1.

CarpetaArrel = fileparts( mfilename('fullpath') );
DirOriginal = cd([CarpetaArrel, filesep, 'mce']);

% entropies per la classificació individual
h1 = infoEntropia(S1);
h2 = infoEntropia(S2);

% informació mutua dels dos vectors
i12 = infoMutua(S1,S2);

% variació de la informació
dvi = h1 + h2 - 2*i12;

cd(DirOriginal);
end

```

```

function ImgConsens = ConsensGT( DirGround, NImg )

Dir = load(strcat(DirGround, NImg));

if length(Dir.groundTruth)>1
    O1 = Dir.groundTruth{1,1}.Boundaries;
    O = cat(3,O1);
end
if length(Dir.groundTruth)>2
    O2 = Dir.groundTruth{1,2}.Boundaries;
    O = cat(3,O1,O2);
end

```

```

end
if length(Dir.groundTruth)>3
    O3 = Dir.groundTruth{1,3}.Boundaries;
    O = cat(3,O1,O2,O3);
end
if length(Dir.groundTruth)>4
    O4 = Dir.groundTruth{1,4}.Boundaries;
    O = cat(3,O1,O2,O3,O4);
end
if length(Dir.groundTruth)>5
    O5 = Dir.groundTruth{1,5}.Boundaries;
    O = cat(3,O1,O2,O3,O4,O5); %Cream matriu 3dim
end

[m,n,z] = size(O);
R = zeros(m,n);
C = zeros(m,n,z);

for i = 1:m
    for j = 1:n
        for k = 1 : z
            if O(i,j,k)==1
                R(i,j) = R(i,j)+1;
            end
        end
    end
end

for k = 1:z
    for i = 1:m
        for j=1:n
            if R(i,j) >= k
                C(i,j,k) = 1;
            end
        end
    end
end

k = 2;
for i=1 : z
    for j=1 : z
        V(j,i) = baddeley(C(:,:,j),O(:,:,i),k);
    end
end

for j=1 : z
    G(j) = 1/z * sum(V(j,:)); %Minimean
end

MIN = min(G);
pos = find(G == MIN);

ImgConsens = Dir.groundTruth{1,pos}.Segmentation;
end

```


BIBLIOGRAFIA

- [1] B. N. Subudhi, V. Thangaraj, E. Sankaralingam, and A. Ghosh, "Tumor or abnormality identification from magnetic resonance images using statistical region fusion based segmentation." *Magnetic resonance imaging*, vol. 34, pp. 1292–1304, 2016. 1
- [2] W. Ji, Z. Qian, B. Xu, Y. Tao, D. Zhao, and S. Ding, "Apple tree branch segmentation from images with small gray-level difference for agricultural harvesting robot," *Optik-International Journal for*, vol. 127, pp. 11 173–11 182, 2016. 1
- [3] J. K. Beleño Sáenz, J. Herrera , S. Medina, G. Martínez, "Clasificación de los frutos de café según su estado de maduración y detección de la broca mediante técnicas de procesamiento de imágenes," *Prospectiva*, vol. 14, pp. 15–22, 2016. 1
- [4] R. Orduna, A. Jurio, D. Paternain, H. Bustince, P. Melo-Pinto, and E. Barrenechea, "Segmentation of color images using a linguistic 2-tuples model," *Information Sciences*, vol. 258, pp. 339–352, 2014. 2, 13, 22
- [5] Y. Deng and B. Manjunath, "Unsupervised segmentation of colour-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 800–810, 2001. 2, 24, 26
- [6] D. Arthur and S. Vassilvitskii, "K-Means++: the Advantages of Careful Seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, vol. 8, pp. 1027–1025, 2007. 2, 9, 29, 30
- [7] J. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer, 2013. 2, 9, 32
- [8] T. Adamek and N. O'Connor, "Stopping region-based image segmentation at meaningful partitions," *Semantic Multimedia*, pp. 15–27, 2007. 2, 7
- [9] M. Rand William, "Objective Criteria for the Evaluation of Clustering Methods," *American Statistical Association*, vol. 66, no. 336, pp. 846–850, 2009. 2, 37
- [10] M. Meila, "Comparing clusterings-an information based distance," *Journal of Multivariate Analysis*, vol. 98, pp. 873–895, 2007. 3
- [11] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, p. 80, 1945. 3, 45
- [12] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 898–916, 2011. 3, 37, 40

- [13] R. C. Gonzalez, R. E. Woods, and B. R. Masters, *Digital Image Processing, Third Edition*. Pearson, 2007. 5
- [14] Y. J. Zhang, *Advances in image and video segmentation*. IRM Press, 2006. 6
- [15] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, pp. 191–203, 1984. 7
- [16] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 100–132, 1985. 8
- [17] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993. 8
- [18] L. Lucchese, S. K. Mitra, and S. Barbara, "Color Image Segmentation : A State-of-the-Art Survey," *PINSA*, 67, A, pp. 207–221, 2000. 8
- [19] H. C. Yung, "Segmentation of color images based on the gravitational clustering concept," *Optical Engineering*, vol. 37, p. 989, 1998. 9
- [20] L. Shafarenko, M. Petrou, and J. Kittler, "Histogram-based segmentation in a perceptually uniform color space," *IEEE Transactions on Image Processing*, vol. 7, no. 9, pp. 1354–1358, 1998. 9
- [21] M. Barni, S. Rossi, and A. Mecocci, "A Fuzzy Expert System for Low Level Image Segmentation," *Proc. of the 8th European Signal Processing Conference (EUSIPCO-96)*, vol. III, pp. 1725–1728, 1996. 10
- [22] X. Ren, "Multi-scale improves boundary detection in natural images," *European Conference on Computer Vision*, p. 545, 2008. 10
- [23] W. Ma and B. Manjunath, "Edge flow: A framework of boundary detection and image segmentation," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, no. 1, pp. 744–749, 1997. 10
- [24] R. Xiaofeng and L. Bo, "Discriminatively trained sparse code gradients for contour detection," *Neural Information Processing Systems*, pp. 593–601, 2012. 10
- [25] P. Mesejo, Ó. Ibáñez, Ó. Cerdón, and S. Cagnoni, "A survey on image segmentation using metaheuristic-based deformable models: State of the art and critical analysis," *Applied Soft Computing*, vol. 44, pp. 1–29, 2016. 11
- [26] N. Covavisaruch and T. Tanatipanond, "Deformable Contour for Brain MR Images by Genetic Algorithm : From Rigid to Training Approaches Faculty of Engineering , Chulalongkorn University," *Proceedings, Image and Vision*, 1999. 11
- [27] S. Karungaru, M. Fukumi, and N. Akamatsu, "Feature extraction for face detection and recognition," *RO-MAN 2004. 13th IEEE Int. Work. Robot Hum. Interact. Commun. (IEEE Cat. No.04TH8759)*, pp. 235–239, 2004. 11
- [28] A. Vidyarthi and N. Mittal, "Brain tumor segmentation approaches: Review, analysis and anticipated solutions in machine learning," *Proceedings of the 2015 39th National Systems Conference, NSC 2015*, 2016. 11

-
- [29] R. Schettinib, "Color image segmentation using Hopfield networks," *Image (Rochester, N.Y.)*, vol. 15, pp. 161–166, 1997. 11
- [30] O. Hironori, N. Kaneki, and O. Koichi, "Automatic color segmentation method using a neural network model for stained images," *IEICE TRANSACTIONS*, 1994. 11
- [31] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 3rd ed. Springer, 2009. 11
- [32] T. M. Nguyen and Q. J. Wu, "Dirichlet Gaussian mixture model: Application to image segmentation," *Image and Vision Computing*, vol. 29, no. 12, pp. 818–828, 2011. 12
- [33] H. Bustince, E. Barrenechea, and M. Pagola, "Restricted equivalence functions," *Fuzzy Sets and Systems*, vol. 157, pp. 2333–2346, 2006. 14
- [34] H. Bustince, M. Pagola, E. Barrenechea, J. Fernandez, P. Melo-Pinto, P. Couto, H. R. Tizhoosh, and J. Montero, "Ignorance functions. An application to the calculation of the threshold in prostate ultrasound images," *Fuzzy Sets and Systems*, vol. 161, pp. 20–36, 2010. 17, 18
- [35] F. Herrera and L. Martínez, "A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, pp. 227–234, 2001. 19, 21
- [36] F. Herrera and L. Martínez, "A 2-Tuple Fuzzy Linguistic Representation Model for Computing with Words," *IEEE Transactions on fuzzy systems*, vol. 33, no. 7, pp. 1117–1122, 2000. 21
- [37] F. Herrera and L. Martínez, "A fuzzy linguistic representation model based on a symbolic translation," *Citeseer*, pp. 25–28, 1999. 21
- [38] E. Herrera-Viedma, a. G. López-Herrera, M. Luque, and C. Porcel, "A Fuzzy Linguistic IRS Model Based on a 2-Tuple Fuzzy Linguistic Approach," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 15, no. 2, pp. 225–250, 2007. 22
- [39] M. E. Yüksel and M. Borlu, "Accurate segmentation of dermoscopic images by image thresholding based on type-2 fuzzy logic," *IEEE Transactions on Fuzzy Systems*, vol. 17, pp. 976–982, 2009. 23
- [40] Y. D. Y. Deng, C. Kenney, M. Moore, and B. Manjunath, "Peer group filtering and perceptual color image quantization," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 21–24, 1999. 25
- [41] Y. Chien, "Pattern Classification and Scene Analysis," *IEEE Transactions on Automatic Control*, no. August, pp. 462–463, 1974. 29
- [42] "Mathworks," p. <https://es.mathworks.com/help/fuzzy/fcm.html?searc>, 2017. 32

- [43] S. Sibi, S. P. R. M, P. Kumar, R. Singh, and A. Baskar, "Image Segmentation by Construction of Shortest Spanning Tree Using Prim's Algorithm," *International Journal of Scientific & Engineering Research*, vol. 4, no. 11, pp. 1588–1594, 2013. 35
- [44] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, pp. 212–225, 2008. 38
- [45] N. L. Fernández-García, A. Carmona-Poyato, R. Medina-Carnicer, and F. J. Madrid-Cuevas, "Automatic generation of consensus ground truth for the comparison of edge detection techniques," *Image and Vision Computing*, vol. 26, pp. 496–511, 2008. 41
- [46] C. Lopez-Molina, H. Bustince, and B. De Baets, "Separability Criteria for the Evaluation of Boundary Detection Benchmarks," *IEEE Transactions on Image Processing*, vol. 25, pp. 1047–1055, 2016. 41
- [47] R. Lowry, *Concepts & Applications of Inferential Statistics*. Vassar College, 1998. 47