Índex

Índex de figures

1. Introducció

1.1 Motivacions:

1.1.1 Els cercadors de text

Actualment els buscadors de text s'han transformat en una eina fonamental en el món de la informàtica en gairebé tots els seus camps. Començant pels simples buscadors de paraules que es poden trobar en processadors de texts com el bloc de notes en els que es limiten a buscar coincidències exactes i acabant pels complexes motors de búsqueda de portals d'Internet com Google o Yahoo, capaços de realitzar cerques temàtiques i dinàmiques en comptes de les obtuses búsquedes de conincidències.

Els grans buscadors d'Internet són capaços de realitzar cerques per temes i de fer correccions en els paràmetres de cerca; poden rectificar errors ortogràfics en la paraula que es busca i obtenir resultats satisfactoris de totes maneres, identificar sinònims de les paraules per incloure'ls com a paràmetre en la búsqueda. Això ho aconsegueixen mitjançant vocabularis, bases de dades on s'emmagatzemen grups de paraules amb un significat semblant, característiques semblants, etc.

Els mateixos buscadors ofereixen la possibilitat de realitzar cerques d'imatges. Aquestes búsquedes es realitzen de la mateixa manera que es realitzen en la cerca de documents web o de text però en lloc d'utilitzar el cos del document utilitzen el nom de l'arxiu. Això pot portar a cerques errònies o poc complertes degut al nom dels fitxers, ja sigui per falta d'informació o per tenir noms enganyosos.

Exemple d'alguns resultats erronis cercant la paraula "moutain" en el cercador d'imatges de Google. A dia 11 de juliol de 2007 les següents imatges apareixien en alguna de les dues primeres pàgines dels resultats:

1. Resultats de Google Imatges amb la pataula "mountain"

1.1.2 Els cercadors de vídeos

A mesura que les noves tecnologies com Internet s'han anat extenent i millorant, la oferta de continguts multimèdia a la xarxa ha anat creixent de forma molt ràpida. Gràcies a pàgines com la popular www.youtube.com o a programes d'intercanvi de fitxers com ara l'Emule qualsevol persona que disposi de connexió a Internet pot accedir sense problemes a una gran quantitat de vídeos de gairebé qualsevol temàtica. Aquest gran volum d'informació fa que els sistemes de búsqueda siguin indispensables per poder obtenir l'arxiu desitjat. No obstant ens trobem amb el mateix problema que amb la búsqueda d'imatges, les cerques s'han de realitzar pel nom del fitxer en lloc de fer-ho per el contingut. Això, que a primera vista pot semblar una nimietat, pot arribar a ser un problema bastant incòmode per a usuaris d'aquest tipus de serveis; un dels casos més freqüents és el de l'intent de descarrega de determinat vídeo a través d'un programa d'intercanvi d'arxius que culmina amb la descàrrega d'un arxiu que no té res a veure amb la búsqueda efectuada. Per això vàrem creure que en lloc de buscar arxius pel seu nom, una cerca on s'analitzés el contingut dels vídeos seria molt més útil.

1.1.3 Els classificadors de vídeo

Un primer pas per a poder buscar fitxers de vídeo a través del seu contingut és aconseguir crear un mètode capaç de classificar-lo. Una cerca on s'analitzés cada fitxer en temps real seria molt lenta i feixuga, cosa que li restaria tota la utilitat. Per a poder obtenir bons resultats de forma àgil una bona solució seria analitzar els vídeos prèviament i assigna'ls-hi etiquetes o classificar-los en diverses categories.

Vàrem creure que idear un mètode que fos capaç de classificar un vídeo dins una

temàtica seria un bon tema per a un projecte final de carrera. A més a més, aquest classificador podria arribar a tenir altres utilitats com, per exemple, controlar els continguts que s'emeten en una televisió en horari infantil; distingint programes adequats de vídeos on apareguessin continguts violents o de caràcter eròtic.

1.1.4 Classificació de diferents tipus d'esports

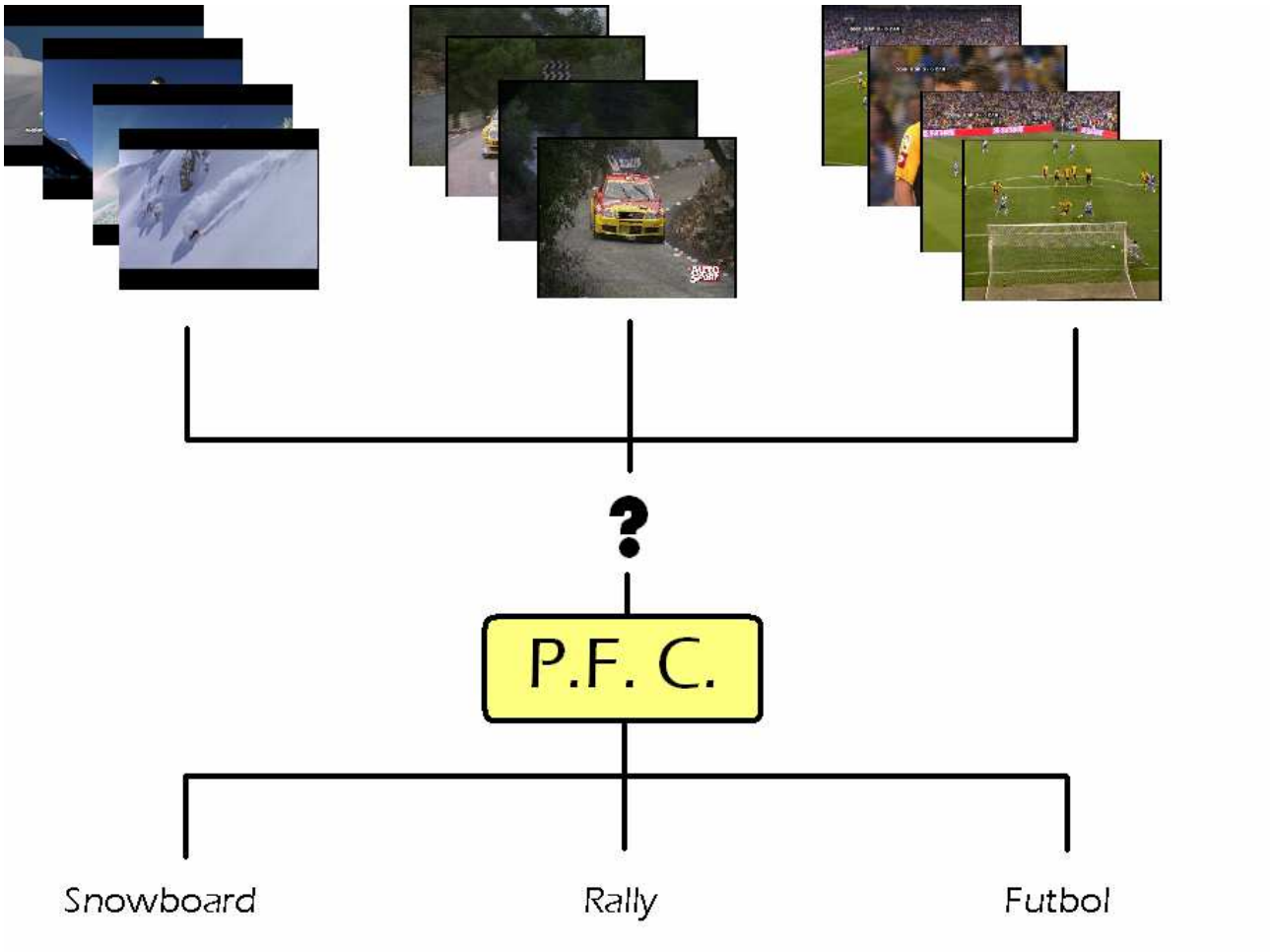
Actualment, amb els ordinadors domèstics de que disposem, no és viable arribar a pensar que es pot classificar qualsevol vídeo ja que el nombre de temàtiques que poden tractar tendeix a infinit. Tenint en compte que 1 minut de vídeo equival aproximadament a 1 megabyte , el volum de dades que es necessitaria per a poder crear models podria arribar a ser de petabytes (2^50 bytes) o exabytes (2^60 bytes). A més a més, el temps que es requeriria per extreure les característiques de totes les característiques seria de dècades (veure resultats).

Per això vàrem creure necessari acotar el camp en el que es desenvoluparia el treball: En lloc de desenvolupar un classificador capaç de diferenciar milers de categories vàrem decidir fer-ne un capaç de distingir diferents activitats esportives. Reduint l'escala del treball era més senzill marcar una estratègia d'acció i decidir quins passos calia seguir.

Degut als bons resultats que obtenen els cercadors de documents de text, com per exemple Google.com o Yahoo.com, vàrem decidir adaptar la seva estratègia per a la classificació i recuperació d'imatges i vídeos. De la mateixa manera que els documents de text estan definits per paraules, podríem dir que un vídeo ho està per paraules visuals. Si en un document que parli sobre informàtica és freqüent trobar-hi les paraules: ordinador, ratolí, pantalla, etc. En un vídeo d'un partit de futbol s'hi podrien trobar paraules visuals, petits fragments d'imatge, com la pilota de futbol, un tros de gespa o l'àrea petita del camp. Un classificador de texts determina la tipologia d'aquests segons la freqüència i la disposició amb què apareixen, seguint el símil, el nostre classificador ho hauria de fer segons la presència de les paraules visuals.

1.2 Objectius:

El present projecte final de carrera s'emmarca en el camp de la visió per computador. Podríem considerar que tracta més l'àmbit de la investigació que no pas el del desenvolupament ja que el seu principal objectiu és intentar determinar els procediments necessaris per a la categorització de diferents tipologies de vídeo. No obstant, si s'arriba a trobar un mètode per a resoldre aquest problema, també es desenvoluparà una senzilla aplicació que permeti classificar un vídeo segons el seu contingut; més a mode demostratiu que pràctic.



2. L'objectiu d'aquest PFC és classificar vídeos segons la seva categoria

Per tal d'afrontar el repte que hem plantejat anteriorment vàrem decidir que calia marcar-nos els següents objectius:

1. Obtenir mostres de vídeos on apareguin diferents disciplines esportives.
2. Escollir un mètode per a detectar regions característiques i punts d'interès d'una

imatge: Paraules visuals. Per a realitzar aquesta tasca vàrem comprovar el funcionament de diferents mètodes de detecció de regions sobre imatges de diferents categories.

3. Crear un vocabulari que serveixi per descriure els diferents frames extrets dels vídeos. Per assolir aquest propòsit extraurem diferents paraules (regions) de les imatges de mostra mitjançant varis detectors de regions que posteriorment seran definides a través d'un descriptor d'imatges SIFT. Utilitzant l'algorisme de clusterització K-means agruparem les paraules que més s'assimilin com a un mateix mot.
4. Descriure els frames dels vídeos de mostra mitjançant els vocabularis obtinguts en el punt anterior. Analitzant els histogrames resultants decidirem amb quin sistema de detecció de regions hem obtingut un millor vocabulari. Els histogrames i el vocabulari que aconsegueixin una millor definició de les imatges seran els usats en les següents tasques.
5. Amb l'ajuda de Support Vector Machines obtenir un model de classificació de les diferents categories de vídeo. Elaborar un script que, mitjançant els models, ens permeti assignar una categoria a un vídeo.
6. Elaborar una senzilla interfície gràfica per a Matlab que faciliti l'ús de l'script anterior. Aquesta interfície tindrà un objectiu purament demostratiu ja que l'objectiu final del projecte és trobar un mètode que permeti classificar categories de vídeos, no el desenvolupament d'una aplicació.

1.3 Entorn de treball

1.3.1 Matlab

Per tal d'assolir els objectius que ens hem marcat vàrem utilitzar, bàsicament, l'entorn de treball Matlab 7.1 que es troba disponible per els tres sistemes operatius més comuns: Windows, Mac Os, i Linux.

El Matlab, abreviatura de Matrix Laboratory, és un programa dissenyat per "The Mathworks" destinat al càlcul i l'anàlisi numèric. Permet treballar amb matrius, i vectors de forma ràpida i intuïtiva. A través d'una línia de comandes dóna la possibilitat de realitzar gairebé múltiples operacions amb matrius i vectors de qualsevol mida i tipus: sumes, restes, representacions, aplicacions de filtres, aplicació de mètodes estadístics, llindaritzacions, i un llarg etc. Tot això fa que sigui molt útil en una gran varietat de camps: tractament i anàlisis d'imatges, adquisició de dades, economia, intel·ligència artificial, etc.

Es calcula que l'any 2004 més d'un milió d'investigadors i estudiants de diferents parts del món utilitzaven aquest software.

Una de les característiques que fan del Matlab una eina molt interessant és la possibilitat de crear noves funcions i afegir-les al programa. Permet la creació d'scripts mitjançant un llenguatge de programació propi anomenat "M-code" o simplement "M". Aquest llenguatge, que és una mescla entre Fortran i C, inclou totes les funcions del Matlab a més d'estructures típiques dels llenguatges iteratius com ara els bucles. Un tret destacable és que permet la interacció amb programes externs al Matlab, cosa que li permet llançar aplicacions i emmagatzemar-ne els resultats.

Un altre virtut destacable del Matlab és la possibilitat d'afegir-hi diferents "Toolbox". Una Toolbox, caixa d'eines en anglès, és un conjunt d'scripts i funcions externes al programa que s'han desenvolupat per atacar un conjunt de problemes. Solen estar especialitzades en determinats camps com ara la estadística, la biologia, el tractament d'imatges, etc. Podríem dir que són l'equivalent a les llibreries en altres llenguatges de programació. Nosaltres hem utilitzat les següents Toolbox:

- Statistic Toolbox: Eina que permet realitzar diferents càlculs estadístics

- Image Toolbox: Permet analitzar i modificar imatges

El Matlab també inclou la possibilitat de crear senzills entorns gràfics per tal de simplificar l'ús de les funcions. Aquest és un camp que no hem explotat massa en aquest projecte ja que vàrem considerar que el principal objectiu era la investigació i no el desenvolupament d'una aplicació concreta.

1.3.1 Elecció del Sistema Operatiu

Tal i com s'indica en l'apartat anterior, el Matlab es troba disponible per els tres sistemes operatius més usats. Això ens donava llibertat a l'hora d'escollir quin utilitzaríem per desenvolupar el nostre projecte.

Després de definir els passos que calia seguir per assolir els objectius proposats vàrem veure que necessitaríem utilitzar petits algorismes com ara detectors de regions o màquines classificadores d'imatges que ja havien estat implementats per altres investigadors i universitats. Vàrem observar que aquests programes, els quals es poden descarregar de forma gratuïta a través d'Internet, estaven fets per a sistemes operatius Unix. Per tal de no perdre temps buscant els codis font d'aquest programes i compilant-

los en un sistema Windows, vàrem decidir desenvolupar el projecte dins d'un entorn Linux. Per tant, tot i que els scripts M de Matlab són multiplataforma, algunes de les funcions implementades en aquest treball no funcionen en els demés sistemes operatius ja que també utilitzen programes i crides pròpies de Linux.

1.4 Planificació

Per tal d'organitzar la feina a realitzar es van temporitzar les tasques. D'aquesta manera es facilitava la consecució dels objectius fixats prèviament. La planificació del projecte es divideix en 16 etapes:

1) Obtenció de diferents vídeos de contingut esportiu
2) Extracció de diferents fragments dels vídeos obtinguts
3) Extracció dels frames dels diferents vídeos
4) Separació dels frames dels vídeos en SHOTS
5) Estudi dels diferents mètodes d'obtenció de regions
6) Obtenció de paraules visuals mitjançant els detectors de regions
7) Descripció de les paraules mitjançant SIFTs
8) Creació dels diferents vocabularis
9) Creació dels histogrames de vocabulari
10) Correspondència d'Imatges
11) Estudi de les SVM
12) Creació dels models de classificació
13) Creació del classificador
14) Realització de proves amb el classificador
15) Realització de la documentació
16) Setmana de Marge

març

	dl	dm	dm	dj	dv	ds	dg
				1	2	3	4
1	5	6	7	8	9	10	11
	12	13	14	15	16	17	18
2	19	20	21	22	23	24	25
3	26	27	28	29	30	31	

abril

	dl	dm	dm	dj	dv	ds	dg
							1
4	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
5	16	17	18	19	20	21	22
6	23	24	25	26	27	28	29
	30						

maig

	dl	dm	dm	dj	dv	ds	dg
		1	2	3	4	5	6
7	7	8	9	10	11	12	13
8	14	15	16	17	18	19	20
9	21	22	23	24	25	26	27
	28	29	30	31			

juny

	dl	dm	dm	dj	dv	ds	dg
					1	2	3
10	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
11	18	19	20	21	22	23	24
12	25	26	27	28	29	30	

juliol

	dl	dm	dm	dj	dv	ds	dg
							1
	2	3	4	5	6	7	8
13	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
14	23	24	25	26	27	28	29
	30	31					

agost

	dl	dm	dm	dj	dv	ds	dg
				1	2	3	4	5
15	6	7	8	9	10	11	12
	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
16	27	28	29	30	31		

setembre

dl	dm	dm	dj	dv	ds	dg	
					1	2	
3	4	5	6	7	8	9	Entrega del P.F.C
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	Presentació del P.F.C
24	25	26	27	28	29	30	

3. Planificació

2. Descripció dels vídeos

2.1 Obtenció dels Vídeos i imatges

La forma més fàcil d'obtenir vídeos de diferents esports és a través d'Internet. Inicialment la meva idea va ser aconseguir-los a través de la xarxa mitjançant programes d'intercanvi p2p però després d'haver-ne descarregat uns quants em vaig adonar que la qualitat era realment baixa ja que la gran majoria eren gravats directament d'emissions televisives. En aquest punt varem decidir que la solució seria extreure directament les imatges i els vídeos de DVDs. Em vaig adreçar a diferents mitjans de comunicació que en ocasions editen DVDs de contingut esportiu: televisions especialitzades, com ara TeleDeportes o Eurosport; premsa escrita, diaris i revistes especialitzades; etc. Però en tots els casos vaig obtenir una negativa com a resposta o, com a molt, ofertes de compra de vídeos esportius. Finalment vaig aconseguir diversos DVDs gràcies a amics i coneguts que, desinteressadament, me'ls van deixar.

Un cop disposàvem de la font per els vídeos calia procedir a l'extracció d'aquests. El volum de dades d'un DVD és molt elevat, per aquest motiu no podíem extreure tot el vídeo d'un DVD; vàrem decidir agafar, només, petits fragments de 5 minuts de diferents parts de les gravacions. Per tal de reduir l'espai que ocuparien els varem decidir utilitzar un sistema de compressió. Com que ens interessava obtenir imatges dels vídeos cada cert temps el sistema de compressió utilitzat havia de fer servir un flux de dades constant (Bit rate estàtic), després de realitzar diferents proves amb varis sistemes de compressió determinàrem que el sistema de compressió més adient i amb el que es perdia menys qualitat era l'XviD. El procés d'extracció dels vídeos el vaig realitzar amb el programa "Dvd2avi", un petit programa de codi obert que permet guardar el contingut d'un DVD en format .avi de forma ràpida i senzilla.

Després d'haver aconseguit els vídeos amb els que es realitzaria el treball era el moment d'extraure'n les imatges. Inicialment intentàrem agafar les imatges dels vídeos mitjançant el programa "Virtual Dub", un potent programa d'edició de vídeo amb llicència GNU que permet transformar el vídeo a altres formats, modificar-ne diferents paràmetres com la velocitat, la freqüència de mostreig dels frames, el format, etc. i extreure els frames (imatges) d'un vídeo determinat. El VirtualDub es pot utilitzar tant a través d'un entorn de finestres com amb una finestra de comandes tot i que amb aquest últim format

el ventall de possibilitats del programa es redueix considerablement. Tot i que és un dels programes més eficients en el seu camp finalment varem descartar el seu ús ja que per realitzar l'extracció de les imatges era necessari entrar a l'entorn gràfic i modificar certs paràmetres de forma manual. Això impedia l'automatització d'aquest procés i, com a conseqüència, no podíem obtenir les imatges dels vídeos directament des de Matlab. Un cop descartat el VirtualDub calia buscar un programa alternatiu capaç de realitzar accions semblants però des d'una línea de comandes, l'alternativa escollida va ser l'mplayer; un altre programa amb llicència GNU disponible tant per a sistemes operatius Windows com per a sistemes Linux. El programa no disposa d'una opció per extreure directament una imatge d'un vídeo cada cert temps, no obstant, això es pot aconseguir mitjançant la comanda:

```
mplayer VideoOrigen -vf decimate=nFrames:diferencia:framescon:percentatge -v
o png:quality=qualitat:outdir=desti
```

4. Comanda que permet l'extracció dels frames d'un vídeo

El paràmetre -vf decimate=nFrames:diferència:framescon:percentatge reprodueix només els frames del vídeo que difereixen un determinat llindar del seu predecessor. Nframes és el nombre màxim de frames que podran ser omesos entre frame i frame. Si en el valor "diferencia" hi posem el seu valor màxim (1.000.000) el programa considerarà que tots els frames són iguals. Sabent que els vídeos dels que disposem es reprodueixen a 25 imatges per segon i cada quants milisegons volem obtenir un frame podem deteteminar el paràmetre nFrames a través de la següent equació: Temps = nFrames/25 --> nFrames = 25*temps.

Amb el paràmetre -vo aconseguim que, en lloc de mostrar la sortida del vídeo per pantalla, l'emmagatzemi en un fitxer de tipus png (també es pot realitzar la mateixa acció amb fitxers .jpg). Amb "quality" i "outdir" podem escollir la qualitat i la carpeta on es guardaran les imatges resultants.

Cal tenir en compte que els frames es guarden i s'anomenen amb l'ordre que apareixen al vídeo, el primer frame serà el 00000001.png, el segon 00000002.png, etc. Aquest detall és bastant rellevant ja que en molts scripts d'aquest projecte s'ha suposat que les imatges extretes seguien aquesta numeració.

2.2 Les categories de Vídeo

Per a realitzar les diferents proves i experiments d'aquest projecte vàrem optar per utilitzar sis tipus de vídeos diferents relacionats amb el món de l'esport. Inicialment

havíem pensat utilitzar un nombre de varietats de vídeo més elevat però la dificultat per obtenir fonts de qualitat i l'alt volum de dades que ocupaven ens van fer reduir aquest nombre.

Per a cada categoria vàrem decidir extreure'n 7 fragments diferents. Aquests 7 fragments els vàrem dividir en dos grups: els vídeos font i els vídeos de test. Els vídeos font tenien una duració d'uns 4 minuts aproximadament; La seva funció era servir d'exemples significatius de la classe a la que pertanyien i ser utilitzats per a crear el vocabulari que la definia. Per altra banda els vídeos de test ,amb una duració d'entre 20 i 40 segons, serien els que s'utilitzarien per a realitzar les proves amb el classificador final.

El nombre de frames que s'han extret de cada fitxer de vídeo font era aproximadament d'uns 240, un frame per cada segon, obtenint així un total de gairebé 500 mostres per categoria. Per a realitzar els experiments i crear els diferents vocabularis, però, només en vàrem emprar 300 ja que dins d'un mateix vídeo hi podem trobar imatges relativament semblants que no aporten nova informació. Per reduir aquest nombre de 500 a 300 vàrem classificar les imatges en shots (posteriorment s'explica detalladament aquest procés).

Dels vídeos de test també vàrem extreure un frame per cada segon, aconseguint unes 30 imatges per a cada vídeo. Aproximadament disposàvem de 150 mostres per classe repartides entre 5 fragments de vídeo diferents que ens permetrien comprovar si els resultats havien estat satisfactoris.

Les sis categories escollides, condicionades per la poca disponibilitat de DVDs on hi apareguessin esports, van ser: Bicicleta tot terreny, Bàsquet, Futbol, competicions de circuit, Rally i Snowboard.

2.3 Breu descripció de les diferents classes

Rally

Els campionats de rally es poden classificar en dos grups segons la superfície en la que es corre: asfalt o terra. Aquesta categoria engloba ambdós competicions sota un mateix grup. Per a poder identificar els dos tipus de rally sota la mateixa classer vàrem decidir crear un vídeo font per a cada superfície.

Els fragments de vídeo d'aquesta classe s'han extret del següents DVDs:

- Las mejores imágenes de los campeonatos de España de Rallyes y Circuitos

2005, *AUTOhebdo SPORT,* any 2005.

- Passats de canto i de sostre a terra “3”, *Jordi Colomer,* any 2003.

Vídeo1: Duració = 4:00 minuts Frames extrets = 240

Vídeo2: Duració = 3:52 minuts Frames extrets = 232

5. Frames dels vídeos de Rally

Competicions en circuit

Les competicions automobilístiques que es celebren dins de circuit tancat també es poden dividir en dos grups: Les competicions de turismes (Per exemple el “Campionat del món de GT”, o copes monomarca com la Copa Renault Clio) i les competicions conegudes com a “Formules” (La Formula1 o les Renault World Series). Tal i com hem fet amb els vídeos de Rally, per tal d’acollir sota la mateixa denominació les dues modalitats hem creat un vídeo font per a cada una d’elles.

Els fragments de vídeo d’aquesta classe s’han extret del següents DVDs:

- Las mejores imágenes de los campeonatos de España de Rallyes y Circuitos 2005, *AUTOhebdo SPORT,* any 2005.
- The Green Flag 2003 MSA British Touring Car Championshib (BTCC), *Duke DVD / Octagon CSI,* any 2003

Vídeo1: Duració = 4:32 minuts Frames extrets = 271

Vídeo2: Duració = 3:40 minuts Frames extrets = 220

6 . Frames dels vídeos d’automobilisme de Circuit

Futbol:

Per a la descripció d’aquesta categoria vàrem intentar obtenir vídeos de més d’un partit de futbol per tal d’intentar reconèixer un major nombre de situacions que puguin aparèixer dins d’un partit de futbol. Com a mostra vàrem utilitzar el partit de la final de la

Copa del Rei 2006, que enfrontava el RCD Espanyol i el RCD. Zaragoza; i el partit de lliga de la temporada 2005/2006 que va enfrontar el R. Madrid CF amb el FC Barcelona, obtingut a través d'Internet.

- La copa de tots, *Santa Monica / Microflux,* any 2006
- R. Madrid – FC Barcelona, *anònim,* any 2005

Vídeo1: Duració = 9:12 (utilitzats 4 minuts) Frames extrets = 552

Vídeo2: Duració = 3:00 minuts Frames extrets = 180

7. Exemples de Frames dels vídeos de futbol

Snowboard:

Tot i que dins d'aquest esport hi ha diverse modalitats, les més populars són el Freestyle (estil lliure) i el Freeride (Snowboard fora de pista). La gran majoria de vídeos que es comercialitzen sobre aquests esports combinen les dues modalitats ja que la les dues es poden practicar a la vegada.

DVDs utilitzats per a l'extracció dels vídeos:

- Riding for a living*, NitroUSA / Onboard Magazine,* any 2006.
- Good times, *ExtremeVideo / Onboard Magazine,* any 2006

Vídeo1: Duració = 3:00 minuts Frames extrets = 180

Vídeo2: Duració = 2:23 minuts Frames extrets = 150

8**. Exemples de Frames dels vídeos d'Snowboard**

Bàsquet:

D'aquesta categoria no vàrem ser capaços de trobar més d'un vídeo: la semifinal de la conferència oest de la NBA entre els Heat de Miami i els Maverics de Dallas. Dins d'un mateix partit de bàsquet no es solen produir situacions molt diverses, degut a aquest

motiu i a que només disposava d'un vídeo d'aquest partit vaig creure adient extreure només un Vídeo Font, però amb el doble de duració.

- Semifinal Conferencia Este 2006: Heat – Maverics, *anònim,* any 2006

Vídeo1: Duració = 8:00 minuts Frames extrets = 480

9. Exemple de Frames del vídeo de bàsquet

Bicicleta Tot Terreny

Tal i com ens va passar en l'anterior classe, només vàrem ser capaços d'aconseguir un sol DVD que tractés aquest esport. A diferència del bàsquet, però, en aquest apareixien situacions molt diverses ja que, per exemple, alguns campionats d'aquest esport es disputen dins del bosc mentre que d'altres ho fan en camps de terra. Per a aquesta categoria vàrem crear dos Vídeos Fonts on els escenaris en els que transcorrien el campionat fossin diferents.

- Apaguen las luces, *Tremendous Entertainment Usa,* any desconegut

Vídeo1: Duració = 3:15 minuts Frames extrets: 195

Vídeo2: Duració = 3:11 minuts Frames extrets: 191

10. Exemple de frames del vídeo de BTT

3. Shots

3.1 Què és un Shot?

Dins d'un vídeo d'una determinada categoria podem trobar-hi tipus d'imatges bastant diferents. Per exemple, en un vídeo d'un partit de futbol podem trobar-hi primers plans dels jugadors, imatges del públic, vistes aèries dels camps de futbol, etc. En la següent figura podem observar diferents frames extrets del dvd de la final de la Copa del Rei 2006 podem observar com la quarta i la cinquena imatge podrien pertànyer a altres categories de vídeo, com per exemple bàsquet.

11. Exemples de diferents imatges típiques d'un partit de futbol

Així doncs, classificar una sola imatge pot ser molt complicat ja que no és una mostra prou significativa del contingut d'un vídeo. No obstant, dins d'un vídeo trobem imatges molt similars entre elles. Generalment, a menys que es produeixi un canvi d'escena o de càmera, la diferència entre un frame i els seu predecessor i successor és bastant petita i no aporta nova informació sobre el vídeo. Per això ens cal trobar un sistema que ens permeti discernir quan es produeix un canvi que aporta nova informació i agrupar les imatges que no ho fan. Un conjunt de frames que no difereixen excessivament entre sí es coneix com a *shot.* En la següent figura es pot observar com es separen tres shots d'un vídeo d'snowboard:

12. En aquesta seqüència d'imatges es pot observar un canvi de shot en el 5è i 9è frame

Quan es produeix un d'aquests canvis les propietats de la imatge, com per exemple el color, acostumen a canviar sensiblement, això fa que sigui relativament senzill comparar-los segons aquestes variacions. Freqüentment s'utilitzen els histogrames de color per descriure una imatge, per això un bon mètode per localitzar els canvis de shot és la comparació d'aquests.

3.2 Els models de color

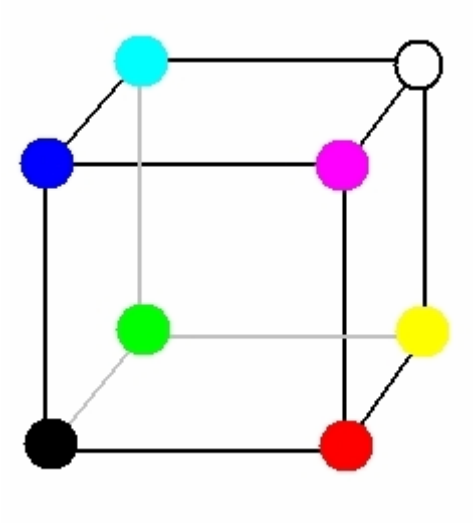
Les imatges de color estan representades per tres capes que es defineixen per model de color. Cada píxel, un punt de la imatge, conté tres components que sumades subministren la informació del color del punt. A continuació es poden observar les tres components RGB d'una imatge en color [1].

13. La unió dels tres plans RGB de la imatge determina la imatge resultant

3.2.2 El Model de color RGB

El model de color RGB prové del nom anglès dels colors vermell, verd i blau (Red,Gren,Blue). A partir dels 3 colors primaris es pot crear qualsevol altre color, partint d'aquest fet el model de color RGB representa els diferents colors que existeixen segons la intensitat dels colors primaris amb la que estan compostos. Per representar el nivell

d'intensitat dels tres colors és necessari escollir una escala, com que aquest model és molt usat dins el món de la informàtica normalment aquesta està compresa entre 0 i 256 (8 bits); com major és el valor més intervé el color en la mescla. Aquest model és el que es sol utilitzar en les pantalles dels ordinadors, les càmeres fotogràfiques digitals senzilles, els televisors, etc.



14. Representació de l'espai RGB en un cub

El model RGB es pot representar mitjançant un cub on cada un dels punts de la superfície o l'interior d'aquest equival a un color:

El principal problema d'aquest model de representació és que no és lineal, és a dir, la distància entre dos colors diferents no és equivalent a la diferència que en percep l'ull humà. En el cub es pot observar com la distància entre els colors blau clar i verd és la mateixa que hi ha entre el groc i el vermell, mentre que per a l'ull humà el color verd i blau cel s'assemblen molt més que el groc i el vermell. Altres models de color, com l'HSV, redueixen parcialment aquest problema aconseguint una representació més pròxima a la de l'ull humà.

3.2.2 El model HSV

Les sigles H.S.V signifiquen Hue,Saturation i Value. En aquest model en lloc d'utilitzar-se els tres colors primaris per descriure un color s'utilitza el to(Hue), la saturació (Saturation) i el valor (Value).

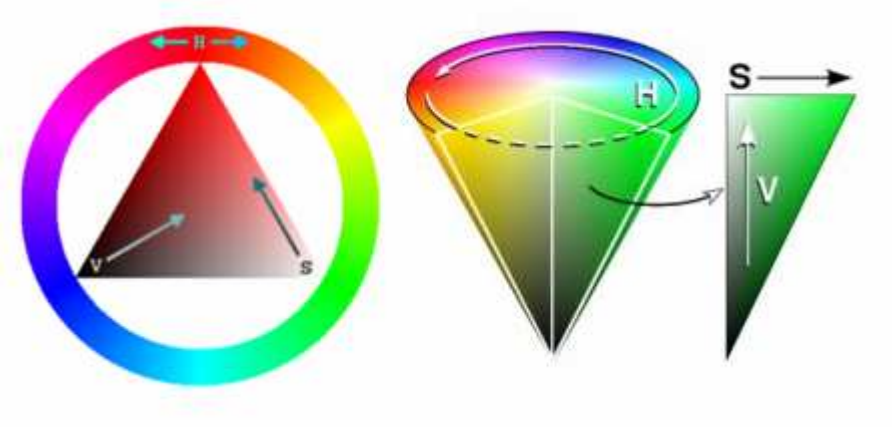
El to és el tipus de color amb el que s'està treballant, es sol representar amb un angle de 360 graus tot i que en algunes aplicacions s'utilitzen valors entre 0 i 100.

La saturació és el nivell d'intensitat amb el que apareix el to dins el punt. Es representa mitjançant una escala entre 0 i 100, si el valor de la saturació és molt baix el color adquireix un to grisos mentre que si és alt el color és molt més viu. En ocasions la

saturació també s'anomena puresa.

El valor és el nivell de "Brillantor" del color, també es representa mitjançant una escala entre 0 i 100. Com menor és el valor més fosc és el color.

A diferència del model RGB el model HSV es pot representar tridimensionalment mitjançant un con ja que els colors s'identifiquen per 2 punts (els eixos de coordenades i absises) i un angle.

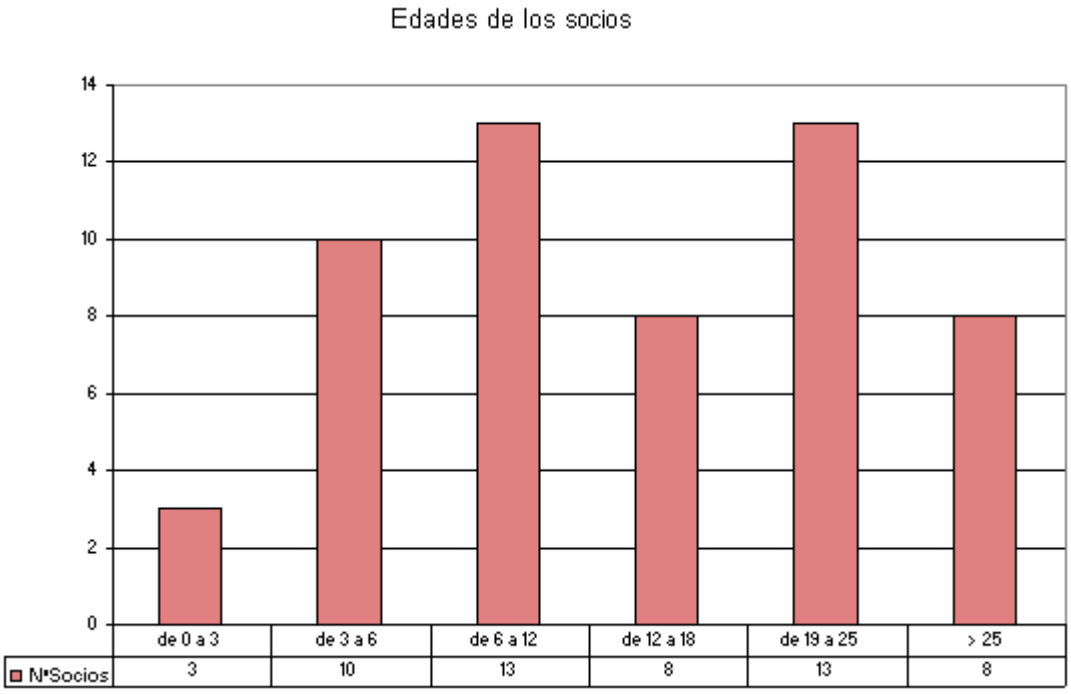


15. Representacions del model de color HSV

Amb aquest model de representació les distàncies entre els diferents punts del con s'aproximen més a la representació que en fa l'ull humà. Això fa que ens sigui molt més útil a l'hora de comparar diferents imatges que el model RGB.

3.3 Histograma HSV

Podem definir un histograma com la representació de la freqüència de repetició dels valors d'una variable. S'acostuma a utilitzar per representar variables que contenen diferents valors escalars que es repeteixen com podrien ser les notes d'una classe o les edats d'una població.



16. Exemple d'histograma dels socis d'un club

La representació més freqüent d'un histograma és mitjançant un diagrama de barres, no obstant en ocasions també es mostra mitjançant una taula on cada casella correspon a un o més valors i el contingut de la cel·la és el nombre de vegades que es repeteix.

Gràcies al model HSV podem representar una imatge com un conjunt de valors enters, això fa que puguem representar les imatges a través d'un histograma i comparar-les entre elles. Podem crear tres sub-histogrames diferents, un per cada component del model (Intensitat, Saturació i Valor), i posteriorment emmagatzemar-los en una sola taula. Per determinar la semblança o la diferència entre dos imatges diferents podem utilitzar la distància euclidiana[eq.1], aquesta funció matemàtica retorna un valor enter que representa la diferència entre dos vectors, en el nostre cas dos histogrames: com més baix és aquest valor més s'assemblen les dos taules.

$$\sum_0^i (a(i)-b(i))^2$$

Equació1: Distància euclidiana

En aquest projecte els histogrames HSV utilitzaven 64 valors per les components d'intensitat i saturació i 32 per el Valor. En total cada un d'ells constava de 160 binds.

3. 4 Separació en shots

Quant es produeix un canvi de càmera o d'escena en un vídeo els histogrames dels dos frames que marquen el canvi acostumen a ser molt diferents ja que en els canvis d'escena els colors solen realitzar un canvi molt accentuat mentre que quan canvia la càmera el que varia és la intensitat i la saturació d'aquests degut, principalment, als canvis d'il·luminació. Mitjançant un llindar, un valor que determini a partir de quin punt una diferència és gran o petita, podem separar un vídeo en shots.

Per tal de poder separar les imatges de les que disposem en shots vàrem crear un script per a matlab en el qual cal introduir la carpeta en la que s'han emmagatzemat les imatges i el llindar que separa un shot de l'altre. L'script realitza l'histograma HSV de totes les imatges i després els compara, un per un, amb l'histograma del frame que el succeeix mitjançant la distància euclidiana. Si el valor de la distància eucleriana és superior a l'indicat per el llindar el programa considera que hi ha un canvi de shot.

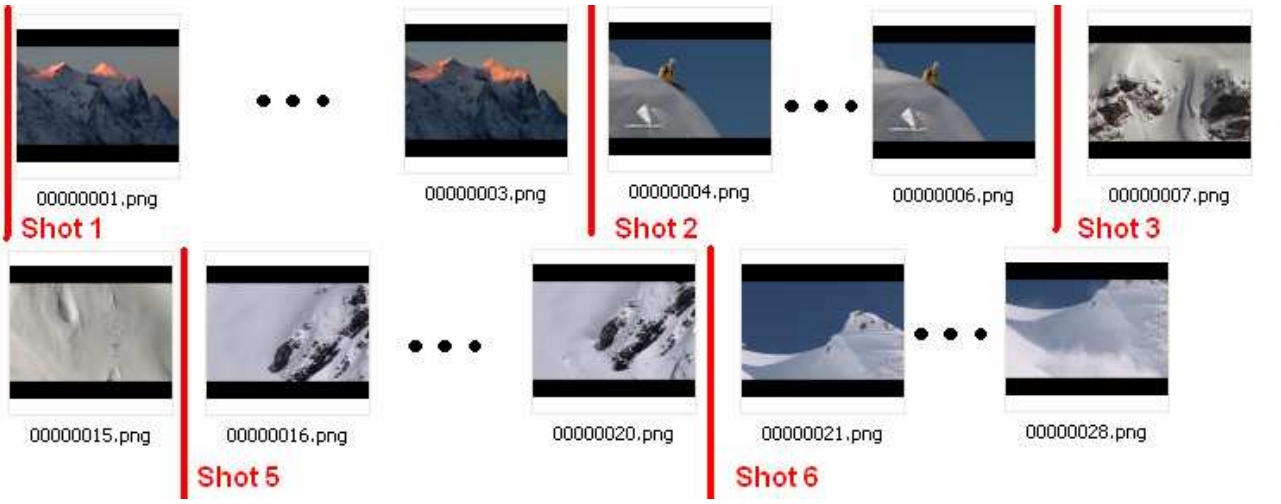
L'script retorna un fitxer ASCII amb diferents valors positius i negatius. Els nombres negatius són les etiquetes que identifiquen l'inici d'un shot (-1 identificarà el primer shot, -2 el segon, -3 el següent i així successivament); els nombres positius fan referència als diferents frames del vídeo. D'aquesta manera podriem dir que si obtenim la següent sortida:

-1 1 2 3 4 5 6 -2 7 8 9 10 11 12 13 -3 14 15 16 -4 17 18 19 20 21...

En el primer shot hi haurà els 6 primers frames, en el segon els frames que van del 7 al 13, al tercer els frames 14, 15 i 16, etc.

Després de realitzar diferents proves amb varis llindars vàrem concloure que el que millor resultats retornava era 600000. A continuació es poden observar exemples d'alguns shots obtinguts.

17. Primers shots del vídeo Btt1.avi



18. Primers shots del vídeo Snow1.avi

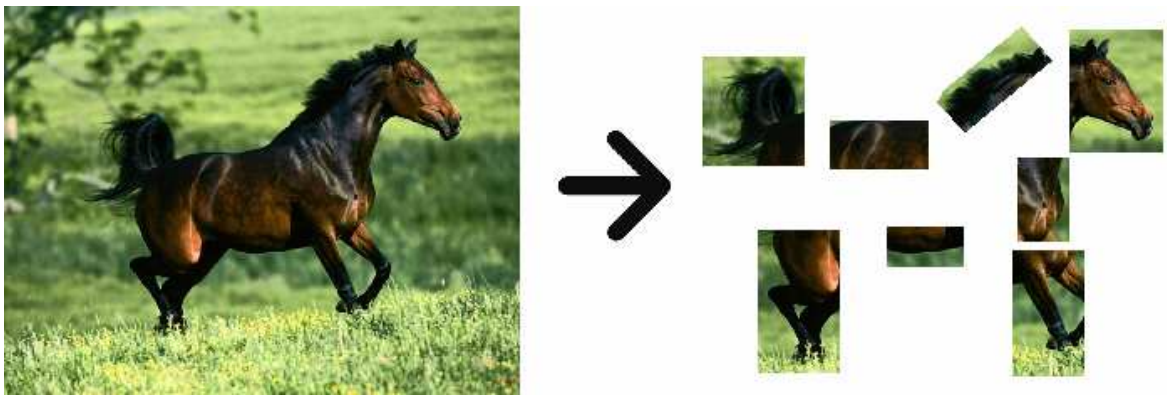
19. Primers shots del vídeo Rally1.avi

4. Vocabulari Visual

4.1 Vocabulari Visual

Per tal de determinar si dos documents tracten un tema semblant o relacionat, els classificadors i cercadors de texts analitzen si apareixen paraules semblants en ells. Creen un vocabulari per tal de decidir si dos paraules tenen un significat semblant, un exemple serien els temps verbals. Totes les conjugacions d'un verb es refereixen a la mateixa acció però són paraules diferents: tant cantaré, com cantàveu, com cantaran fan referència a l'acció de cantar per tant podrien ser agrupades sota un mateix significat representat per l'infinitiu del verb. També podríem dir, per exemple, que tant cançó, com melodia, com ritme estan representades per "música"; o que maco, bonic, bell i preciós es troben dins del mateix grup. D'aquesta manera un classificador de text és capaç d'entendre que la frase "Ell canta una cançó preciosa" és semblant a "Jo cantava una bella melodia".

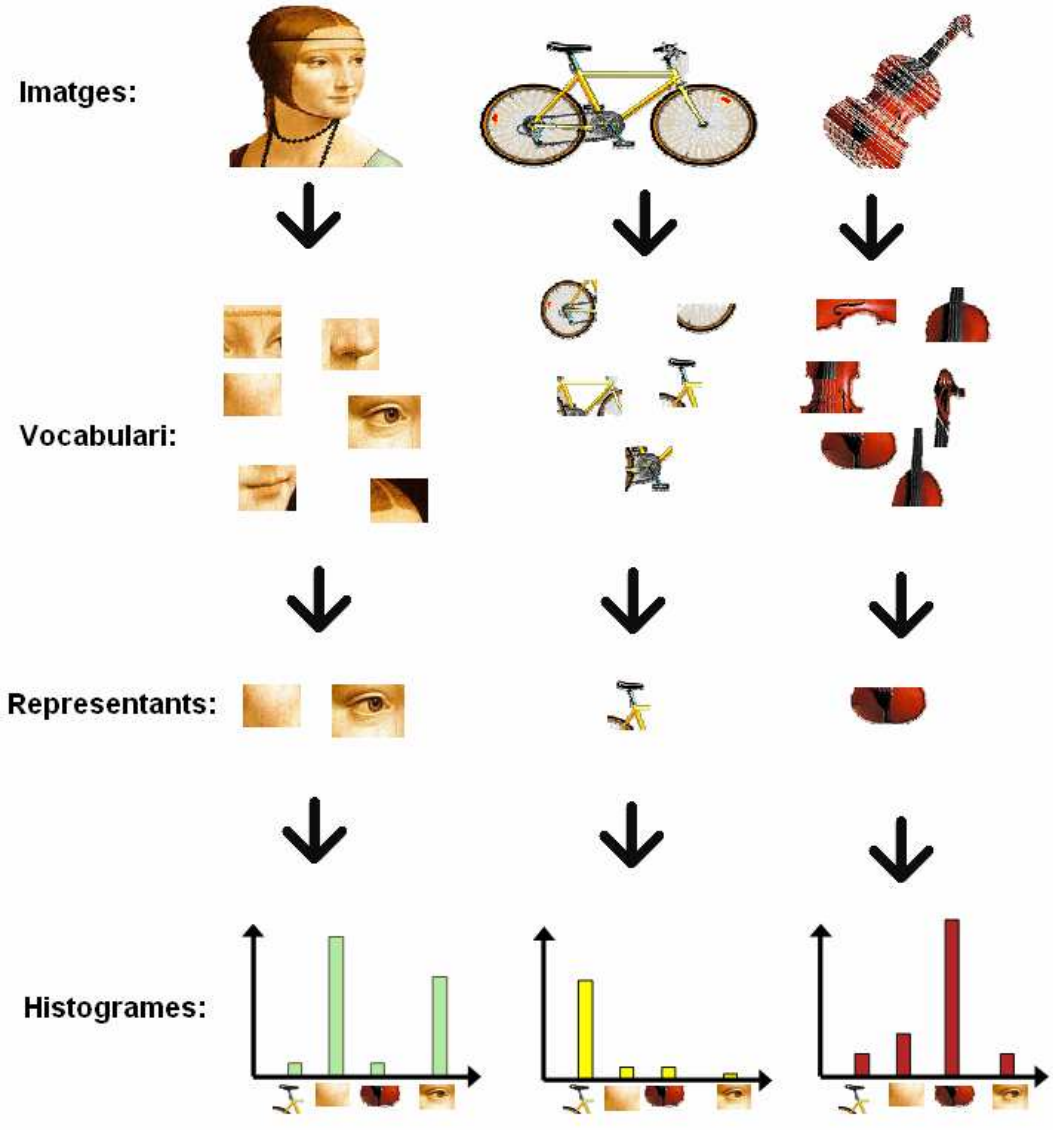
En aquest projecte la idea és exactament la mateixa però es canvien els conjunts lèxics per fragments d'imatges. Així, doncs, intentarem trobar un representant de diferents trossos d'imatges que representin el mateix objecte o un de semblant: un representant per tots els tipus de roda que apareixen en una cursa de cotxes, un exemple de l'herba que hi ha en un camp de futbol o un anella d'una cistella de bàsquet que identifiqui a les demés anelles. Crear un vocabulari que, en lloc de estar format per paraules i lletres, ho estigui per fragments d'imatges i les seves característiques: un vocabulari visual.



20. Possible vocabulari d'una imatge on aparagués un cavall

21. Possible Vocabulari d'una imatge on aparegui un cotxe

Tal i com es pot veure en la següent figura, comptabilitzant el nombre de vegades que apareix una paraula del vocabulari en un imatge i guardant-ho en un histograma es pot deduir a quina categoria pertany una imatge..



22. Procés que segueix la imatge fins elaborar-ne el seu histograma

Per tal de crear un vocabulari el primer que necessitàvem era trobar les paraules que hi havia als frames que havíem extret del vídeo. Per obtenir els millors resultats possibles vàrem decidir seguir dos estratègies diferents. La primera buscar les paraules a través d'un detector de regions i posteriorment descriure-les amb un descriptor d'imatges SIFT. La segona era semblant a la primera, però en lloc de trobar les paraules mitjançant un detector de regions ho vàrem fer mitjançant una graella regular: comprovant sempre les mateixes zones d'una imatge.

23. Exemples de diferents mètodes de detecció de regions

Per crear el vocabulari de les paraules del conjunt d'imatges de mostra vàrem utilitzar l'algorisme de clusterització anomenat K-means. Amb el vocabulari resultant de l'agrupació vàrem realitzar els histogrames que ens vàren servir per fer la descripció final de la imatge.

4.2 Detectors de regions i Descriptors d'imatges

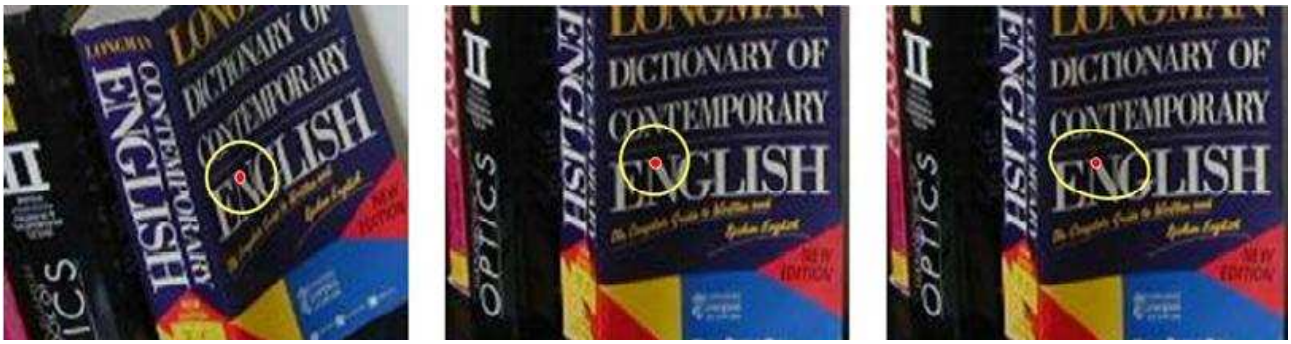
Per tal de poder identificar i definir els diferents tipus d'imatges que hem obtingut anteriorment necessitem una eina capaç d'extraure'n característiques i informació que sigui fàcilment reconeixible en altres imatges. Una eina que pugui, per exemple, capaç d'obtenir la descripció de l'anella d'una cistella de bàsquet i mantenir-la inalterable independent de la càmera que l'enfoqui. Per a això utilitzarem algorismes de detecció de regions i descriptors d'imatges. Els primers són capaços d'identificar possibles punts d'interès que poden resultar útils per descriure una imatge mentre que els segons intenten obtenir informació sobre aquests de tal manera que no depengui de la posició i de la lluminositat. Per exemple, obtenir la descripció de la roda d'un cotxe independentment de l'angle amb què es miri.

4.3 Detectors de regions

Els detectors de regions, que es van començar a utilitzar i desenvolupar a mitjans de la dècada dels 90 [2], s'han utilitzat en diverses aplicacions i treballs com ara la reconstrucció en 3 dimensions d'imatges, la separació de shots, la localització i orientació de robots, reconeixement de textures, reconstrucció d'imatges panoràmiques, etc.

El tret comú en totes aquestes aplicacions és que treballen amb diferents grups d'imatges i que totes les imatges tenen, o haurien de tenir, punts semblants amb la seva predecessora. La forma i la orientació d'aquests punts no té perquè ser la mateixa però l'objecte al que pertanyen sí; per exemple en una seqüència en la que un cotxe traça una corba les òptiques canviaran de posició, d'orientació i aparentment de forma però continuaran sent els mateixos fars del mateix cotxe: l'algorisme ha de ser capaç de detectar el mateix punt després d'un canvi de vista. Aquests punts o zones es coneixen com a regions invariants o regions covariants afins. A diferència dels algorismes de segmentació, en aquest cas quant parlem de "regions" ens referim a un conjunt de píxels que no tenen perquè complir cap condició de canvi respecte el seu entorn.

Els detectors de regions retornen 5 valors per cada punt rellevant trobat[2]. Els dos primers equivalen a les coordenades que ens indiquen la situació del punt, mentre que els tres últims ens defineixen l'el·lipse que determina la regió. S'utilitzen el·lipses per definir la regió perquè es podrien considerar com la deformació d'un cercle, la regió en una imatge inicial està definida com un cercle però quan aquesta canvia de posició o se'n deforma la superfície un altre cercle no pot indicar la mateixa zona mentre que una el·lipse si pot:



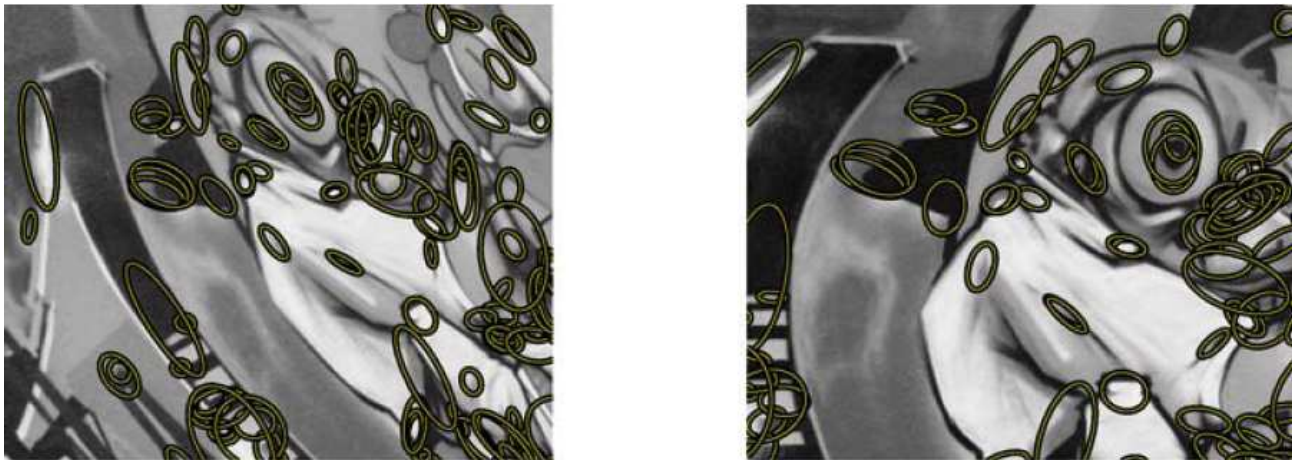
24. En aquesta imatge es pot observar com un cercle no pot utilitzar-se per senyalitzar una determinada zona un cop s'ha deformat mentre que una el·lipse sí.

No obstant amb la detecció de regions per ella mateixa és suficient, en una imatge qualsevol el més probable és que s'obtinguin centenars o milers de regions diferents solapant-se entre elles i complicant moltíssim la comparació entre dues imatges. És per això que són necessaris els descriptors d'imatge, per definir unes característiques de la

regió que romanguin inalterables i facilitin el reconeixement d'aquesta en una altre imatge.

Existeixen diferents mètodes per determinar les regions d'una imatge: els detectors "harris affine" i "Hessian-Affine", els MSER (Maximally Stable Extremal Region), els basats en contorns, basats en intensitats, etc. Per a estudiar les imatges de les que disposàvem els meus tutors del projecte em varen aconsellar utilitzar els MSER i els de "Harris-Affine" ja que eren els que millor resultat ens proporcionarien. Aquesta elecció va estar condicionada al fet que aquests dos detectors es podrien considerar complementaris o compatibles: mentre que el de Harris-Affine es basa en interseccions de rectes i contorns per trobar les regions, l'MSER ho fa amb zones on la intensitat es manté relativament constant (blobs).

4.3.1 Harris Affine

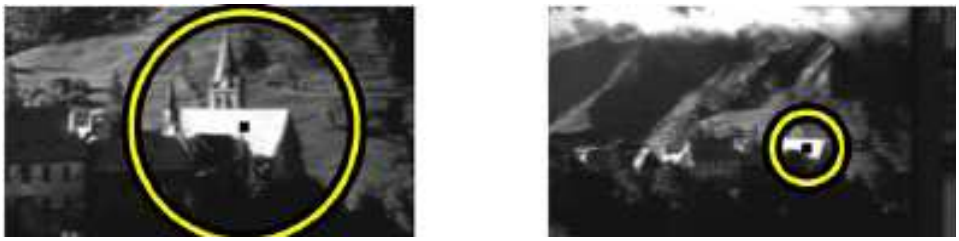


25. Regions d'una imatge detectades mitjançant Harris-Affine

El sistema de detecció de regions "Harris Affine" [2], es basa en la obtenció de contorns i punts de creuament per detectar els punts d'interès i en el gradient per determinar el tamany de l'el·lipse.

Si s'aplica la derivada en una imatge se n'obtenen els canvis de contrast, és a dir, els seus contorns. Aquests contorns estan determinats per dos píxels ja que amb la primera derivada es detecten dos cops. Per exemple si tenim una imatge que en un punt k canvia de blanc a negre, el filtre de la primera derivada obtindrà tant el canvi de blanc a negre com de negre a blanc. Per afinar més aquest contorn existeix el que es coneix com a Laplacià [3] o segona derivada, que és molt més sensible als canvis de contrast. Aplicant la segona derivada sobre una imatge obtenim més contorns i més ben definits. Cada cop que dos o més línees de contorn es creuen el detector les identifica com un

punt d'interès, d'aquesta manera s'intenta evitar que els punts siguin sensibles a la orientació de la imatge.

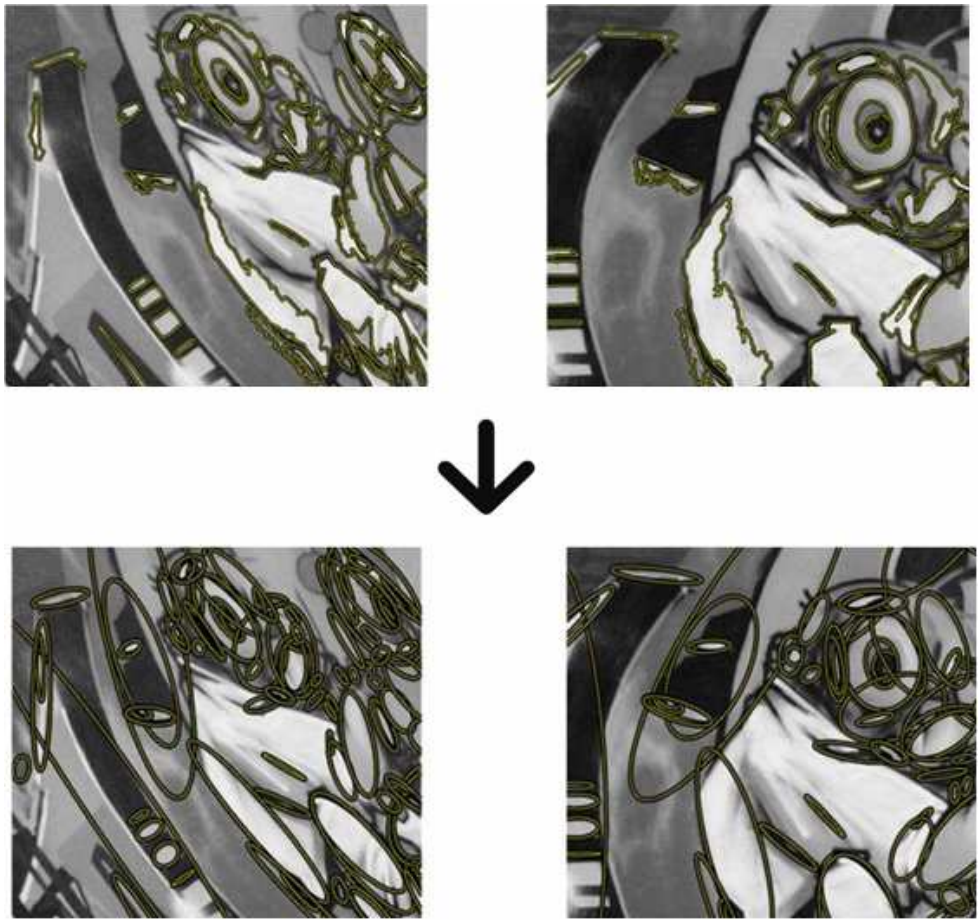


26. El detector de Harris - Affine no depèn de l'escala de la imatge

Per determinar l'àrea de l'el·lipse que definirà la regió s'utilitza el gradient, o primera derivada, de la intensitat de la imatge. Com hem dit en el paràgraf anterior, la derivada s'utilitza per detectar canvis de contorn però un canvi de contrast petit no queda pràcticament reflexat en la sortida d'aquesta, en canvi quan el canvi és considerable hi queda marcat de forma clara. Mitjançant un llindar i el contrast obtingut mitjançant la primera derivada el detector de Harris determina on acaba la regió d'un punt. Un cop ha obtingut el centre de la regió i l'àrea aproximada d'aquesta genera l'el·lipse que millor l'encercla. D'aquesta manera es pretén aconseguir que els canvi d'escala no afectin la detecció de regions.

4.3.2 MSER

Les sigles MSER[2] signifiquen Maximally Stable Extremal Region, (Màxima regió extrema estable). En aquest cas la paraula “extrema” es refereix a que tots els píxels d’una regió tenen un valor d’intensitat més alt (més brillants) o més baix (més foscs) que tots els del seu voltant. Aquest algorisme primerament utilitza un mecanisme molt semblant al de l’algorisme de segmentació “region – growing”,enumera tots els píxels i els enumera segons la seva intensitat, posteriorment comprova quina és la intensitat dels seus veïns i, en cas de ser la mateixa, els tracta com sí fossin un sol píxel.



27. El detector de regions MSER primer cerca regions estables i posteriorment les engloba dins el·lipses

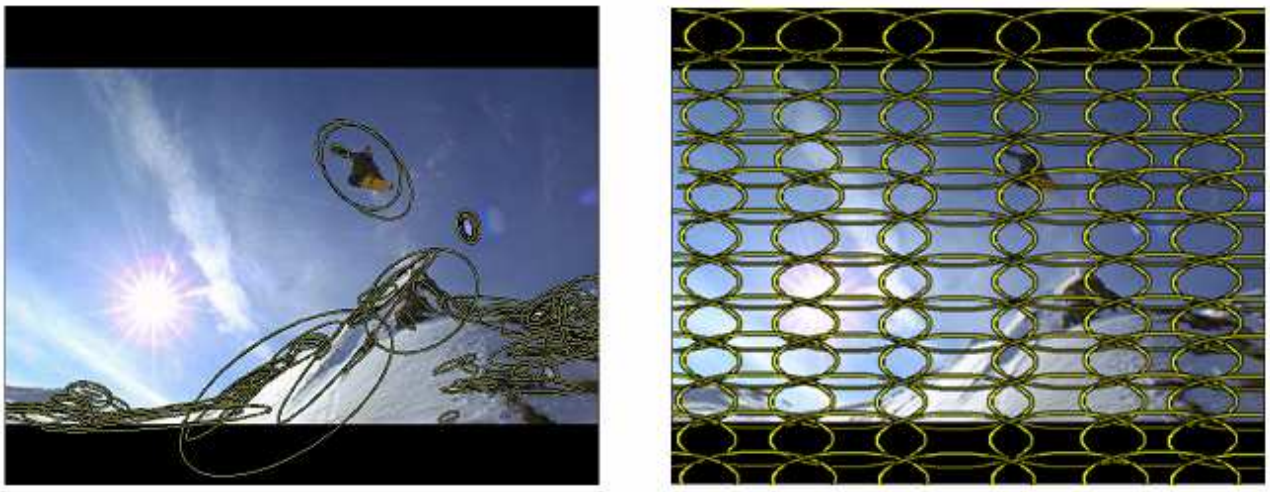
Un dels avantatges d’aquest sistema és que es pot conèixer prèviament el nombre màxim i mínim de regions que es trobaran ja que, en els casos més extrems, es consideraria cada píxel com una regió diferent o tots els píxels com una de sola. Un altre aspecte positiu d’aquest algorisme és la seva poca sensibilitat a les deformacions i als canvis d’orientació ja que contempla el nivell d’intensitat dels píxels i aquests no canvien amb la orientació. A més a més és un dels mètodes més ràpids i que menys potència del processador necessiten.

Com a contrapunt cal ressaltar que els canvis d'il·luminació d'una imatge poden afectar la detecció de les regions ja que, per exemple, l'ombra d'un objecte farà variar la intensitat de la superfície on està projectada. Un altre aspecte a tenir en compte, tot i que aquest no és forçosament negatiu, és que el nombre de regions que s'identifiquen en una imatge és notablement inferior al de altres detectors.

4.3.3 Regular Grids

Un altre sistema que s'utilitza per a determinar regions és el "Regular Grid" o graella regular. Aquest mètode no es pot considerar com un detector de Regions ja que, a diferència dels detectors Harris-affine o MSER, no selecciona els punts d'interès de forma dinàmica segons les característiques de la imatge si no que ho fa de forma estàtica: Cada x píxels es genera una nova el·lipse. Podríem dir que buscar regions d'una imatge emprant un Regular Grid consisteix en aplicar una graella.

Les regions resultants queden definides per 5 valors que representen una el·lipse, els dos primers marquen el centre d'aquesta i els altres tres el tamany i la orientació. Abans d'executar una regular grid cal determinar el tamany de l'el·lipse que marcarà la regió d'interès i cada quants píxels s'executarà. En la següent figura es pot comparar com els detectors de regions no proporcionen informació sobre la totalitat de la imatge, per exemple podem veure que amb el detector MSER no s'obté cap regió que descrigui el cel mentre que utilitzant un Regular Grid sí.



28 Regions trobades amb MSER i possibles regions trobades amb un Regular Grid

4.4 Descriptors d'imatge

En l'anterior operació hem aconseguit detectar regions d'interès d'una imatge, ara el que necessitem és un mètode que ens permeti descriure aquests punts per què ens

siguin útils per identificar-la o obtenir imatges semblants. Com hem comentat anteriorment els mètodes de detecció de regions són sensibles als canvis de vista, a les deformacions i a la brillantor de la imatge,etc. Per això necessitem una eina que ens permeti descriure aquests punts de tal manera que no els afectin aquests paràmetres, aquí és on intervenen els descriptors d'imatges. Molts dels descriptors d'imatges més utilitzats es solen basar en els diferents gradients que hi ha en una determinada regió de la imatge, per tant primer de tot cal comprendre què és un gradient.

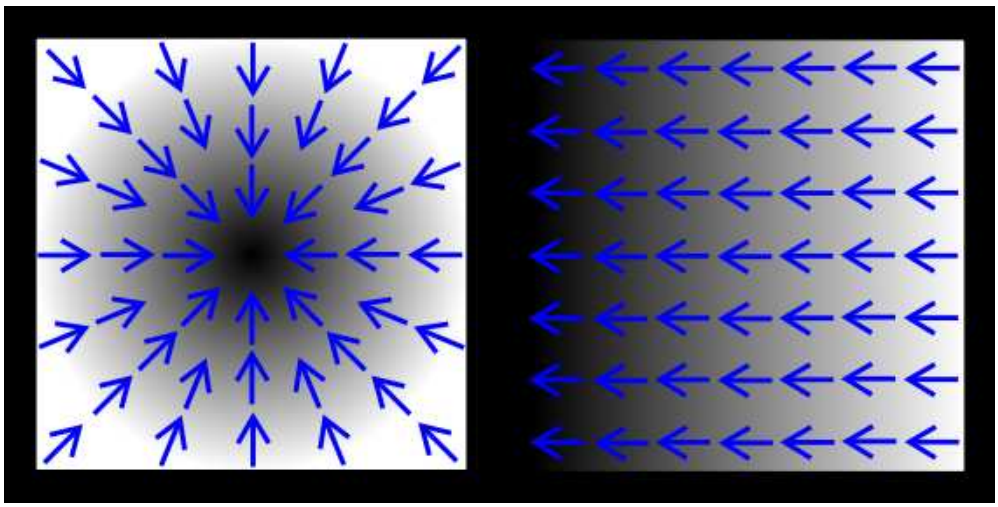
4.4.1 Gradient:

Tècnicament podem definir un gradient d'un punt en un camp escalar com l'únic vector que ens permet trobar la derivada direccional en qualsevol direcció com en l'equació 2:

$$\frac{\partial \phi}{\partial n} = (\mathrm{grad}\phi) \cdot \hat{\mathrm{n}}$$

Equació 2: Gradient

On n és un vector unitari. I d(alfa)/d(n) la derivada direccional de l'angle en la direcció del vector unitari que informa de la variació de la magnitud escalar segons ens movem en la direcció del vector[4]. En altres paraules, un gradient ens indica, des d'un punt, cap a quina direcció els valors escalars canvien d'una forma més ràpida. Per exemple, si disposéssim d'una làmina de ferro i n'escalféssim una part el gradient de qualsevol punt de la làmina ens indicaria la direcció en que el metall canvia de temperatura de forma més sobtada. En les següents imatges s'observa com els gradients dels diferents punts es posicionen en la direcció on el canvi és més brusc (considerant el color negre com la magnitud més alta i el blanc com la més baixa).



29 Representació gràfica dels gradients d'intensitat

En una imatge els gradients poden seguir els canvis en la tonalitat, la intensitat, la saturació, les components de color, etc.

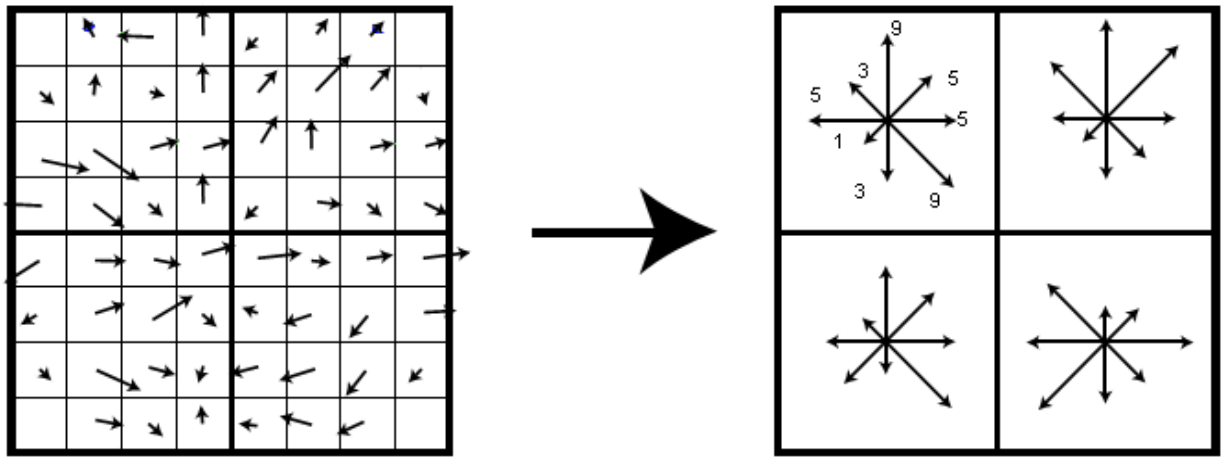
30. Gradients d'una imatge representada en Blanc i negre

4.4.2 Descrivint la imatge mitjançant SIFTS

Per descriure les regions obtingudes mitjançant el Regular Grid o MSER utitlizarem el descriptor d'imatges que es coneix com a SIFT: Scaled-Invariant Features Transformer (Transformador de característiques escalarment estables). El primer pas que realitza el descriptor per descriure la imatge és buscar els gradients de tots els punts que es troben dins la regió determinada per el punt d'interès i la el·lipse. Mitjançant un filtre gaussià [5] s'atorguen pesos als diferents punts que es troben dins del descriptor per tal de no donar una excessiva importància als punts que es troben allunyats del punt d'interès ja que els punts més exteriors solen estar més afectats per el soroll. Aquests peso s'assignen mitjançant una fórmula molt senzilla:

$$Pes = 1 - d \qquad \text{on } d = \text{distància del punt d'interès}$$

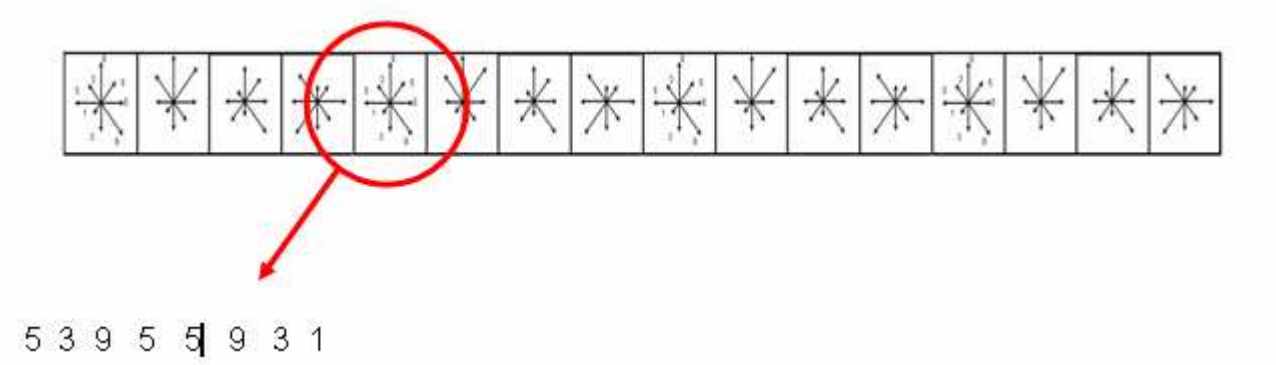
Els mòdul dels diferents gradients és multiplicat pel pes que se li ha donat. A continuació el descriptor es divideix en k zones d'igual tamany i es realitza un histograma amb el valor de cada una de les zones, aquest histograma es pot realitzar amb un nombre diferent de components però el més freqüent és utilitzar-ne 8. Amb aquest pas es passa de tenir una matriu amb moltes cel·les a tenir una matriu de k/2 x k/2 on cada una de les cel·les conté un vector amb l'histograma de la regió.



31. El conjunt de gradients es divideixen en k grans zones de les que es realitzen histogrames

En la figura anterior es pot observar com una matriu de 64 cel·les (8x8) es divideix en 4 zones i en resulta una matriu de 2x2 amb histogrames de 8 components.

El més comú és dividir la matriu inicial amb 16 zones (una matriu de 4x4) i utilitzar histogrames de 8 components, el resultat d'això és obtenir una matriu de 4x4x8 o el que és el mateix, un vector de 128 posicions.

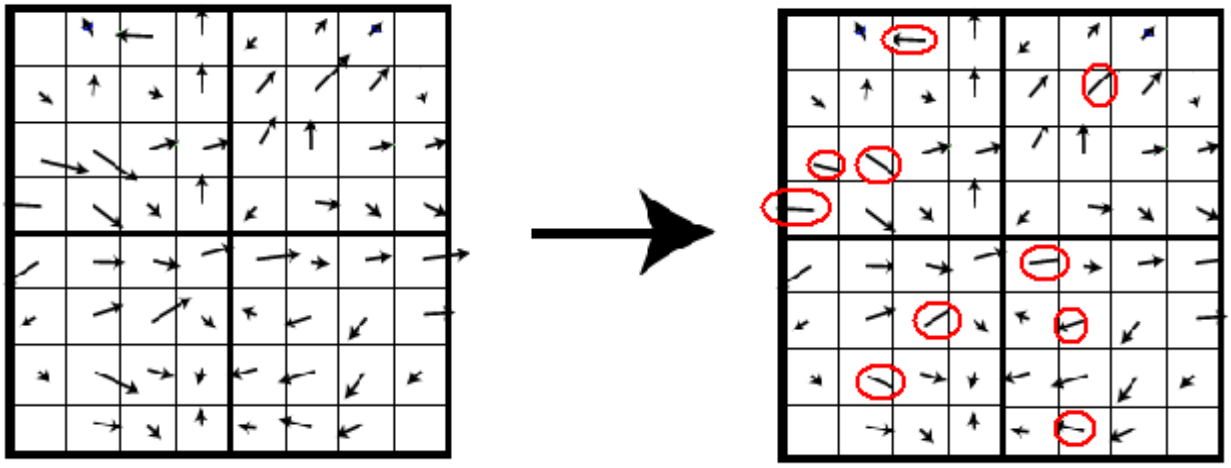


32. Detall del vector resultant

Seguidament el vector es normalitza a una unitat per tal de poder-lo comparar amb els demés vectors. Arribats en aquest punt podem determinar que el vector de descripció no és sensible als canvis homogenis d'orientació , d'il·luminació, i de contrast. Els problemes d'orientació han desaparegut prèviament, quan hem detectat les regions, al determinar-les per el·lipses. En cas de que la il·luminació canviï homogèniament a tota la regió no afectarà el valor dels gradients ja que la intensitat incrementarà d'una manera lineal en tots els píxels, per tant els gradients no canviaran d'orientació. En un canvi de contrast homogeni el valor de tots els píxels serà multiplicat per un valor constant per tant l'increment del mòdul dels gradients serà lineal i no afectarà en la creació de l'histograma. No obstant els canvis en la il·luminació no uniformes o sobre superfícies en 3 dimensions

continuen sent un problema.

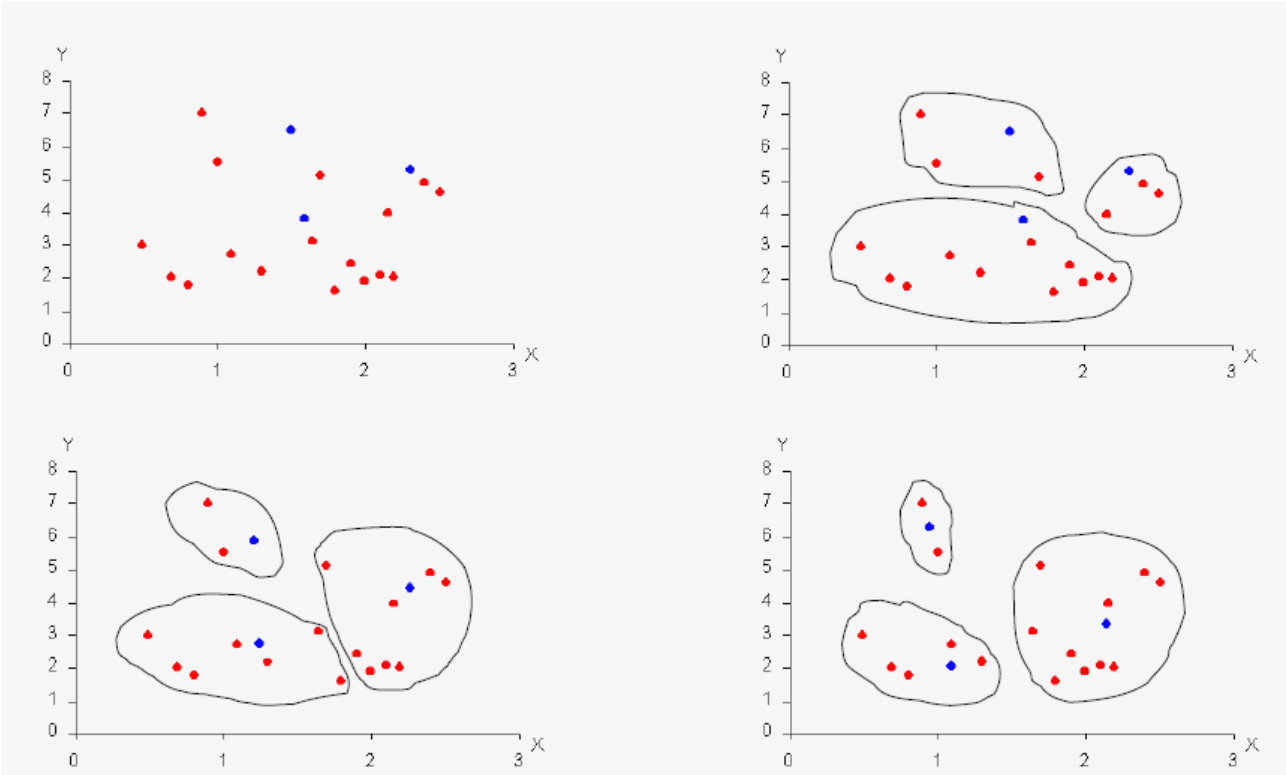
Els canvis d'il·luminació no lineals solen produir alteracions en el mòdul dels gradients però rarament en varien l'angle. Per tant, la millor manera de reduir la influència d'aquests canvis és restar importància a la llargada del mòdul del gradient. Per tal de donar molta més importància a la orientació del gradient per sobre del seu mòdul es va decidir establir un llindar que determinés a partir de quina llargada el gradient no aportava nova informació. Després de realitzar varis estudis sobre diferents imatges amb diferents il·luminacions i figures es va decidir que els mòduls superiors a 0.2 [6] no aportaven nova informació i per tant es podien reduir a aquest valor.



33. El módul dels gradients es redueix per disminuir la seva influència sobre el resultat

4.4.3 L'algorisme K-means

L'algorisme k-means[7][8] és un algorisme seqüencial que intenta dividir un conjunt de dades en K diferents grups d'informació. Intenta trobar un determinat nombre de nuclis representatius de les dades que tenen al seu voltant. Inicialment l'algorisme escull k punts a l'atzar i crea un grup amb els punts que estan a una menor distància dels punts escollits. Un cop els grups s'han creat es calcula quin és el punt més cèntric del grup, quin està a menor distància de tots el demés; quan s'han determinat els nous centres es realitza, de nou, la operació d'agrupar-hi els que estan més a prop. Aquest seguit d'iteracions es va realitzant fins que el grup no varia, moment en el qual s'han trobat els grups ideals.



34. Exemple de l'evolució de 3 clústers en l'algorisme de k-means

Per esquematitzar l'algorisme el podem dividir en 5 fases o etapes:

1. Donat un nombre K, elegir K nombre de centroides aleatòriament o mitjançant algorismes com el de Lloyd*.
2. Assignem cada punt al centroide més proper, estem creant un cluster.
3. Calculem els nous centroides. Busquem els centres dels clústers.
4. Assignem els punts als nous centroides.
5. Si algun dels punts ha canviat de grup tornem al punt 3, altrament la agrupació ha finalitzat.

*L'algorisme de Lloyd[9] ajuda a trobar diversos centroides que facilitin la creació de clústers en el k-means i reduir el nombre d'iteracions.

4.5 Anàlisis de la detecció de regions:

Per seguir endavant amb el projecte era necessari comprovar que els detectors de regions que havíem escollit ens proporcionaven uns bons resultats. Calia, doncs, analitzar quin era el comportament dels mètodes anteriors si els aplicàvem sobre les imatges que havíem obtingut dels Vídeos Font i determinar si s'havia de prescindir d'algun d'ells.

4.5.1 Experiment1

En aquesta primera prova vàrem aplicar el detector de regions Harris-affine sobre tots els frames que havíem extret dels diferents DVDs. Un cop extretes les regions de les diferents imatges de mostra en vàrem escollir, aleatòriament, 10 de cada Vídeo Font per comprovar quins havien estat els resultats ja que comprovar una per una les 1800 imatges de les que disposàvem ens va semblar una feina molt feixuga i innecessària. A continuació es mostren, juntament amb una breu ressenya, les imatges que vàrem utilitzar per determinar si aquest detector ens podria ser útil. Les paraules o regions obtingudes estan encerclades mitjançant una el·lipse de color groc. Els diferents conjunts de frames es troben classificades segons la categoria a la que pertanyen i el Vídeo Font del que procedeixen:

Automobilisme: Circuit

35. Exemples de regions detectades en la cateogira Circuit mitjançant Harris-Affine

Observant les diferents imatges de la figura 29 podem veure com amb el detector de Harris-Affine podem trobar regions que formen part de les diferents parts d'un cotxe (imatge central esquerra) , no obstant, en moltes imatges veiem que es troba un nombre molt baix de punts d'interès. Un tret característic que comparteixen gairebé totes les figures anteriors és l'absència de paraules que facin referència a l'entorn on es

desenvolupen les competicions: L'asfalt, la gespa, les línies que delimiten la carretera, etc.

Automobilisme: Rally

36**. Exemples de regions detectades en la cateogira Circuit mitjançant Harris-Affine**

Tal i com es pot comprovar en la figura 25, amb les imatges de la categoria Rally succeeix quelcom semblant que amb la categoria Circuit. S'obté informació sobre petits

detalls dels automòbils (òptiques, rodes, propaganda, vidres, etc.) però la informació sobre l'entorn on es desenvolupa l'acció és mínima. A part, en imatges fosques (Fig z.1 i z.8) el nombre de regions trobades es redueix considerablement. A més a més, podem notar que el nombre de regions trobades ha disminuït considerablement. Aquests fets ens fan dubtar de la possibilitat d'aconseguir diferenciar les categories Rally i Circuit utilitzant el detector de Harris-Affine ja que els cotxes que apareixen a les dos categories són molt semblants i no hem aconseguit obtenir informació de l'entorn. Això farà que el vocabulari trobat dins les dues tipologies sigui pràcticament idèntic i dificulti molt la seva distinció.

Bàsquet:

37**. Exemples de regions detectades en la cateogira Bàsquet mitjançant Harris-Affine**

En les imatges de la categoria Basquet s'ha trobat un nombre de detectors satisfactori. Es pot comprovar com s'obtenen punts d'interès de gairebé la totalitat de la imatge i com elements característics d'aquest esport com els jugadors, les cistelles o la pilota queden descrits per diferents regions.

Bicicleta tot terreny:

38. Exemples de regions detectades en la cateogira Btt mitjançant Harris-Affine

Tot i que en algunes imatges anteriors podem observar com el ciclista que participa en l'activitat queda totalment descrit per les regions obtingudes, en altres observem com els elements més importants de la imatge són ignorats. També podem veure com en alguns frames la quantitat de descriptors trobats ha estat gairebé nul·la. Malgrat tot l'anterior, el detall més destacable d'aquests resultats és l'absència de descriptors en la segona imatge de la figura. Això pot ser un gran inconvenient a l'hora de poder identificar aquesta categoria ja que molts campionats d'aquesta disciplina del ciclisme es disputen dins boscos on la il·luminació és escassa.

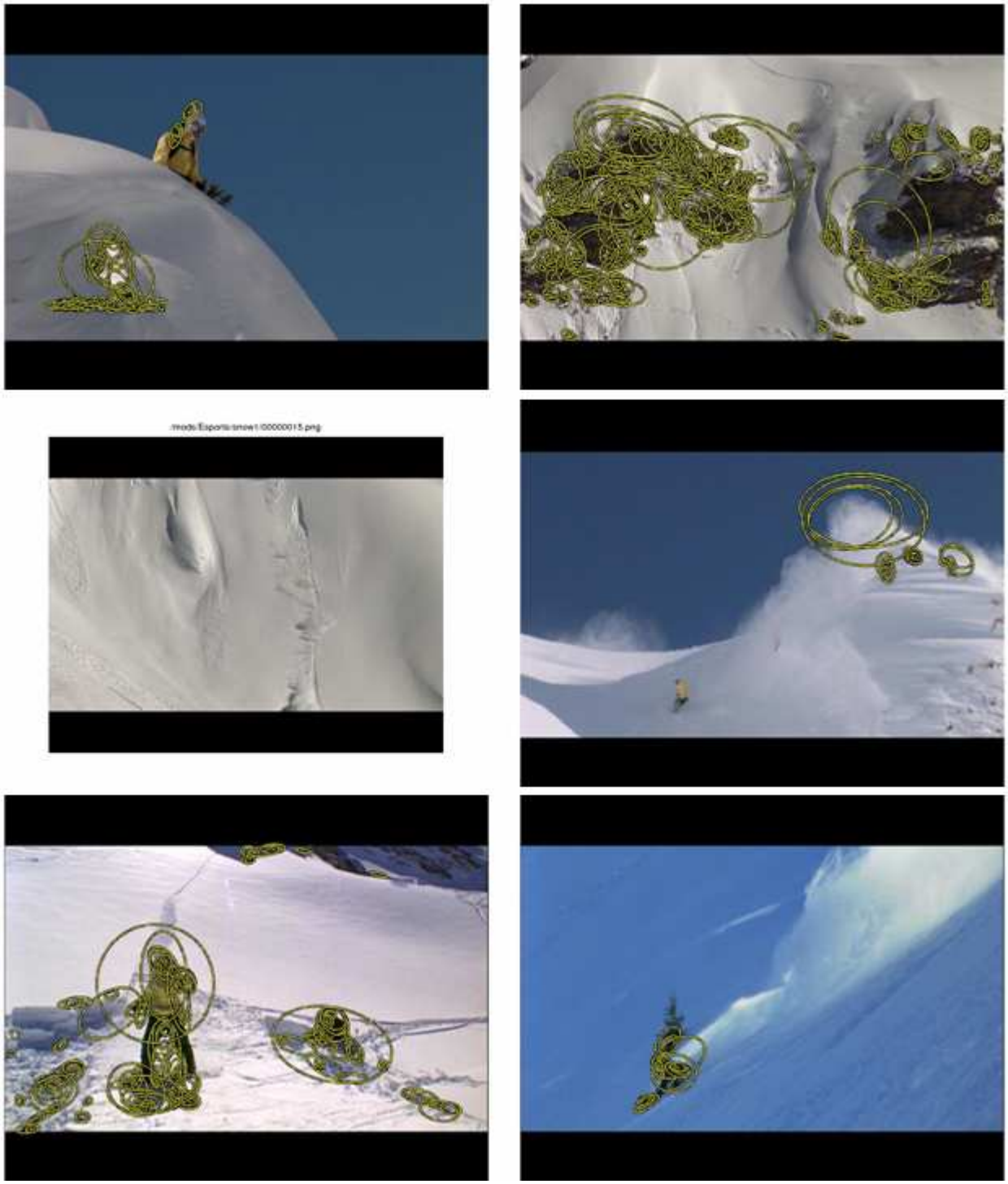
Futbol:

39. Exemples de regions detectades en la cateogira Futbol mitjançant Harris-Affine

En el conjunt de frames pertanyents al vídeo Futbol1.avi podem observar com en les imatges on s'enfoquen els jugadors de més a prop s'obté un gran nombre de regions que els descriuen, no obstant no passa el mateix amb els elements del terreny de joc. Aquest fet s'accentua quan es realitzen plans allunyats del que passa en el terreny de joc: no hi ha cap regió que defineixi alguna àrea de l'herba del camp de futbol i el nombre de punts

d’interès que descriuen els jugadors disminueix sensiblement. També podem observar com en alguns casos elements típics d’aquest esport com ara la pilota de futbol no són reconeguts com a paraula (Imatge inferior esquerra).

Snowboard



40. Exemples de regions detectades en la cateogira Futbol mitjançant Harris-Affine

El fet que l’Snowboard es practiqui en paisatges nevats on els colors predominants

siguin diferents tonalitats de blanc fa que el nombre de regions que es trobin en els vídeos d'aquesta categoria sigui bastant baix, arribant a no detectar cap regió en algunes imatges. Una constant en totes les imatges és que no apareix gairebé cap regió que aporti informació sobre la neu, l'element que més apareix en aquests vídeos. Podem observar, també, que els demés elements que apareixen en les imatges acostumen a tenir paraules que sí els descriuen.

4.5.2 Conclusions de l'experiment 1

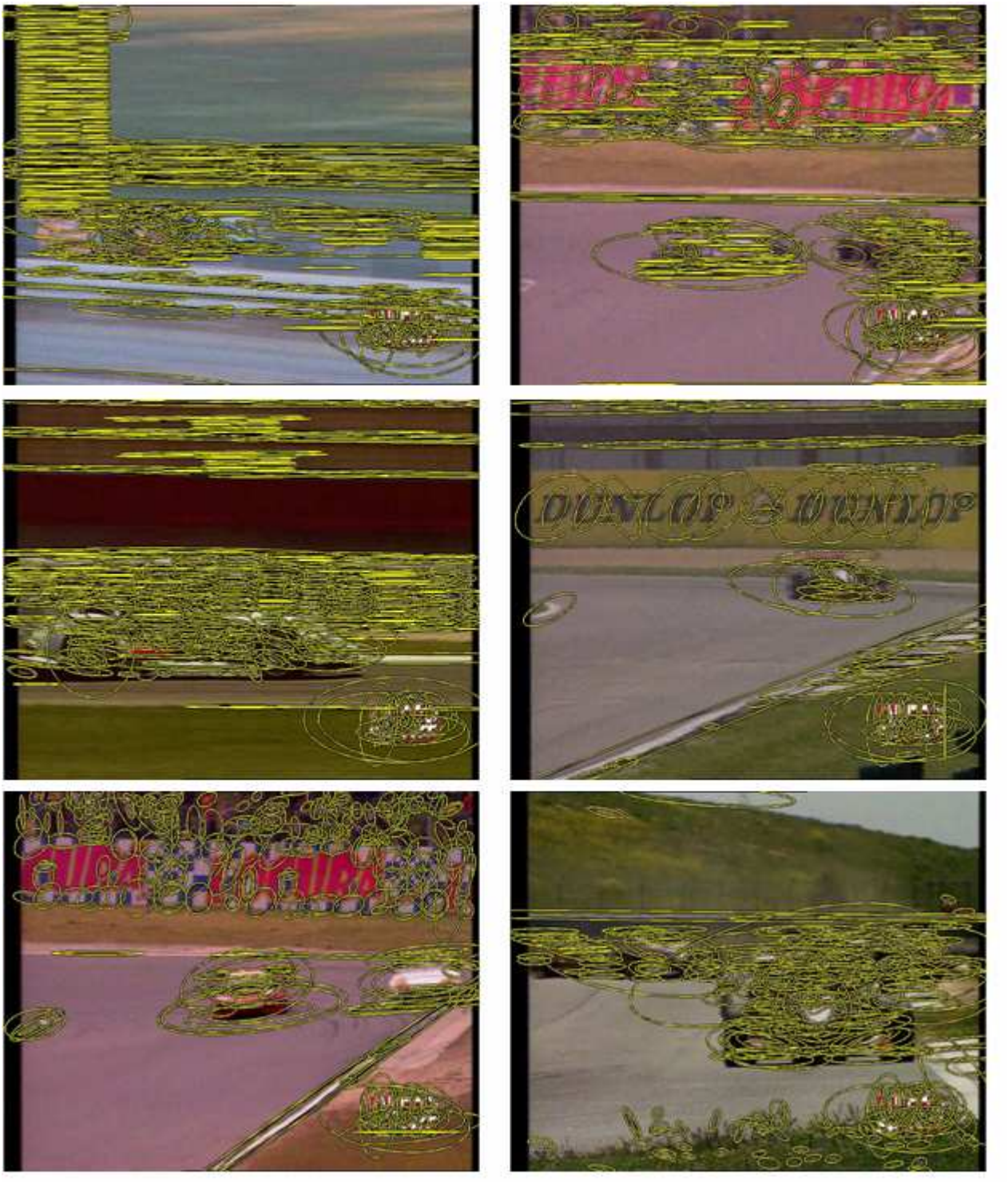
Després d'analitzar detalladament les imatges resultants d'aplicar el detector de regions Harris-Affine podem observar-ne les següents característiques:

- El detector de Harris-Affine detecta bastants punts d'interès en els principals objectes de la imatge (futbolistes, ciclistes, cotxes, etc.) però pràcticament no obté informació de l'entorn on es situen els elements.
- Quant la imatge és molt fosca o molt clara es detecta un nombre de regions molt baix. Té problemes per identificar diferents punts d'interès quan els canvis d'intensitat dels elements de la imatge son baixos (Per exemple quan un ciclista es troba dins un bosc poc il·luminat, o un surfista amb roba clara enmig de la neu).
- En algunes imatges no detecta cap regió. Això produirà errors en la descripció a través de SIFTS i dificultarà l'automatització del procés.
- Els detectors de regions de Harris-Affine han obtingut aproximadament una mitjana de 250 regions per imatge.

4.5.3 Experiment 2

L'experiment 2 seguia la mateixa dinàmica que el primer experiment, però en lloc d'utilitzar el detector de regions Harris-Affine utilitzava el MSER. Es varen analitzar totes les 1800 imatges de mostra però, tal i com es va fer anteriorment, només vàrem observar els resultats en 10 imatges de cada classe per facilitar el procés. Els frames utilitzats per comprovar els resultats varen ser els mateixos que vàrem emprar en el primer experiment per, posteriorment, poder comparar els resultats.

Automobilisme: Circuit



41. Exemples de regions detectades en la categoria Circuit mitjançant MSER

El primer punt destacable que observem en aquest primer vídeo és que el nombre de regions obtingudes ha augmentat considerablement respecte a les trobades mitjançant Harris-Affine. L'altre fet ressaltable és el fet que els punts d'interès trobats per el MSER descriuen tant els objectes principals de la imatge com el seu entorn, tot i que aquest amb menor detall. A més a més el nombre de regions detectades en imatges fosques també

es sensiblement superior.

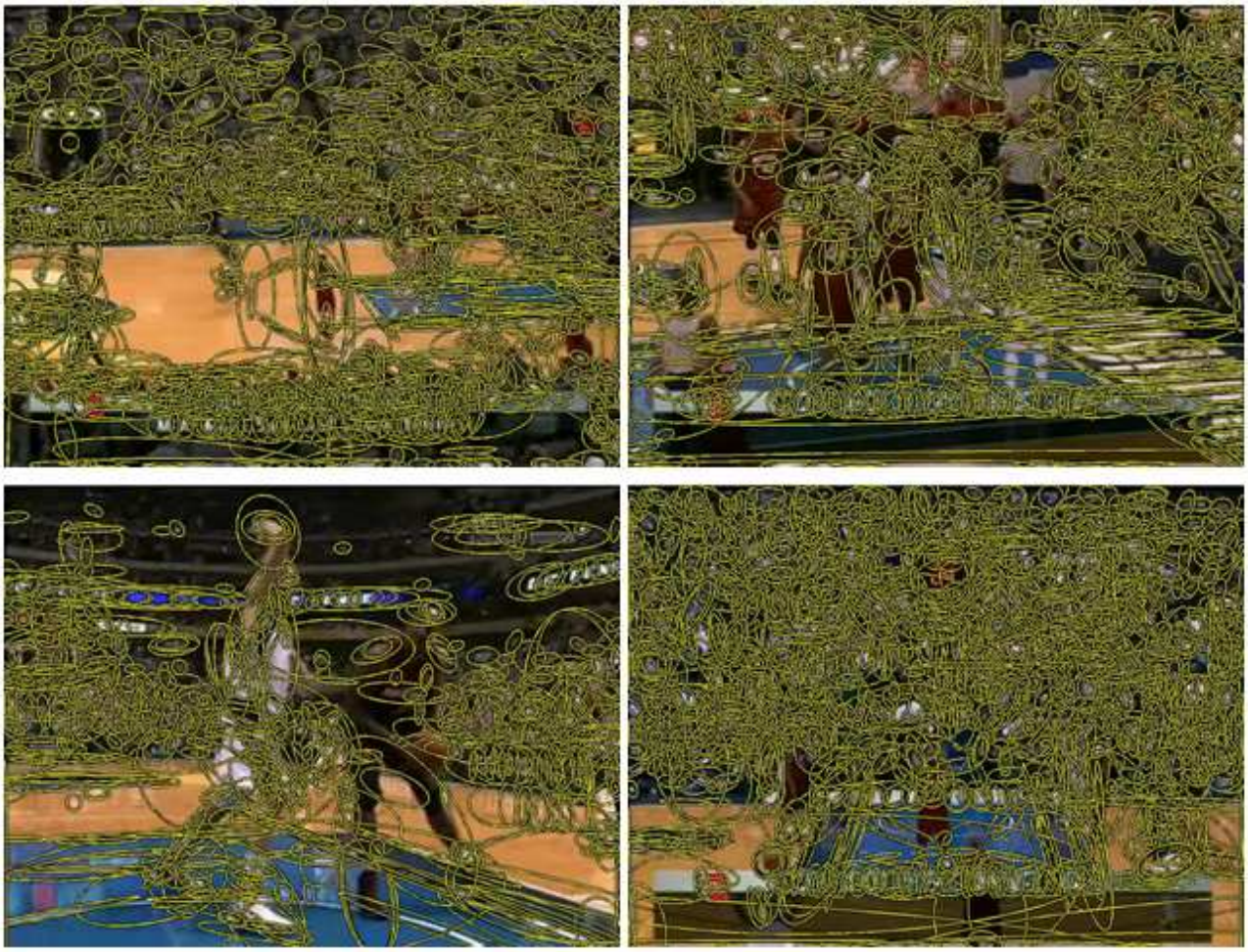
Automobilisme: Rally

42. Exemples de regions detectades en la categoria Rally mitjançant MSER

En aquests frames també podem comprovar que el detector MSER és capaç de detectar punts d'interès en l'objecte i en l'entorn tot i que en algunes imatges aquest fet és molt més accentuat (frame superior esquerra). Aquest fet ens facilitarà la diferenciació

entre les classes Rally i Circuit ja que, amb aquest detector, a part de tenir informació de l'element central també l'obtenim de l'entorn.

Bàsquet:



43. Exemples de regions detectades en la categoria Basquet mitjançant MSER

Els resultats que s'han obtingut en el vídeo de Bàsquet són bastant semblants, es troben un alt nombre de paraules i distribuïdes per tota la imatge. La diferència més significativa és l'augment de regions que descriuen característiques del terreny de joc com les línies o la cistella.

Bicicleta Tot Terreny

44. Exemples de regions detectades en la categoria BTT mitjançant MSER

Tal i com passava aplicant el detector de Harris en el vídeo BTT1.avi, en alguns frames on la imatge és poc nítida degut a la pols del circuit s'obtenen poques paraules. No obstant amb el detector MSER aquest nombre és superior i en les imatges més clares s'obté una bona descripció del ciclista i l'entorn que l'envolta. En aquesta mostra també es pot observar com, encara que la imatge sigui fosca, el detector és capaç de trobar un bon nombre de regions; aquest fet és especialment destacable en la el frame superior esquerra on el detector de Harris-Affine no hi ha trobat cap paraula.

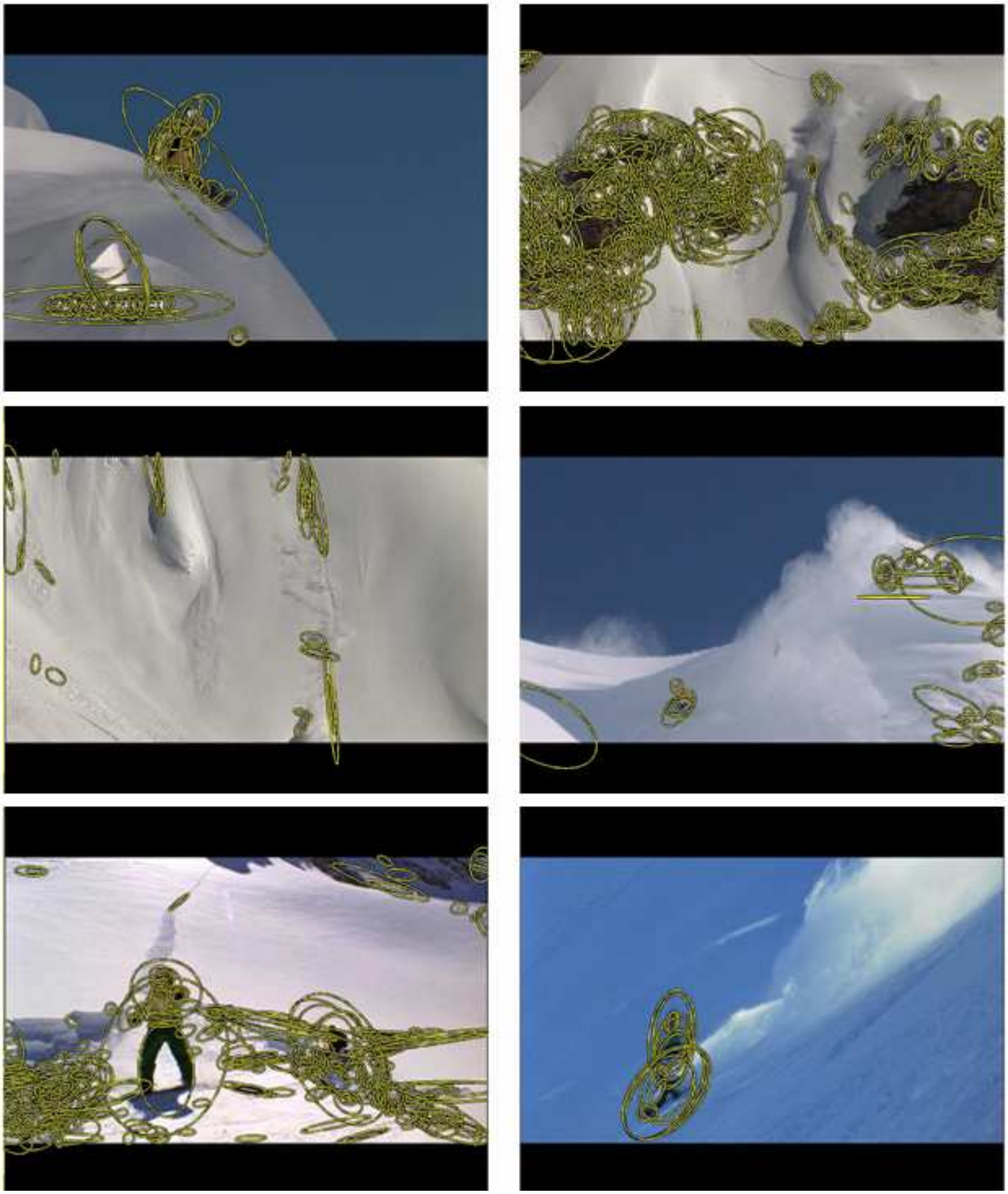
Futbol

45. Exemples de regions detectades en la categoria Futbol mitjançant MSER

En els frames dels vídeos observem que en les imatges ne primer pla el nombre de regions detectades és semblant que amb el de Harris. No obstant, quan les imatges són més llunyanes el detector MSER obté un major nombre de paraules aconseguint així descriure parts d'elements típics d'un camp de futbol com el cercle central o les línees que defineixen l'àrea. Un altre factor destacable que trobem en aquests frames és que en

tots ells apareix la pilota com a paraula, cosa que no passava amb Harris-Affine.

Snowboard



46. Exemples de regions detectades en la categoria Futbol mitjançant MSER

El nombre de regions que s'han detectat mitjançant aquest detector, en els vídeos d'snowboard, és molt més alt que en l'anterior experiment. Això es pot comprovar clarament en la figura superior dreta. A més a més també ha aconseguit trobar paraules

on l'anterior no ho ha aconseguit (frame central esquerra). A més a més, aquest increment del nombre de punts d'interès, ha servit per poder detectar elements que anteriorment no s'han trobat, com el surfista que apareix en la imatge central dreta.

4.5.4 Conclusions de l'experiment 2

Un cop examinats els resultats obtinguts amb el detector MSER podem deduir els següents aspectes:

- El detector MSER és capaç d'obtenir, generalment, un nombre més alt de regions dins la imatge que el de Harris-Affine. A més a més, aquests solen quedar més distribuïts en la imatge aconseguint, així, descriure una major part d'aquesta.
- En imatges on els canvis d'intensitat són baixos acostuma a ser més efectiu que el seu homònim.
- Tal i com s'ha explicat anteriorment, els detectors MSER sempre detecten, com a mínim, una regió. Això evita que trobem imatges sense informació i facilita l'automatització del procés de descripció de les regions mitjançant SIFTS.
- El detector de regions MSER ha obtingut aproximadament una mitjana de 400 regions per imatge.

4.5.5 Experiment 3

En el tercer experiment s'han buscat les regions a les diferents imatges mitjançant un Regular Grid. Tal i com s'esperava s'ha trobat un gran nombre de regions per a totes les mostres utilitzades: de 3000 a 6000 depenent de la resolució que s'utilitzava en el vídeo Font. Les imatges resultats d'aquests experiments no seran mostrats ja que, degut a l'alt nombre de regions trobades, aquestes apareixen pintades totalment de color groc. S'ha utilitzat una graella de mida 10, cada 10 píxels creava una el·lipse.

El principal objectiu d'aquest experiment era comprovar que no es produïa cap problema durant l'aplicació del Regular Grid i que totes les imatges obtenien un determinat nombre de regions. Tal i com s'esperava, això ha estat així.

4.5.6 Conclusions de l'experiment 3

Un cop s'ha realitzat l'experiment 3 s'han pogut realitzar les següents observacions:

- Les regions que es troben en els diferents frames d'un mateix vídeo són exactament les mateixes. Així, doncs, si fos necessari es podria optar per analitzar només el primer frame de cada vídeo.
- Tal com s'esperava els canvis de qualsevol tipus en la imatge no afecten les regions detectades mitjançant un regular Grid.
- La detecció de regions mitjançant un Regular Grid ha obtingut aproximadament una mitjana de 4000 regions per imatge.

4.5.7 Anàlisis dels resultats

Gràcies als experiments realitzats hem pogut veure que la detecció de regions mitjançant el detector Harris-Affine detectava un nombre de regions bastant baix arribant, fins i tot, a no trobar-ne cap. Aquest fet ens ha fet prendre la decisió de descartar l'ús d'aquest detector per a la realització del vocabulari ja que ens trobaríem amb imatges que no es podrien arribar a descriure degut a l'absència de paraules.

Així doncs la creació del vocabulari es realitzarà mitjançant les paraules visuals obtingudes amb el detector MSER i el Regular Grid.

4.6 Resultats del vocabulari:

Per a realitzar diferents proves sobre el funcionament del programa havíem pensat en crear diferents tamanys de vocabulari: de 300 paraules, de 500 i de 1000. Aquesta operació calia realitzar-la dues vegades: una per les paraules aconseguides a través del detector de regions MSER i una altra per les paraules aconseguides mitjançant el Regular Grid.

Per reduir el nombre de dades que s'utilitzarien per a la creació del vocabulari i agilitzar, d'aquesta manera, el temps de computació vàrem utilitzar 20 imatges per a cada vídeo font (excepte del vídeo de bàsquet, del que en vàrem utilitzar 40). Aquestes imatges varen ser escollides totes, aleatòriament, de diferents shots per tal de minimitzar la informació repetida.

El primer vocabulari es crearia a partir de, aproximadament, 96000 vectors que s'havien identificat mitjançant el detector de regions MSER.

El segon vocabulari es crearia mitjançant uns 800000 vectors que s'havien aconseguit aplicant un regular Grid sobre les diferents imatges de mostra.

4.6.1 Experiment 1

Mitjançant la funció de Matlab [A,B] = kmeans(matriu,300) [10] preteníem agrupar tots els vectors aconseguits a través dels descriptors de fitxer SIFT en 300 representants de les diferents paraules.

El primer vocabulari es va generar en un ordinador portàtil que disposava d'un processador Pentium IV 2700Mhz i 256 MB de memòria RAM. El vocabulari es va crear satisfactòriament però l'ordinador va necessitar 6 hores per a poder-lo calcular.

El segon vocabulari es va intentar generar en el mateix ordinador però, després de 2 hores d'execució, el procés va fallar per falta de memòria. Vàrem decidir realitzar aquest càlcul en un ordinador més potent: Pentium IV 3,2Ghz i 3 GB de ram. Amb una màquina amb més memòria vàrem aconseguir calcular el segon vocabulari; el temps d'execució d'aquest segon càlcul va ser de 40 hores.

4.6.2 Experiment 2

El segon experiment consistia en obtenir un vocabulari de 500 paraules. Després d'haver vist la poca capacitat de l'ordinador portàtil vàrem decidir executar directament la funció kmeans en la màquina més potent.

El primer grup de vocabulari no es va poder crear ja que, després d'aproximadament 24 hores de càlcul, el Matlab retornà error per falta de memòria.

Vàrem creure que el segon grup de vocabulari, que s'hauria d'originar amb 9 vegades més vectors que el primer, ens retornaria el mateix error i ho vàrem voler comprovar. Després de 10 minuts de càlcul la màquina va quedar sense memòria.

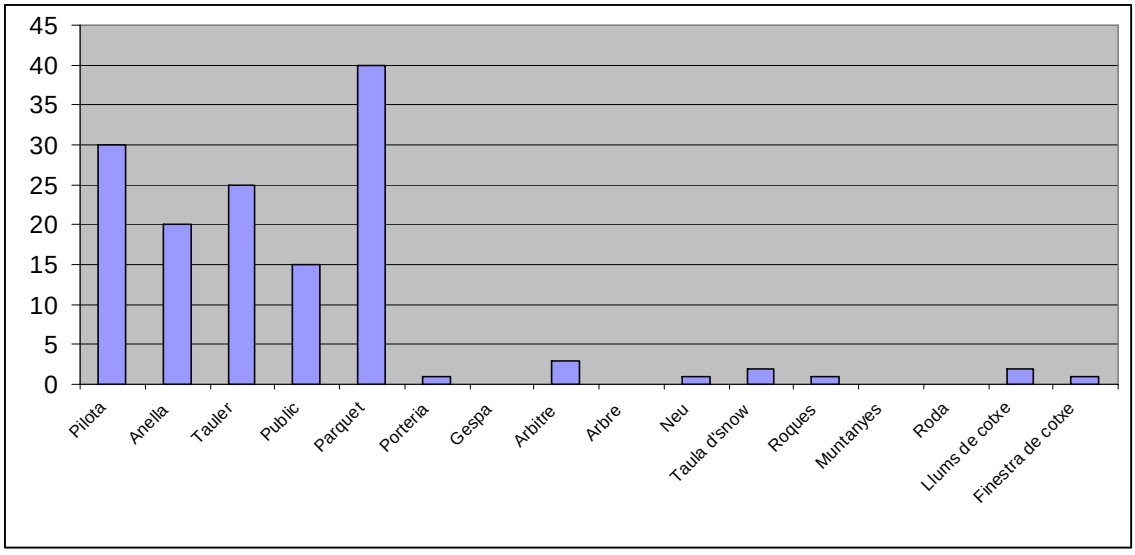
Degut a aquest fracàs ja no vàrem intentar crear un vocabulari de 1000 paraules. Així, doncs, vàrem continuar el projecte únicament amb el vocabulari de 300 paraules.

4.7 Histograma dels descriptors d'imatges

Amb els diferents representants dels grups de vectors varem procedir a elaborar histogrames de les diferents imatges que teníem. D'aquesta manera esperàvem poder descriure els diferents frames comptabilitzant les paraules visuals que hi apareixien. Teòricament el resultat de l'algorisme d'agrupació havia de ser un seguit de vectors que representessin situacions típiques dels diferents esports, representants de les paraules. Comparant les regions obtingudes en una imatge amb les paraules resultants del k-means mitjançant la distància euclidiana podíem saber a quin grup pertanyien. Guardant en una taula la freqüència amb que es repetia un grup de descriptors dins d'una mateixa imatge esperàvem trobar un bon mètode per diferenciar imatges dels distints tipus d'esports.

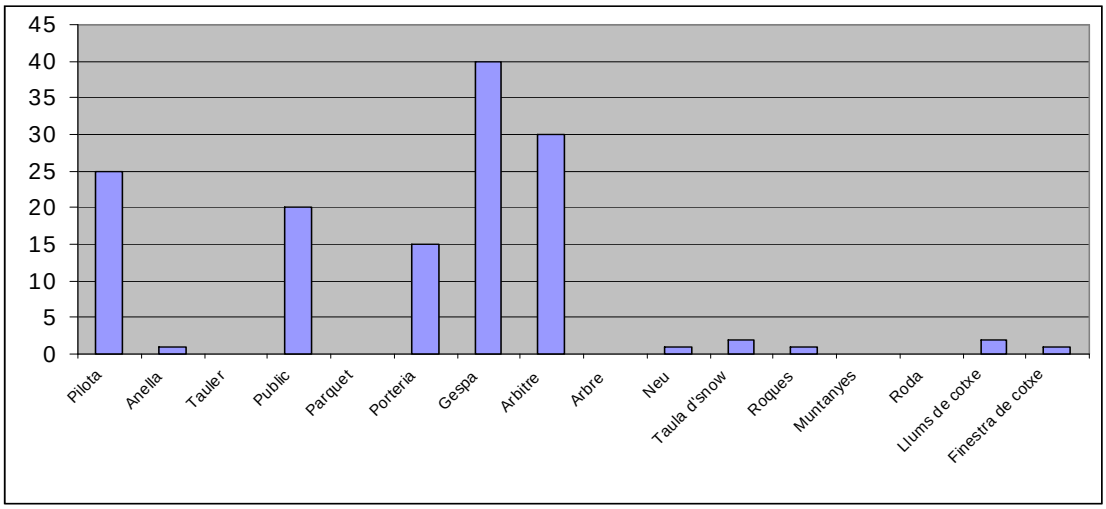
Simplificant-ho podríem dir que esperàvem trobar, per exemple, els següents histogrames per una imatge de basquet, una de futbol, una d'snowboard o una d'un rally:

Bàsquet:



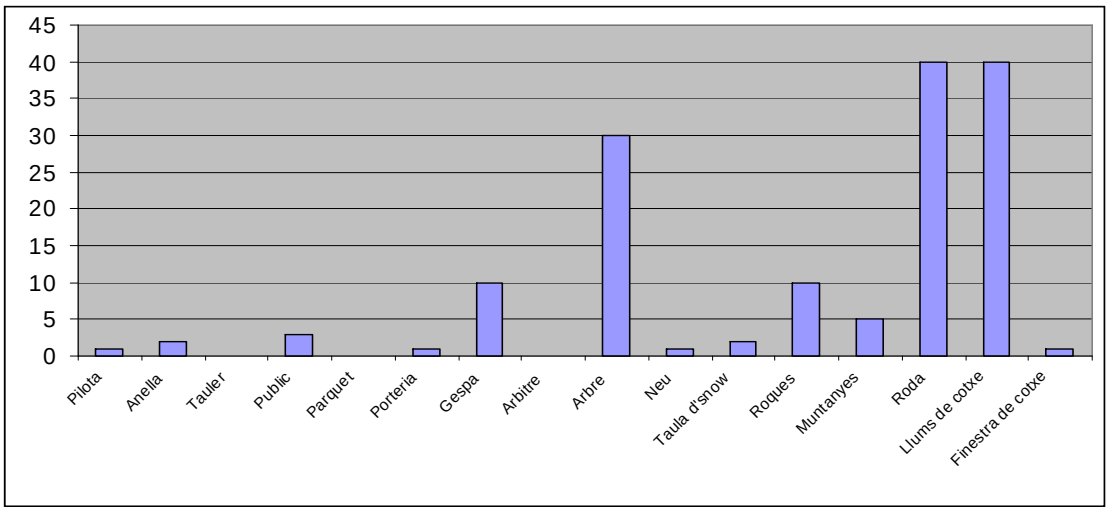
47. Simplificació d'un possible histograma d'un frame de Bàsquet

Futbol:



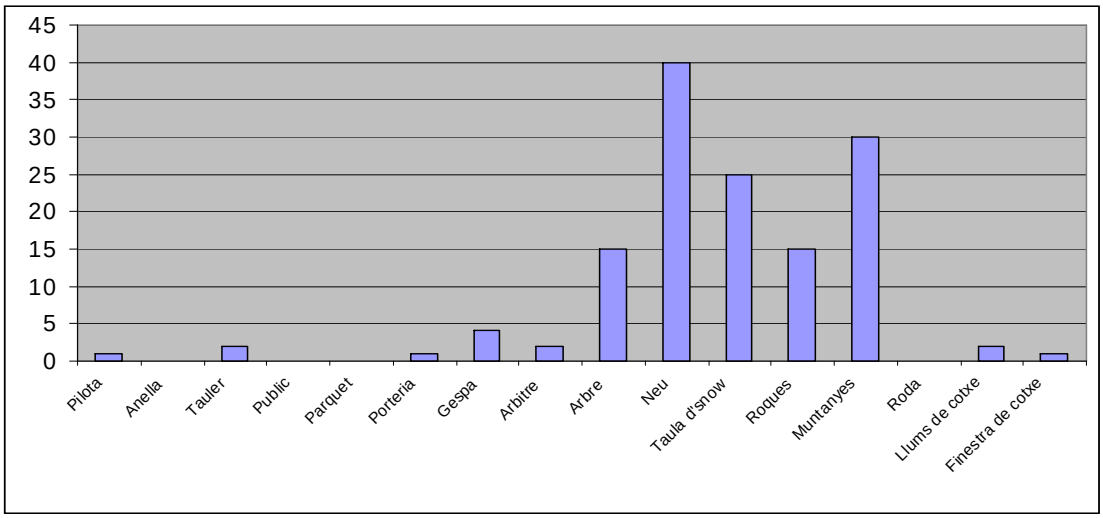
48.Simplificació d'un possible histograma d'un frame de Futbol

Rally:



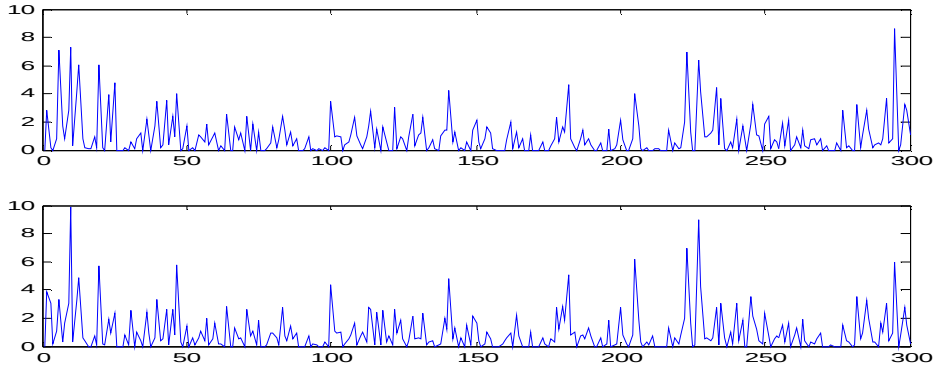
49. Simplificació d'un possible histograma d'un frame de Rally

Snow:

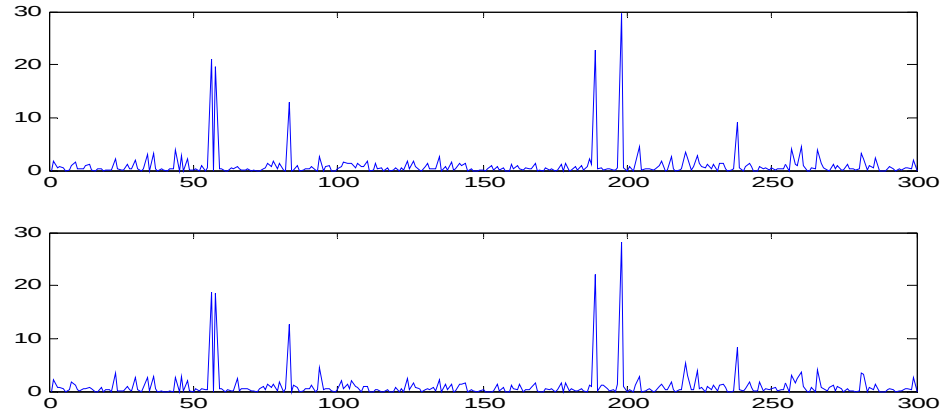


50. Simplificació d'un possible histograma d'un frame de Snowboard

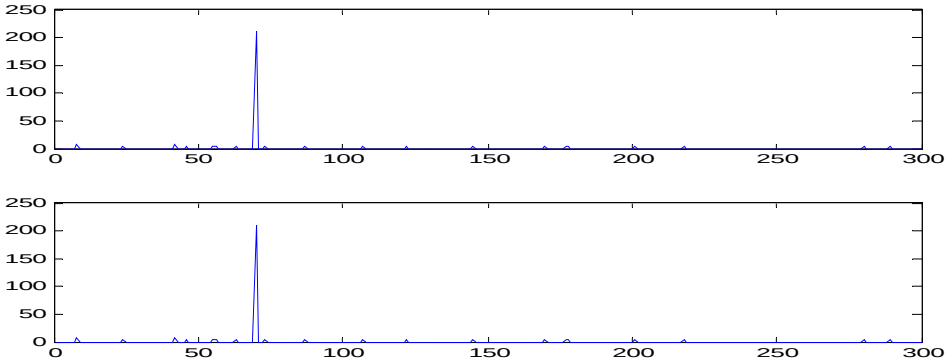
No obstant els resultats no foren tan fàcils de comprovar com en l'exemple anterior. En lloc de noms d'objectes disposàvem de vectors de 128 bytes i en comptes de tenir-ne 20 en teníem 300. A continuació es mostren alguns exemples d'histogrames que hem obtingut:



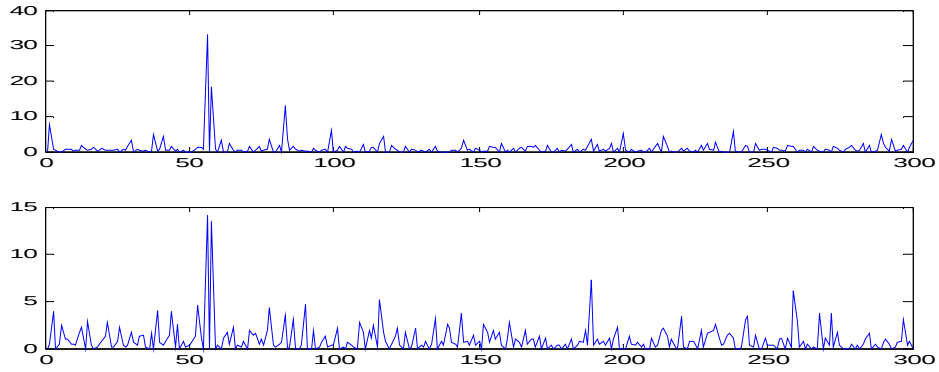
51. Histogrames de Bàsquet realitzats amb el vocabulari Regular Grid



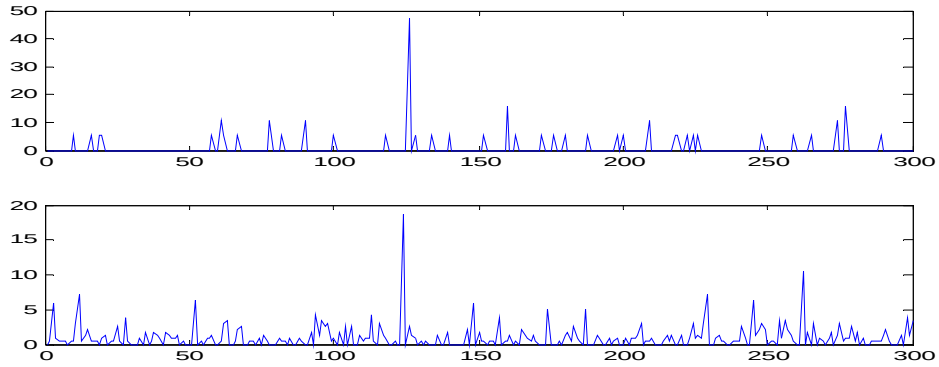
52. Histogrames d'Snowboard realitzas amb el vocabulari Regular Grid



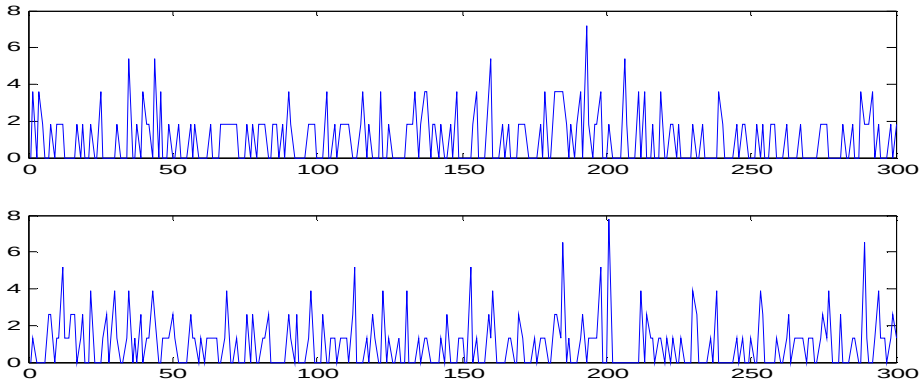
53. Histogrames de BTT realitzats amb el vocabulari MSER



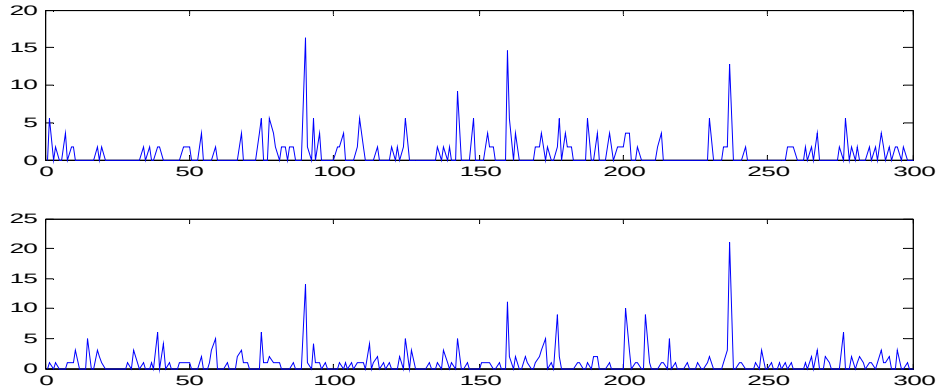
54. Histogrames de Futbolrealitzats amb el vocabulari Regular Grid



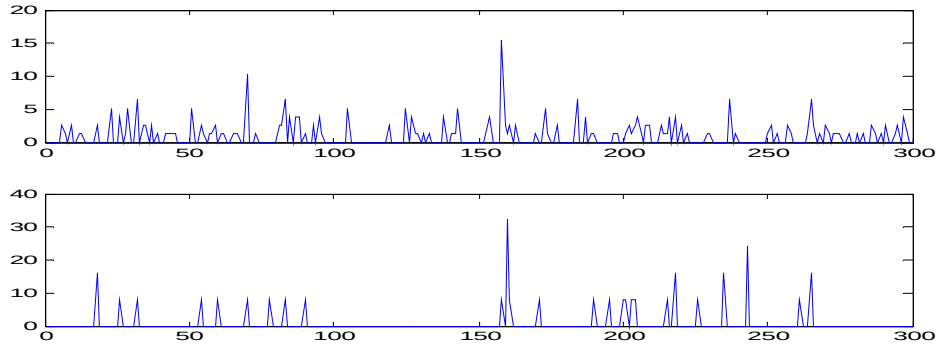
55. Histogrames de Futbol realitzats amb el vocabulari MSER



56. Histogrames de Rally realitzats amb el vocabulari MSER



57. Histogrames de Circuit realitzats amb el vocabulri MSER



58. Histogrames d'Snowboard realtizats amb el vocabulari MSER

Si observem els diferents histogrames de les classes (figures 51 a 57) podem veure com els histogrames d'una mateixa categoria s'assemblen entre ells mentre que si es comparen amb la resta són bastant diferents. Per poder classificar les diferents imatges i trobar-ne de semblants ens vàrem basar en aquest fet.

5 Correspondència entre imatges

5.1 Image Retrieval

El primer pas necessari per determinar si els resultats havien estat satisfactoris era comparar tots els histogrames entre ells per observar si els del mateix esport s'assemblaven. Per a fer-ho vàrem decidir utilitzar el que es coneix com a "Image Retrieval": A partir d'una determinada imatge anomenada "query" retornar les k següents que més si assemblen ("retrievals").

59. Exemple d'Image Retrieval d'una imatge

Per a fer-ho vam programar un script en Matlab que comparava la distància euclidiana entre un determinat histograma K i la resta i els ordenava de menor a major distància, creant una taula amb la direcció de centenars de fitxer ordenats de menor a major semblança. Posteriorment vàrem modificar aquest script per tal de que també pogués calcular la distància Xi Quadràtica per tal de poder contrastar diferents mètodes i resultats.

Com ja s'ha comentat anteriorment treballàvem amb un gran volum de fitxers i això feia molt pesat i lent el procés d'ordenació de la taula de fitxers cada cop que se n'hi havia d'afegir un de nou. Per tal d'agilitzar el procés de comparació dels histogrames vàrem decidir calcular la distància entre l'histograma K i la resta d'histogrames i guardar-ne els resultats en una taula de dos columnes on en la primera es guardava el nom de l'histograma i en la segona la distància en la que es trobava:

Ruta de l'histograma	Distància
/mnt/dades/frames/imatge001.hist	50018
/mnt/dades/frames/imatge002.hist	36246
/mnt/dades/frames/imatge003.hist	15655
/mnt/dades/frames/imatge004.hist	78972
/mnt/dades/frames/imatge005.hist	59464
...	...
...	...
...	...
/mnt/dades/frames/imatge_n.hist	123688

Seguidament es procedia a l'ordenació d'aquesta taula. Per agilitzar aquest pas vàrem utilitzar l'algorisme d'ordenació – Bombolla [11] que redueix de forma considerable el temps necessari per ordenar una taula. D'aquesta manera obteníem una taula d'on podíem extreure els K-frames més semblants .

Després d'haver aconseguit una taula amb la relació dels histogrames més semblants era el moment d'avaluar els resultats. Si el sistemes utilitzats havien estat els adequats les primeres posicions de la taula, o un alt percentatge d'elles, havien de pertànyer a frames del mateix vídeo o de vídeos del mateix esport.

5.2 Precision & Recall

Per avaluar l'eficiència les consultes d'una base de dades o d'un buscador és comú utilitzar el que es coneix com a "Precision & Recall", un seguit d'equacions que retornen valors entre 0 i 1 o percentatges que representen l'efectivitat d'una consulta[12]. La precisió [Eq. 3] és el nombre d'encerts d'una consulta respecte el nombre de consultes totals mentre que el Recall[Eq. 4] és el nombre d'encerts respecte el nombre màxim d'encerts que es podien realitzar.

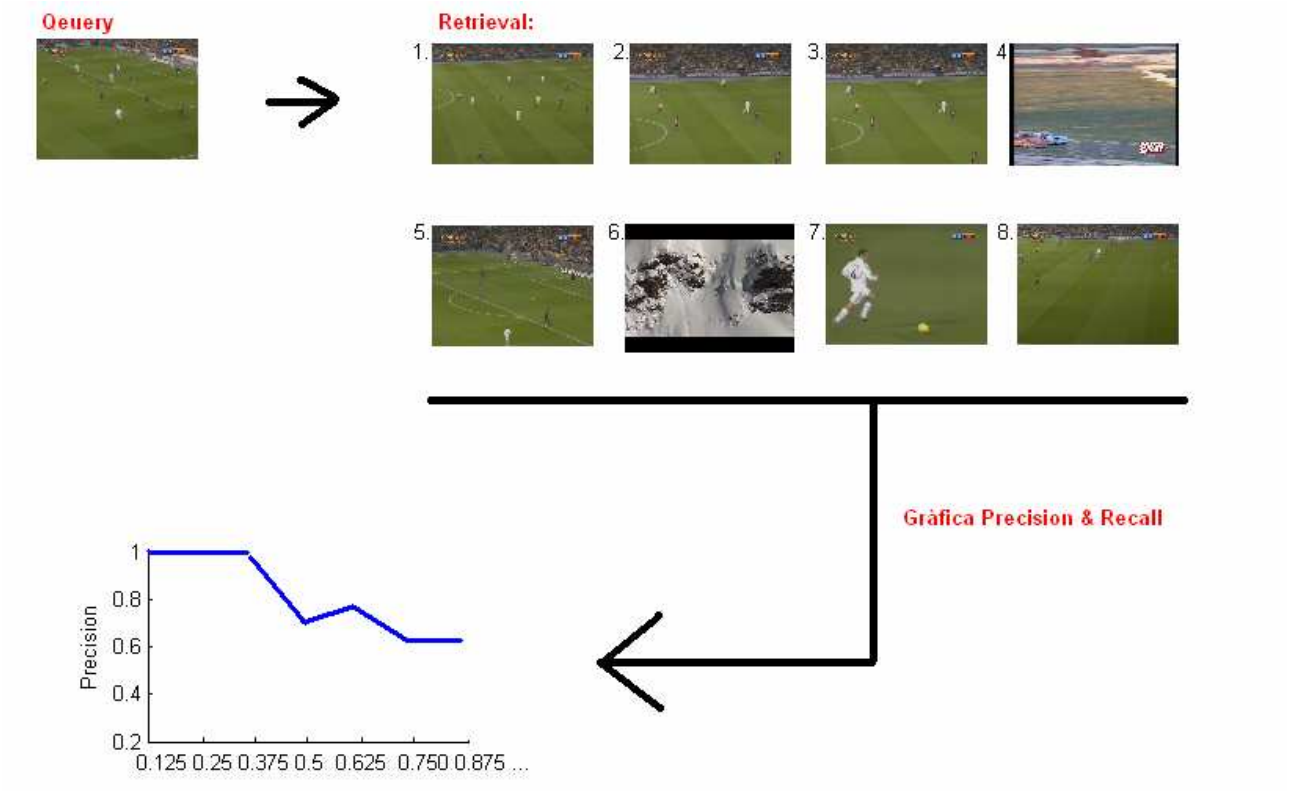
$$\text{Precision} = \frac{\text{nº Retorns correctes}}{\text{nº de Retorns}}$$

Equació 3: Precision

$$\text{Recall} = \frac{\text{nº Retorns correctes}}{\text{base de dades}}$$

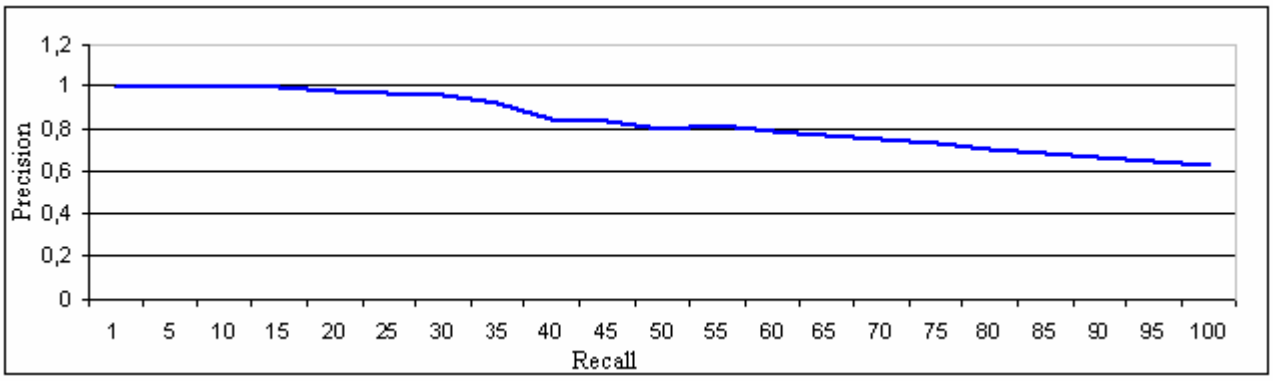
Equació4: Recall

Per tenir una visió del resultat de la precisió d'un conjunt de dades es consulta la precisió per n diferents subconjunts de amb n diferents cardinal: Es realitza la consulta inicialment amb una sola dada, seguidament amb dos, després amb tres fins arribar al nombre n.

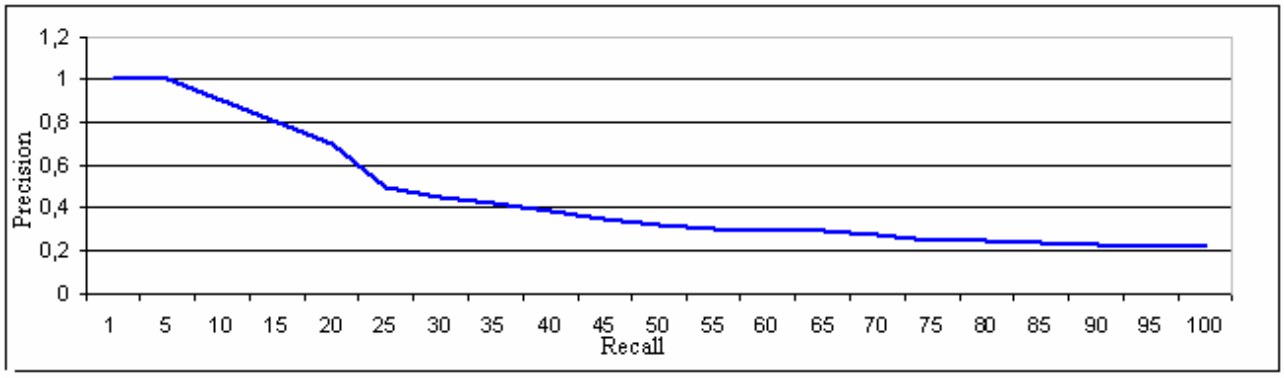


60. Exemple de precision & Recall

A continuació es procedeix a la representació dels resultats amb una gràfica, si l'àrea d'aquesta és superior al 50% es considera que els resultats de la consulta és correcte i s'accepten els resultats. En cas de què sigui inferior al 50% els resultats es desestimen, això significa que cal millorar els procediments de la consulta. Si aquesta gràfica fos una línea recta en el valor 1 obtindríem el resultat ideal, ja que l'àrea seria del 100%.



61. Exemple de gràfica on s'acceptarien els resultats: Àrea = 85,3%



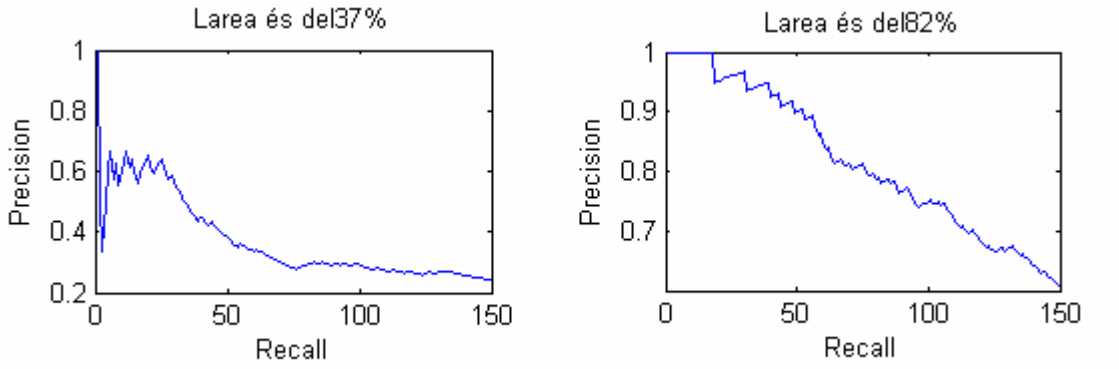
62. Exemple de gràfica on es desestimarien els resultats: Àrea = 44,5%

Calia, doncs, comprovar quins resultats obteníem amb els diferents vocabularis.

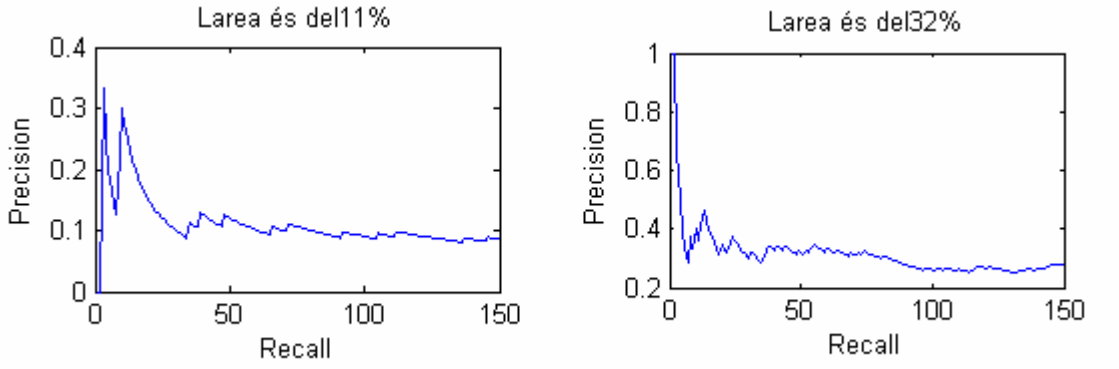
5.3 Anàlisis de la correspondència d'imatges

5.3.1Experiment 1

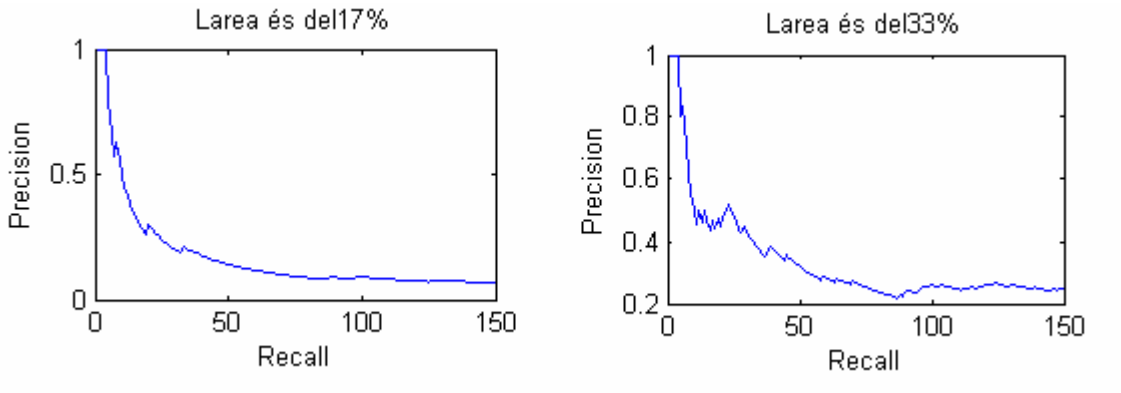
Tal i com s'ha comentat, per tal de determinar si el retrieval d'una imatge és satisfactori utilitzarem les gràfiques de precision & recall i l'àrea d'encerts d'aquesta. A continuació es poden veure alguns exemples de les gràfiques obtingudes:



63. Gràfiques Precision & Recall dels histogramas MSER de les imatges Futbol1/160.png i Basquet/160.png



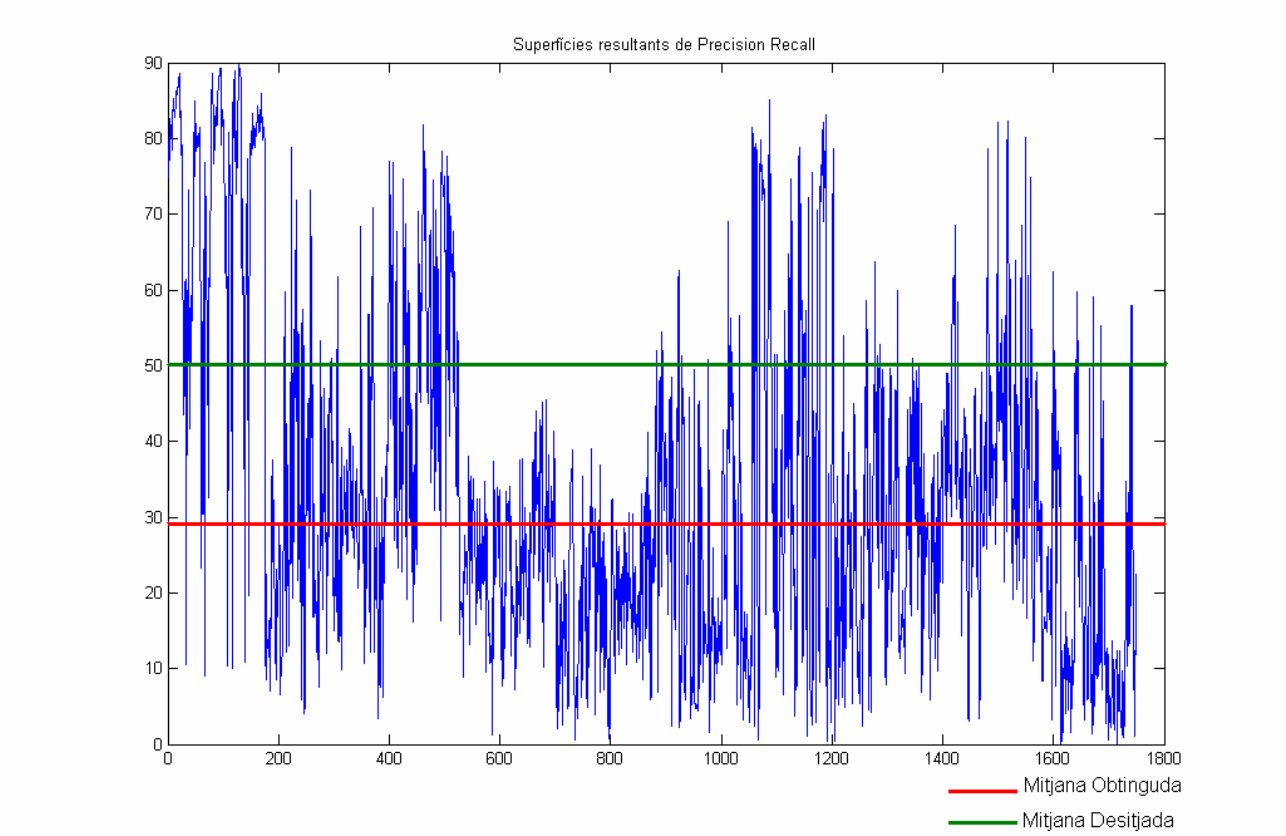
64. Gràfiques Precision & Recall dels histogramas MSER de les imatges Snow1/110.png i Btt1/100.png



65. Gràfiques Precision & Recall dels histogramas MSER de les imatges Rally1/100.png i Circuit2/30.png

Per poder observar i analitzar els resultats obtinguts el primer que vàrem fer va ser guardar en una taula totes les àrees de les gràfiques obtingudes mitjançant el procés de "precision & recall" per fer-ne la mitjana i representar-les en forma de gràfica. El resultat va ser molt pitjor de l'esperat, la mitjana de la superfície de les gràfiques era del 29,24%, molt inferior al 50% per tant no era un resultat satisfactòri. Ara calia esbrinar les possibles causes d'aquests mals resultats.

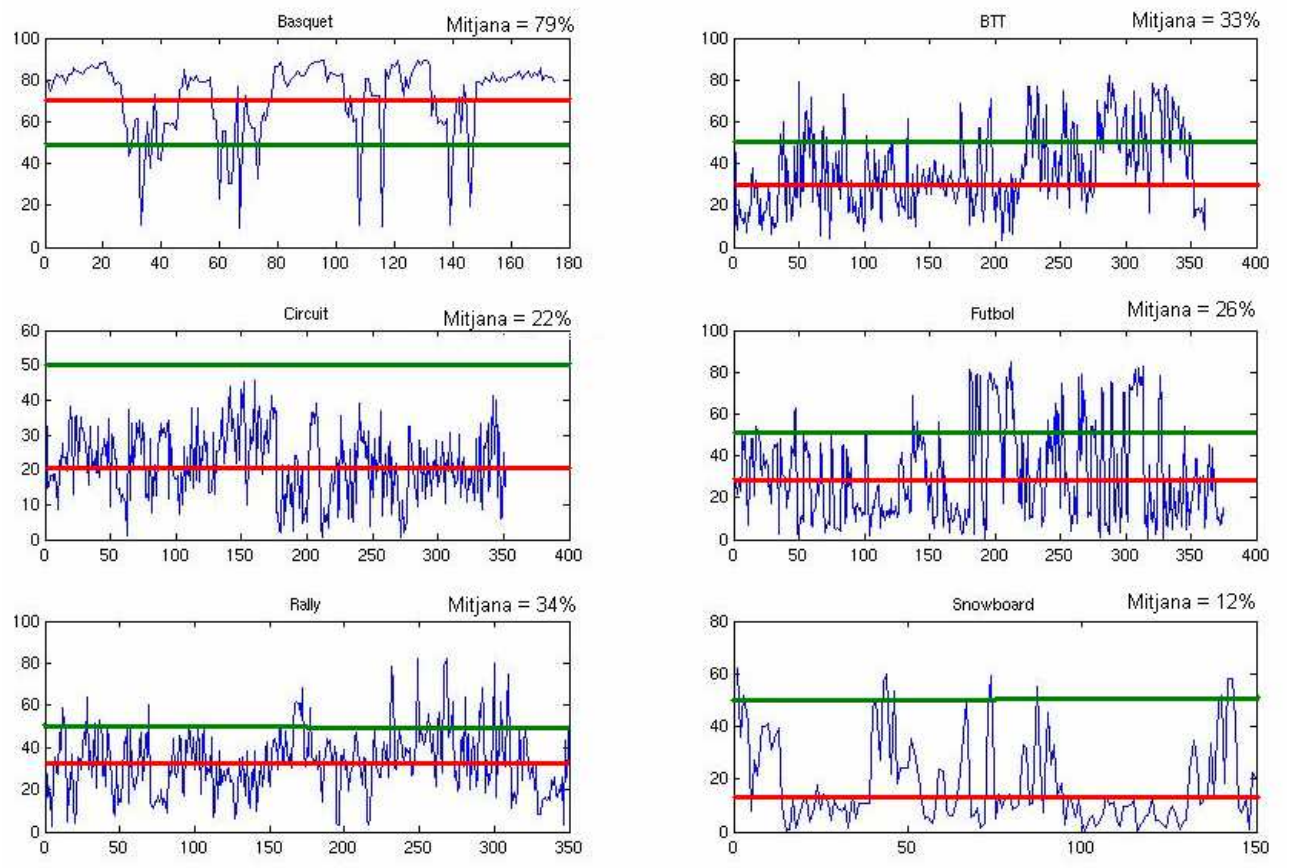
En la següent figura podem observar una gràfica on es representa l'àrea d'encert assolida per a cada una de les 1750 imatges resultants.



66. Àrees d'encert obtingudes en cada una de les imatges d'entrenament

Observant la gràfica anterior es veu clarament que els resultats obtinguts no eren, ni

de bon tros, els desitjats però també si observa un altre detall important. Es pot observar com hi ha grups d'imatges on les àrees obtingudes són clarament més altes que a la resta, com per exemple les primeres 150 imatges. Com que les diferents imatges estaven distribuïdes per carpetes i s'han analitzat seguin l'ordre alfabètic d'aquestes, ens plantejàrem la hipòtesis de que alguns tipus de vídeos, per alguna raó, eren més senzills d'identificar que els altres. Vàrem calcular les mitjanes de superfície de forma independent pels diferents esports i, tal com pensàvem, observàrem que aquesta variable era molt depenent de la tipologia de vídeo que estàvem tractant, per exemple en els frames pertanyien a vídeos de bàsquet la mitjana era de prop del 80% mentre que en els que apareixien esports d'hivern aquesta era només del 12%; en la resta de vídeos la mitjana oscil·lava entre un 20 i un 40%..

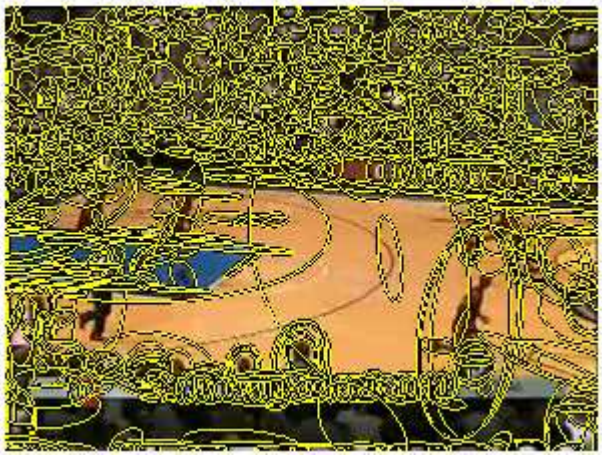


67. Àrees d'encert obtingudes en les diferents imatges ordenades segons la classe

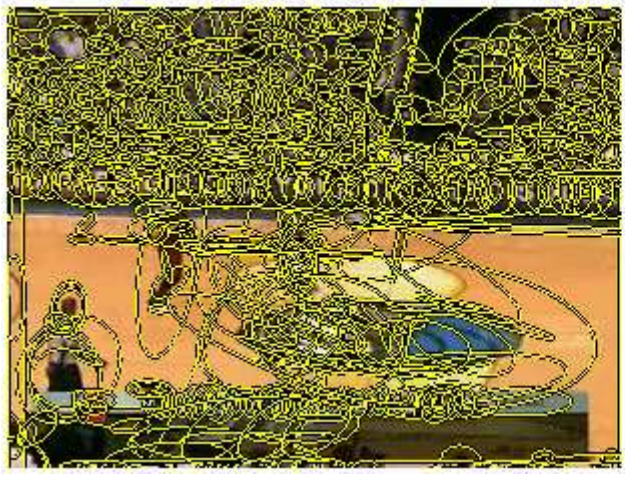
En aquestes gràfiques s'observa clarament la diferència entre les diferents classes de vídeos. Un cop confirmada aquesta hipòtesis calia esbrinar quin era el motiu d'aquesta diferència.

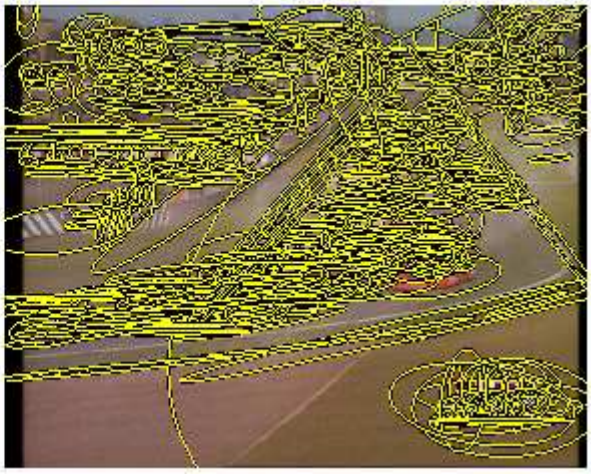
Fixant-nos en els diferents fitxers on es guardaven els descriptors de les imatges vàrem poder observar que el nombre d'aquests era molt més alt en les imatges que pertanyien a vídeos de bàsquet que a la resta. Això ens va fer pensar que, probablement, els detectors de regions que utilitzàvem no trobaven prous punts d'interès dins la imatge i

que no extreien prou informació de la imatge. Per comprovar aquesta teoria vàrem mirar les regions que s'havien generat en algunes de les imatges on l'àrea del precision-recall era més alta i les vàrem comparar amb algunes de les que tenien una àrea més baixa:

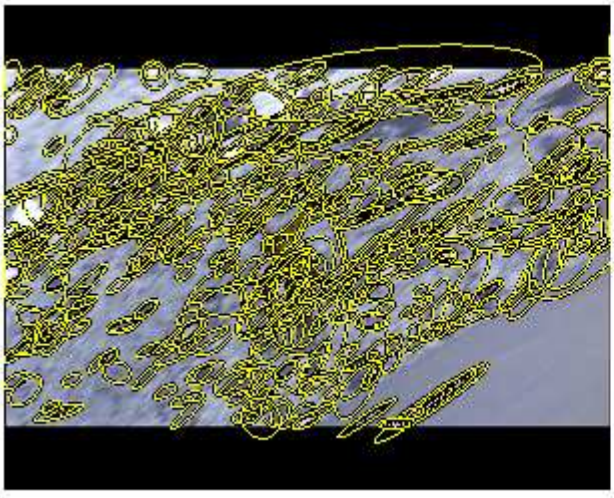


68. Bàsquet: Àrea = 85 **Futbol Àrea = 71**

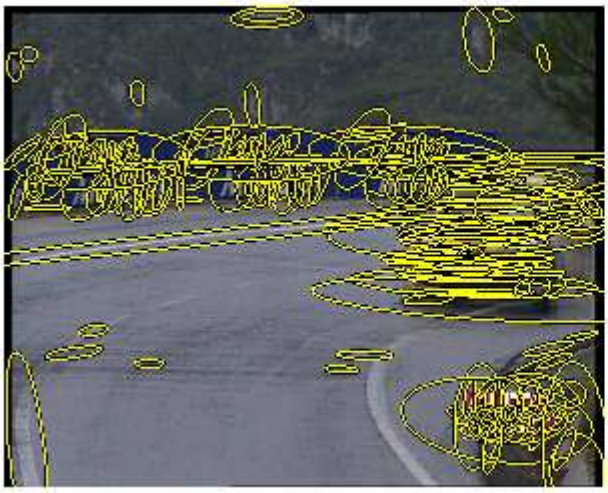




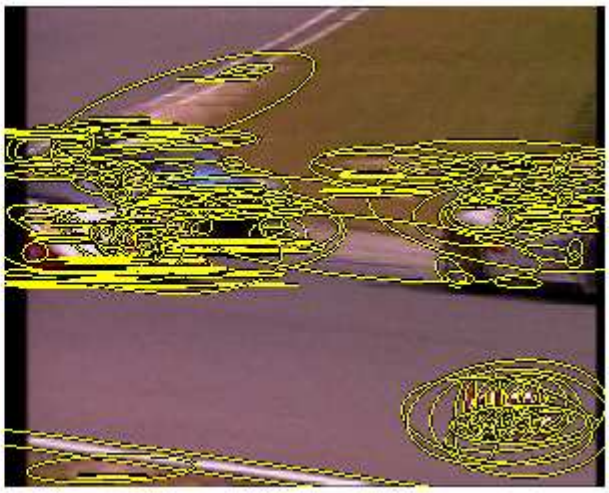
69. Bàsquet: Àrea = 80 **Circuit Àrea = 65**



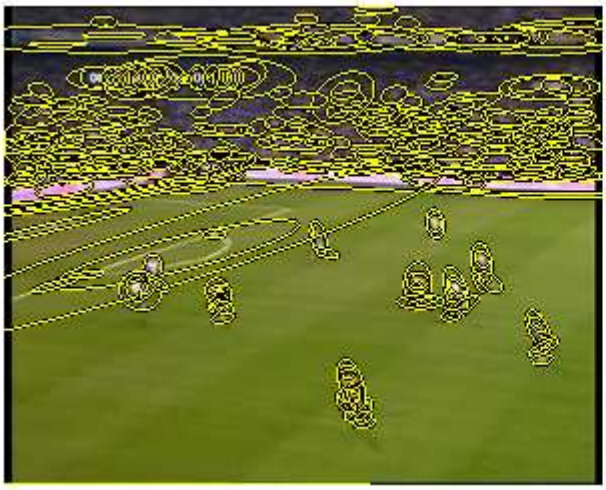
70. Snowboard Àrea = 67 **6. Rally Àrea = 53**



71. Rally: **Àrea = 31**

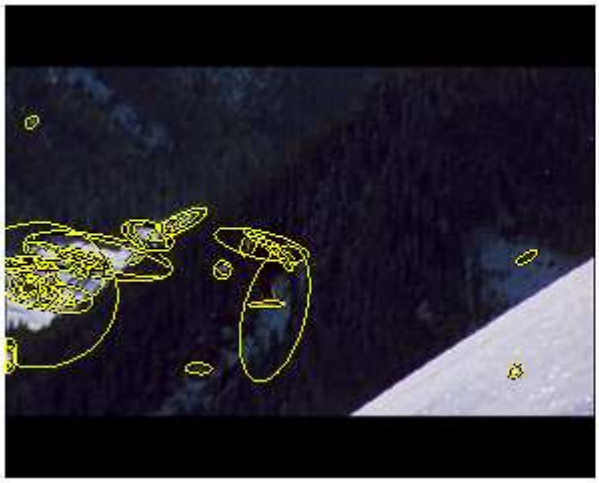


Circuit **Àrea = 27**

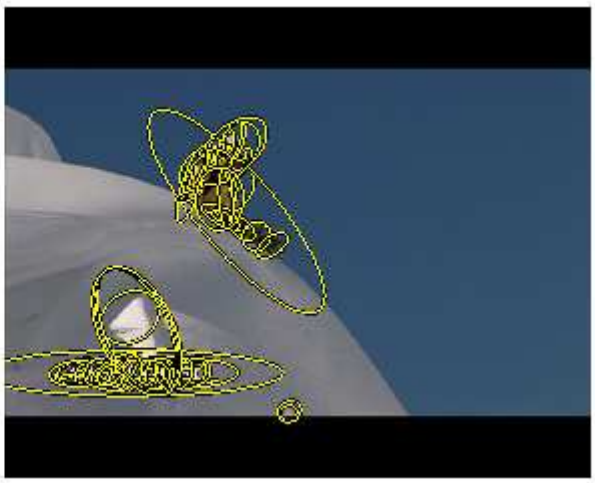


72.Futbol: **Àrea = 19**

BTT **Àrea = 18**



73. Snowboard: **Àrea = 13**

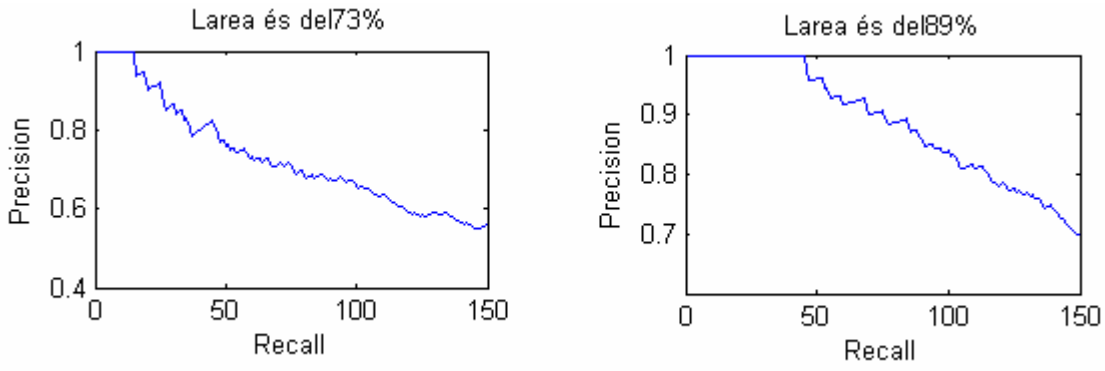


Snowboard: **Àrea = 8**

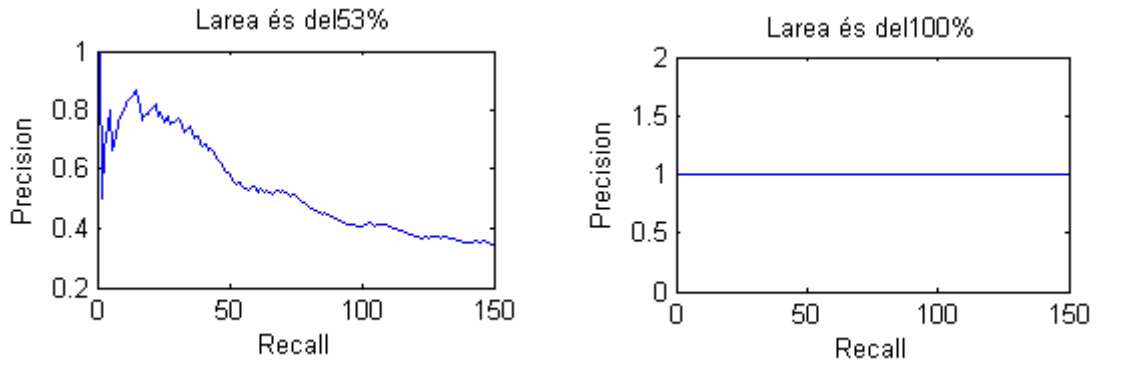
Tal i com es pot veure en les 12 mostres anteriors, generalment, les imatges que han obtingut un millor resultat a l'hora d'avaluar-ne els resultats també són les que disposen de més descriptors. D'aquest fet vàrem concloure que el baix nivell de correspondència entre imatges era causada per la falta de regions. Un altre possible motiu d'aquest resultats és la distribució heterogènia de les regions i l'absència d'aquestes en algunes zones de determinades imatges. Per exemple les imatges 5 i 6 no tenen un nombre excessivament alt de regions però estan distribuïdes per tota la imatge i varen obtenir àrees superiors al 50%, en canvi altres imatges com la 8 o la 9 tot i disposar d'un alt nombre de regions obtenen una àrea baixa ja que aquestes es troben molt concentrades en una mateixa zona. Aquesta absència de regions fa que parts de la imatge que es repeteixen freqüentment en determinades classes de vídeo, com per exemple la gespa en el futbol (imatge 9) o la neu en l'snowboard (Imatge 12) no s'utilitzin per realitzar els histogrames.

5.3.2 Experiment 2

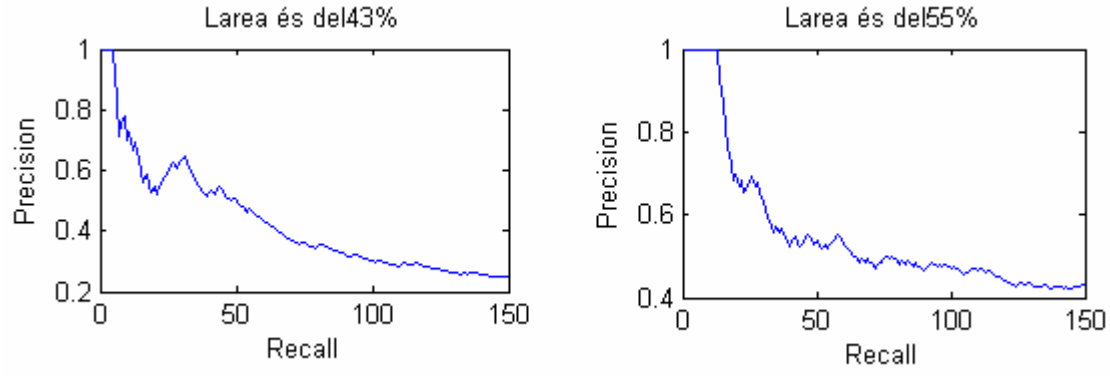
La segona prova, igual que la primera, consistia en crear i analitzar els resultats dels histogrames de les diferents classes però, aquest cop, mitjançant el vocabulari obtingut amb el Regular Grid. En la següent figura es poden observar exemples de diferents histogrames creats amb el segon vocabulari:



74. Gràfiques Precision & Recall resultants dels histogrames Regular Grid de les imatges Futbol/160.png i Basquet/160.png



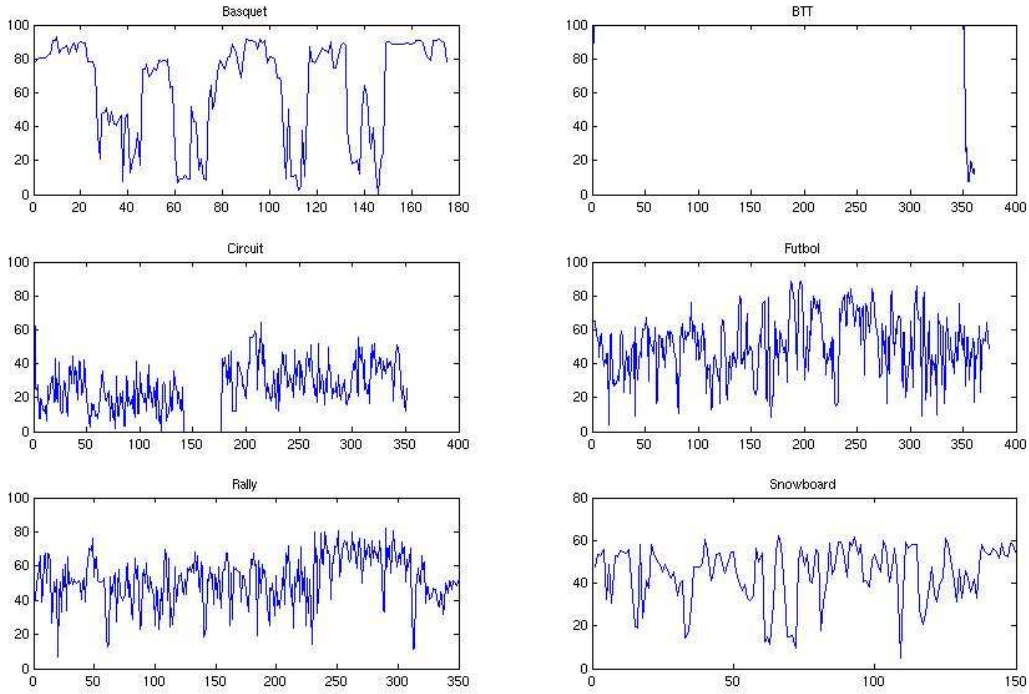
75. Gràfiques Precision & Recall resultants dels histogrames Regular Grid de les imatges Snow1/110.png i Btt1/100.png



76. Gràfiques Precision & Recall resultants dels histogrames Regular Grid de les imatges Rally1/100.png i Circuit2/30.png

Utilitzant el mateix script que havíem fet servir anteriorment vàrem poder observar que els resultats eren molt més satisfactoris ja que les àrees resultants eren notablement més altes que amb l'altre mètode. A diferència dels resultats obtinguts prèviament la mitjana de l'àrea ascendia a més del 50%, per tant podíem considerar els resultats obtinguts com a satisfactoris. En cinc de les sis categories de vídeo utilitzades la mitjana era superior al 50%, la única tipologia que no arribava a aquest llindar era l'"Automovilisme dins de circuit" degut, bàsicament, a que moltes de les consultes retornades pertanyien a la classe "Rally", la mitjana de l'àrea d'encerts era del 48%. En altres classes, com per exemple en la categoria de Bicicleta Tot Terreny, els resultats obtinguts eren increïblement bons, essent la mitjana molt propera al 100%. Les àrees d'encert segons el tipus de vídeo van ser les següents: Bàsquet 59%, Bicicleta tot Terreny: 91%, Futbol = 53%, Automobilisme = 51%, Rallies = 48% i Snowboard = 52%.

A continuació s'inclouen les gràfiques d'encerts amb les respectives mitjanes de cada tipologia:

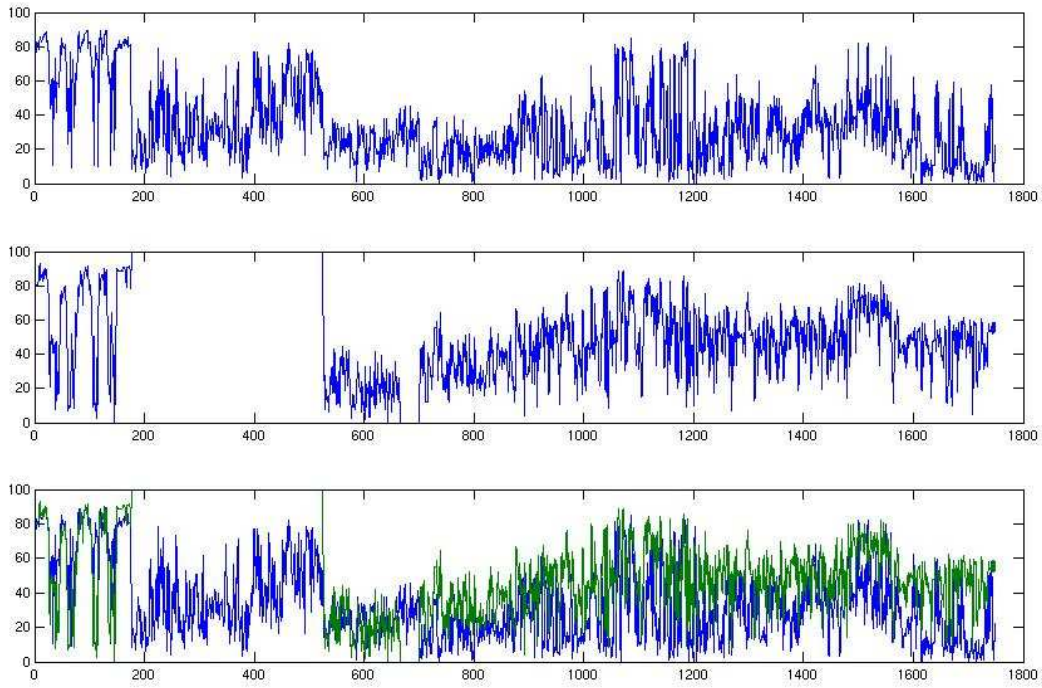


77. Àrees d'encert de les imatges ordenades per categories

5.3.3 Análisis dels resultats

Un cop hem realitzat els dos experiments observem que l'àrea mitjana de les gràfiques de precision & recall de les diferents categories és més alta quan utilitzem els histogrames que s'han creat a partir del vocabulari Regular Grid. També cal afegir-hi, per altra banda, que l'àrea mitjana que hem obtingut amb els retrievals dels histogrames creats amb el vocabulari MSER és inferior a 0.5. Aquests dos fets han comportat la desestimació de la utilització del vocabulari MSER per el següent pas del projecte: crear els models que ens serviran per classificar els diferents vídeos.

En les següents figures es comparen els resultats obtinguts segons els mètodes utilitzats per determinar els punts d'interès:



78. Comparació de les àrees d'encerts de les imatges mijançant el vocabulari MSER i el de Regular Grid

En la primera gràfica es pot observar l'àrea d'encerts obtinguda per a tots els frames utilitzats per a la creació del primer vocabulari en el que s'havia utilitzat el mètode MSER. En la segona s'observa la mateixa gràfica però amb el vocabulari creat mitjançant el Regular Grid. Es pot observar clarament que els resultats obtinguts són millors amb el segon mètode ja que en gairebé tots els punts de la gràfica les àrees d'encert de la segona representació són més alts que les de la primera. Això es veu amb més claredat en la tercera figura on s'han representat les dues gràfiques anteriors: de color blau la primera i de color verd la segona.

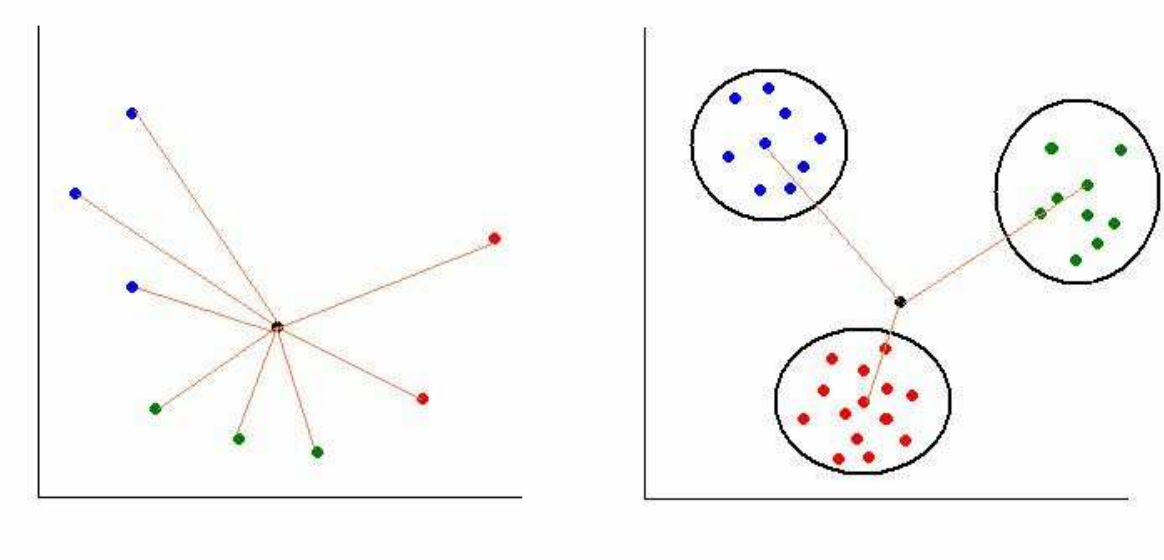
6. Classificaició segons classes:

L'objectiu d'aquest projecte era aconseguir classificar 6 classes de vídeo diferents. Per això en els subapartats 6.1 i 6.2 s'analitzen diferents tipus de classificadors que ens poden ajudar a obtenir un model de classificació satisfactori.

6.1 K-nearest neighbours

Fins ara cada cop que hem hagut de classificar un vector o un punt respecte un seguit de conjunt d'elements de la seva mateixa tipologia ho hem fet mitjançant les distàncies.

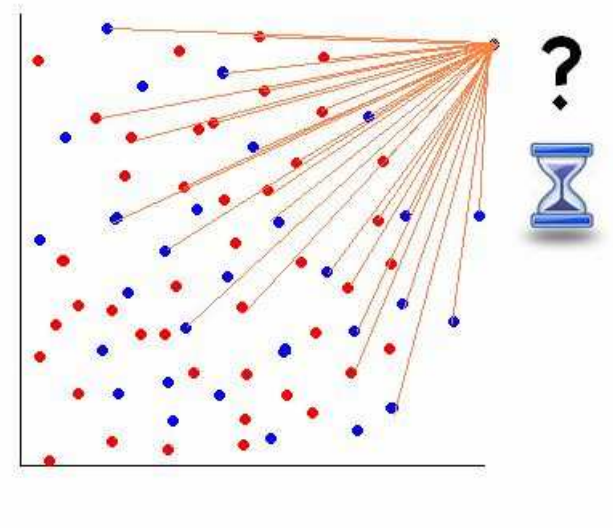
Quan hem necessitat saber quins eren els veïns més propers a l'element ho hem fet calculant la distància amb tots els elements dels que disposàvem i escollint els que retornaven una menor llunyania. De la mateixa manera quan hem necessitat determinar quines eren les imatges més semblants a un frame ho hem fet calculant la menor distància amb els centroides dels conjunts, els representants del grup. La dinàmica de comparar les distàncies d'un element amb els seus veïns també es pot utilitzar per a classificar-lo dins d'un grup, aquest sistema es coneix com a K-Nearest neighbours (k veïns més propers). Consisteix en buscar els K veïns més propers i observar quina és la seva categoria, la categoria que més vegades es repeteixi s'assigna a l'element que es vol classificar.



79. Exemple de funcionament del classificador "nearest neighbour"

Aquest mètode pot ser bastant eficaç en alguns casos ja que els resultats que

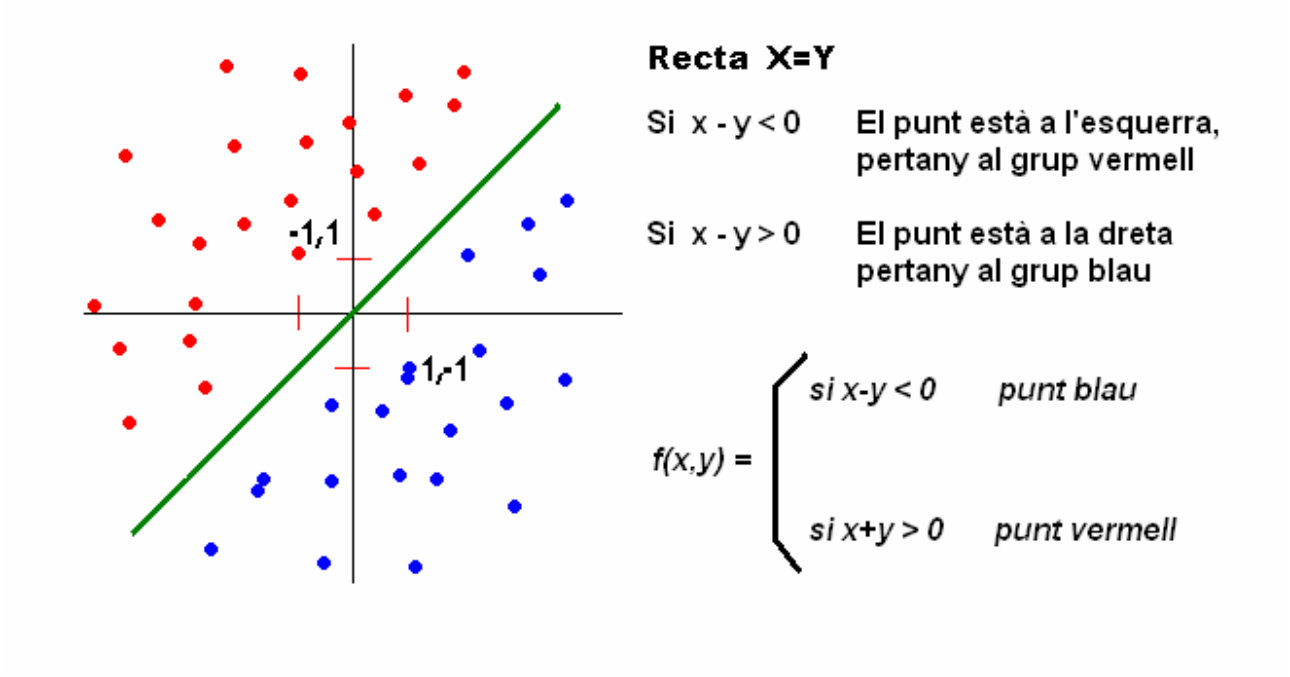
retorna no depenen de cap factor extern ni necessiten que un agent extern, humà o virtual, hagi de participar en el procés i prendre alguna decisió. No obstant és molt poc eficient i, tot i que pot resultar útil si s'utilitza amb un nombre d'elements baix, el temps d'execució pot ser excessiu si el nombre d'elements és molt elevat.



80. La utilització de distàncies comporta un temps d'execució molt alt

En aquest projecte es treballa amb un gran nombre d'imatges, aproximadament 2.000. Tenint en compte que per a cada imatge es crea un histograma de 300 camps, obtenim un total de 600.000 possibles distàncies a buscar. Aquest nombre és molt elevat, això ens impedeix utilitzar la distància euclídia a l'hora de classificar els frames d'un vídeo. A més a més, cal tenir en compte que a aquest nombre de mostres cal afegir-hi les imatges del vídeo que volem classificar. Per tal de poder analitzar una petit fragment de 30 segons d'una pel·lícula necessitem, com a mínim, extreure un frame per segon. Això significa que el nombre de distàncies que avaluaríem seria proper als 18 milions (600.000 x 30). Aquest fet deixava clar que calia trobar un mètode menys costós computacionalemnt que ens permetés determinar a quina categoria d'histogrames pertanyien els histogrames extrets dels frames d'un vídeo.

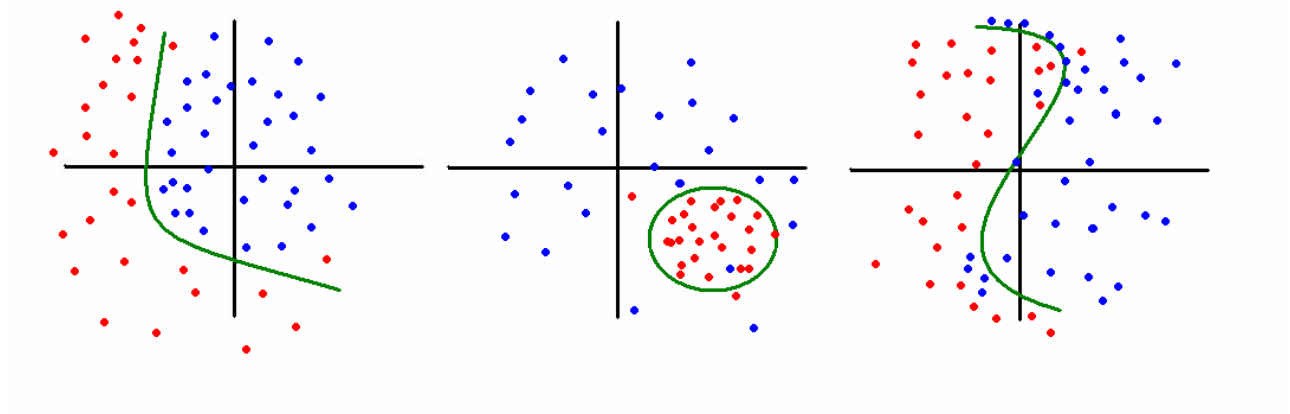
Si analitzem dos grups de vectors de dos components i els representem a través de dos eixos de coordenades en ocasions podrem observar que una recta pot delimitar quins punts pertanyen a un grup i quins pertanyen a l'altre. Si som capaços de determinar l'equació que defineix aquesta recta podrem classificar els punts a través de l'equació substituint les variables X i Y per les coordenades dels punts, podem determinar si el punt està a la dreta o a l'esquerra de la recta. En altres paraules, podem determinar a quin grup pertany.



81. Alguns grups de vectors es poden classificar mitjançant una recta

En la figura anterior es pot comprovar com es pot determinar a quin grup pertany un punt segons a quin costat es troba. Ara per saber a quin grup pertany un vector ja no és necessari buscar les distàncies amb els seus veïns, ara tan sols cal resoldre una funció. Per tant el temps que es necessita per ubicar un vector dins d'un grup ha passat a ser un temps constant. Aquest sistema també es pot utilitzar amb vectors de més de dos dimensions, l'únic que canvia és el nombre de variables que apareixen a l'equació ja que cada variable representa una component del vector.

No obstant no sempre resulta tant senzill determinar una equació que separi els diferents tipus de vectors ja que no solen correspondre's a rectes. Normalment aquesta equació no sol ser una recta, acostuma a ser una corba, un cercle, una equació de segon grau, etc. A més a més, en moltes ocasions és impossible trobar aquesta funció, per aquest motiu s'utilitzen aproximacions.



82. Exemples de línees no rectes que separen grups de vectors

Un altre problema de l'algorisme K-nearest neighbours és que només és capaç de separar classes de vectors mitjançant línies rectes i, com acabem de veure, això no sempre és possible. No obstant, existeixen altres classificadors de vectors com per exemple les Support Vector Machines[13] que solucionen aquest problema.

6.2 Support Vector Machines

A grans trets podríem definir les "Support Vector Machines", a les que anomenarem SVM a partir d'aquest moment, com un conjunt de mètodes supervisats d'aprenentatge[14][15] utilitzats per a la classificació de vectors de grans dimensions. El seu origen prové d'un camp bastant diferent al que es tracta en aquest projecte, la investigació genètica. Les SVM Van ser desenvolupades per el científic rus Vladimir Vapnik[16] en els laboratoris AT&T.inc a mitjans dels anys 90, coincidint amb l'explosió d'aquest tipus d'investigació on necessitaven un mètode ràpid i eficaç per classificar la informació extreta d'un gen o la manera d'agrupar diferents grups de proteïnes segons la composició dels seus aminoàcids. Tot i això, el seu ús s'ha extès en altres camps de la informàtica on es precisa classificar vectors de grans dimensions, com ara en la visió per computador.

L'objectiu d'aquestes es trobar una funció matemàtica, un model, que permeti classificar de forma ràpida i eficaç un vector.

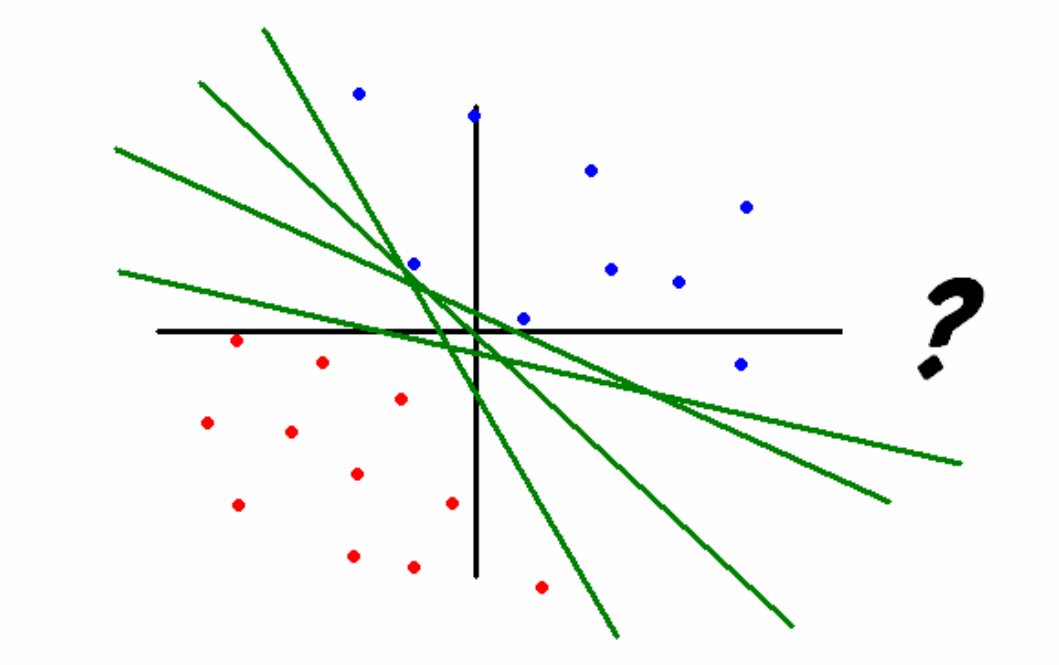
Les SVM per elles mateixes no són capaces de determinar quins vectors pertanyen a cada categoria. Necessiten un entrenament on un agent extern els indiqui el grup del que formen part els vectors, per això diem que utilitzen mètodes supervisats d'aprenentatge.

Existeixen dos tipus de SVM, les simples i les complexes[13]. Les primeres només permeten diferenciar entre dos classes de vectors mentre que les segones poden classificar-los en tants grups com es desitgi. No obstant, la fiabilitat d'aquestes és sensiblement més baixa.

6.2.1 Simple Support Vector Machines

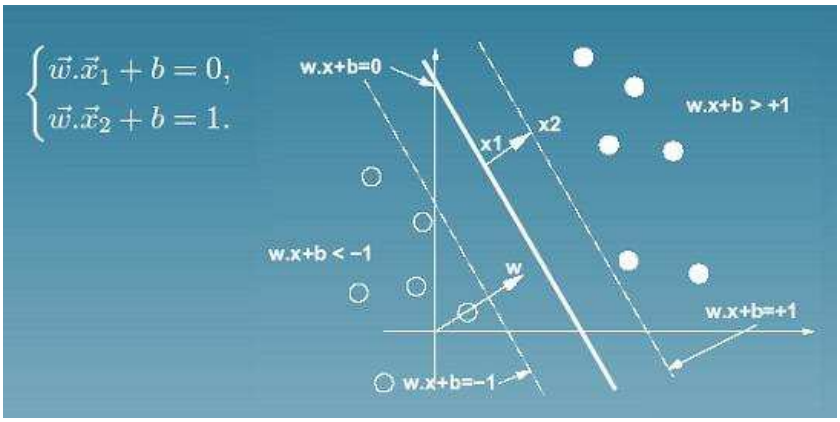
Com hem explicat anteriorment les SVM simples només són capaces de distingir entre dos classes de vectors. Inicialment es varen desenvolupar les SVM lineals. En

aquestes s'intentava trobar una recta, un pla o un hiperplà (un pla en un espai de més de 3 dimensions) capaços de separar els diferents vectors en dos grups.



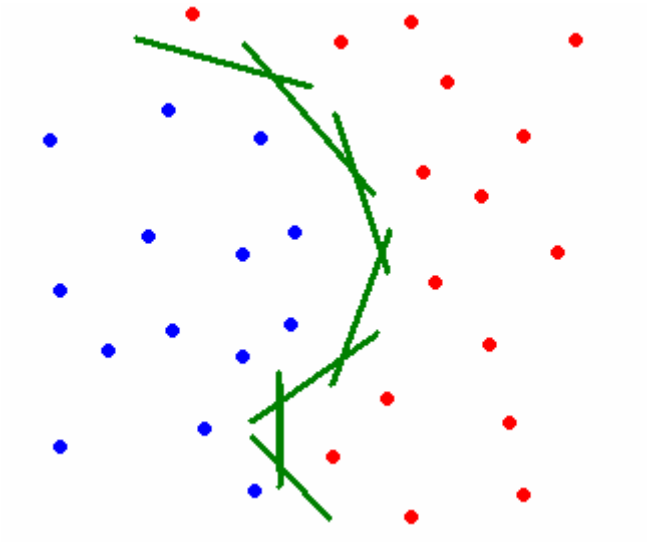
83. Existeixen diverses línies que poden separar grups de vectors. Cal determinar quina és la millor

Existeixen diferents rectes que poden separar dos núvols de punts, per això el primer que cal fer és determinar quina és la òptima. Vladimir Vapnik determinà que la millor recta per separar dos grups de vectors era la que obtenia un major marge envers els dos punts més externs dels núvols de punts, els punts de support. A més a més la recta òptima ha de ser equidistant als dos punts de support.



84. La millor línia de separció es troba mitjançant l'ajuda dels vectors de suport. [13]

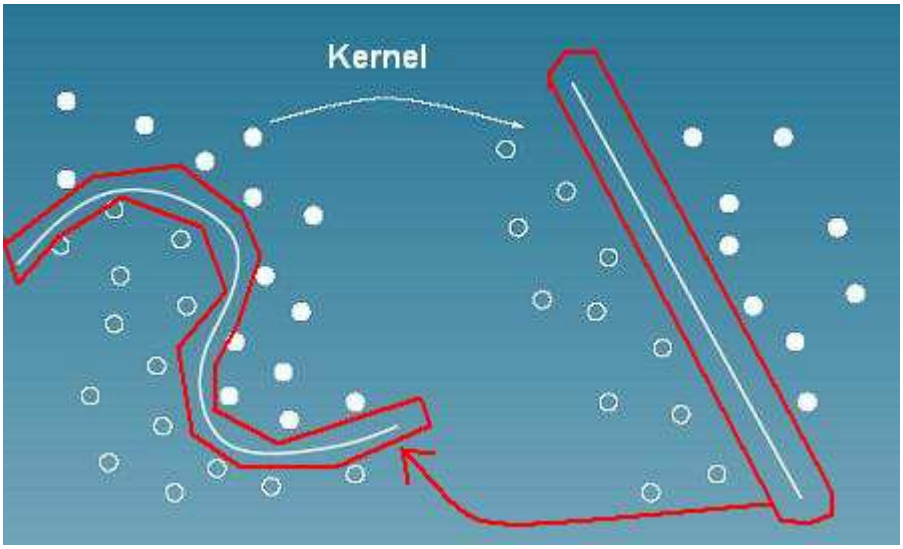
No obstant, com s'ha comentat prèviament (veure figura 82) no tots els núvols de punts es poden separar amb una recta. Algunes classificacions es poden solucionar mitjançant un arc convex, o combinant diferents rectes.



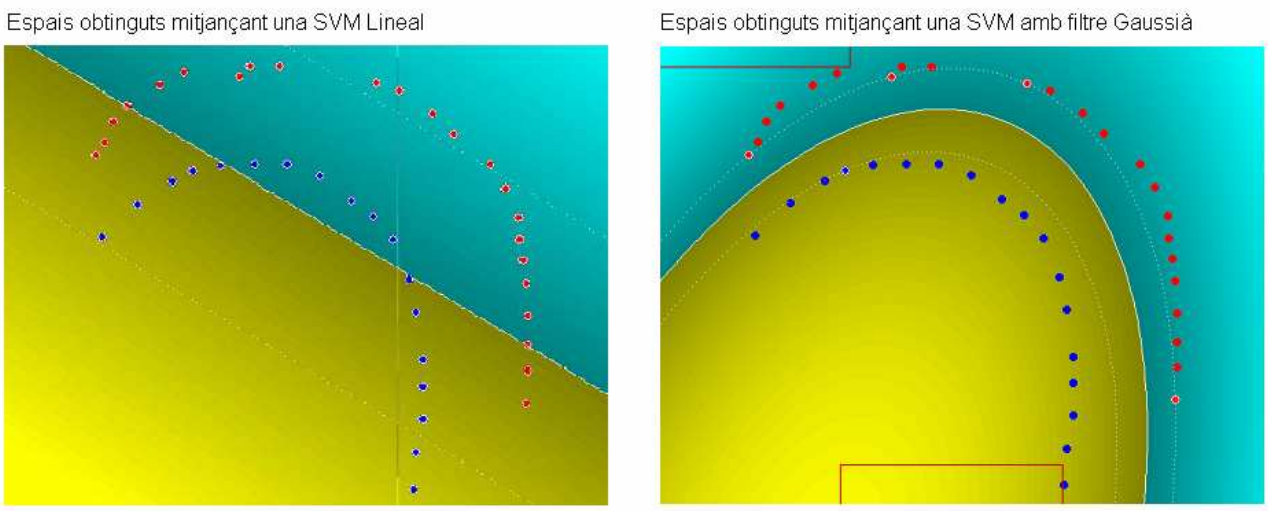
85. Alguns núvols de vectors es poden separar mijançant diverses línies rectes.

Una altra manera d'aconseguir una equació que separi dos grups és mapejant-los mitjançant un "kernel" o una funció. Amb aquest mètode les SVM el que fan és redistribuir els punts en una imatge de la representació de tal manera que la seva separació sigui més senzilla mitjançant una línea recta. Un cop s'ha determinat la recta que separa els dos conjunts en la imatge s'inverteix el procés per tal de representar la recta que s'ha obtingut en la antiimatge, la antiimatge de la recta és la funció que separarà els dos tipus de vectors.

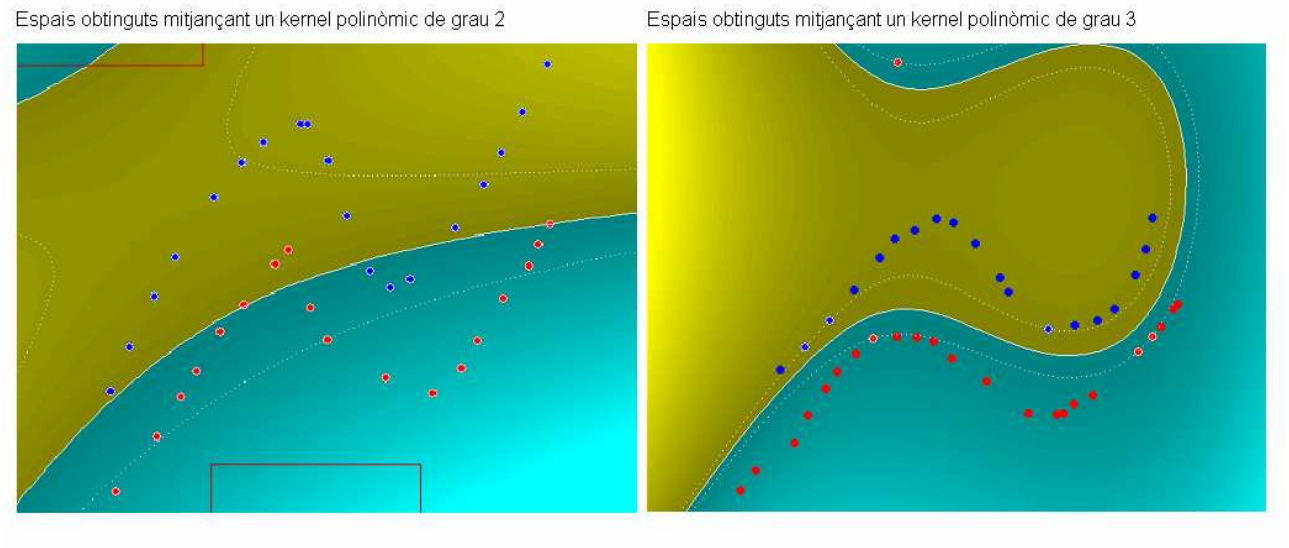
Existeixen molts tipus de kernel diferents: exponencials, polinòmics, circulars, sinusoïdals, gaussians, etc. Les SVM van aplicant diferents modificacions d'un determinat kernel, com per exemple el grau del polinomi, fins que obtenen una bona imatge dels punts.



86. Exemple de kernel



87. Exemples de diferents models obtinguts mijançant diferents kernels [13]



88. Exemples de diferents models obtinguts mijançant diferents kernels[13]

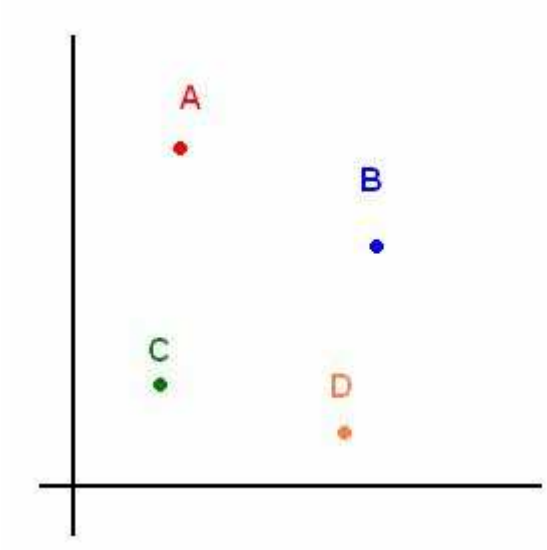
Tal i com es pot observar en les anterior figures es pot comprovar que, si s'utilitza el kernel adequat, es poden obtenir resultats bastant satisfactoris mitjançant les svm simples.

6.2.2 SVM MultiClasse: 1 contra 1

Tal i com el seu nom indica, les SVM Multiclasse son capaces de distingir més de 2 tipologies de vectors dins d'un espai multidimensional. La filosofia de reconeixement d'aquestes es basa en el sistema "Most voted"[13], el més votat. Les MSVM analitzen les diferents classes per separat i creen n-1 x n-2 models on n és el nombre de classes de què disposa, en lloc de crea un model que identifiqui directament les zones que pertanyen als tipus de vectors ho fa individualment. Si en un espai vectorial tenim les classes de

vectors 1, 2 i 3 una MSVM crearà 6 models diferents: 1 contra 2, 1 contra 3 i 2 contra 3

L'algorisme, que ja sap prèviament amb quin nombre de classes està tractant, realitza les comparacions de classes una contra una i n'emmagatzema les sortides. Posteriorment considera que aquell determinat punt correspon al resultat que més vegades s'ha repetit. Per exemple, si suposem que tenim les calsses A, B, C i D i que cada un d'ells representa un color diferent la SVM funcionaria de la següent manera.



	A	B	C	D
Vermell vs Blau	Vl.	Bl.	Bl.	Vl.
Vermell vs Verd	Vl.	Vd.	Vd.	Vl.
Vermell vs Taronja	Vl	Vl.	Vl.	Tr.
Blau vs Verd	Bl.	Bl.	Vd.	Vd.
Blau vs Taronja	Tr.	Bl.	Bl.	Tr.
Verd vs Taronja	Tr.	Vd.	Vd.	Tr.

Vl = Vermell Vd = Verd Bl. = Blau Tr = Taronja

89. Exemple del funcionament d'una SVM Multiclasse

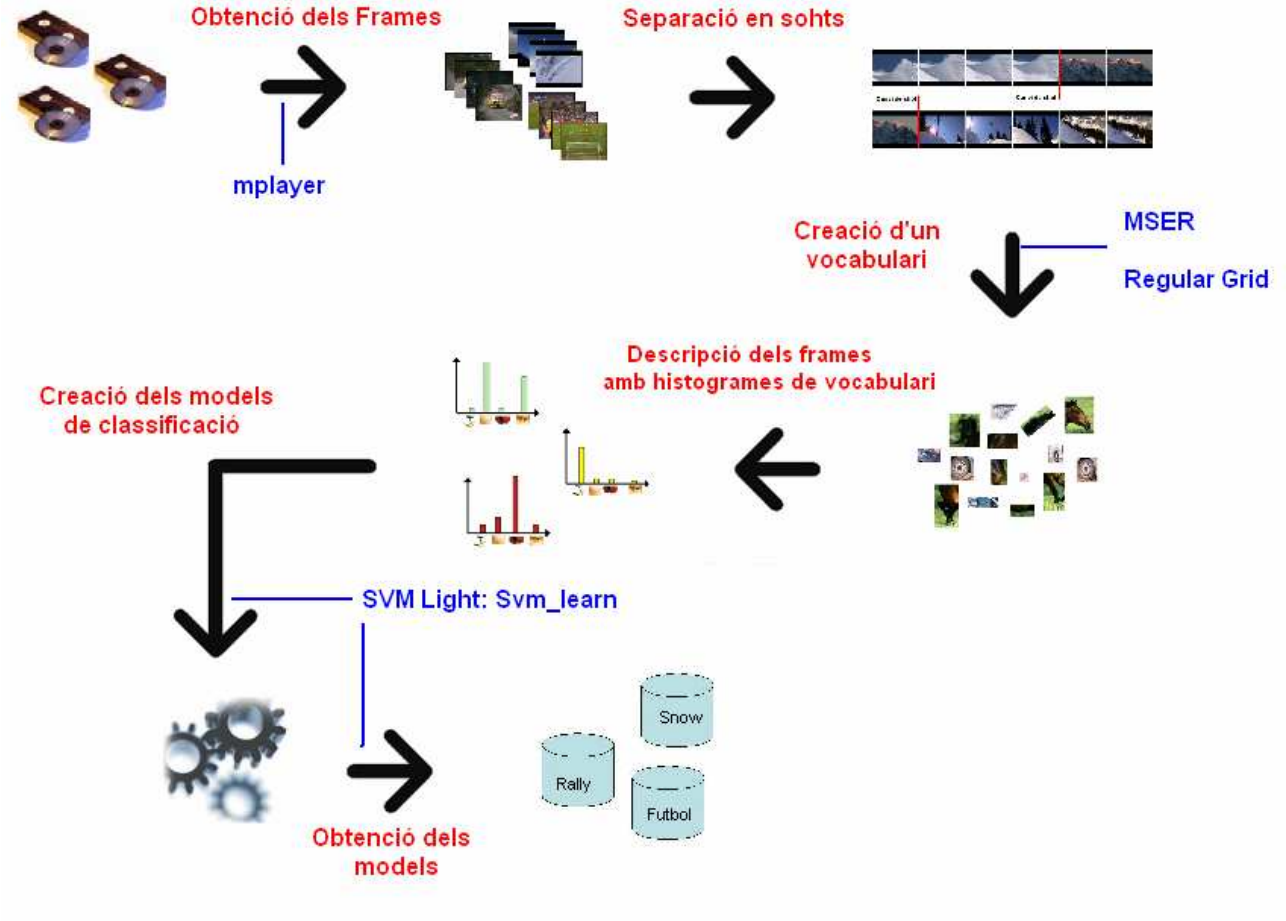
La SVM diria que el punt A és vermell ja que és el resultat que més s'ha repetit, el B blau i així successivament.

Aquest mètode, però, és molt sensible als errors i per això és molt menys precís que les SVM lineals, si per exemple en la comparació vermell contra taronja del punt D hagués donat com a resultat Vermell el sistema "most voted" hagués determinat que el punt D pertanyia a la classe vermell.

6.3 Implementació del model mitjançant SVMs

Després d'haver realitzat els histogrames de les imatges extretes dels vídeos disposàvem de 1200 histogrames classificats en 6 categories diferents. Com en el capítol anterior ja s'ha explicat, cada histograma està compost de 300 enters diferents, per tant cada histograma és un vector que defineix una imatge. Per tal de poder identificar futurs frames obtinguts de nous vídeos era necessari crear un model que ens permetés relacionar els histogrames de les noves imatges amb els histogrames ja existents o

descartar-los si no pertanyen a cap categoria coneguda.



90. Creació del model

Per crear aquest model vàrem decidir utilitzar una mescla entre els SVM lineals i els multiclasse ja que, tot i que aquests últims són molt sensibles als errors, disposem de sis tipologies de vectors diferents. Per tal d'intentar reduir l'error resultant de la creació dels models vàrem decidir realitzar una petita modificació al sistema "most voted". En lloc d'enfrontar els diferents models individualment vàrem decidir crear un sistema "un contra tots".

6.3 SVM Multiclasse: Un contra tots

El sistema "most voted" no ens semblava un mètode adient ja que, dins del marc en el que treballàvem,retornaria almenys 15 resultats erronis cada cop que hi introduíssim un vector histograma. A més a més requeria crear un gran nombre de models (6 classes, models = n-1 x n-2 = 20): Bàsquet contra Btt, Bàsquet contra Futbol, Bàsquet contra Snowboard, etc.

En lloc d'això vàrem creure que el més convenient per assolir els nostres objectius era enfrontar una categoria envers els histogrames resultants. D'aquesta manera podríem obtenir uns resultats bastant més clars i entenedors. A més a més, mitjançant aquesta filosofia obríem la possibilitat a descartar tipologies que no havien estat classificades prèviament, cosa que no succeix amb el "most voted".

No obstant, érem conscients que amb aquest sistema també existia la possibilitat d'obtenir falsos resultats tal i com es pot comprovar en el següent exemple:

Entrant el vector "v" que pertany a la classe futbol obtenim els següents resultats:

Bàsquet contra resta d'imatges = resta.

Btt contra resta d'imatges = resta.

Futbol contra resta d'imatges = Futbol.

Rally contra resta d'imatges = Rally.

Automobilisme de circuit contra resta d'imatges = resta.

Snowboard contra resta d'imatges = resta.

Com es pot observar obtenim dos resultats que es contradiuen. Per resoldre aquest tipus de problemes vàrem pensar en utilitzar una SVM simple ja que només caldria discernir entres dos esports. Al reduir-ne el nombre de candidats es facilita considerablement la feina a la SVM ja que hi ha menys punts que dificultin el mapejat, per tant la solució és més fiable; així doncs també vàrem crear un model de SVM per a totes les possibles combinacions d'un resultat doble (15 en total):

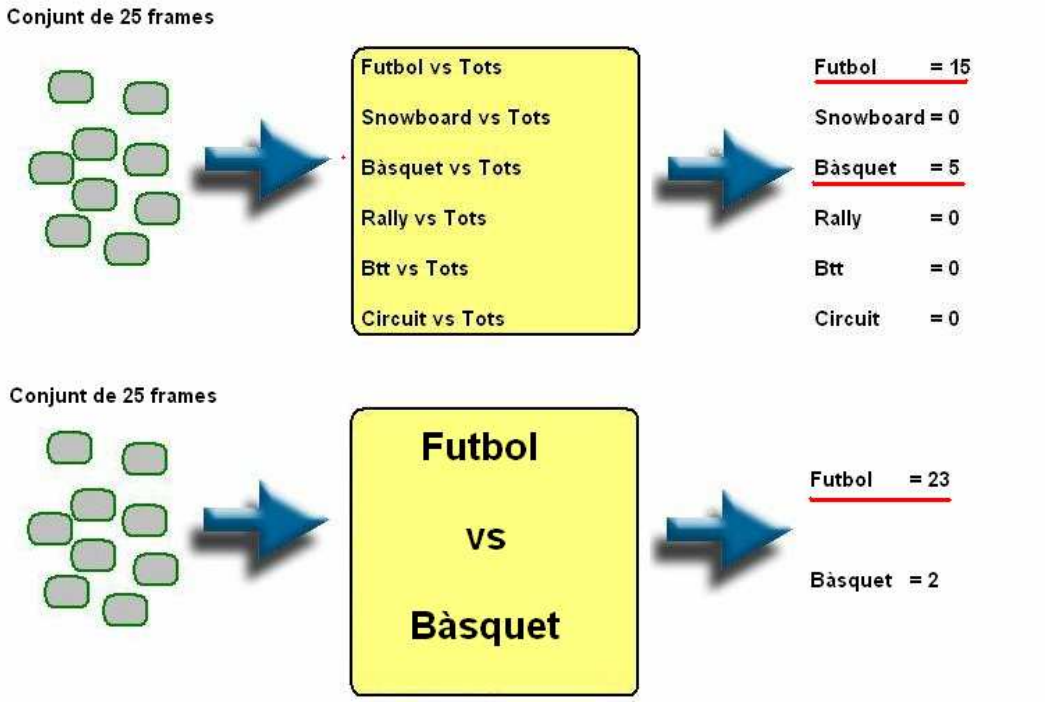
Bàsquet vs Futbol	Bàsquet v S.Board	Bàsquet vs Rally	Bàsquet vs Circuit	Bàsquet vs Btt
Futbol vs S.Board	Futbol vs. Rally	Futbol vs. Circuit	Futbol vs Btt	S.Board vs. Rally
S.Board vs Circuit	S.Board vs Btt	Rally vs Btt	Rally vs Circuit	Circuit vs Btt

En les següents figures es pot observar com funcionaria el nostre classificador per dos possibles entrades de 25 frames d'un vídeo de futbol:

A la primera observem com el detector detecta 15 de les 25 imatges com a imatges de la classe futbol mentre que a les deu restants no els hi assigna cap categoria, això fa que no s'hagi de recórrer a un segon model de classificació per decidir que la classe és de la tipologia futbol.

91. Exemple del funcionament d'una SVM "un contra tots"

En la següent figura observem com el classificador també identifica 15 de les 25 imatges com a futbol, no obstant, també n'identifica 5 com a Bàsquet. Això fa que calgui tornar a analitzar el conjunt d'imatges, aquest cop mitjançant el model de comparació entre els dos esports que han determinat els primers classificadors. Els resultats del segon classificador determinen que 23 de les 25 imatges pertanyen al tipus futbol mentre que només 2 ho fan a la classe Bàsquet. Aquest resultats, realitzant un símil esportiu, donen com la classe Futbol com a guanyador del partit.



92. Exemple del funcionament d'una SVM "un contra tots" (2)

6.3.2 Les SVM Light

Des que es van idear les primeres SVM se n'han desenvolupat un gran nombre. Totes es solen basar en els mateixos algorismes però solen canviar els filtres que s'utilitzen per mapejar els vectors. Per crear els nostres models vàrem utilitzar, aconsellats pels tutors del projecte, una sèrie d'utilitats que es coneixen com a "SVMLight"[17]. Varen ser desenvolupades en un projecte de col·laboració entre les universitats de Cornell (Nova York, EUA.) i de Dortmund (Alemanya) liderat per el doctor Thorsten Joachims [18]. Dins d'aquestes utilitats es troben els dos programes anomenats "Svm_learn" i "Svm_classify" que serveixen per obtenir el model de distinció entre els dos conjunts de vectors i, a través d'un model, assignar un nou vector a un dels dos conjunts. Les SVM Light s'engloben dins la categoria de SVM simples ja que només poden crear models capaços de distingir dos tipus de vectors.

El programa Svm learn és el que s'encarrega de crear el model que defineix els dos tipus de vectors. Cal executar-lo des de la línea de comandes de Linux o bé cridar-lo des del Matlab a través de la següent comanda:

./svm_learn exemples model

El paràmetre exemples és l'adreça d'un fitxer de text que conté els diferents vectors que s'utilitzaran per crear el model. Cada línea d'aquest fitxer correspon a un vector; la línea comença amb una etiqueta (-1 o 1) que indica a quina classe de vector pertany i el segueixen les components del vector enumerades segons la seva posició:

Fitxer exemples:

1	1:0.15	2:0.75	3:0.83	4:0.1
1	1:0.32	2:0.32	3:0.12	4:0.45
-1	1:3.43	2:3.34	3:7.0	4:1.34
-1	1:32	2:322	3:43	4:11

El paràmetre model és l'adreça on es guardarà el fitxer que servirà per determinar si un vector pertany a l'un o l'altre grup i que s'utilitzarà en el programa Svm Classify.

L'SVM Classify també s'executà des de la línea de comandes.

./svm_classify fitVectors model fitSortida

El primer paràmetre ha de contenir la direcció del fitxer on hi ha els vector que es volen classificar. Aquest fitxer mantindrà el mateix format dels exemples però el valor de l'etiqueta no tindrà cap tipus d'influència en el resultat, per aquest motiu s'acostuma a

assignar-hi l'enter 0. El segon paràmetre, model, correspon a l'adreça de l'arxiu de classificació resultant de L'SVM_learn. Finalment el paràmetre fitSortida contindrà la direcció de l'arxiu on es guardaran els resultats de la classificació. El fitxer fitSortida contindrà un determinat nombre de valors decimals, cada valor correspondrà a un vector diferent. El signe, positiu o negatiu, de cada un d'aquests nombres identificarà a quina classe pertany. En el següent exemple es pot observar la relació entres dos possibles fitxers de vectors i fitxers de sortida:

Fitxer de vectors	*Fitxer de Sortida*	*Classe de vector*
0 1:21 2:43 3:34 4:11	-0.93	A
0 1:01 2:02 3:34 4:43	0.23	B
0 1:43 2:94 3:03 4:54	0.01	B
0 1:28 2:01 3:30 4:48	1.1	B
0 1:15 2:63 3:23 4:12	-0.12	A
0 1:45 2:05 3:06 4:07	-2.11	A

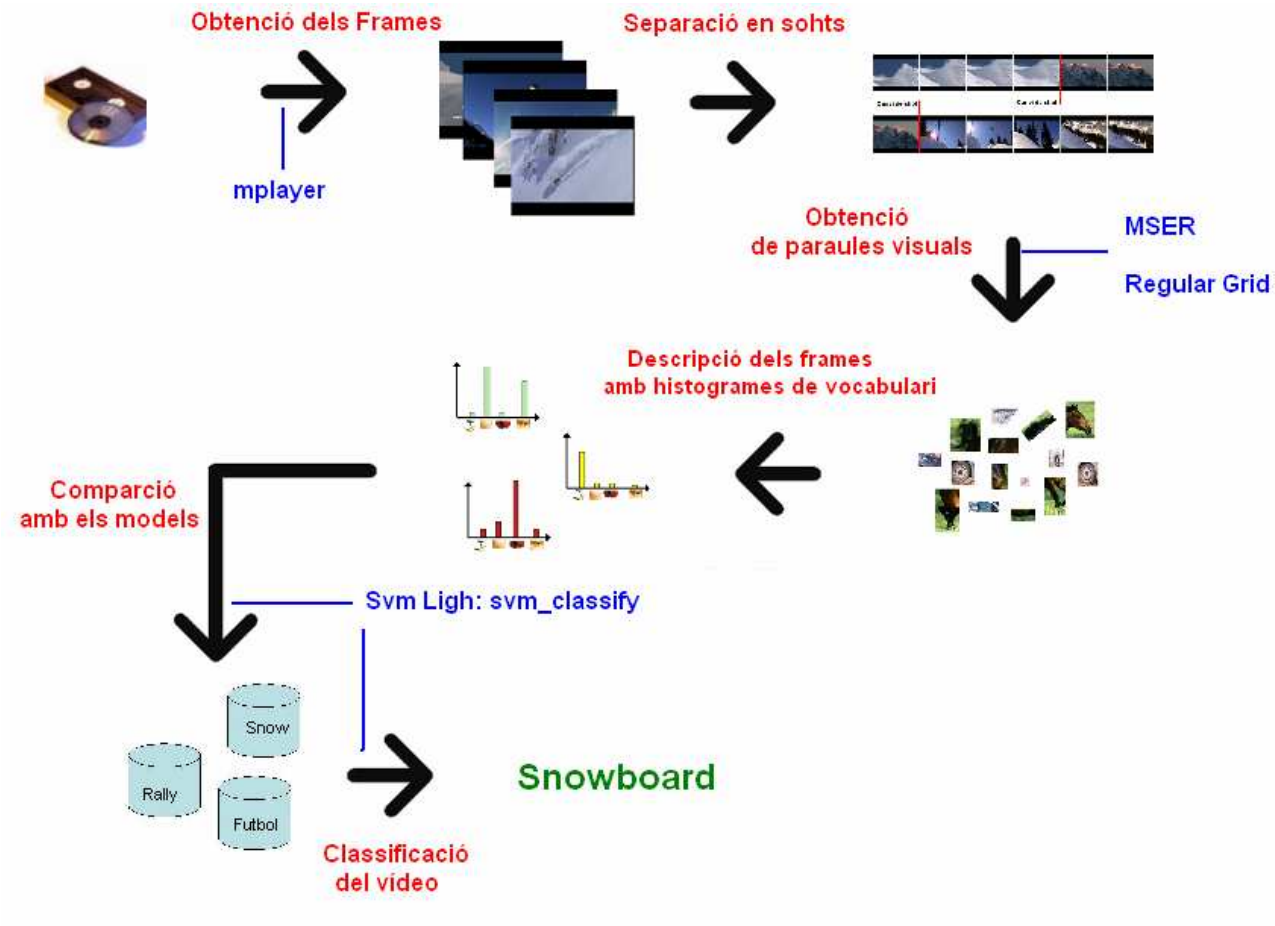
6.4 Classificació de vídeos

Per tal de poder classificar un vídeo cal seguir els següents passos:

1. Obtenció dels frames (2.1)
2. Separació en shots (només si el vídeo és molt llarg) (3)
3. Obtenció de les paraules visuals (4.3.3)
4. Creació d'histogrames de les paraules visuals (4.3.7)
5. Comparar-lo amb els diferents models aconseguits mitjançant una SVM

Els 4 primers passos ja han estat explicats en apartats anteriors, per aquest motiu en aquesta part del projecte ens centrarem en la implementació del classificador.

Per tal de poder classificar un vídeo a partir dels histogrames resultants primer era necessari emmagatzemar-los tots dins d'un mateix fitxer anomenat fitxer de Test. Per aquest vàrem implementar una funció de Matlab anomenada CreaTest a la qual se li indicava la carpeta en la que es trobaven els histogrames i el nom del fitxer on es guardarien tots els vectors. Per exemple: CreaTest('./Snowboard/','TestSnow');



93. Procés de classificació d'un vídeo

Vàrem implementar una altra funció a la qual se l'hi indicava el nom del fitxer Test i el comparava, mitjançant les svm light, amb els models obtinguts anteriorment. Utilitzant el sistema "Un contra tots" explicat anteriorment es determinava a quina classe pertanyia el fitxer Test i, per extensió, quina era la categoria del vídeo (6.3.1).

6.3 Resultats

Per a comprovar el funcionament del classificador vàrem decidir realitzar tres proves diferents.

En la primera vàrem utilitzar 10 frames de diferents shots de cada vídeo font (excepte de bàsquet, en que en vàrem agafar 20) per a crear el models de classificació. Per a la comprovació del seu funcionament vàrem utilitzar 30 frames de cada una de les diferents categories que no s'haguessin emprat ni per a crear el vocabulari ni per elaborar el model.

En el segon experiment vàrem utilitzar el mateix model que en la primera prova però vàrem decidir utilitzar els Vídeos Test per a comprovar el funcionament. D'aquesta

manera volíem comprovar el funcionament del classificador quan s'intentaven classificar fragments de vídeos que compartien categoria amb els vídeos font però que no compartien origen.

En el tercer experiment vàrem decidir ampliar els models utilitzats per a la classificació afegint-hi els frames d'un dels vídeos test de cada una de les categories. D'aquesta manera esperàvem augmentar la capacitat de detecció dels models.

6.3.1 Experiment 1

Vídeos utilitzats per a la creació del model:

Futbol1.avi, 10 frames	Futbol2.avi, 10 frames
Basquet.avi, 20 frames	
Circuit1.avi, 10 frames	Circuit2.avi, 10 frames
Rally1.avi, 10 frames	Rally2.avi, 10 frames
Snow1.avi, 10 frames	Snow2.avi, 10 frames
Btt1.avi, 10 frames	Btt2.avi, 10 frames

Vídeos utilitzats per a la creació dels arxius de test:

Per a la creació dels arxius de test s'han utilitzat frames dels anteriors vídeos que no s'haguessin utilitzat ni per a la creació del model ni per a la creació del vocabulari. Els arxius s'han anomenat testBasquet, testBtt, testFutbol, testCircuit, testRally i testFutbol

Resultats

Tal i com s'esperava el classificador ha estat capaç d'identificar tots els arxius de test de forma correcta. Aquest fet era relativament previsible ja que els frames que s'han classificat formaven part dels vídeos que s'han utilitzat per la creació del model de classificació. Així doncs, l'encert ha estat del 100%.

6.3.2 Experiment 2

Vídeos utilitzats per a la classificació del model:

En aquest experiment s'ha utilitzat el mateix model que en la prova anterior.

Vídeos utilitzats per a la creació dels arxius de Test

Per a la creació dels arxius de test s'han utilitzat els 30 primers frames de cada un

dels Vídeos de Test. Els diferents fitxers de test s'han identificat amb la paraula "Test" seguida del nom del vídeo (per exemple testBasquet1)

Resultats

En la següent taula es poden observar els resultats obtinguts. Per a facilitar la seva comprensió els vídeos identificats correctament s'han escrit amb verd mentre que els que s'han classificat incorrectament es troben en vermell.

Test	Classificat	Test	Classificat
testBasquet1	**Bàsquet**	testCircuit1	**Circuit**
testBasquet2	**Bàsquet**	testCircuit2	**Futbol**
testBasquet3	**Bàsquet**	testCircuit3	**Futbol**
testBasquet4	**Bàsquet**	testCircuit4	**Futbol**
testBasquet5	**Bàsquet**	testCircuit5	**Futbol**
testBtt1	**Bàsquet**	testFutbol1	**Bàsquet**
testBtt2	**Bàsquet**	testFutbol2	**Bàsquet**
testBtt3	**Bàsquet**	testFutbol3	**Bàsquet**
testBtt4	**Bàsquet**	testFutbol4	**Bàsquet**
testBtt5	**Bàsquet**	testFutbol5	**Bàsquet**
testRally1	**Futbol**	testSnow1	**Futbol**
testRally2	**Futbol**	testSnow2	**Futbol**
testRally3	**Futbol**	testSnow3	**Circuit**
testRally4	**Futbol**	testSnow4	**Circuit**
testRally5	**Futbol**	testSnow5	**Snow**

Observant els resultats es pot comprovar clarament com el classificador no ha pogut identificar nous vídeos, només un 23% dels resultats han estat correctes.

La justificació d'aquesta situació es podria trobar en el fet que els Vídeos de Test que vàrem seleccionar eren bastant diferents dels Vídeos Font, per tant podria ser que les mostres amb les que es va confeccionar el model no fossin prou àmplies. Si volem entrenar una SVM per a identificar automòbils no es podem limitar a entrenar-la només amb fotos laterals de diferents cotxes ja que aquesta, després, no podrà identificar com a cotxe una foto del cul d'un automòbil. Vàrem suposar, doncs, que aquests problemes d'identificació es podrien resoldre ampliant les fonts dels vídeos per a la creació dels

models.

6.6.3 Experiment 3

Vídeos utilitzats per a la classificació del model:

Després d'observar el baix percentatge d'encert de l'anterior model de classificació vàrem decidir ampliar les mostres utilitzades per a la creació d'aquest. A part dels vídeos ja utilitzats anteriorment vàrem decidir afegir-hi els primers 30 frames dels següents vídeos de Test:

Bàsquet1.avi	Futbol1.avi	Circuit1.avi
Btt1.avi	Rally1.avi	Snowboard1.avi

Vídeos utilitzats per a la creació dels arxius de Test

En aquest apartat s'han utilitzat els mateixos fitxers de test que en l'experiment 2.

Resultats

Després d'haver ampliat la base de dades esperàvem solucionar parcialment els resultats obtinguts en l'experiment 2. En la següent taula es poden observar els resultats obtinguts amb els nous models de classifcació.

Test	Classificat	Test	Classificat
testBasquet1	**Bàsquet**	testCircuit1	**Rally**
testBasquet2	**Bàsquet**	testCircuit2	**Circuit**
testBasquet3	**Bàsquet**	testCircuit3	**Circuit**
testBasquet4	**Bàsquet**	testCircuit4	**Circuit**
testBasquet5	**Bàsquet**	testCircuit5	**Circuit**
testBtt1	**Btt**	testFutbol1	**Futbol**
testBtt2	**Bàsquet**	testFutbol2	**Desconegut**
testBtt3	**Btt**	testFutbol3	**Desconegut**
testBtt4	**Rally**	testFutbol4	**Futbol**
testBtt5	**Btt**	testFutbol5	**Futbol**
testRally1	**Rally**	testSnow1	**Snow**
testRally2	**Rally**	testSnow2	**Snow**

testRally3	**Rally**	testSnow3	**Snow**
testRally4	**Rally**	testSnow4	**Snow**
testRally5	**Rally**	testSnow5	**Snow**

Els resultats obtinguts un cop s'ha ampliat la base de dades són molt satisfactoris. S'han aconseguit classificar correctament 25 dels 30 vídeos de test (un 83%). Així doncs podem relacionar clarament la falta d'encert de l'experiment2 amb un dèficit d'imatges utilitzades en la creació del model ja que al augmentar aquesta mostra els resultats han millorat considerablement.

Els errors que s'han produït durant la classificació són, en part, comprensibles ja que quan no ha reconegut el futbol com a tal tampoc l'ha identificat com a cap classe. I les categories dels arxius testBtt4 i testCircuit1, que s'han classificat com a Rally, tenen certa relació amb aquesta categoria. Les competicions de BTT habitualment es practiquen en circuits de terra o al mig del bosc, com en els Rallys de terra. I tant a la categoria Circuit com a Rally hi apareixen cotxes.

No obstant són sorprenents algunes confusions que es produeixen en la classificació: Per exemple en el fitxer testBtt2 és classificat com a Bàsquet, una categoria que a priori sembla que no té cap tipus de relació amb la Bicicleta de Muntanya.

7. Conclusions:

7.1 Conclusions

Un cop acabat el projecte podríem concloure que s'ha assolit el principal objectiu d'aquest, desenvolupar un mètode capaç de classificar vídeos de diferents disciplines esportives, ja que els resultats obtinguts han estat bastant bons. A més a més, podríem extreure les següents conclusions:

- Els detectors de regions ens poden servir per a obtenir paraules dels elements principals d'una imatge (ciclistes, arbres, cotxes, etc.) i descriure'ls, tal i com s'ha pogut observar en l'apartat 4.4. No obstant, no són tan útils si també es necessita obtenir informació sobre altres elements de la imatge com ara el fons d'aquesta o zones que no estan definides amb nitidesa (Per exemple paraules que defineixin la gespa del camp de futbol, o el cel en un vídeo d'snowboard). Aquest problema es pot solucionar utilitzant un Regular Grid ja que obté paraules de totes les parts d'una imatge.

- El sistema d'Image Retrieval té un funcionament lògic ja que els conjunts d'imatges amb els que s'obtenen pitjors resultats són part de classes de vídeo molt semblants i que comparteixen trets en comú com per exemple rally i competicions de circuit.

- El mateix succeeix amb el classificador de vídeos, les classes on es produeixen menys encerts són classes en les que existeix certa relació com el Rally i les competicions de Circuit.

- Aquest sistema de classificació o recuperació de vídeos i imatges no es podria utilitzar en temps real ja que els seus temps d'execució són molt alts. Per exemple, la classificació d'un vídeo amb una durada d'aproximadament mig minut tarda aproximadament uns 50 minuts degut al procés d'extracció i anàlisis dels diferents frames. No obstant aquest procés es podria realitzar offline.

- Els processos previs a la creació de les SVM que permeten la classificació de les diferents classes suposen un temps d'execució molt alt. Crear un model capaç de classificar 6 classes diferents ha necessitat aproximadament una setmana de càlculs: D'aquests 7 dies 4 s'han dedicat a extreure els frames i trobar-ne les paraules que hi apareixien; 2 a la creació del vocabulari i aproximadament 1 en la generació dels histogrames. Altra vegada tot aquest procés es podria realitzar "off-line" i emmagatzemar-ne els resultats en una base de dades, tal i com fan els cercadors google i yahoo.

7.2 Treballs Futurs

Un cop acabat el projecte se'ns acudeixen diferents tasques que es podrien realitzar en un futur per tal de millorar el seu funcionament i el seu rendiment.

- Ampliació de les classes:

La primera, i la més obvia, d'aquestes tasques és l'ampliació del nombre de classes amb les que treballa el projecte. Seria interessant comprovar el funcionament del classificador amb un nombre més elevat de classes. L'ampliació de les categories més enllà de l'àmbit esportiu també seria un bon punt a treballar en un futur.

- Millora del temps d'execució:

Part de la lentitud d'execució d'aquest projecte es deu a que aquest es desenvolupa mitjançant scripts de Matlab. El fet que aquests scripts necessitin executar-se juntament amb el Matlab fa que la seva execució sigui lenta i pesada. Seria, doncs, una bona idea intentar implementar els diferents scripts en un llenguatge més ràpid i àgil com per exemple c++

Altres causants de la lentitud d'aquests processos són els programes externs utilitzats on es mostren moltes sortides de pantalla, com per exemple els detectors de regions o els descriptors d'imatge SIFT. Aconseguir versions més ràpides d'aquests executables on no es mostressin tants missatges per pantalla augmentaria considerablement l'eficiència dels mètodes.

- Creació d'una base de dades;

Per tal de millorar l'accés a les dades i a les imatge seria convenient emmagatzemar tota la informació en una base de dades. Posteriorment caldria indexar-la per així millorar

l’eficàcia i el temps d’accés a aquesta.

- Millora de les paraules visuals:

Els vocabularis que s’han utilitzat en aquest projecte han utilitzat com a paraules visuals les diferents regions que hem trobat en una imatge, ja fos mitjançant un detector de regions o un Regular Grid. Seria convenient afegir nova informació a les paraules visuals, com per exemple la seva situació en la imatge, per tal d’experimentar si influeixen en els resultats.

- Classificació dels frames mitjançant altres aspectes:

Els frames obtinguts dels diferents vídeos s’han descrit mitjançant histogrames de paraules visuals i s’han classificat en base a aquests. Un possible treball futur seria l’experimentació amb diferents formes de descripció de la imatge com per exemple segons les formes que hi apareixein o la direccionalitat dels gradients de la imatge.

- Experimentació amb noves Suport Vector Machines

Actualment existeixen un gran nombre de SVM. En aquest projecte s’ha utilitzat el conjunt d’utilitats anomenat “SVM Light”. Seria interessant observar el comportament del classificador si en lloc de les SVM Light utilitzessim una màquina classificadora de vectors diferent.

8. Bibliografia

8.1 Referències

1. http://pserv.udg.edu/Portal/Uploads/4103862/Tema3.pdf Apunts de Visió per Computador Tema3, ETIS 2006-2007 *Lladó, Xavier* Visitat estiu 2007

2. http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/vibes_ijcv200 4.pdf Comparision of Affine-Invariant Local Detectors and Descriptors, *Mikolajczyk,Krystian; Schmid, Cordelia* Visitat primavera 2007

3. http://en.wikipedia.org/wiki/Laplace_operator Visitat estiu 2007

4. http://en.wikipedia.org/wiki/gradient Visitat primavera 2007

5. http://en.wikipedia.org/wiki/gaussian_filter Visitat estiu 2007

6. http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf Distinctive Image Features from Sacale-Invariant Keypoints, *Lowe, David G.* Visitat primavera 2007

7. http://pserv.udg.edu/Portal/Uploads/4103862/Tema4-Segmentacio.pdf Visió per Computador Tema 4, Etis 2006-2007 *Lladó, Xavier*

8. http://en.wikipedia.org/wiki/K-means_algorithm Visitat primavera 2007

9. http://es.wikipedia.org/wiki/lloyd%27s_algorism Visitat estiu 2007

10. http://www.mathworks.com/support/functions/alpha_list.html

The MathWorks Support function list, Visitat primavera 2007

11. http://es.wikipedia.org/wiki/Ordenamiento_de_burbuja Visitat primavera 2007

12. http://en.wikipedia.org/wiki/Information_retrieval Visitat estiu 2007

13. http://www.cvc.uab.es/~jordi/ Classification: From NN to Adaboost *Vitria,Jordi,* Visitat estiu 2007

14. http://www.cs.ucla.edu/~eeskin/papers/mismatch-nips02.pdf Mismatch String Kernels

for SVM Protein Classification *Leslie, Christina; Eskin, Eleazar; Jason, Weston; Stafford, William,* Visitat estiu 2007

15*. V. N. Vapnik*. Statistical Learning Theory. Springer, 1998.

16. http://en.wikipedia.org/wiki/Vapnik, Visitat estiu 2007

17. http://svmlight.joachims.org/ Visitat estiu 2007

18. http://www.cs.cornell.edu/People/tj/ Visitat estiu 2007

8.2 Altres fonts consultades

http://courses.ece.uiuc.edu/ece598/ffl/paper_presentations/JilinTu_evaluations.pdf A performance evaluation of local detector and descriptors, *Mikolajczyk,Krystian; Schmid, Cordelia,* Visitat primavera 2007

http://eia.udg.es/~aboschr/Publicacions/ivc06b.pdf A review: Which is the best way to organize/classify images by content?, *Bosch, Anna; Muñoz, Xavier; Martí, Robert;* Visitat estiu 2007

http://eia.udg.es/~aboschr/Publicacions/pami.pdf Scene classification using a hybrid generative/discriminative approach, *Bosch, Anna; Zisserman, Andrew; Muñoz, Xavier* Visitat estiu 2007*.*

http://eia.udg.es/~aboschr/Publicacions/trecvid06.pdf Oxford TRECVID 2006 – Notebook paper *Philbin, James; Bosch, Anna; Chum, Ondrei; Geusebroek, Jan-Mark; Sivic, Josef; Zisserman, Andres.* Visitat estiu 2007

http://eia.udg.es/~aboschr/Publicacions/civr07.pdf Representing Shape with a spatial pyramid Kernel, *Bosch, Anna; Ziesserman, Andrew; Muñoz, Xavier*. Visitat estiu 2007

http://www2.tuebingen.mpg.de/agbs/lcvii/wiki/lect15.pdf Parts-Based Categorization. *Franz, M. O.* Visitat estiu 2007

http://users.ecs.soton.ac.uk/srg/publications/pdf/SVM.pdf Support Vector Machines for Classification and Regression *R. Gunn, Steve* Visitat estiu 2007

Annex: Toolbox de Matlab

En aquest annex es descriuen algunes de les diferents funcions que s'han implementat per assolir els objectius del projecte. Són un conjunt d'eines, una toolbox, que té com a funció principal facilitar el procés de classficació d'un vídeo.

Funcions de descripció

Def_Sifts

DEF_sifts(carpetaIm,extIm,carpetaReg,extReg,carpetaDesti,IMMAX)

INPUTS:

String carpetaReg = carpeta d'origen

String extReg = extensio de les regions (mser, har, support...)

Enter IMMAX = nombre d'iamtges maxim

String carpetaIm = carpeta on hi ha les imatges

String extIm = extensio de les imatges .png, .jpg... etc.

String CarpetaDesti = carpeta on es guardaran els sifts.

FUNCIO:

Extreu les descripcions SIFT dels fitxers de regions de la carpeta "carpetaReg" i les guarda a la Carpeta Desti

def_Extreu

DEF_extreu(origen,desti,qualitat,frames)

INPUT

String origen = vídeo d'origen

String carpeta = carpeta on es guardaran els videos

Enter qualitat = qualitat del vídeo de 1 a 10

Enter frames = quantiat per la que es dividiran els frames (exemple: 25 = 1 cada segon, 50 = 1 cada 2 segons)

FUNCO

Extreu els frames d'un video

def_supportRegion

DEF_supportRegion(directoriImatges,directoriDesti)

INPUT

String directoriImatges = directori on es troben les imatges

String directoriDesti = directori on es guarden les imatges

FUNCIO

aplica un regular grid sobre les imatges del directori directoriImatges

regions

regions(imatge)

INPUT

string imatge = imatge de la que es volen extreure regions

OUTPUT

extreu les regions de la imatge amb Harris i MSER

Funcions de càlcul

Def_Semblants

def_semblants(nombre,imatge,taulaArxius)

INPUTS:

Enter Nombre = nombre d'imatges que retornarà,

String Imatge = Imatge imatge nom de la imatge font

Taula Strings taulaArxius = taula amb la llsita de tots els arxsius a comparar

OUTPUTS:

Taula Strings xprimers = Taula amb lesX imatges més semblants a imatge

Taula Enters xdistancies = Taula amb les distancies a les X imatges més semblants

FUNCIO

Retorna les X imatges mes semblants a la imatge "imatge".

Def_arees

Def_arees(directori,extensio)

INPUTS:

String directori = Directori on es troben els fitxers que guarden els encerts (.totencerts)

String extensio = extensio dels fitxers dels encerts (generalment .totencerts)

OUTPUTS:

Taula Enters TaulaArea = Àrees d'encerts de les diferents imatges

FUNCIO:

Calcula l'àrea d'encerts dels fitxers d'encerts

Def_bombolla

DEF_bombolla(taulaD,taulaA)

INPUTS:

Taula Enters taulaD: taula on hi ha les distancies d'arxius desordenades

Taula Arxius taulaA: taula on hi ha els fitxers desordenats

OUTPUTS:

Taula Enters tauladist: taula on hi ha les distancies d'arxius ordenades

Taula Arxius taulaarx: taula on hi ha els fitxers ordenats

FUNCIO

Ordena la taula de distancies i darxius en funció de la distancia

def_exits

DEF_exits(stringa,taula)

INPUTS

String stringa = cadena a buscar

Taula strings taula = taula on es buscara la cadena

OUTPUTS

Enter exits = nombre de vegades que apareix stringa en taula

FUNCIO

Contabilitza el nombre de vegades que apareix una cadena semblant a Stringa en una taula

def_histogramesC

DEF_histogramesC(carpeta,desti,nombre,matriu);

INPUT

String carpeta = carepta on es troben els arxius i on es deixara l'histograma

Enter nombre = nombre d'imatges maxim

String matriu = vocabulari

FUNCIO

Realitza els histogrames de vocabulari dels arxius sift de la carpeta "carpeta" i els guarda a la carpeta "desti".

def_ordena

DEF_ordena(taulaArxius)

INPUTS

taula string TaulaArxius = Llista amb les imatges

FUNCIO

A partir dels arxius de distancies .tot crea un arxiu amb els "encerts per posicio"

taulaArxius es una taula amb el nom dels histogrames a comprovar

busca les imatges mes semblants de totes les posicions de taula arxius.

guarda en un fitxer .tot.distnum i .tot.distarx les distàncies de cada imatge amb la resta d'imatges

guarda els encerts en un fitxer .totencerts

def_precionRecall

DEF_precisionrecall(matriu)

INPUT

Taula enters matriu = taula que conte els encerts per a cada posicio

OUTPUT

taula reals a = taula que conte el percentatge d'encerts per a posicio

FUNCIO

retorna una taula amb els percentatges d'encerts per posicio

def_Histograma

DEF_histogramaC(sift,taula)

INPUT

string sift = arxiu . sift d'una imatge

taula vectors taula= vocabulari, kmeans

OUTPUT

taula enters histo = histograma del sift

FUNCIO

retorna l'histograma de les característiques d'una imatge. dist eucleriana

histoHSV

histoHSV(imatge)

INPUT

string imatge = direccio on es troba la imatge

OUTPUT

matriu enters histoH = hisgotgrama component H

matriu enters histoS = hisgotgrama component S

matriu enters histoV = hisgotgrama component V
FUNCIO
realitza l'histograma hsv en format 64,64,32

restaR
restaR(a,b)
INPUT
vector a;
vector b;
OUTPUT
float resultat = distancia euclídiana de a a b
FUNCIO
calcula la distancia euclidiana de a a b

escenesp
escenesp(carpeta,imatges,fitxer,limit,extSMax)
INPUT
string carpeta = lloc on hi han les imatges
enter imatges = nombre d'imatges de la carpeta
string fitxer = fitxer de sortida on es guardaran els shots, es sobreescriu o es crea
int limit = llindar de les escenes
string ext = extensio de les imatges
enter SMax = màxim nombre imatges per shot. -1 = maxim (1000000).
FUNCIO
separa les imatges en frames

transforma
transforma(nombre)
INPUT
enter nombre = nombre
OUTPUT
string stringim = nombre en format string de 8 posicions
FUNCIONS
transforma nombre en una string de 8 posicions

def_encerts

def_encerts(matriu)
INPUT
taula enters matriu = taula amb els resultats de svm_classify
OUTPUT
enter n = encerts
FUNCIO
retorna el nombre de vectors amb una etiqueta superior a zero

def_analitza
def_analitza(test)
INPUT
Strig test = direccio del fitxer de vectors del vídeo
OUTPUT
String res = Classe del vídeo
FUNCIO
Analitza els vectors histograma d'un vídeo i n'identifica la classe

Funcions de mostreig

def_mostra
def_mostra(arxiuG,arxiuM)
INPUT
string arxiuM = directori del fitxer totencerts
string arxiuG = directori del fitxer totencerts
OUTPUT
enter areaa = area d'encerts del precision recall
enter areab = area d'encerts del precision recall
FUNCIO
mostra les grafiques precision&recall dels arxius arxiuM i arxiuG

Funcions i programes externs utilitzats

Mplayer

Oxford libraries: Harris-Affine

Oxford libraries: MSER

Oxford libraries: Compute descriptors

Oxford libraries: display features

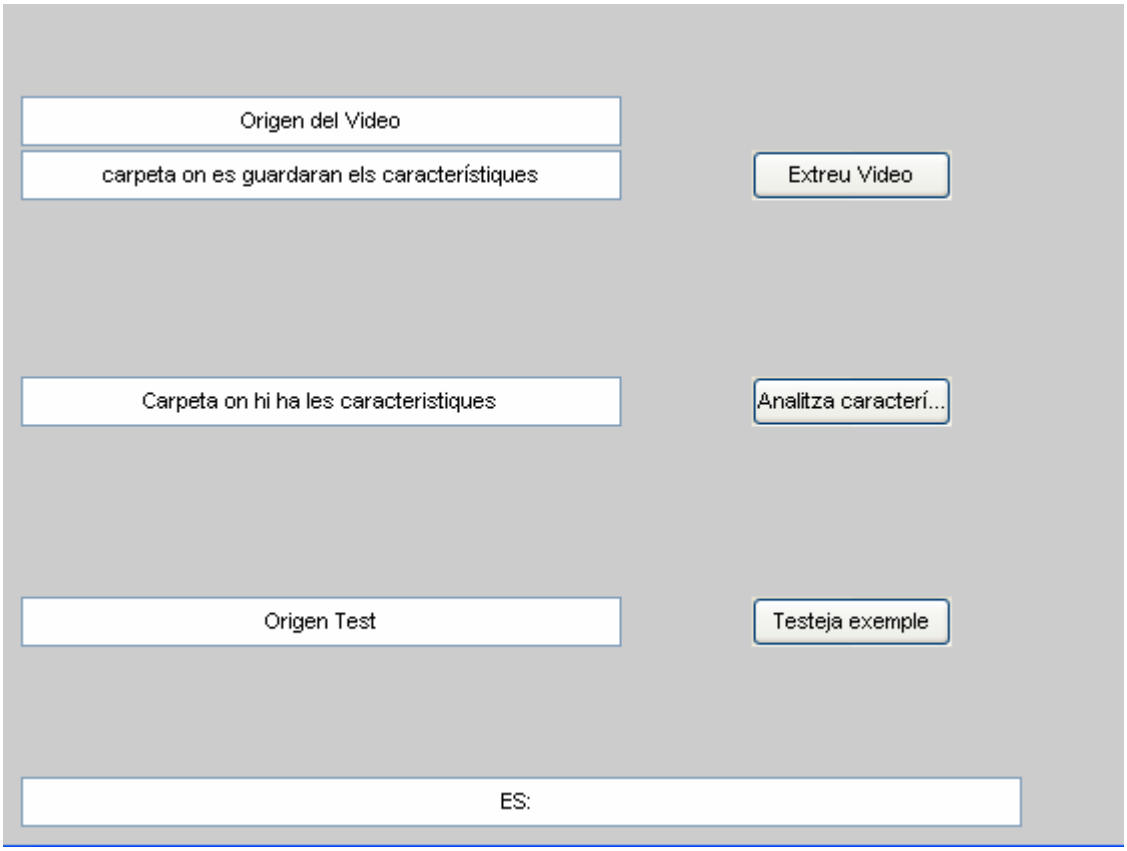
SVM Light: Svm learn

SVM Light: Svm Classify

Interfície gràfica

La interfície gràfica de la Toolbox, que s'invoca mitjançant la comanda Interfície, permet classificar un vídeo de tres formes diferents.

Amb el botó "Extreu Vídeo" es realitza el procés complet de classificació, des de l'extracció dels frames. Amb el botó "Analitza característiques" el Matlab analitza el vídeo a partir dels fitxers .SIFT que es trobin en una determinada carpeta. El botó "Testeja Exemple" ofereix la opció de classificar directament l'arxiu de vectors del vídeo.



Interfície gràfica