Indexació de continguts televisius: Anàlisis de Vídeo

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Jordi Hernàndez Puigdellívol
i dirigit per
Jordi Vitrià
Bellaterra, 10 de Juny de 2007

Universitat Autònoma de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Jordi Vitrià

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Jordi Hernàndez Puigdellívol

I per tal que consti firma la present.

Signat: .....

Bellaterra, ...de.....de 2007

ÍNDEX

INTRODUCCIÓ

1.Introducció

1.1. Motivacions

Una de les principals motivacions que ens ha portat a fer aquest projecte és que ens sembla sorprenent veure com un ordinador pot arribar a aprendre i a fer coses que fa uns anys semblaven de ciència ficció. D'altra banda la televisió ha estat present en les nostres vides des de sempre i creiem que ara esta a punt d'experimentar grans canvis respecte el que estem acostumats a veure.

La visió per computador, juntament amb la intel·ligència artificial, és una de les àrees que més ens atreuen en el camp de la informàtica ja que la visió és un dels fenòmens naturals més interessants i complexos, però a la vegada necessari. Moltes vegades s'ha imitat el comportament de la natura amb gran èxit i encert. N'és un clar exemple la visió dels ratpenats, del quals s'ha copiat el seu funcionament en els sonars dels submarins.
Creiem que relacionant la visió per computador, la intel·ligència artificial i la televisió, podem arribar a assolir grans objectius.

Amb aquest nou futur imminent que li espera a la televisió, l'ús d'aquestes branques de la informàtica ajudaran a crear la nova revolució de la informació que estem a punt de viure.

1.2. Indexació de continguts

1.2.1. Què és?

La indexació de continguts consisteix en l'etiquetatge de material (ja sigui material audiovisual, pàgines web o qualsevol altre contingut) de manera que la cerca del material que ens interessa es faci de manera ràpida i eficient a partir d'aquestes etiquetes prèviament creades.
És un concepte molt senzill, però a la vegada és una eina clau en la ordenació de la informació. Si disposem de grans quantitats de dades i d'informació es fa totalment imprescindible disposar d'algun mètode per a la recuperació de la informació que desitgem.

Aquestes etiquetes que associem a la informació sovint són anomenades metadades (o metadata) . Metadata no és res més que dades sobre les dades. Pot semblar una definició buida i sense sentit, però realment és informació (dades) sobre un recurs concret (dades).

1.2.2. Com es realitza la indexació?

La indexació de continguts es pot realitzar de diverses formes. A continuació expliquem com es pot realitzar aquesta feina:

Indexació manual: És el mètode més simple i que requereix més temps per part de les persones encarregades de gestionar les etiquetes. Consisteix en trobar aquestes característiques importants que defineixen el contingut i etiquetar-ho manualment mitjançant una base de dades.

Indexació social: Aquest mètode és també manual, però es diferencia de l'anterior en el fet que els usuaris de la base de dades tenen la possibilitat d'etiquetar el material en el moment en que el visualitzin. L'etiquetatge no és realitzat per un grup de persones tancat. Aquest mètode està guanyant popularitat gràcies a la forta expansió que ha sofert Internet en aquest últims anys. L'exemple més clarificador d'aquesta tècnica d'etiquetatge el trobem en el portal de vídeos YouTube.

Indexació automàtica: Aquí és on realment entren en joc les diferents eines que ens proporciona la intel·ligència artificial i la visió per computador. Aquest mètode consisteix en la indexació automàtica dels continguts mentre aquests son capturats, o bé amb un processament de les dades posterior, mitjançant tècniques informàtiques que siguin capaces d'extreure les característiques diferenciadores del contingut.

1.2.3. Tipus d'indexacions

Segons el nivell en que analitzem les dades podem dividir la indexació en els següents grups:

Indexació d'alt nivell: En aquest tipus d'indexació el que s'emmagatzema com a metadades és el contingut d'alt nivell del material. Aquest alt nivell normalment consisteix en l'acció, el temps i l'espai i altres dades referents al significat del material.

Indexació de baix nivell: Aquest altre tipus d'indexació no treballa en el significat del material sinó en les seves característiques bàsiques. En el cas que estiguem analitzant material audiovisual, aquestes dades poden consistir en el color mig de la imatge, el to i sonoritat mitjana del so, o bé els nivells de contrast i saturació.

Indexació en un domini específic: Aquesta tècnica fa servir les característiques d'alt nivell per a obtenir les dades de baix nivell. Aquest tipus de tècniques només resulten efectives en un domini d'aplicació molt específic, el qual és una limitació prou important.

1.2.4. Origen i usos de la indexació

Actualment hi ha un gran nombre d'empreses que treballen amb grans quantitats de dades, com ara les dades audiovisuals. Sovint tenen un gran repositori d'informació on guarden l'històric dels diferents treballs realitzats, així com material necessari per la realització dels productes. En les empreses el temps és or, i és necessària la cerca ràpida del material que es desitja i la recuperació d'aquest en el mínim temps possible. No és viable navegar per les carpetes del repositori fins a trobar el material que es necessita. Per tant, cal un mètode que solucioni aquestes mancances en l'àrea de recerca de grans quantitats d'informació. La solució adoptada per la majoria consisteix en relacionar els continguts amb metadades.
Així doncs, si una agencia de publicitat desitja recuperar un anunci realitzat fa uns quants anys en el que apareixia un nen que menjava un gelat, faran servir un buscador que a partir de les paraules clau "nen" i "gelat" trobarà l'anunci en un temps acceptable. En canvi, imaginem que haguéssim de mirar els anuncis realitzats en els últims anys per a localitzar-lo, seria totalment ineficient i poc productiu.

En l'actualitat, la indexació de continguts no s'ha limitat únicament a l'ús empresarial. En el marc d'Internet, és ben coneguda la pàgina Youtube [Referència a la bibliografia]. Aquesta pàgina permet la recerca i visualització de vídeos a partir de certes paraules clau

que introduïm en el buscador. Aquestes paraules clau consisteixen en metadades sobre el vídeo a visualitzar. Cada usuari que decideix posar un vídeo a la xarxa definirà les paraules clau sobre l'arxiu i el pujarà a la web per a ser accessible per a tothom a qui li interessi.

1.2.5. El futur de la indexació

Amb la forta demanda dels continguts multimèdia, la indexació està creixent i guanyant gran popularitat. Gran part d'aquesta popularitat es deguda a l'ús d'Internet.

Inicialment, la indexació de pàgines web per a ser localitzades pels buscadors, la feien els mateixos creadors de la pàgina, això va ser el que va posar els ciments a tot el procés que vindria després. És clar que quan va començar tot, la tecnologia de l'època no permetia compartir continguts multimèdia degut a la lentitud de les xarxes de comunicació i que els formats de compressió d'arxius multimèdia es trobaven a les beceroles.
Els temps han canviat, i actualment la compartició d'arxius multimèdia a través de la xarxa gaudeix de bona salut. És evident que aquest ha estat un punt d'inflexió que ha contribuït notablement al creixement de la indexació i la gestió de continguts.
Creiem que el futur a curt termini que li espera a la indexació és realment impactant. Amb la entrada de la nova televisió digital i interactiva, on disposarem d'una quantitat de canals increïblement gran, serà necessari un mètode per a seleccionar què volem visualitzar segons el que ens vingui de gust en aquell moment.
Amb una amplia oferta de canals com prometen que tindrem, al voltant d'uns cinc mil, practicar el Zapping serà poc productiu, ja que quan haguem visualitzat l'últim d'aquests canals segurament ja haurà canviat la programació del primer que haurem vist. És a dir, la gestió de continguts serà més una necessitat que no pas una possibilitat.

1.3. Televisió

1.3.1. On ens trobem en la televisió

Avui en dia, encara predomina la televisió analògica, on tan sols ens arriba una senyal de vídeo i veu, més unes dades de teletext que s'aprofiten a enviar en els moments de rastreig de la imatge. A més, no hi ha cap forma fàcil i ràpidament accessible per poder enviar dades en la direcció oposada, es a dir, de casa nostra a la cadena de televisió. Això és un gran inconvenient tenint en compte com està el món actualment, ja que cada vegada més, es vol una interacció entre l'espectador i la cadena. La forma tradicional de fer aquesta comunicació és utilitzant el telèfon, i actualment també els missatges SMS o els correus electrònics. Tot i així, encara hi ha un gran desfasament comparat amb la gran revolució en la forma de comunicació humana d'aquest segle, Internet.
És per això que ha sorgit la TDT, televisió digital terrestre, la qual intenta poder cobrir aquestes mancances. La TDT permetrà una millora considerable de la imatge i el so, utilitzant molt més amplada de banda per poder transmetre el seguit de dades, i a més, també permetrà transferir altres dades digitals per poder fer una interacció molt més còmoda amb l'usuari i la cadena.
Tot això es pot aconseguir fent un aprofitament de l'amplada de banda gracies a transmetre la informació digitalment, la qual cosa ens permet poder comprimir totes les dades a ser enviades. Gràcies a la computació d'avui en dia no suposa cap pèrdua considerable de temps.
Algunes d'aquests serveis interactius que la TDT ens permet, combinant-ho amb l'estàndard MHP, són les Guies de programació electròniques (EPG), anuncis interactius i serveis d'informació com ara les últimes noticies, el temps de la teva zona, informes del tràfic, etc.

S'ha de tenir en compte que actualment la televisió està perdent terreny respecte a Internet, ja que ja hi ha molta gent que prefereix passar el seu temps lliure navegant per la web i mirant els continguts que ell tria en cada moment que no pas estar davant de la televisió avera que emeten. A més, amb la capacitat d'amplada de banda disponible actualment, ja és possible la descàrrega de programes o series per Internet i veure-les justament quan vols, evitant dependre de l'horari de la televisió. Aquest serà un fet que

haurà de canviar en els pròxims anys si la televisió vol seguir sent dels primers en el negoci de l'entreteniment.

1.3.2. Cap a on avança la televisió

És sempre incert el que ens depara el futur, però es poden intuir certs aspectes que segur que predominaran en la futura televisió. Una de les principals novetats serà la televisió sota demanda, en la qual es podrà tirar que veure en cada moment, no com ara que hi ha una programació lineal i has d'estar a l'hora que comença el programa per poder-ho veure, sinó que hi haurà com una mena de carta televisiva on podràs tirar el que vols veure i quan ho vols veure. De fet, ara ha sortit un programa que fa exactament això, es diu Joost [Referència a la bibliografia]. Encara està a la fase inicial, però els creadors afirmen que serà una revolució en la forma de veure la televisió.

Una altra novetat completament relacionada amb la que acabem de comentar, serà com triar el que vols veure a cada moment. Aquí és on entra clarament el nostre projecte, ja que hi haurà d'haver mètodes de cerca basats en contingut molt més específics que no els que hi ha avui en dia, ja que hi haurà una gran quantitat de possibles opcions a l'hora de triar el que vols veure.

Altres novetats que revolucionaran el món de la televisió, seran una interacció molt més lleugera entre la cadena i l'espectador, permetent la comunicació a temps real, com per exemple programes de concursos interactius amb tots els espectadors, on des de casa amb el telecomandament a distància podran marcar la resposta correcta i fins i tot guanyar premis. També hi podrà haver comunicació entre espectadors, com fòrums de debat directament incrustats en el mateix programa.

També hi haurà una comunicació totalment directa entre la televisió i Internet, sent possible tenir un navegador web just al costat del programa que estem mirant, tenir un programa de missatgeria instantània per poder a la vegada parlar amb els teus amics o que s'obrís una nova finestra quan arribés un correu electrònic nou, etc.

Els anuncis publicitaris de televisió, tal com els concebem actualment, per força hauran de canviar. Com que són la principal font d'ingrés de les cadenes televisives, farà falta

buscar nous mètodes per poder posar la publicitat. Un mètode que s'està proposant actualment, no per solucionar aquest problema, però si per crear una forma d'anunciar més eficient i potser menys pesada, és utilitzar el mètode que fa servir Google amb el seu sistema de publicitat en pàgines web, Google Ads. Aquest mètode consisteix en posar anuncis relacionats amb el contingut que estàs visualitzant.

Per una banda, a l'afegir interacció amb els programes, es podran crear anuncis molt més dinàmics, i més orientats a l'espectador en concret. Es podran emetre anuncis diferents per cada usuari, fent que aquests corresponguin a les aficions i interessos de l'usuari concret. Això es podria realitzar mantenint un historial dels programes preferits per poder crear un perfil d'usuari i mostrar anuncis que poguessin ser del teu interès. Inclús es podran recollir dades de les reaccions dels usuaris davant d'un anunci en concret (si decideix interactuar, si simplement l'observa, si canvia de canal...).

Per altra banda, la gent intentarà evitar els anuncis tant com sigui possible. Existeixen sistemes de detecció d'anuncis, com ara els plantejats a [4]. Això farà que els usuaris evitin els anuncis, suprimint-los de les seves gravacions televisives. Les empreses de publicitat hauran d'innovar i plantejar nous mètodes de publicitat, ja que sinó ningú veurà el seu treball.

Amb la nova televisió digital, on es podran descarregar els programes a més velocitat que no pas a temps real, la eliminació dels anuncis i la publicitat sense haver d'esperar a disposar d'una gravació (en cinta VHS, DVD o disc dur) serà una realitat.

Segurament és pot pensar la televisió del futur, com una barreja entre Internet i la televisió convencional, la qual cosa ens porta directament a pensar que hi haurà noves cadenes de televisió, nous programes, que ara són del tot impossibles. La televisió d'avui en dia està pensada per agradar a gent geogràficament propera, i per nacions. Amb Internet la cosa canvia, i hi haurà una televisió per a tot el mon, la qual cosa permetrà crear programes que aquí només serien d'interès d'uns quants, però en tot el món, seran una gran quantitat a tenir en compte.

1.3.3. Gestió de continguts

Hem vist doncs, que un punt clau d'aquesta nova televisió serà el de poder triar els programes del nostre gust a cada moment. Per tant, la indexació de contingut serà un dels temes més importants a tenir en compte per a poder cercar còmoda i fàcilment els

programes que desitgem. No cal dir que també serà de vital importància establir quines poden ser les característiques a indexar, ja que en un entorn on hi poden haver més de 5.000 canals amb diferents programes a triar a qualsevol hora, amb quatre característiques no en tenim ni per començar, ja que només podríem tenir un nombre petit de tipus de programes diferents i no s'ajustaria massa a les diferents necessitats de cada persona.

És per això que cal buscar noves característiques per a indexar, però ens porta a un nou problema, com indexar tota aquesta gran quantitat de dades. La solució recau en la informàtica, en utilitzar tècniques de visió per computador i intel·ligència artificial per poder fer una indexació automàtica eficient i si pot ser, ràpida.

1.3.4. Convergència de tecnologies

Com ja hem comentat anteriorment, el que segurament el futur ens prepara, és una convergència entre totes les tecnologies, amb Internet com a l'element intermediari. És un gran pas que encara trigarà uns anys a ser del tot quotidià, però el més probable és que tingui una gran acceptació per part de tothom. Així doncs, se'ns obra un ventall enorme de possibilitats, ja siguin d'interacció amb la televisió com de tasques totalment automàtiques a la hora de buscar programes. Serà possible, per exemple, poder utilitzar el telèfon mòbil per comunicar-se amb l'orinador de casa, perquè aquest a la vegada es connecti amb el televisor i comenci a buscar per tu el programa que més de gust et pot venir. Així només arribar ja ho tindràs preparat. Aquest fet pot ser del tot viable si tenim en compte tot el que hem comentat en aquestes pàgines. Un perfil d'usuari amb els gustos, una televisió digital on es poden veure tots els programes quan vols, més un sistema d'indexació i cerca de continguts ens porta directament a aquest futur no tan llunyà.

També podríem passar a ser esporàdicament els protagonistes d'un concurs a través de la webcam o amb una simple resposta del comandament a distància.

Gràcies a aquestes noves possibilitats, sorgiran noves aplicacions per fer cada vegada més interactiva la televisió, utilitzant les diferents tecnologies ja conegudes i les que estan per venir. Això ens portarà a un sistema d'entreteniment completament nou i ple de noves experiències.

Clars exemples d'aquesta convergència de tecnologies els podem trobar a la cadena de televisió americana NBC, que permet descarregar-se els capítols de totes les sèries que emeten per a ser visualitzats amb un iPod.

És un gran encert per part d'aquesta cadena de televisió. És evident que si ells no posen disponibles aquestes sèries per Internet, algú altre s'encarregarà de fer-ho.
Per altra banda, trobem l'exemple de Joost, esmentat anteriorment, un projecte dels creadors de Kazaa i Skype. Permet visualitzar continguts audiovisuals en Streaming a través d'Internet. Aquests continguts són compartits per els diferents usuaris de la aplicació, com si es tractés d'un programa d'intercanvi d'arxius P2P.

1.4. Treball

1.4.1. Explicació dels objectius inicials

L'objectiu principal d'aquest projecte és poder demostrar que hi ha certes característiques en els programes de televisió (ja siguin pel·lícules, sèries, programes, telenotícies...) que ens han de permetre poder classificar aquests programes per altres categories apart de les característiques típiques de gènere, actors principals, idioma, etc. sinó que se'n poden extreure d'altres més relacionades amb sensacions i sentiments. Tenint en compte que s'acosta la nova televisió digital, on buscar programes del nostre gust serà del tot habitual, poder basar-se en més característiques que no pas les típiques serà molt interessant.

A més, hi ha molts programes semblants de contingut, uns que triomfen completament i d'altres en canvi, que queden en l'oblit. Això és degut segurament, a característiques imperceptibles per nosaltres mateixos, però que si féssim una recerca exhaustiva utilitzant intel·ligència artificial, veuríem que hi són, i que hi tenen molt a veure. Si sabéssim exactament quines són les variables del subconscient que fan que ens agradi més un programa que l'altre, i que fan que un programa tingui més audiència que un altre, podríem arribar a entendre una mica més com funcionen els gustos dels humans i tirar endavant una línia de recerca completament nova pel que fa a aquest aspecte.

Nosaltres ens hem basat en els telenotícies, ja que és un programa comú en totes les cadenes amb diferents formats, convertint-se en un clar exemple per poder diferenciar aquestes trets esmentats anteriorment. En Jordi Hernández ha treballat la part d'anàlisi de la imatge, mentre que el Sergi Espinar s'ha centrat en l'anàlisi del so. La fusió d'aquests dos treballs farà possible la creació d'un sistema complet d'anàlisi i extracció de característiques.

Tot i que aquest és un gran objectiu, nosaltres limitarem el nostre treball a una primera fase, que serà crear un sistema ràpid per poder extreure tot un seguit de descriptors i demostrar que és possible diferenciar i classificar els telenotícies a partir d'unes certes característiques que creiem que poden influir en les esmentades sensacions.
D'altra banda també volem demostrar que les diferències s'accentuen en cadenes de televisió amb més i menys pressupost, i veure que és el que ens fa veure de seguida de que es tracta d'una d'aquestes.

1.4.2. Estructuració del treball

Aquesta memòria està estructurada en quatre grans blocs:

El primer bloc és aquesta introducció, on expliquem les motivacions que ens portat a la creació d'aquest projecte, els objectius que hem volgut assolir i una breu descripció de l'organització d'aquest document.

Seguidament trobem l'apartat de mètodes, on expliquem els coneixements teòrics necessaris que he emprat per fer el projecte, incloent el perquè i com els hem utilitzat.
A continuació hi ha l'apartat de la implementació on expliquem detalladament el procediment que hem seguit per a la realització del projecte, els diferents programes que hem fet i la relació entre ells, així com els resultats obtinguts en els nostres experiments.

Finalment tenim l'apartat de les conclusions, on resumim el que s'ha aconseguit i el que es podria millorar del treball d'aquests darrers mesos, així com les principals aportacions del nostre treball i les diverses línies de continuació d'aquest projecte.

Hem de recalcar que aquest projecte consta de dos parts ben diferenciades però que estan fortament relacionades, a més hem treballat conjuntament a l'hora de planificar el projecte i sobretot en la part final del classificador, la qual cosa ens ha dut a fer una memòria amb algunes parts compartides i d'altres de molt semblants, com ara aquesta introducció i l'apartat dels resultats conjunts.

MÈTODES

2. Mètodes

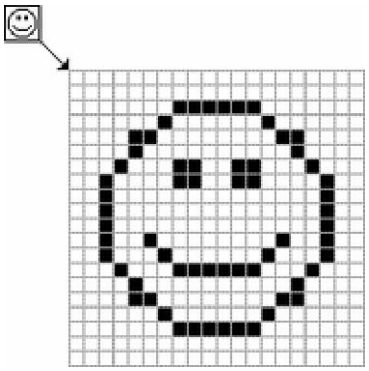
2.1. Conceptes previs

A continuació explicaré una sèrie de conceptes bàsics que apareixeran al llarg d'aquesta memòria, necessaris per poder comprendre-la.

Imatge

Una imatge és una figura formada pel conjunt d'interseccions dels raigs lluminosos procedents de cadascun dels punts d'un objecte (imatge real), o bé formada pel conjunt d'interseccions de les prolongacions dels raigs lluminosos (imatge virtual), després d'ésser modificada la seva trajectòria en incidir o travessar un sistema òptic.

Píxel



És la unitat mínima que forma una imatge digital. Normalment, es defineix com un punt o un quadrat, però realment no posseeix una forma determinada, es pot dir que es tracta d'una mostra abstracta o una unitat de mesura de resolució.

Cal fer notar que el píxel no té una mida concreta sinó que, és establerta pel fabricant de la maquinària, tot i això, acostuma a ser de l'ordre de mil·límetres.

Els píxels són unitats independents, però el fet que els percebem com un conjunt és degut a la limitació de la nostra visió, que per punts tant petits, fa que tinguem aquesta sensació de continuïtat.

A partir del concepte de píxel en podem d'altres com ara el *vòxel* (element de volum), el *texel* (element de textura) i el *sufel* (element de superfície), utilitzats per altres processaments de la imatge.

El valor de cada píxel, és una mesura quantatitzada que determina la intensitat del punt de la imatge que representa. Pot prendre diferents valors depenent del sistema de color que es faci servir, els més comuns són: el *blanc i negre,* on cada píxel té només una

dimensió i és la lluminositat del punt que representa, el *RGB* on el píxel té tres dimensions i cada una representa la intensitat dels seus respectius colors, vermell verd i blau. I finalment el *CYMK* on cada píxel té quatre dimensions i representen el cian, magenta, groc i negre.

El nombre diferents que podem representar a cada píxel depèn de la quantització. És a dir, que podrem representar 2^q possibles colors, on q és el nombre de bits utilitzats per la quantització.

Imatge Digital

Una imatge digital és la representació d'una imatge real com un conjunt finit de píxels. Estan disposats en forma de matriu on cada posició ens dona la localització de cada element de la composició. Com més píxels utilitzem, i més bits utilitzem per la quantització, obtindrem una imatge més fidedigne a la realitat.

Figura: A l'esquerra quantització amb 2 bits, a la dreta imatge real

Normalment es creu que les imatges només són les dels colors, però qualsevol aspecte real que es pugui posar en una matriu pot ser una imatge digital, com en l'anàlisi del so, l'espectrograma, o en aspectes mèdics, una exploració del funcionament neuronal del cervell.

Les imatges digitals, poden ser tan de dos com de tres dimensions, i es poden obtenir amb càmeres digitals, escàners, escàners 3D, radars aeris i altres de més complexos com els sonars o màquines de control sísmic.

Histograma d'una imatge

És la representació de la distribució dels colors en una imatge, es calcula sumant el número de píxels d'un determinat rang de colors, normalment en espais de color unidimensionals (blanc i negre), bidimensionals o tridimensionals (RGB).

Un histograma és la descripció estadística estàndard de la distribució en termes de freqüència d'aparicions en diferents classes, en el color, les classes son les regions en l'espai de colors.

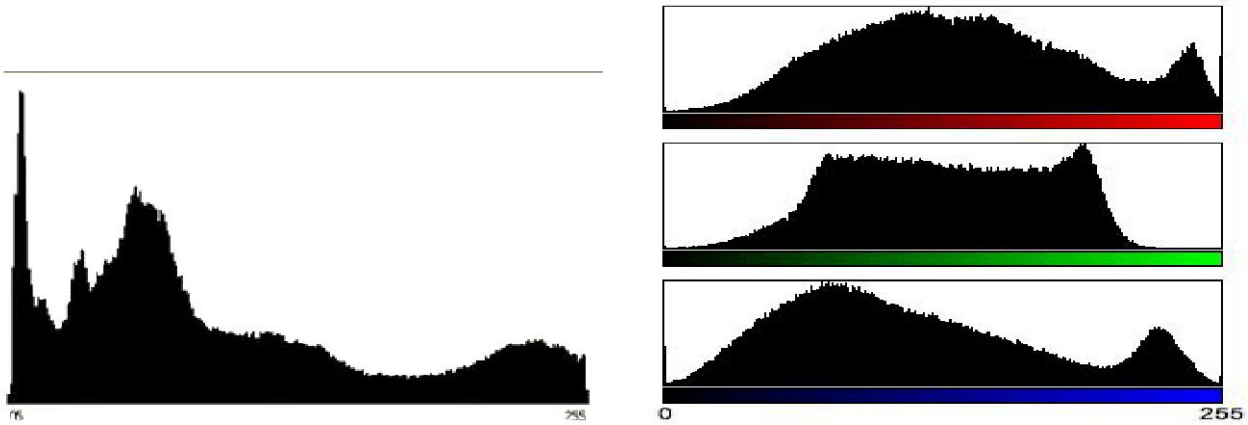


Figura: Histograma unidimensional(blanc i negre) i histograma tridimensional (RGB)

Així doncs, En el sentit més general, un histograma és el mapejat de cada observació en una categoria anomenada *bin*, i el gràfic n'és tan sols una representació. Si *n* és el nombre total d'observacions, *k* el numero total de bins, el histograma m_i seguirà les següents condicions:

$$n = \sum_{i=1}^{k} m_i$$

No hi ha una manera per determinar el millor nombre de bins, a més, diferents mides de cada categoria, poden donar-nos diferents característiques de les dades. Alguns teòrics han provat de determinar un nombre òptim de bins, però aquests mètodes utilitzen grans assumpcions en la forma de la distribució. En general sempre s'hauria de provar diferents amplades de bins per finalment triar el que més s'avingui a les postres dades.

El número de bins pot ser calculat directament, o bé utilitzant una amplada de bins triada *h*:

$$k = \left\lceil \frac{\max x - \min x}{n} \right\rceil$$

On n és el nombre d'observacions en la mostra $x = (x_1, x_2, \ldots, x_n)$

Algunes aplicacions dels histogrames són, per exemple, en les càmeres digitals, per estimar la il·luminació de l'escena com a part de l'algorisme automàtic de balanceig de blancs. En visió per computador pot ser utilitzat per solucionar el problema de reconeixement d'objectes.

2.2. Característiques visuals

Per poder assolir el nostre objectiu, ens hem de basar en característiques que puguem extreure de cada programa o sèrie. Cal tenir en compte que hi ha una gran quantitat de descriptors, i com que no els podem pas extreure tots, hem de saber triar quins seran els més rellevants per la nostra causa.

Aquests, han de ser prou importants com per poder d'alguna manera o altra, canviar la sensació general de la imatge o vídeo, la qual cosa redueix significativament el seu número. Un bon resultat seria, a partir d'agafar tot un seguit de característiques, poder esbrinar quines són les que causen el canvi més fort, quines són les que poden fer fluctuar l'audiència d'un programa, o fins i tot, quines són les que et poden fer canviar l'estat d'ànim.
En el nostre cas en Sergi Espinar es centrarà en el contingut auditiu, i jo, en el contingut visual del vídeo, per així finalment obtenir els descriptors audiovisuals que creiem més importants.

Figura: Exemple de tres imatges que transmeten sensacions totalment diferents.

Cal tenir en compte que el cervell humà te un gran sistema cognitiu, i la típica frase de que una imatge diu molt més que mil paraules, és totalment certa, ja que la informació que se'n pot extreure és immensa. No obstant això, la sensació que ens transmet també depèn molt de cada persona i de les seves experiències personals. Per tant es tractaria

de poder trobar els descriptors el més generals possibles per minimitzar aquesta diferència, però més endavant seria interessant poder-la tenir en compte.

2.2.1. Descripció de les característiques

Cares

Una de les primeres coses que detectem i reconeixem recent nascuts són les cares, primerament la de la mare, però ja de seguida, la resta, a més, cada dia en veiem moltes i de molt diferents, la qual cosa segur que influencia en el que estem buscant. Per tant la característica visual en que més ens centrarem, seran les cares.

Dins d'un vídeo, com a la realitat, les cares apareixen de moltes maneres diferents, ja siguin més grans o més petites (més a prop o més lluny), més al centre o més desplaçades, més pàl·lides o més acolorides, amb els cabells llargs o curts, amb les faccions més marcades o menys etc... però tot això són característiques massa elaborades per el que a nosaltres ens interessa, així doncs, ens basarem en:

- Número de cares: Com més cares apareixien, significa que hi ha més gent la qual cosa dona una certa sensació d'ambient de festa, de passar desapercebut, o també fins i tot d'estrès.

- Escala: Va fortament lligada amb la proximitat, com més gran sigui, significa més propera, i això, pot transmetre més confiança que no pas una cara petita, la qual ens en transmetrà significativament menys.

- Posició: La posició on es trobin les cares també és important, una cara totalment centrada mirant cap a la càmera, dona una impressió molt més directe, com si estes mirant i parlant-te.

- Moviment de les cares: Una cara completament quieta, o una que no es para de moure, es una altre característica que creiem que hem de tenir en compte, perquè per exemple, en el cas dels telenotícies, pot reflexar el caràcter del presentador, i això influenciar en el nostre gust per aquest. El fet d'estar força

quiet ens pot mostrar una certa inseguretat, o una grau d'experiència i seguretat si es mou d'una forma natural.

Color

Es comunament conegut que els colors poden transmetre diferents sensacions, per exemple, el verd esperança, el vermell passió, el groc alerta i el blau calma. Aquestes sensacions tenen una explicació lògica, ja que en èpoques primitives, aquests colors estaven relacionats directament amb aquestes impressions. El vermell es solia veure en fruites sucoses o en la sang, el blau en el mar i el cel, i el verd en la vida de les plantes i els arbres. Actualment es veuen reflexades en molts aspectes, com ara en les senyals de transit, en els productes del supermercat entre d'altres, per això creiem que és un dels trets importants a l'hora d'aconseguir aquesta sensació general de l'escena o vídeo.

Una manera per calcular el color de la imatge és utilitzant el histograma de color, o bé calculant el color mitjà de la imatge.
Tenint *im* com la matriu de la imatge de dimensions *i,j* i amb píxels de tres dimensions *RGB, el color mitjà serà doncs:*

$$R = \frac{\sum_{ii=1}^{i}\sum_{jj=1}^{j} im_{ii,jj,1}}{i*j} \quad G = \frac{\sum_{ii=1}^{i}\sum_{jj=1}^{j} im_{ii,jj,2}}{i*j} \quad B = \frac{\sum_{ii=1}^{i}\sum_{jj=1}^{j} im_{ii,jj,3}}{i*j}$$

Intensitat

És totalment diferent la impressió que ens transmet una escena fosca o clara. Una imatge més fosca tendeix a donar-nos una sensació de por i desconfiança, fortament relacionada amb la nit, on el nostre ull no és gaire eficient ja que per naturalesa és la hora de dormir, la qual cosa fa que no distingim massa bé el que passa el nostra voltant i ens faci sentir aquesta por i desconfiança. Per altra banda una imatge clara ens pot transmetre clarament el contrari, calma i confiança, degut a que a la llum del dia, podem veure perfectament el nostre entorn. Segurament també per aquests motius, han aparegut noves sensacions, no per això menys importants, com ara els culturals. Un blanc abundant relacionat amb el cel, o un negre més fosc i angoixant, relacionat amb el infern.

Figura:Diferencia entra la mateixa imatge amb menys intensitat, i amb més intensitat

Contrast

El contrast d'una imatge ens pot donar una informació de més o menys suavitat com també de tristesa, ja que el fet de que hi hagi poc contrast significa que els colors tenen una tonalitat semblant, es a dir, els valors dels píxels propers tenen valors molt iguals. Aquest fet es pot relacionar directament amb la boira o un dia totalment solejat. La boira fa un efecte suavitzador en el contrast de l'escena, la qual cosa dona aquesta sensació de tristesa perquè en un dia amb aquestes característiques, no es sol sortir tan al carrer, hi no hi ha cant dels ocells, la qual cosa és un sinònim de poca vida, totalment el contrari d'un dia solejat, on es pot veure molta gent passejant, o en èpoques primitives, forces animals i ocells voltant i cantant, el que dona una gran sensació de vida, i conseqüentment, una certa alegria.

Figura: Diferencia entre la mateixa imatge amb menys i més contrast

Canvi d'escena

Un vídeo on tota l'estona és la mateixa escena és molt diferent d'un en el que es canvia cada pocs segons. En el primer cas, pot ser força avorrit, pesat, o fins i tot relaxant, en canvi, en el segon cas pot ser excitant, apassionat o estressant. Caldrà doncs, tenir un descriptor també d'aquesta característica totalment important en el conjunt del vídeo.

Altres descriptors

Seria molt interessant poder extreure quins objectes apareixen en cada imatge, ja siguin cases, ciutats, arbres, cotxes etc..., ja que aquests, tenen un gran significat cognitiu en cada persona. Actualment però, això és totalment impossible, ja que caldria entrenar un classificador amb tots els objectes possibles en el món real, i en totes les posicions en que es poden trobar, a més, imaginant que aconseguíssim tenir-lo, el temps de càlcul per buscar cada objecte en una sola imatge seria del tot intractable.

També seria interessant poder estimar la bellesa de les cares que apareixen, o la bellesa estètica de la imatge en general, però són termes que avui en dia no estan encara gaire ben definits.

En el cas dels telenotícies, creiem que és un molt bon exemple per demostrar que es possible extreure un seguit de descriptors que ens serviran per poder fer una classificació. Són una gran referència de les cadenes de televisió i el seu objectiu es transmetre informació d'una forma clara, objectiva i creïble. Volem veure com aquests descriptors, tan visuals com auditius, ens poden permetre diferenciar els diferents telenotícies, la qual cosa ens confirmarà que anem pel bon camí.

Estem segurs que l'escala del presentador, la posició en que apareix, el color de l'escenari, la quantitat de llum de l'escena, els segons que apareixen i com és mouen, juntament amb la bellesa de la imatge (presentador més escenari) i els objectes que hi apareixen, són els principals descriptors que ens fan que un telenotícies ens agradi més o menys, i també, cosa molt important, que ens el creguem és o menys.

Tenint en compte això, podríem ampliar-ho a qualsevol programa o sèrie de la televisió i veure quines podrien ser les causes de que canviï tant l'audiència si tenen el mateix contingut.

2.3. OpenCV

La paraula OpenCV significa *Open Source Computer Vision library,* i com el seu nom indica, és una llibreria de codi obert per c++ originalment desenvolupada per Intel però que actualment tothom pot contribuir amb les seves aportacions per millorar-la. La llibreria és multiplataforma i funciona tan en Windows com Linux. Està enfocada per a fer processament d'imatge a temps real. Conté tot un seguit de funcions, tan d'alt com de més baix nivell, totalment optimitzades per a processadors Intel, fent així que el seu objectiu sigui possible.

Alguns exemples d'utilització d'aquesta llibreria són:

- Interacció Humà - Ordinador (HCI)
- Identificació d'objectes
- Segmentació i Reconeixement
- Reconeixement de Cares
- Motion Tracking

En el nostre cas, ens hem centrat en les funcions de més baix nivell per extreure les característiques desitjades com ara la mitjana, amb les funcions d'histograma i amb el reconeixement d'objectes per poder detectar cares i extreure'n trets relacionats.

2.3.1. Detector d'objectes

El detector d'objectes de OpenCV està basat una tècnica publicada per Paul Viola i Michael Jones el 2001, coneguda com a **Cascading Haar Classifiers** la qual està basada en quatre conceptes clau:

- Característiques simples rectangulars anomenades *haar-features*
- Una imatge integral per la ràpida detecció de característiques
- El mètode *Adaboost machine-learning*
- Un *cascade-classifier* per combinar les diferents característiques eficientment

Les característiques que Viola i Jones utilitzen, estan basades en *haar wavelets*. Un haar wavelet és un conjunt de *square wave*, i un square wave és un parell de rectangles, un de clar i un de fosc. Però la combinació rectangular utilitzada per la detecció d'objectes visual en OpenCV, no són realment haar wavelet, sinó que contenen combinacions més adequades per les tasques de reconeixement visual, per aquesta diferencia, les característiques s'anomenen *haar features* o *haarlike features.*

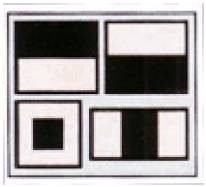


Figura: Haarlike features

Un cop sabem que són les haarlike features, seguirem explicant com funciona l'algorisme per detectar objectes. Primer, un classificador (Anomenat "*cascade of boosted classifiers working with haar-like features"* i que esta explicat més endavant) és entrenat am uns quants centenars de imatges de mostra de l'objecte que volem detectar, per exemple cares o cotxes, anomenats exemples positius, aquestes imatges estan escalades a la mateixa mida (per exemple 20x20), i amb uns quants centenars més d'exemples negatius que seran imatges aleatòries de la mateixa mida.

Després d'haver entrenat el classificador, aquest pot ser aplicat a una regió d'interès, de la mateixa mida que la utilitzada durant l'entrenament, de la imatge que vulguem. El classificador ens retornarà un "1" si la regió és semblant amb el que hem entrenat , ja sigui una cara o un cotxe, o un "0" en el cas contrari. Si volem buscar-ho en una imatge sencera podem moure la finestra de cerca a través de la imatge i comprovar cada posició utilitzant el classificador, el qual està dissenyat de tal forma que és fàcilment escalable per poder trobar l'objecte a diferents mides, i que a més, és més eficient que escalar la mateixa imatge. Així doncs, si busquem un objecte d'una mida desconeguda, haurem de fer el procediment varies vegades a diferents escales i movent la finestra per poder cobrir totes les possibilitats.

La presencia d'una Haar feature és determinada restant la mitjana del valor fosc del píxel de la mitjana del valor clar. Si la diferencia supera un llindar (valor establert durant l'entrenament), llavors direm que la característica és present.
Per determinar la presencia o no de centenars de haar features a cada posició de la imatge i a diferents escales eficientment. Viola i Jones utilitza una tècnica anomenada *Imatge integral.* En general, integrar significa sumar petites unitats, i en aquest cas, les petites unitats són els valors dels píxels. El valor integral de cada píxel es la suma de tots

els píxels de sobre seu i de la seva esquerre, començant per dalt a l'esquerra i anant cap a la dreta i avall. Així doncs, la imatge sencera pot ser integrada utilitzant tan sols unes poques operacions per píxel.

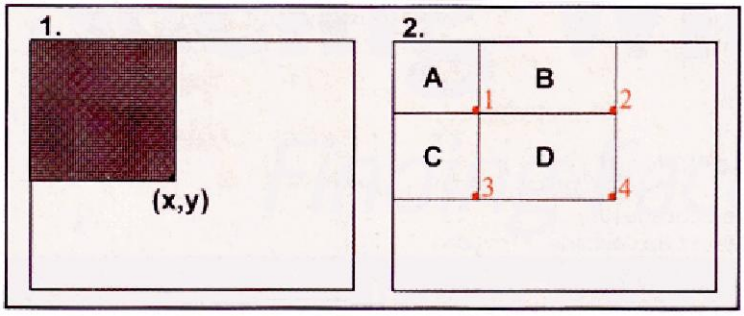


Figura: 1. Integració de la imatge. 2.Divisió de la imatge per rectangles que no comencen a la cantonada

En aquesta imatge (1) podem veure que després de la integració, el valor de cada píxel (x,y) conté la suma de tots els píxels en la regió rectangular per sobre i l'esquerra. Per trobar el valor mitjà d'aquest píxel, només cal dividir el seu valor per l'àrea del rectangle. Ara bé, si volem trobar el valor de un altre rectangle que no comenci en la cantonada, llavors haurem de dividir la imatge com en la imatge(2). Podríem obtenir el valor D de la següent manera:

$$D = A + B + C + D – (A+B) – (A+C) + A$$

On A+B+C+D és el valor de la integral a la posició 4, A+B és el valor a la posició 2, A+C és el valor a la posició 3 i A és el valor a la posició 1. així doncs podem trobar el valor de qualsevol rectangle utilitzant la següent formula:

$$\mathbf{(x4,y4) – (x2,y2) – (x3,y3) + (x1,y1)}$$

Centrant-nos en el classificador, la paraula *"cascade"*, significa que el classificador resultant consisteix en un conjunt de classificadors més simples (etapes) que són aplicats subseqüentment a una regió d'interès fins que en alguna d'aquestes etapes, el candidat és refusat o bé les passa totes. La paraula *"boosted"* significa que els classificadors de cada etapa són complexos i estan fets a la vegada de classificadors bàsics utilitzant una de les quatre diferents tècniques de boosting.

- Currently Discrete Adaboost
- Real Adaboost
- Gentle Adaboost
- Leogitboost

Els classificadors bàsics son arbres de decisió amb al menys 2 fulles. Les característiques *Haar-like* són les seves entrades i estan calculades tal com s'explica seguidament. L'algorisme actual de OpenCV utilitza les següents característiques *Haar-Like.*

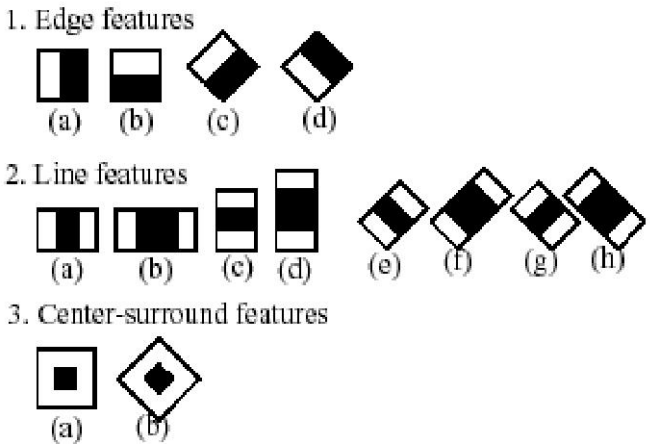


Figura: Diferents caracteristiques haar-like

Els trets usats en un classificador en concret ve definit per la posició de la seva forma (1a, 2b ec..), amb la regió d'interès i la escala. (Aquesta escala no és la mateixa que la utilitzada en l'etapa de detecció, tot i això, aquestes dos són multiplicades). Per exemple, en el cas de la tercer tret (2c) la resposta és calculada com la diferencia entre la suma dels píxels de la imatge dins del rectangle que cobreix tot el tret (incloent les dos tires blanques i la negre del mig) i la suma dels píxels de la imatge dins de la tira negre multiplicat per tres per compensar les diferencies entre les mides de les àrees. Les sumes dels valors dels píxels en les regions rectangulars són calculades ràpidament utilitzant imatges integrals (funció cvIntegral de la llibreria OpenCV)

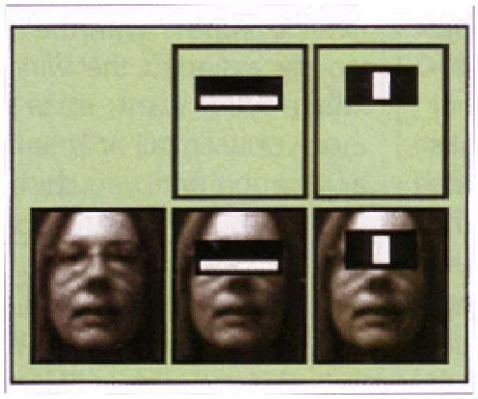


Figura: Exemple de detecció d'una cara basat en les haarlike features

2.3.1.1. Detector de cares

Utilitzant l'esmentat detector d'objectes, openCV ens proporciona uns fitxers XML ja entrenats amb centenars de cares utilitzant el classificador. Llavors, només caldrà cridar la funció "*cvHaarDetectObjects"* de openCV amb els paràmetres de la imatge a reconèixer i el fitxer XML amb les cares entrenades. Aquesta funció troba les regions rectangulars a la imatge semblants a la entrenada en el fitxer XML (en aquest cas les cares), i ens retornarà una estructura amb la posició de tots aquests rectangles. Això ho fa, escanejant la imatge unes quantes vegades a diferents escales, solapant regions per no quedar-se en mig d'una cara i aplicant el detector d'objectes a cada una d'elles. També utilitza heurístiques per reduir el nombre de regions a analitzar, com ara el *Canny prunning* que descarta regions on és probabilisticament improbable que hi apareix-hi una cara. Un cop fet això, ens retorna les regions que han passat el detector com a rectangles. La funció ens permet dir-li la mida de la cara més petita, es a dir, l'escala més petita a analitzar, i també el pas entre escales, la qual cosa ens pot accelerar molt el procés, cosa interessant per poder analitzar vídeo a temps real.

Paràmetres

- Escala de detecció mínima: El setè paràmetre a la crida de la funció *cvHaarDetectObjects,* correspon amb la mida de la cara més petita. Es pot ometre el paràmetre i llavors agafarà la mida per defecte, que si mirem el XML que utilitzem per detectar les cares, podem veure que la mida es 24x24. Depenent del vídeo que utilitzem aquesta mida pot ser molt petita i conseqüentment alentir el procés. Una bona manera de triar la escala per al nostre vídeo, és utilitzar una fracció d'aquest, per exemple ¼ de l'amplada.

- Llindar mínim de veïns: Una de les coses internes que passa en el detector de cares, es que detecta la mateixa cara més d'un cop. N'hi ha que estan molt allunyades i haurien de ser descartades, i les altres s'haurien d'ajuntar en una sola detecció. Aquesta funció dons, fa les dues coses i el valor de llindar mínim de veïns és el que ens diu quantes deteccions s'han de fer en una regió per considerar-ho una cara. El valor per defecte és 3, si per exemple posem 0, ens mostrarà totes les deteccions.

Figura: Detecció de amb llindar mínim de veïns 0

- Increment de l'escala: El quart paràmetre de la funció ens permet dir quan ràpid incrementar l'escala, això ens farà anar el detector força més ràpid, però si posem un valor massa gran, ens podem deixar algunes cares. El valor per defecte es 1.1, es a dir, que augmenta un 10% a cada iteració.

- Canny Pruning Flag: El sisè paràmetre consisteix en un flag que pot ser 0 o *CV_HAAR_DO_CANNY_PRUNING.* Si aquesta opció és seleccionada, el detector no fa cas a regions que heuristicament, no semblen contenir cares i ens pot ajudar a no detectar falsos positius i a reduir la quantitat de computació. Aquestes regions són detectades utilitzant un detector de costats, el *canny edge detector* abans de passar el detector de cares.

2.3.2. ProxyTrans

La llibreria OpenCV ens ofereix un filtre especial anomenat ProxyTrans, és un filtre per l'api multimèdia de Microsoft DirectShow (apartat 2.4) amb una entrada i una sortida, l'entrada necessita imatges (frames) ja descodificades.

El que aquest filtre ens permet, és que per cada frame és cridi una funció que nosaltres vulguem, amb un argument que serà un apuntador a la imatge d'aquest frame en el format d'imatges d'OpenCV, la qual cosa ens permetrà poder treballar amb ella d'una forma totalment transparent, còmode i ràpida.

Finalment la sortida d'aquest filtre serà la mateixa imatge, però tenint en compte que hi podem haver fet totes les modificacions desitjades i haver-ne extret la informació que vulguem.

2.4. DirectShow

DirectShow és una API desenvolupada per Microsoft pels fer més còmode als desenvolupadors de software, poder manejar i realitzar un seguit d'operacions als fitxers multimèdia. És la següent versió de l'antiga tecnologia *Video for Windows,* basada en *Microsoft Windows Component Object Model (COM) framework.* DirectShow proveeix una interfície per diferents llenguatges de programació de Microsoft i és extensible. Es un framework basat en filtres que pot renderitzar o gravar fitxers multimèdia sota la demanda de les necessitats del programador.
Les eines i documentació de DirectShow estan distribuïdes com a part de la plataforma *Microsoft Plataform SDK*. La majoria d'aplicacions relacionades amb vídeo a Windows l'utilitzen per gestionar els continguts multimèdia.

2.4.1. Arquitectura de filtres

Com ja he dit en l'apartat anterior, l'arquitectura de DirectShow està basada en filtres. Cada un d'ells representa una etapa en el processament de les dades i té un número determinat d'entrades i sortides, les quals els connecten entre ells. Això significa que aquest filtres poden estar connectats de maneres diferents per finalment, crear un graf de filtres, gràcies a això, cada desenvolupador pot afegir el seu propi filtre (la qual cosa ens interessa per poder afegir aquí el detector de cares) a qualsevol etapa i després renderitzar el vídeo procedent d'un fitxer, d'una webcam, o d'un socket per Internet.

Per poder crear o testejar grafs de filtres podem utilitzar una aplicació que ve incorporada amb les eines de DirectShow, **GraphEdit**. El que fa és buscar en el registre de Windows els filtres registrats i en crea el graf.

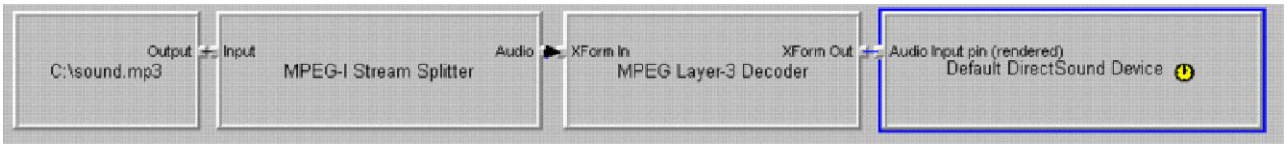


Figura: Exemple del graf de filtres creat per un fitxer MP3

En l'exemple de la figura podem veure com el graf conte quatre filtres, el primer és el fitxer d'entrada a llegir (En aquest cas un fitxer MP3), seguidament hi ha el *Stream Splitter* que s'encarrega de passar el flux de dades que va llegint al següent filtre, el *MP3 decoder* que descodificarà el fitxer MP3. Finalment les dades descodificades i ja preparades per ser escoltades s'envien al filtre *render* que s'encarregarà d'enviar-ho a la targeta de so.
En aquest cas per exemple, podríem afegir un altre filtre que eliminés les freqüències de la veu, l'afegiríem entre els filtres *MP3 decoder* i *Render.*

2.4.2 Característiques

A DirectShow hi ha un seguit de nivells d'abstracció, la més simple d'elles és que el programador creï el graf afegint manualment, connectant cada filtre desitjat en el graf. El següent nivell de complexitat, és que el mateix programador creï el seu graf, per exemple d'un fitxer font i llavors deixar que el DirectShow el completi. I finalment és pot deixar que DirectShow creï tot el graf automàtica i directament des del fitxer, webcam o web.

Per defecte, DirectShow suporta la majoria de formats més comuns, com ara *MPEG1, MP3, WMV* etc... Com ja he comentat, és totalment extensible i es possible afegir extensions que permetin suportar qualsevol format disponible incloent qualsevol códec de vídeo o so, per exemple, s'han fet filtres per el formats *Ogg Vorbis* i *AC3*.

Una altre de les avantatges que té el DirectShow, és que ell mateix gestiona d'una manera transparent el fet d'haver de fer un bucle per anar carregant el fitxer multimèdia. Crea tot un seguit de *threads* que fan el procés, això facilita molt la feina del programador i ajuntant-ho amb el graf de filtres es poden arribar a fer grans coses amb molt poca feina.

Així doncs el DirectShow ens servirà per poder processar els vídeos dels telenotícies, posant el filtre abans esmentat d'OpenCV (Apartat 2.3.2), i en ell, aplicar el detector de cares per poder-ne extreure les característiques desitjades.

2.5. Detector de canvi d'escena

Ja hem comentar que els canvis d'escena són una característica important que caldrà extreure dels vídeo que processem.
Per determinar el canvi d'escena de cop, es a dir, sense *fades in*, *fades out* o *disolves,* i que es el que predomina en els telenotícies, hi ha tres possibles algorismes:

Frame Negre

Aquest és un mètode molt fàcil de calcular però a la vegada molt poc eficient. Consisteix en comprovar cada frame per veure si n'hi ha un de completament negre ja que en forces programes, quan hi ha un canvi d'escena insereixen un frame del tot negre. El gran inconvenient, és que no sempre s'utilitza aquesta tècnica per marcar els canvis d'escena i per això no ens servirà per al nostre extractor de descriptors.

Histograma de color

Aquesta tècnica està basada en la diferencia entre els tres histogrames RGB de dos frames consecutius dels vídeo. Es tracta de marcar un valor llindar el qual si es sobrepassa direm que hi ha hagut un canvi d'escena. Aquesta tècnica és particularment eficaç quan les dues escenes tenen colors significativament diferents i també es comporta molt bé en escenes on hi ha un moviment ràpid de càmera. Tot i això és quantitativament més lent que el mètode de diferència de frames.

Diferència de frames

Està basat en la diferència de la intensitat espacial entre frames successius del vídeo.
Es força més ràpid que el mètode del histograma de color i pot detectar canvis d'escena en ocasions en que no varia el color. Tot i això pot fallar molt en casos que hi hagi un moviment ràpid de càmera, on donarà molts falsos positius.

En el nostre cas hem decidit utilitzar la tècnica del histograma de color, però modificat, en comptes de fer servir els tres histogrames, un per a cada color, utilitzarem només el histograma de la imatge en blanc i negre. Vindria a ser com una mescla dels dos mètodes ja que per una banda utilitza el histograma, i per l'altre, la mesura de la intensitat. És una mica més lent que no pas la tècnica de la diferència de frames, però com que també hem

d'utilitzar aquest histograma per calcular la intensitat mitjana de cada frame en podrem aprofitarem el càlcul.

En la següent imatge podem veure la seqüència de frames de dos canvis d'escena on funcionen perfectament tant la tècnica del histograma de color com la tècnica de diferencia de frames, ja que es pot observar que els colors son substantivament diferent en les tres escenes cosa que afavoreix el primer mètode i d'altra banda la intensitat en el conjunt de la imatge es també prou diferent com perquè el segon mètode també funcioni correctament. A més la nostra versió utilitzant el histograma de la imatge en blanc i negre també detecta els dos canvis d'escena sense cap problema.

Figura: tres canvis d'escena de fàcil detecció amb la tècnica del histograma de color i amb la de diferència de frames.

2.6. Classificador (PrTools)

Per demostrar que la diferenciació dels telenotícies a partir dels descriptors visuals i auditius en que ens hem basat és possible, hem decidit crear un classificador de telenotícies basat en ells. Gracies a això, podrem concretar en quin grau influencia cada

característica en la seva discriminació, i llavors, analitzar-ho més concretament per poder veure quin efecte tenen les variacions en aquests trets.

Per classificar els diferents telenotícies hem utilitzat la eina PrTools4 per Matlab, que té llicència gratuïta per us acadèmic de recerca no comercial.

La principal motivació d'aquesta eina és que en l'àrea de reconeixement de patrons, hi ha una gran quantitat de computació i un gran nombre d'algorismes (i en van apareixent de nous), una plataforma de programació que permeti una implementació rapida i flexible es força necessària.

Conté aproximadament unes dues-centes rutines de reconeixement de patrons i el seu suport corresponent. En l'estat actual, PRTools4 representa el gran conjunt bàsic que cobreix l'àrea dels patrons estadístics. Tot i això, alguns mètodes encara no estan implementats i la part de xarxes neuronals ha estat deixada de banda ja que el propi Matlab ja inclou una molt bona eina referent a aquesta area.

PRTools4, te algunes limitacions, com ara la gran quantitat de memòria que requereix. A més, conjunts de més de deu mil objectes no poden ser processats en màquines normals.

IMPLEMENTACIÓ

3. Implementació

Per assolir el nostre objectiu, ens hem plantejat de seguir el següent procediment, primer de tot, caldrà obtenir tots els vídeos a processar, definir una estructura per guardar els diferents descriptors audiovisuals d'una forma fàcil, ràpida i entenedora, seguidament, caldrà fer un programa que els extregui i els guardi en el format decidit.

Després d'això, vindrà la part d'extracció de dades, la qual cosa requerirà utilitzar el programa que extreu els descriptors, a tot el seguit de vídeos de diferents telenotícies i cadenes per poder tenir una gran varietat i nombre de dades.

Com que un dels nostres objectius és poder reconèixer telenotícies, cal aprendre quin és cada un, i per aprendre fa falta aquesta gran quantitat de dades.

Després de tenir totes les dades necessàries, passarem a fer un programa que les analitzi, i en puguem extreure algunes conclusions, poder veure les diferencies entre les característiques dels diferents telenotícies, i finalment, per demostrar que certament, aquests trets influencien en el resultat final, farem el classificador de telenotícies que ens permetrà, amb pocs segons d'un fragment de telenotícies, dir amb quina cadena es correspon.

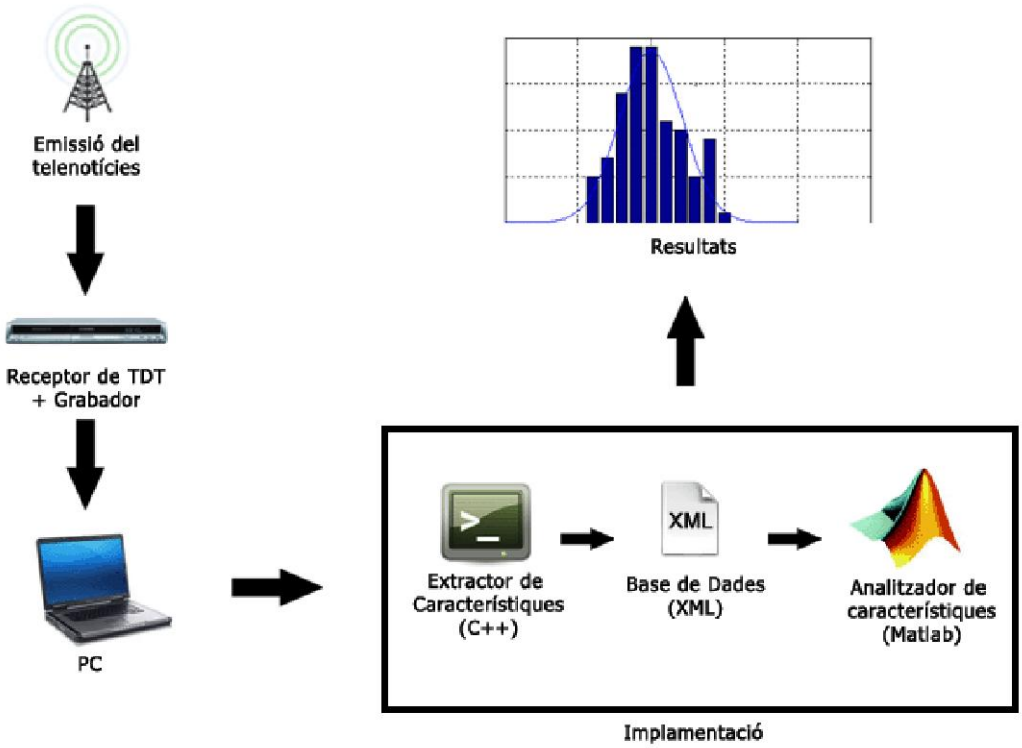
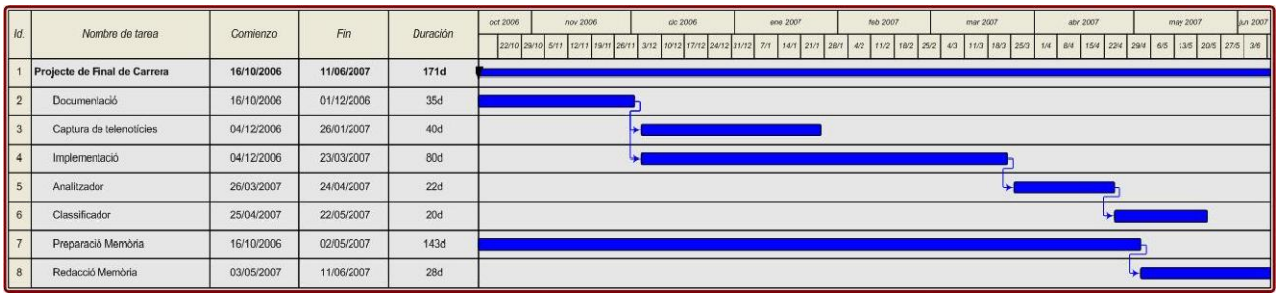


Figura: Procés a seguir per a la implementació del projecte.

3.1. Planificació del projecte

El primer que necessitem és fer una planificació temporal del projecte. Començant a mitjans d'octubre a obtenir documentació. Un cop ja en tenim suficient, comencem l'etapa de implementació, juntament amb la captura dels telenotícies a través del receptor TDT.
Un cop ja tenim la part de l'extractor de característiques i els telenotícies capturats, serà hora de analitzar els descriptors i finalment fer el classificador.
Durant tot aquest temps s'haurà començat a preparar la memòria final, per finalment acabar redactant-la completament.



Id.	Nombre de tarea	Comienzo	Fin	Duración
1	**Projecte de Final de Carrera**	**16/10/2006**	**11/06/2007**	**171d**
2	Documentació	16/10/2006	01/12/2006	35d
3	Captura de telenotícies	04/12/2006	26/01/2007	40d
4	Implementació	04/12/2006	23/03/2007	80d
5	Analitzador	26/03/2007	24/04/2007	22d
6	Classificador	25/04/2007	22/05/2007	20d
7	Preparació Memòria	16/10/2006	02/05/2007	143d
8	Redacció Memòria	03/05/2007	11/06/2007	28d

3.2. Recopilació de vídeos

Per poder realitzar el projecte, necessitem una gran quantitat de dades, la qual cosa implica capturar un gran nombre de telenotícies. Per anar bé, volíem tenir uns 20 telenotícies de quatre cadenes diferents, en total 80 telenotícies,i a més volíem que dues cadenes fossin d'àmbit nacional, i les altres dues de ciutat, tenint així dos tipus ben diferenciats, per poder veure si hi ha grans diferències en la sensació que transmeten entre les cadenes amb més pressupost i amb les altres dues. Així dons, les cadenes que hem triat per la indexació de continguts dels seus telenotícies han estat, *Antena 3* i *TV3* per les dos amb és pressupost, i *Televisió Sant Cugat* i *Canal Latino* per les dues altres amb menys pressupost.

La obtenció dels arxius de vídeo dels diferents telenotícies no ha estat fàcil. Hem tingut certs problemes tècnics que hem solucionat de la millor manera possible. Inicialment, la idea era capturar aquests vídeos mitjançant una capturadora de TDT. La varem provar des de Manresa i des de Sabadell. A cap d'aquests dos llocs rebíem suficient senyal per a poder gravar les emissions en unes mínimes condicions. Per tant, varem haver de buscar una alternativa. Finalment, la obtenció la varem realitzar amb un DVD gravador de

senyal analògica. El problema és que el procés va ser molt lent i requeria un excessiu nombre de passos per a obtenir un arxiu de vídeo en un format acceptable per a treballar-hi, la qual cosa, juntament amb la de veure que el volum de dades era suficient per el que volíem fer, ens va decidir a agafar 10 telenotícies de cada cadena en comptes dels 20 que teníem pensat inicialment.

3.3. Estructura dels descriptors

Es tracta de tenir una base de dades amb tots els descriptors de cada telenotícies, però aquesta no ha de ser ni molt menys concurrent, ni amb control de consistència, ja que el que necessitem és tan sols una forma de generar i guardar les dades, sense la necessitat de ser modificades més endavant. Tampoc ha de ser accessible des de cap xarxa, ha de ser una base de dades local, senzilla i ràpida. Per tot això, hem cregut convenient utilitzar els coneguts fitxers XML, els quals ens donen exactament el que necessitem, senzillesa d'us, i sense controls de cap tipus que no necessitem.

Un cop ja sabem on guardar els descriptors caldrà definir com guardar-los. Hem de fer-ho d'una forma que sigui senzilla i ràpida de calcular ja que ens interessa poder-ho fer a temps real.

Per les cares utilitzarem el rectangle que ens proporciona la mateixa funció de l'OpenCV, un punt inicial, una amplada i una alçada, amb això ja tenim tot el que necessitem perquè en podem treure l'escala i la posició, i amb un seguit de frames, n'obtindrem el moviment.

També guardarem el número de cares de l'escena, i en el cas que no n'hi hagi cap, els valors de la posició seran tots zero, però no es tindran en compte en el posterior anàlisis.

Per el color, utilitzarem la mitjana dels valors de cada canal de cada píxel, així doncs, obtindrem un valor mitjà pel vermell, un pel blau, i un pel verd que estarà entre 0 i 255.
Si volguéssim una informació del color més acurada, ho podríem fer utilitzant el histograma de color, on ens diu quin color predomina en cada regió, però en el nostre cas no ho creiem convenient ja que amb el color general de la imatge ja en tenim la informació suficient.

Per la intensitat i el contrast si que utilitzarem el histograma, però en aquest cas de la imatge en blanc i negre, ja que el color no hi te a veure, i en calcularem la mitjana dels valors que serà la intensitat mitjà de la imatge, i la desviació típica que en serà el contrast mitjà.

Finalment ens caldrà detectar els canvis d'escena utilitzant l'algorisme explicat anteriorment, i guardarem un "1" si es el frame d'una escena nova i un "0" si no ho és.

3.4. Processament del vídeo

Per al processament del vídeo, com que ens caldrà extreure els descriptors de la imatge, i ens interessa que pugui ser a temps real, hem decidit fer el programa en C++ que ens permet una gran rapidesa ja que es pot treballar a baix nivell i a més, hi podem afegir la llibreria de OpenCV que ens permetrà utilitzar les seves funcions de visió per computador totalment optimitzades.
Llavors doncs, ens faltarà poder processar cada frame d'un vídeo, i ho farem utilitzant la api de Microsoft DirectShow, que ens proporciona la interfície adequada per poder-ho fer.

Primerament, el que cal crear és una funció la qual passant-li una imatge, ens detecti les cares i les enquadri per poder-ho veure. Utilitzant la funció de OpenCV *cvHaarDetectObjects,* el fitxer entrenat que ell mateix ens proporciona, i la funció *cvRectangle* per enquadrar les cares.

En la següent imatge, podem veure que el detector de cares que ens ofereix OpenCV amb els paràmetres per defecte funciona força correctament, de les vint-i-cinc cares que hi ha, les troba totes, però també podem apreciar que en tres casos, troba dos cops la mateixa cara. Això és degut a que l'algorisme del detector, recorre la imatge diferents vegades a diferents escales, i es dona la casualitat que la mateixa cara dona positiu en

dues escales diferents i sobrepassa el valor de mínims veïns. Es podria solucionar fàcilment si canviéssim el valor de pas d'escala, o el valor de veïns però llavors, hem pogut comprovar que hi ha molts falsos negatius, ja que en forces casos la regió que avaluarà, es correspon amb un tros de la cara, o amb massa fons, a més, pel nostre objectiu ens interessa més que detecti dos cops la mateixa cara que no pas que no la detecti, i en el cas que es donés aquesta situació, es pot detectar i tractar d'una forma molt fàcil, només cal mirar les posicions i mides del conjunt rectangles que ens troba el detector i comprovar que no n'hi hagi de superposats. Tot i això, a la part de l'extractor de característiques, he modificat els paràmetres per defecte per aconseguir una detecció més ràpida i precisa, a més, en el cas dels telenotícies, el detector és on funciona més bé, perquè es tracta d'una escena amb una sola persona, on la proporció de la cara respecte la imatge és força gran i que mira frontalment la gran majoria del temps.

Com hem vist, el detector funciona molt bé, però hi ha casos en que pot arribar a fallar bastant, com ara quan hi ha cares parcialment ocultes, quan estan força de perfil o quan tenen un tros de cara a la ombra i l'altre al sol.

Un cop ja tenim una funció que ens detecta cares passant-li una imatge, ara cal fer que d'un vídeo, passem cada frame per aquesta funció. Això ho farem utilitzant l'api de Microsoft DirectShow. El que s'ha de fer primer és crear el graf de filtres per fitxers .MPG i .AVI, que seran els dos tipus possibles de vídeo que admetrem. Utilitzant l'aplicació GraphEdit obtenim el següent.

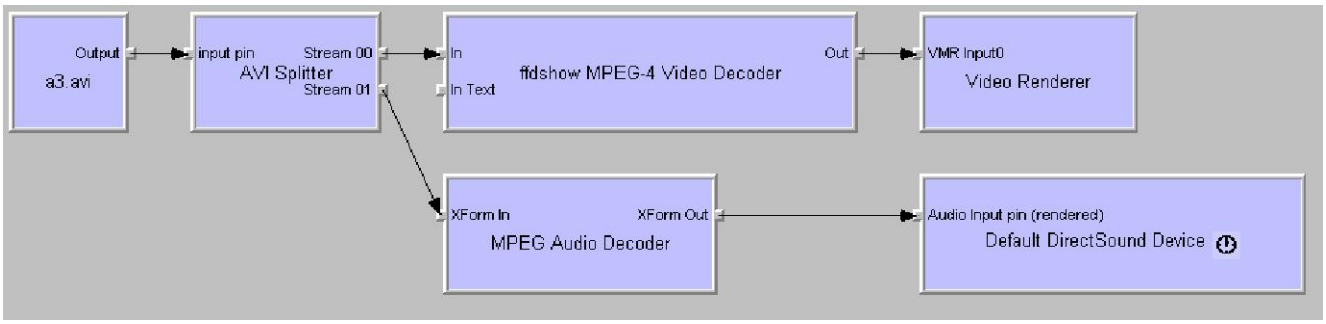


Figura: Graf de filtres per un fitxer .avi

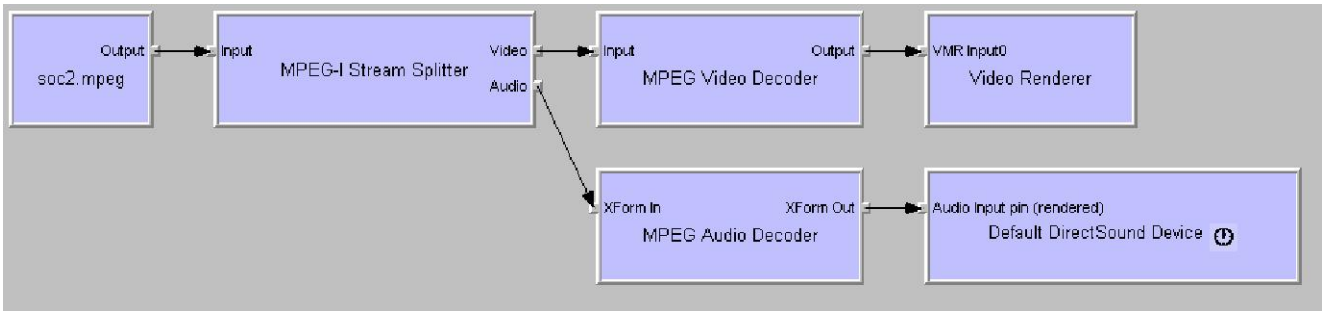


Figura: Graf de filtres per un fitxer .mpg

Com podem veure, només canvia el primer i segon filtre i el que farem és que segons l'extensió en carregui uns o els altres.

D'aquest graf de filtres, el so no ens interessa, i a més, volem posar el filtre de openCV *proxytrans* just després del Video Decoder, per així tenir el frame com una imatge qualsevol i poder-lo enviar a la funció que hem creat que ens detecta les cares i les enquadra (per poder utilitzar aquest filtre, primer s'ha de registrar amb l'aplicació `regsvr32`) per finalment enviar-ho a la sortida i poder-ho visualitzar.

Figura: Graf de filtre sense la part de so i amb el filtre ProxyTrans

Així doncs ja tenim el graf final que haurem de crear amb les funcions que ens proporciona l'api de DirectShow, i dir en el filtre proxytrans que cridi la funció de detectar cares.

Però la cosa no és tan simple, aquesta funció que ha de cridar el proxytrans, ha de ser una funció global, que rebrà com a argument un apuntador de tipus *void* a la imatge del frame, i aquest serà transformat utilitzant un cas a la imatge estàndard de OpenCV amb la qual podrem treballar còmodament i serà la que passarem a la nostra funció que detecta les cares, i que ampliarem per extreure el conjunt de descriptors que hem cregut necessaris per el nostre objectiu.

Figura: Exemple de l'aplicació d'extreure característiques.

En aquesta imatge podem veure com queda l'aplicació finalment. Hi ha el botó *Obrir* que ens permetrà triar el fitxer a processar, el botó *Processar* que començarà a reproduir el vídeo extraient-ne els descriptors explicats en el següent aparat.
També i ha una llista amb els diferents filtres utilitzats per al processament del vídeo, és exactament el graf de filtres que hem creat en DirectShow posat a la pràctica, el primer és el fitxer d'origen, seguit del descodificador, el detector de cares i la sortida, ja que hem descartat els filtres que feien referència al so perquè no ens interessen.
Finalment podem veure la finestra de reproducció, on es pot veure el vídeo i les cares que troba en cada moment, força interessant per fer proves i comprovar que funciona correctament, d'altra banda, aquest últim filtre de render (que mostra la finestra del reproductor) es podria treure per guanyar una mica més de rapidesa a l'hora de processar tots els vídeos.

Un cop ja tenim tota l'aplicació, ens queda definir la funció *detectFaces()* de la classe *FaceDetetc* que serà l'encarregada d'extreure els descriptors de cada frame i que explico en el següent apartat.

3.5. Extractor de característiques

Aquesta és una de les parts més importants del projecte, ja que caldrà extreure les característiques més importats del vídeo. Com ja hem vist en l'aplicació, l'extractor de

característiques serà una funció que rebrà cada frame del vídeo en el format de OpenCV. La qual cosa ens permetrà utilitzar les funcions de la llibreria que estan optimitzades per a processadors Intel, això ens ajudarà a fer que l'aplicació sigui força més ràpida que si ho féssim en un format estàndard com ara el jpg gif o bmp.

L'algorisme del extractor serà doncs el següent:

Inicialitzar valors

Per a cada frame

1. Actualitzar número de frame
2. Detectar cares i dibuixar el rectangle que les senyalitza
3. Calcular histograma en blanc i negre
4. Calcular intensitat mitjana de la imatge
5. Calcular contrast de la imatge
6. Calcular la mitjana del color
7. Detectar canvis d'escena
8. Escriure el fitxer XML amb tota la informació

Fi per

Amb això ja tenim un sistema que extraurà els descriptors que hem decidit, ara cal doncs, explicar detalladament els passos més importants de l'algorisme i les seves peculiaritats.

3.5.1. Característiques

Actualitzar número de frame

Aquest és el pas més senzill de tots i consisteix en incrementar el frame en u cada vegada. Aquest valor el farem servir per a dos coses, la primera per guardar cada xml amb el número de frame amb que es correspon, i la segona per fer una optimització que explico més endavant.

Detectar cares i dibuixar el rectangle que les senyalitza

Aquest pas consisteix en detectar les cares utilitzant la funció de OpenCV anteriorment explicada. Els paràmetres que finalment he utilitzat han estat, el pas d'escala 1.2, és a dir, que augmenta un 20% a cada iteració. El flag de canny pruning activat per evitar alguns falsos negatius que sense ell detectava, el número de veïns mínims per defecte 3, ja que amb 2 no acabava de trobar totes les cares que més ens interessaven. Finalment la mida de cara més petita, la hem posat més gran que la mida per defecte perquè segons la mida del vídeo que utilitzem (320x240), una escala de cara mínima de 24x24 és massa petita per els nostre interès, que són els presentadors dels telenotícies que acostumen a ocupar una gran part de l'escena. Així doncs, hem decidit utilitzar una escala mínima de 80x80 que és ¼ de l'amplada del vídeo.

Gràcies a canviar els paràmetres per defecte, aconseguim poder processar el vídeo a més velocitat que temps real, i amb més exactitud a l'hora de detectar les cares.

Com ja hem dit anteriorment, de cada cara que troba ens en dona les coordenades i l'amplada i alçada, aquestes dos últimes sempre són iguals ja que ens retorna quadrats. Aquestes quatre dades seran doncs, les que guardarem en el fitxer XML per després poder-ne extreure les característiques desitjades utilitzant l'analitzador.

Calcular histograma en blanc i negre

Farem una funció a part per calcular el histograma de la imatge, passant-li com a paràmetre la imatge original. OpenCV ens proporciona una funció per calcular el histograma, a més podem tirar l'amplada dels bins. Primer caldrà doncs, convertir la imatge en blanc i negre i quantatitzada a 8bits per així tenir 256 possibles valors de cada píxel, definir el número de bins i el rang de cada un d'ells. Per el nostre cas hem utilitzat un tan el número de bins com el rang de 256, així tenim cada possible valor en una columna que serà tan alta com píxels amb el seu valor hi hagi.

La formula que utilitza OpenCV per calcular aquest histograma és

$$Bin_i = \sum_{k=1}^{h} \sum_{l=1}^{w} im_{k,l} | im_{k,l} = i$$

On Bin_i és cada bin del histograma, *im* la imatge, i *h* i *w* la alçada i amplada de la imatge respectivament.

Calcular intensitat mitjana de la imatge

Per calcular la intensitat mitjana de la imatge, cal recorre el histograma, i fer la mitjana dels seus valors, com que hem escollit el número de bins a 256, número de diferents intensitats amb una imatge quantatitzada a 8bits, llavors tenim el número total de píxels a cada intensitat i la mitjana del histograma ens donarà exactament el que busquem.

$$I = \frac{\sum_{i=1}^{w} Bins_i}{w}$$

On *I* és la intensitat mitjana i *w*, el número de bins.

Calcular contrast de la imatge

El contrast és una mesura de la diferencia d'intensitats d'una imatge, el qual podem trobar d'una forma general, calculant la variància del valor de cada píxel, és a dir, de la intensitat en cada punt de la imatge.

$$C^2 = \frac{\sum_{i=1}^{w} (Bins_i - I)^2}{w}$$

C és la variància al quadrat, *I* la intensitat mitjana i *w*, el número de bins

Calcular la mitjana del color

Per calcular la mitjana del color només cal sumar els valors de cada color de cada píxel, i dividir entre el número de píxels, tot i això, OpenCV guarda la imatge en color en un format un xic especial, que consisteix en guardar els tres valors de cada píxel, consecutivament, tenint així un vector d'una dimensió tres vegades la mida de la imatge.

Però OpenCV també ens proporciona una funció que calcula la mitjana d'una imatge directament i ens retorna els valors, en el cas de color, en una estructura que conté un vector amb 3 camps, un per cada color.

$$R = \frac{\sum_{ii=1}^{i}\sum_{jj=1}^{j} im_{ii,jj,1}}{i*j} \quad G = \frac{\sum_{ii=1}^{i}\sum_{jj=1}^{j} im_{ii,jj,2}}{i*j} \quad B = \frac{\sum_{ii=1}^{i}\sum_{jj=1}^{j} im_{ii,jj,3}}{i*j}$$

Detectar canvis d'escena

Per detectar els canvis d'escena, utilitzem un una barreja entre els dos mètodes explicats anteriorment, el *Histograma de color* i la *Diferència de frames***.** El primer consisteix en calcular la diferencia entre els tres histogrames de color de frames consecutius, i el segon en calcular la diferencia de frames consecutius píxel a píxel, els dos necessiten tenir un valor llindar, el qual si es supera, és dirà que hi ha canvi d'escena.

El que he utilitzat jo és una diferencia frame a frame, però a partir del histograma en blanc i negre, és com una barreja dels dos, perquè el primer utilitza el histograma per veure canvis en el color, i el segon fa una diferencia píxel a píxel que esta basada en la intensitat, així dons el mètode utilitzat en el nostre cas, fa una diferencia en la intensitat, però basant-se en el histograma.

Fent varies proves hem trobat que un valor llindar adequat era el 15, ja que valors més petits donava molts falsos positius, i valors més grans molts falsos negatius. S'ha de tenir en compte que hi ha canvis d'escena en els quals les dos escenes poden ser força semblants i per això que costi poder-los detectar, i com hem dit abans, un dels inconvenients de la diferencia de frames, és que en escenes on la càmera és mou suficientment ràpid, pot crear falsos positius, ja que els dos frames consecutius poden arribar a ser molt diferents. Tot i això, el detector de canvi d'escena funciona molt bé i només falla en pocs casos, els quals són tolerables per el nostre objectiu final.

Escriure el fitxer XML amb tota la informació

A la base de dades volem tenir tots els descriptors de cada frame i vídeo. Això és un nombre molt gran de dades que cal manejar ràpida i eficaçment.

Per tant en aquesta part, hi ha un gran factor de disseny, cal plantejar bé com crear i guardar aquests fitxers XML on guardarem els descriptors i que serà la nostra base de dades per després fer l'anàlisi. A primer cop d'ull hi ha tres possibles solucions, la primera de les quals consistiria en crear un sol fitxer XML per a tots els vídeos. La segona seria crear un fitxer XML per a cada vídeo, i la tercera, crear un fitxer XML per a cada vídeo i frame.

La primera opció no és gens viable ja que ens quedaria un fitxer immensament gran, a més, el fet d'anar fent proves suposaria un gran inconvenient a l'hora de buscar errors o modificar només algunes parts del fitxer, per tant aquesta opció queda descartada.

La segona, sembla més bona però també s'ha de tenir en compte que per a la realització del programa, i a l'hora de fer les proves necessàries, el fet de tenir un fitxer XML tan gran, és un problema, sobretot per anar a una part en concret del fitxer. Per aquests motius ens hem decidit per la tercera opció.

Tenint un fitxer per cada frame i vídeo, i en carpetes per a cada vídeo, ens és molt fàcil veure quins descriptors extreu en cada moment. A més de la facilitat de modificar només aquells que necessitem.

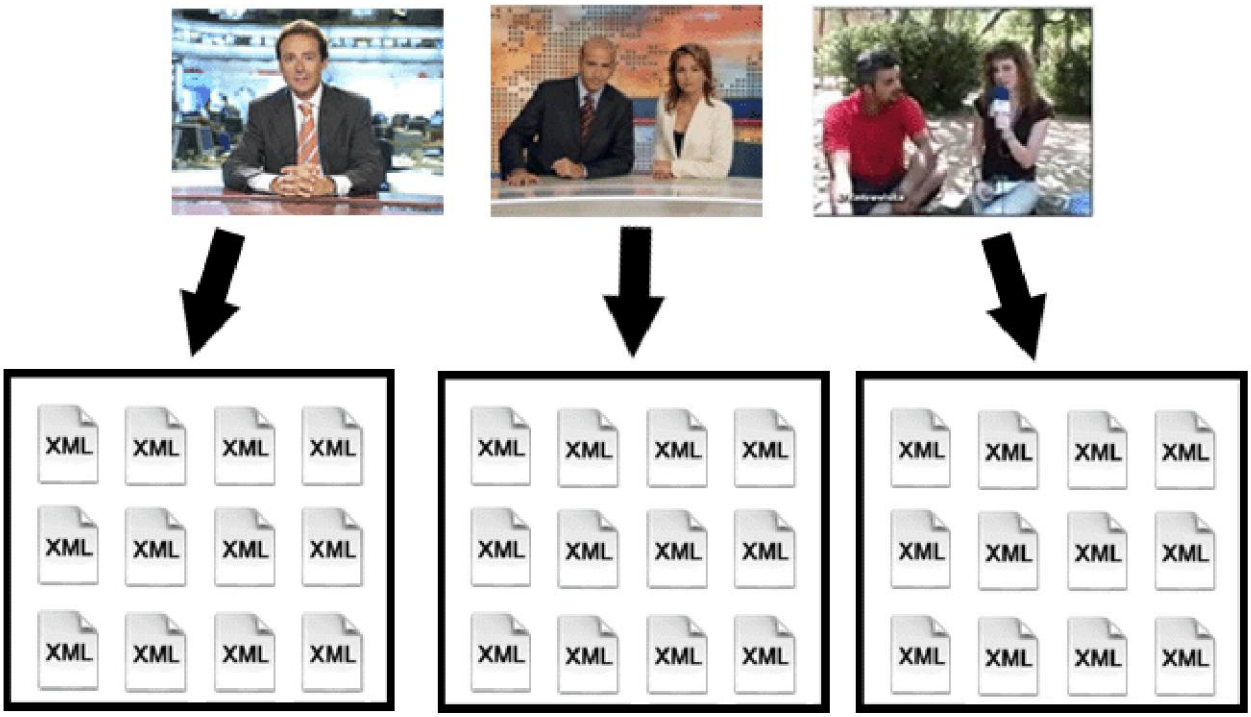


Figura: Esquema de l'estructura de la base de dades

3.5.2. Millores

Un cop ja tenim l'estructura bàsica de l'aplicació, cal veure quines coses es poden millorar per poder fer-lo encara més eficient i ràpid.

El principal problema que té, és la gran quantitat de fitxers XML que genera, tenint en compte que un programa de telenotícies pot arribar a durar una mica més d'una hora (en el cas de TV3) ens trobem que el sistema de fitxers de Windows no suporta tal quantitat de vídeos. Un telenotícies de TV3 generaria: una hora són 60 minuts, i 60 minuts són 3600 segons, i tenint en compte que hem capturat els vídeos a 25 frames/segon, obtenim 90.000 fitxers XML, sense dubte, una barbaritat. A més en el so, per calcular l'espectrograma, s'utilitzen 62.5 frames per segon, la qual cosa encara multiplica més aquest número de fitxers XML.

Així dons, hem utilitzat una opció que ens soluciona els dos problemes, el de el nombre tan gran de fitxers, i el de les mesures diferents en vídeo i so. Es tracta de crear fitxers, en comptes de per a cada frame, per a cada segon. Així, tenim una mesura estàndard i se'ns redueix notablement el número de fitxers a generar. Els valors que hi haurà ara en cada fitxer, serà la mitja del conjunt de valors de cada frame.

La part del extractor de característiques que alenteix més el programa (tot i que es processa el vídeo més ràpid que a temps real) és el detector de cares. Per tant, una altra millora per poder accelerar el procés, consistiria en reduir el nombre de vegades que s'executa aquesta funció. Si pensem una mica en l'estructura dels programes de televisió, podem veure que acostuma a aparèixer la mateixa cara un cert instant de temps seguit, es a dir, no apareix durant només un o dos frames, com també te un moviment més o menys suau, la qual cosa ens diu que ens dos o tres frames seguits, no s'haurà mogut gairebé gens, i menys en el cas dels telenotícies. Això ens porta a creure que si executem el detector de cares un frame cada *X*, podem accelerar el procés sense perjudicar la validesa de les característiques extretes. A més tenint en compte que ara guardem per segons i no per frames ens facilita encara més la tasca. Finalment només cal trobar quin valor de *X* és el més adequat, i hi haurà un compromís entre rapidesa i exactitud. En el nostre cas, com que primerament volem unes dades exactes i no ens interessa tant que les extregui ràpidament, farem que executi la funció de detectar cares a cada frame.

3.5.3. Extreure les característiques

Un cop ja tenim l'extractor de característiques funcionant, el que caldrà fer, és obtenir aquestes característiques. Aquest es el procés de més durada del projecte, perquè s'han de processar tots els vídeos dels telenotícies. Tenint en compte que tenim vora els 40 vídeos i cada un d'una durada aproximada de 50 minuts, són uns 2000 minuts a processar, es a dir, 33 hores de vídeo. Com que hem aconseguit fer anar l'extractor de característiques un pel més ràpid que temps real, ho podrem arribar a processar amb unes 25 hores, tot i això, segueix sent un procés llarg.

Finalment, després d'haver processat tots els vídeos, hem obtingut 36 carpetes, una per cada vídeo, amb una mitjana de 2.700 fitxers XML per els telenotícies de *Antena 3* i *Televisió Sant Cugat*, 3.800 fitxers XML per els de *TV3* i 1.300 per els del *Canal Latino.* Tot plegat són un total de **93.340 fitxers XML** els quals ocupen un espai de 30Mb però en l'estructura de fitxers de Windows, on cada els blocs mínims son de 4bytes, ens trobem que aquestes **30Mb** es converteixen en **1.42Gb** d'espai a disc. Aquí doncs ens podríem plantejar utilitzar l'altre opció mencionada anteriorment que es tractava de crear un sol XML per cada vídeo, reduint així aquesta fragmentació notablement, però com que avui en dia aquesta quantitat de memòria no ens és imprescindible, hem decidit seguir així perquè ens es molt més còmode a l'hora de treballar amb els descriptors extrets de cada frame.

3.6. Analitzador de telenotícies

Ara que ja tenim la bases de dades plena amb els descriptors extrets de cada vídeo, cal analitzar aquestes dades per poder-ne extreure algunes conclusions.

El primer que farem, serà crear les gràfiques de cada descriptor, així podrem veure com es comporten els telenotícies de cada cadena segons aquests descriptors. Després crearem un comparador de telenotícies, que segons els descriptors d'un frame, ens dirà ordenadament els frames dels altres telenotícies que més s'hi assemblen, així podrem veure fàcilment quines semblances i diferencies hi ha, i que a simple vista costaria força poder-les detectar.

Per fer tot això utilitzarem el programa Matlab, que ens permet treballar molt fàcilment i eficaç amb grans volums de dades, ens permet crear gràfiques de tot tipus que ens seran molt útils a l'hora de comparar i extreure conclusions.

3.6.1. Gràfiques dels descriptors

Primerament, per poder crear unes gràfiques més acurades, extraurem els descriptors per a cada frame d'un vídeo de cada telenotícies, en comptes de fer-ho de cada segon, ja que així podrem veure el funcionament amb més detall i comprovar que l'extractor de característiques funciona correctament.

Per utilitzar aquestes dades amb el Matlab, el que farem, és crear una funció que ens llegeixi tots els fitxers XML de un vídeo i ens ho guardi en una matriu on cada fila serà un descriptor, aquesta matriu doncs, tindrà tantes files com descriptors i tantes columnes com frames, es a dir, com fitxers XML. Un cop ja tinguem aquesta matriu serà trivial mostrar les gràfiques i calcular mitjanes per així poder veure d'una manera general, les diferències entre els telenotícies de cada cadena.

Número de cares

El número de cares de cada frame ens mostra exactament això, quantes cares apareixen en cada frame. Aquesta característica tampoc no és massa rellevant en el nostre cas, perquè en els telenotícies acostuma a haver-hi el presentador al centre de l'escena parlant frontalment cap a la càmera, pot ser a vegades que hi apareixen els dos presentadors, o en les imatges d'explicació d'una noticia, que aparegui més gent, però nosaltres ens centrarem en les escenes on hi hagi un determinat presentador parlant, ja que sinó, per ajuntar-ho amb les característiques de la veu, seria una tasca realment complexa.

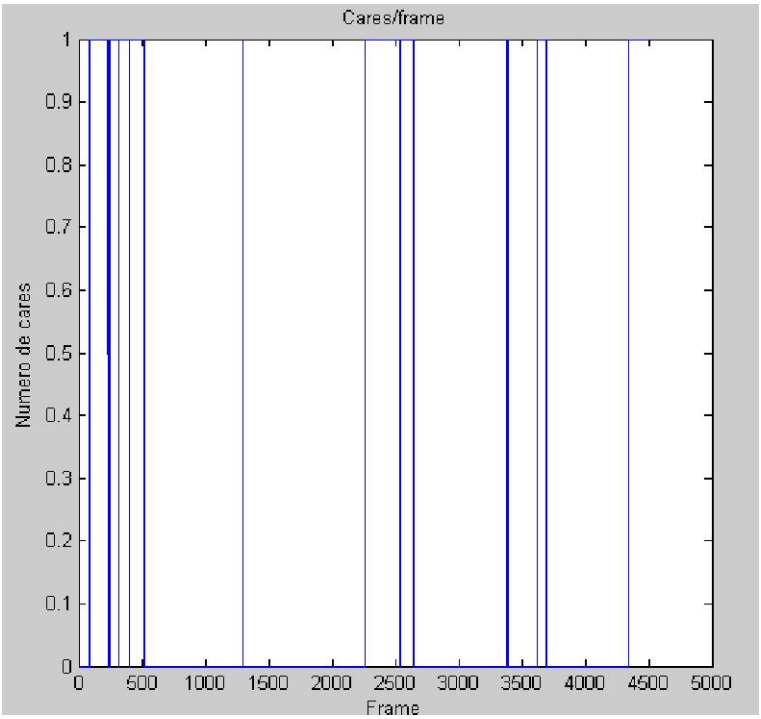


Figura: Gràfica del número de cares d'un telenotícies de Antena 3

En la gràfica es poden deduir clarament els trossos on hi ha el presentador del telenotícies, i en els casos que no. També es pot veure que hi ha alguns falsos negatius però que són pocs i no ens influiran gaire en el resultat final.

Escala de la cara

Aquesta característica si que és realment important, ja que creiem que és una de les que tenen més impacte a l'hora de mirar i caracteritzar els telenotícies.

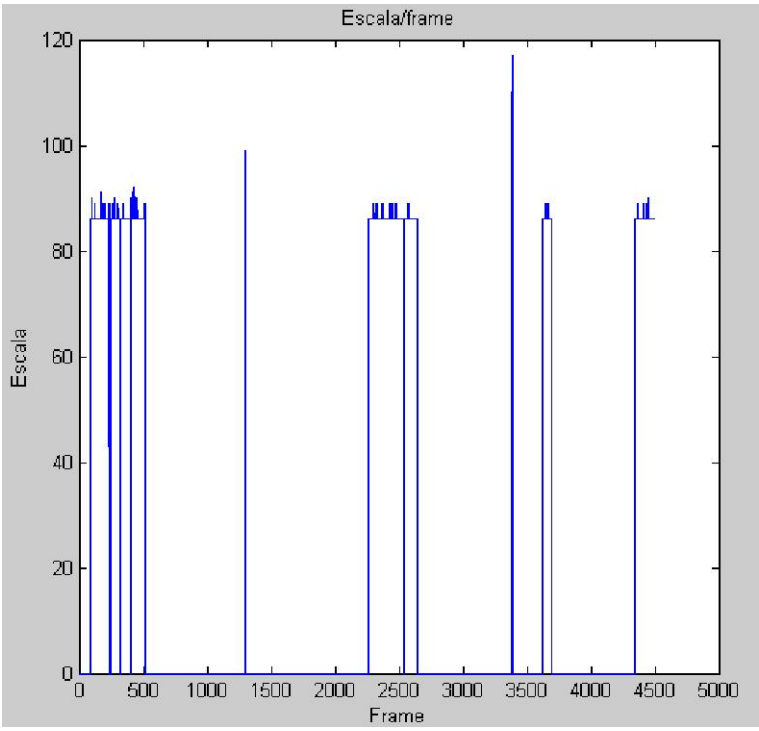
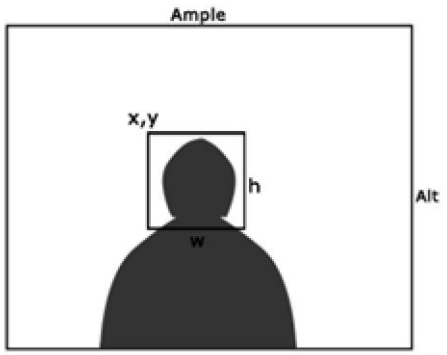


Figura: Gràfica l'escala de les cares en un telenotícies de Antena 3

En aquesta gràfica si que es pot apreciar molt bé en quins espais hi ha el presentador del telenotícies, a més, podem veure que manté l'escala de la cara molt constant vora els 85-90 píxels. També podem veure un parell de falsos positius a l'hora de detectar les cares però més endavant explicarem perquè no ens hi hem d'amoïnar.

Posició de les cares

OpenCV ens torna la posició de la cara amb les coordenades del punt superior esquerre juntament amb l'amplada i alçada. A nosaltres ens interessa el punt mig de la cara per poder dir exactament on es troba i veure si està centrada o no, per això utilitzarem la següent: fórmula:



$$X = x + w/2 - Ample/2$$
$$Y = y + h/2 - Alt/2$$

Així doncs, obtenim les següents gràfiques:

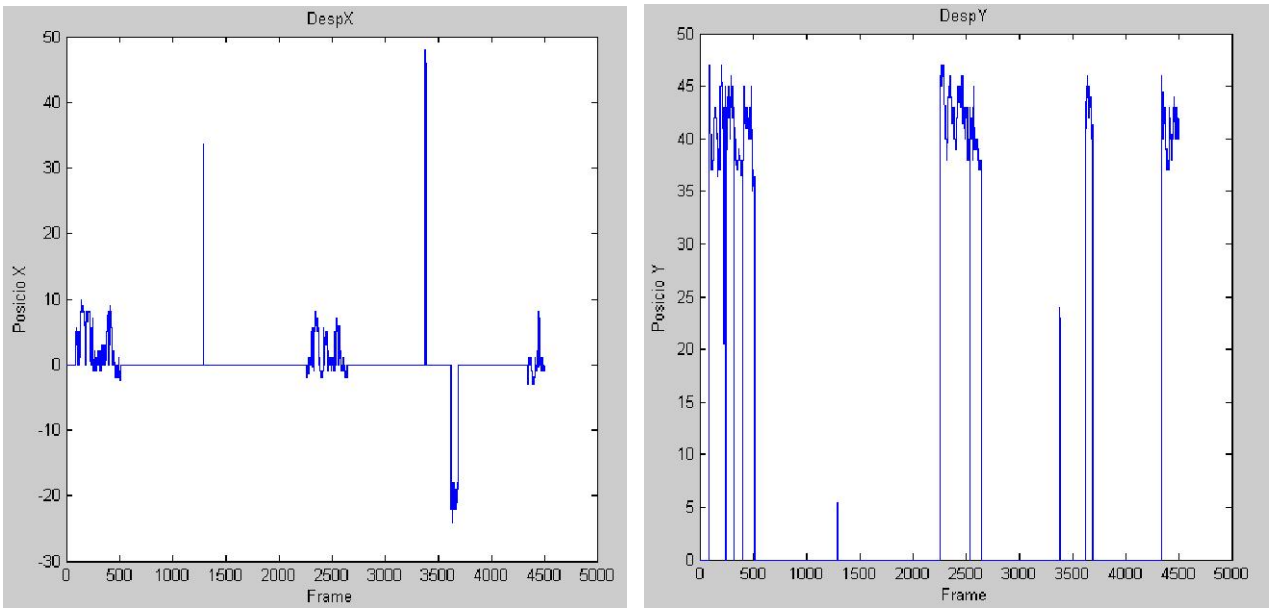


Figura: Gràfica de la posició el presentador d'un telenotícies de Antena 3

En la gràfica de l'esquerra, podem veure que certament, el presentador es va movent horitzontalment, tenint una tendència a la dreta. En el interval on està desplaçat a -20 cap a l'esquerra no es tracta del presentador sinó d'unes imatges d'entremig.

En la gràfica de la dreta podem veure que el presentador també es va desplaçant verticalment, força més que no pas horitzontalment, però s'ha de tenir en compte que el fet de mirar avall, encara que estiguis en el mateix lloc, OpenCV detecta la cara un pel per sota, ja que els ulls nas i boca s'han desplaçat mínimament.

Color

Per poder veure bé el cas del color, no ens serveix una gràfica dels valors de cada valor RGB, sinó que ens aniria bé poder veure una espècie de mosaic amb el color mitjà de cada frame. Per fer això, he creat una matriu quadrada de dimensions segons el número total de frames. A cada valor de la matriu, hi poso un vector de tres dimensions amb els valors RGB del frame corresponent, i finalment, se l'hi ha de dir al Matlab que ens ho mostri com una imatge, el que obtenim és el següent:

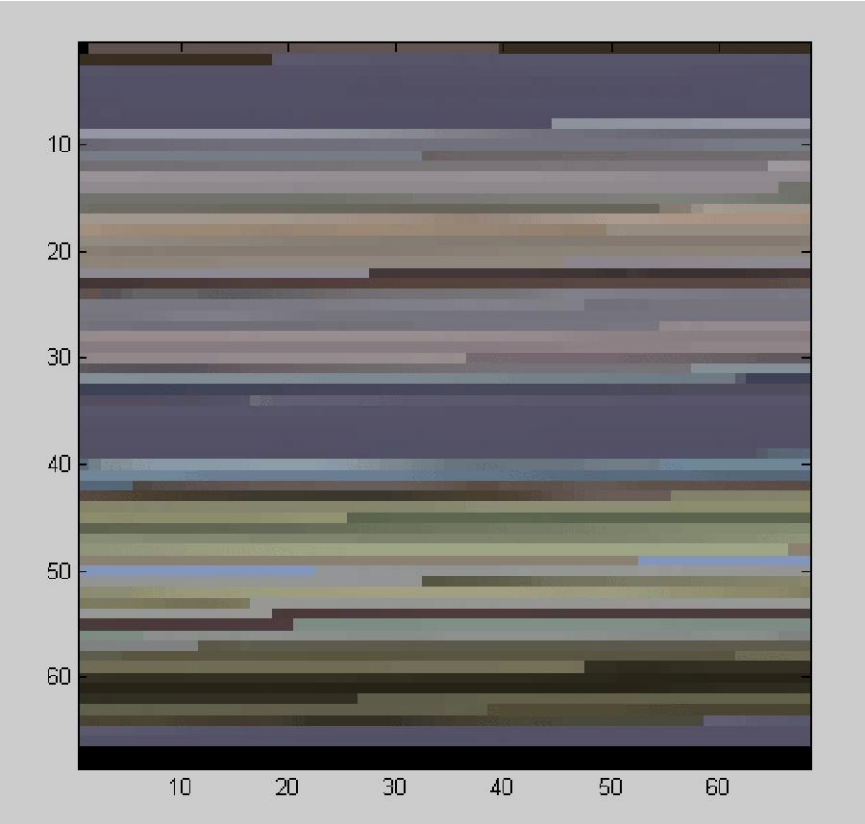


Figura: Mosaic dels colors de cada frame d'un telenotícies d'Antena 3

En aquest mosaic també es pot veure clarament a quins trossos hi ha el presentador del telenotícies. El color mitjà de l'escena dels presentadors és doncs, la mitja entre el color de fons i el color del vestit del presentador ja que ocupa gran part d'aquesta.

Intensitat

La gràfica de la intensitat ens mostra la il·luminació de cada frame, podem veure que en els trossos que hi ha el presentador és força constant, però a la resta no se'n pot extreure cap conclusió, a més, només ens interessarà els trossos on hi hagi el presentador del telenotícies.

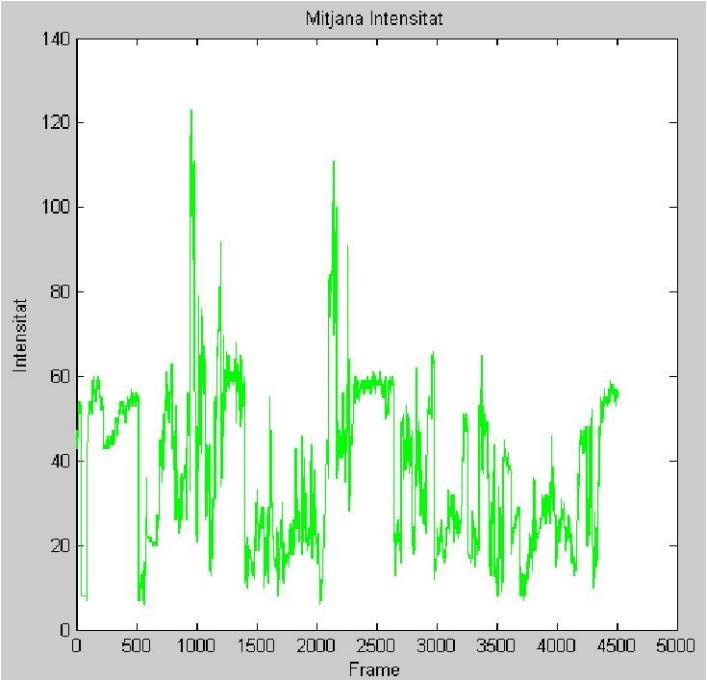


Figura: Gràfica de la intensitat de cada frame d'un telenotícies d'Antena 3

Contrast

El contrast de cada frame es força semblant a la intensitat, en els trossos on hi ha el presentador és força constant i ens pot servir per diferenciar els diferents telenotícies, per veure-ho més clarament, haurem de utilitzar el classificador i veurem en quin grau ajuda a diferenciar-los.

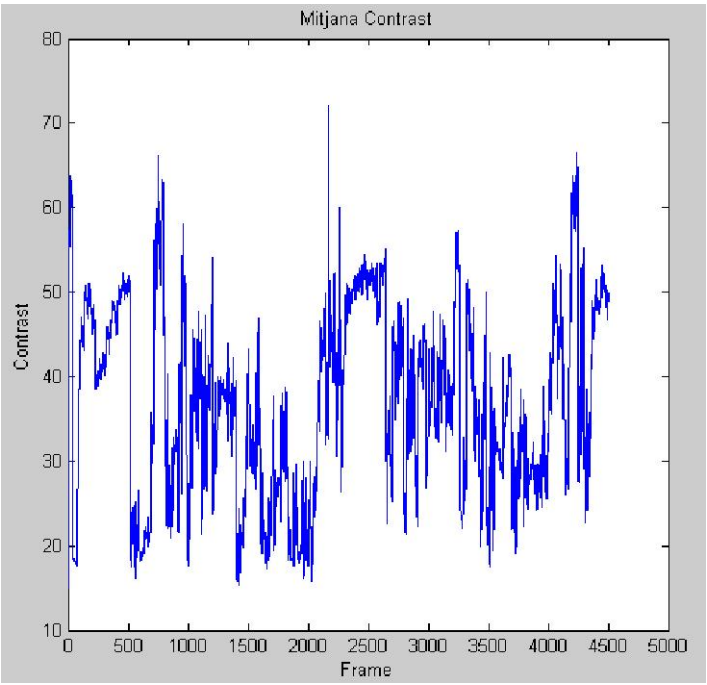


Figura: Gràfica del contrast de cada frame d'un telenotícies d'Antena 3

Canvis d'escena

Aquesta gràfica és un pel confusa, podem veure també clarament els llocs on hi ha el presentador, que és on no hi ha canvi d'escena, però també podem veure els falsos positius que ens dona a causa de moviments suficientment ràpids de la càmera que fan saltar al detector. Tot i això, com ja he comentat en les altres gràfiques, només ens interessen els trossos on hi ha els presentadors per tant, no ens hem de preocupar per aquests espais entremitjos.

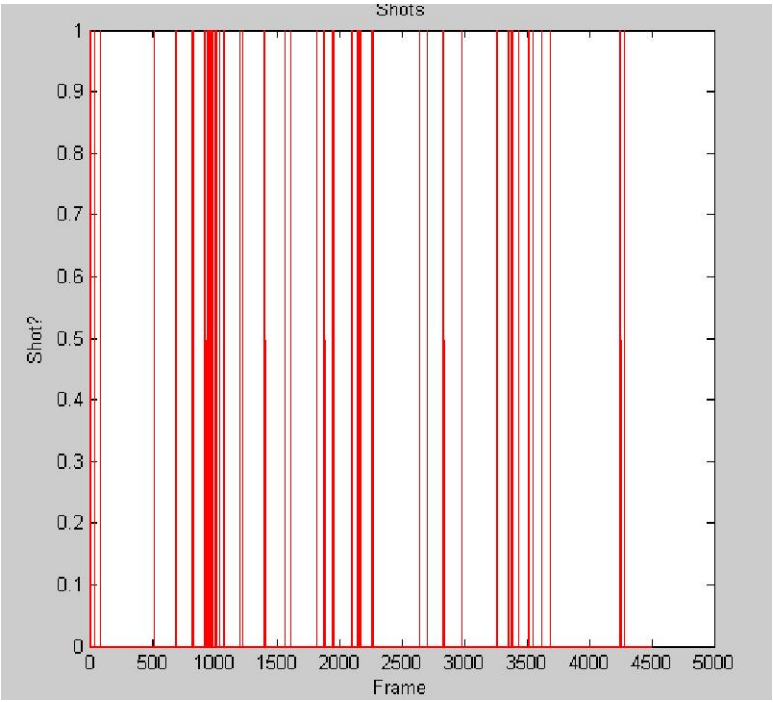


Figura: Gràfica dels canvis d'escena d'un telenotícies d'Antena 3

Vistes aquestes gràfiques podem dir que la generació dels descriptors és totalment correcte i el que ara caldrà comprovar, es que efectivament ens poden servir per diferenciar els diferents telenotícies, i en quin grau o poden arribar a fer.

3.6.2. Comparador

Aquest comparador ens servirà per poder veure quins són els frames més semblants entre telenotícies de diferents cadenes i lo diferents que son entre ells. Per realitzar aquest comparador, ens hem basat amb la distància euclidiana, que consisteix en la següent fórmula.

$$Dist = \sqrt{\left((Car1_1 - Car1_2)^2 + (Car2_1 - Car2_2)^2 ... (Car9_1 - Car9_2)^2\right)}$$

Aquesta distància obtinguda ens donarà lo lluny que esta cada frame del que vulguem comparar, però s'ha de tenir en compte que s'han de ponderar aquestes característiques, ja que potser un frame en que la posició escala i desplaçaments son molt semblants, però el color de l'escena és molt diferent ens dirà que esta molt més llunya que un en que la posició i escala és més diferent però el color és molt més semblant. Aquí cal dons, marcar unes prioritats en les característiques, quines volem que tinguin més pes i quines menys.

El que hem fet és, d'un frame d'origen, obtenir els seus descriptor, tasca senzilla ja que tenim un fitxer XML per a cada frame amb els seus descriptors, i seguidament, recorre de dalt a baix el telenotícies en que vulguem buscar els frames semblants, anant calculant la distància amb el frame d'origen, i a la vegada, guardant el número de frame i la seva distància en un vector, per finalment ordenar-lo i obtenir els frames més semblants.

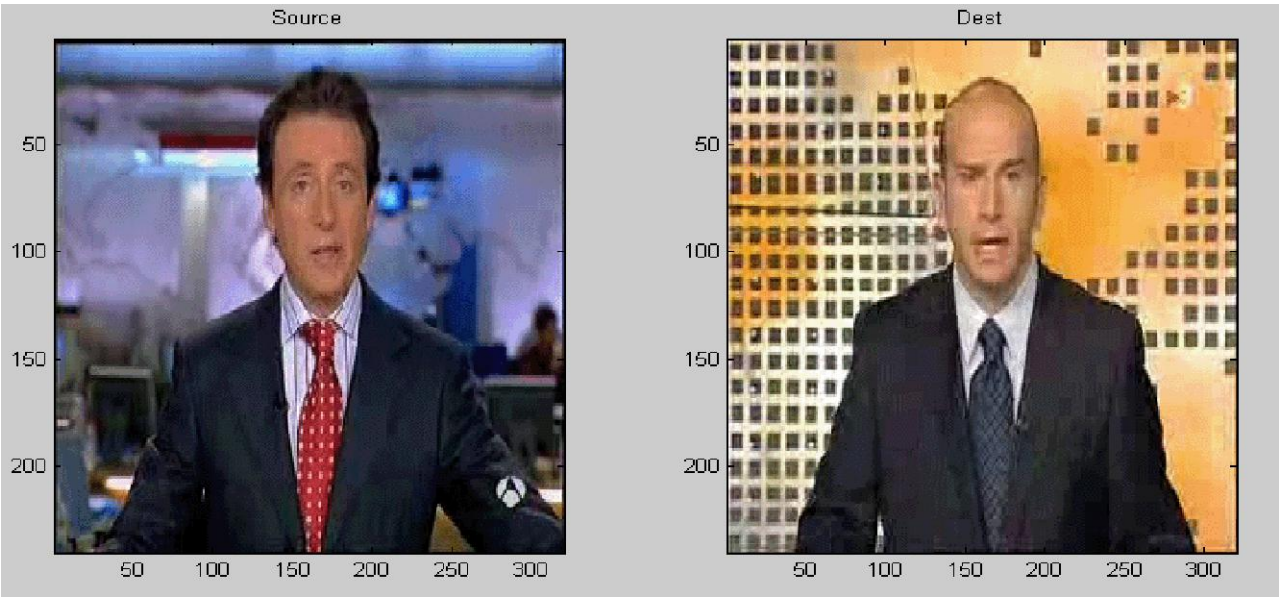


Figura: Resultat del comparador amb un frame del presentador d'Antena 3 sobre un telenotícies de TV3

En la següent imatge podem veure el frame d'origen d'un telenotícies d'antena 3 i el frame que ens troba com a més semblant de TV3 utilitzant pesos equivalents per a cada descriptor. Podem observar que efectivament ens dona un frame on la posició i escala de la cara és si més no la mateixa, però el color de fons és considerablement diferent, si en comptes d'agafar aquest frame en comprovem un altre, el resultat és totalment diferent i

el que ens retorna és un frame on el color és més semblant que no pas la posició i escala de la cara.

3.7. Classificador

Finalment, només ens queda realitzar el classificador de telenotícies, ho farem primer amb els descriptors de vídeo i so separadament, i després tots conjuntament. Així podrem veure l'eficàcia de cada part, i finalment la conjunta que hauria de ser la més bona.

Per poder utilitzar els classificadors conjuntament necessitem tenir les dades en un format estàndard, i el que varem decidir va ser utilitzar espais de 5 segons, els quals estarien solapats un segon amb el anterior. Això ho varem escollir d'aquesta forma perquè per les dades del so, temps inferiors són gairebé insignificatius. D'altra banda, com que les dades del vídeo estan en XML per a cada segon, serà trivial aconseguir aquesta nova representació.
Una altre modificació que varem haver de fer, i que ens elimina tots els errors esmentats anteriorment en cada una de les gràfiques, es que per poder realitzar un bon classificador, ens centrarem només en els presentadors dels telenotícies, abandonant completament la resta, per això caldrà doncs, triar manualment en quins intervals de temps apareixen per finalment escollir només els descriptors d'aquests intervals, si no fos així, poc tindríem a fer amb les característiques de parla i de vídeo, ja que en tot el telenotícies pot estar parlant força gent i en escenes completament variades.

Per classificar doncs, hem utilitzat el la suite d'eines PRTools4, la qual ens proporciona diferents classificadors els quals son:

1. Bayesià lineal
2. Bayesià quadràtic
3. Parzen
4. Xarxa neural amb Backpropagation de 3 unitats ocultes.
5. Lineal amb expansió Karhunen–Loève de la matriu de convolució
6. Lineal amb expansió PCA
7. Lineal logístic
8. Lineal Least Squared Error

9. Mixtura de gaussianes
10. K veí més proper
11. Densitat Parzen
12. Xarxa neural amb Levenberg-Marquardt
13. Xarxa neural amb radial basis

En els classificadors hem dividit les mostres en aprenentatge i test. El número de mostres d'aprenentatge ha estat inicialment 250, ja que la teoria ens diu que per el número de característiques que tenim fan falta aquest número de mostres, però pot variar molt depenent de la validesa d'aquestes.

3.8. Resultats

A continuació mostrarem els diferents resultats obtinguts, primer mostrarem les diferències entre els quatre telenotícies, seguidament, la classificació obtinguda amb només els descriptors de vídeo i finalment, la classificació obtinguda amb els descriptors de vídeo i so.

3.8.1. Diferències entre telenotícies: Vídeo

Un cop ja tenim tots els vídeos analitzats i tenim els descriptors en matrius del Matlab, es convenient veure gràficament les principals diferencies entre ells.

En les següents gràfiques les barres estan ordenades de la següent manera:

1. TV3
2. Antena 3
3. Canal Latino
4. Televisió Sant Cugat

Escala

Podem observar que la escala de la cara varia una mica entre els telenotícies de TV3, Antena 3 i el Canal Latino, dels quals TV3 és en el que apareix la cara més gran. En canvi, es pot veure clarament que en els telenotícies de Televisió Sant Cugat, és on l'escala de la cara és més petita, i amb una diferencia força notable a les altres cadenes.

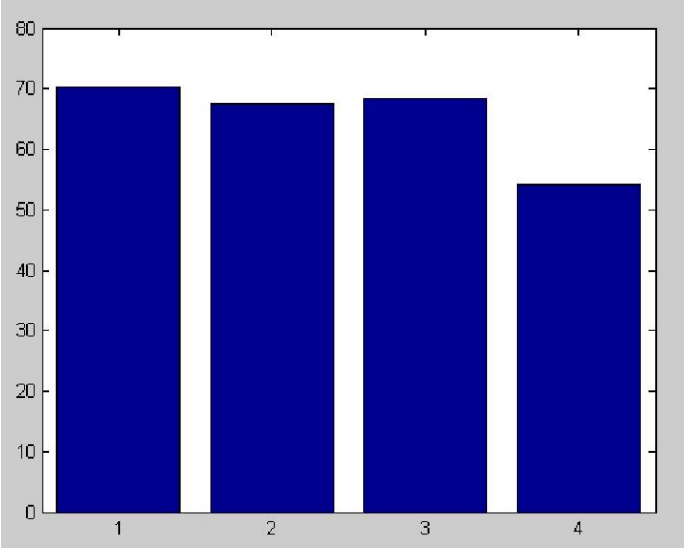


Figura: Gràfica de l'escala dels diferents telenotícies

Desplaçament Horitzontal

Aquesta característica si que es força impressionant, ja que es pot veure com varia el desplaçament horitzontal en cada cadena. Podem veure que el presentador que està més centrat es el d'Antena 3 amb un lleuger desplaçament a la dreta, seguit del Canal Latino, després podem veure que Televisió sant Cugat és l'únic que té un desplaçament cap a l'esquerra, i més considerable que en les altres dos cadenes. Però on hi ha la gran sorpresa és en els telenotícies de TV3, on hi ha un gran desplaçament a la dreta, molt més significatiu que en els altres. Observant els telenotícies hem pogut veure que en els de tv3, molts cops que entra la càmera, està desviada a la dreta i ha de rectificar. Aquest segurament es la causa d'aquest gran desnivell.

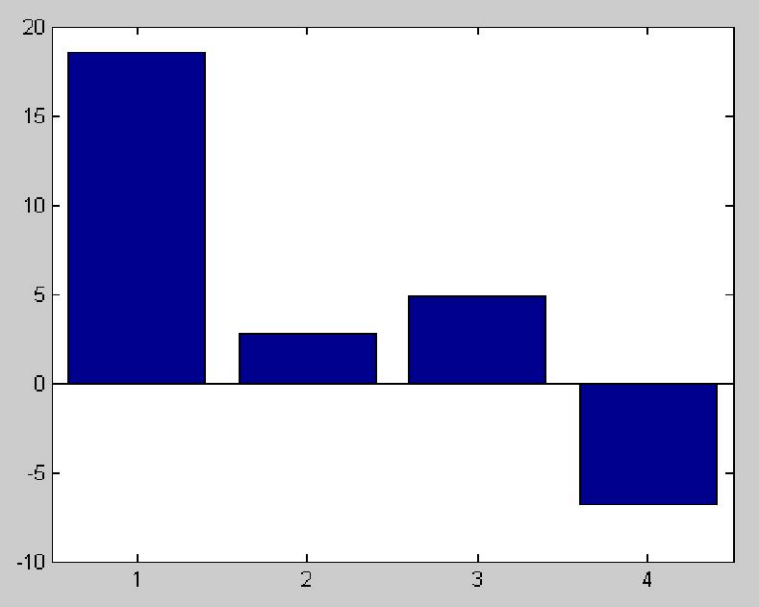


Figura: Gràfica del desplaçament horitzontal dels diferents telenotícies

Desplaçament Vertical

A diferència del desplaçament horitzontal, aquí podem veure que les quatre cadenes situen les cares per la meitat superior de l'escena, sent TV3 qui està més amunt i Antena 3 qui menys.

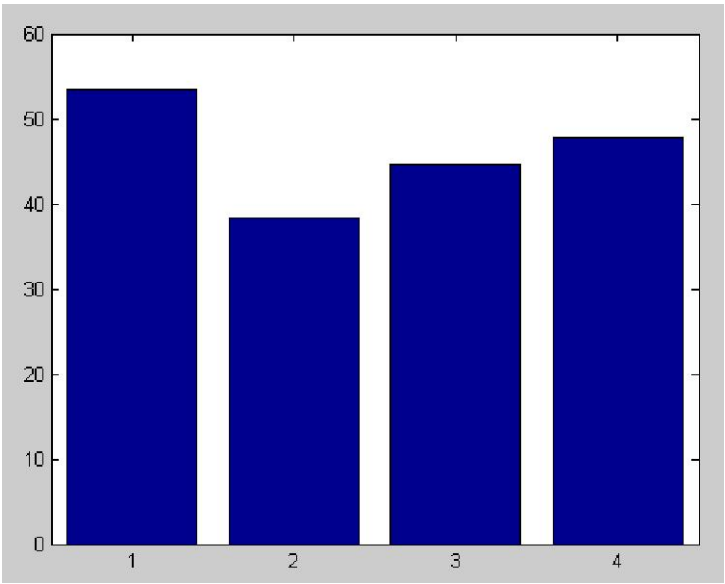


Figura: Gràfica del desplaçament vertical dels diferents telenotícies

Intensitat

La intensitat ens diu quin telenotícies és més clar i quin més fosc. En aquesta gràfica i ha grans diferències i TV3 és el altre vegada la que en té més, però podem veure també, que Canal Latino és força més fosc que tots els altres.

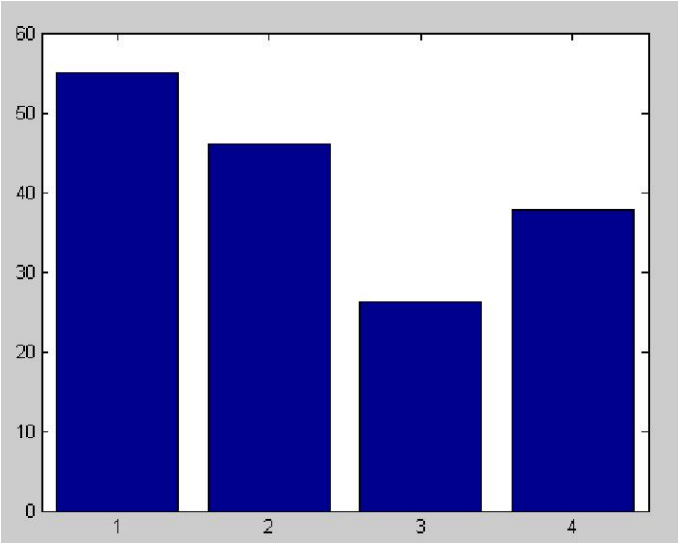


Figura: Gràfica de la intensitat dels diferents telenotícies

Contrast

En aquesta gràfica podem apreciar que les quatre cadenes tenen un contrast força semblant, sent altre cop el Canal Latino el que en te menys, segurament degut a l'escena general més fosca vista prèviament.

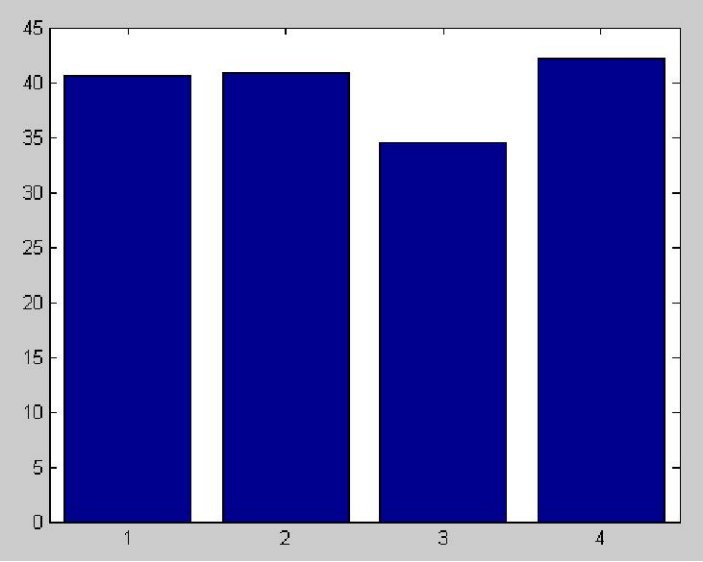


Figura: Gràfica del contrast dels diferents telenotícies

Color

Finalment, podem veure les grans diferències entre el color, fortament lligades amb la intensitat de la escena.

Figura: Gràfica del color mitjà dels diferents telenotícies

Cal notar, que les gràfiques de cada cadena per separat, hi ha grans variacions, sobretot en el Canal Latino i en Televisió Sant Cugat. Això ens afirma que hi ha diferències entre les cadenes amb més pressupost i amb menys, sent unes d'elles la poca uniformitat en les diferents emissions.

A la vista d'aquests resultats, només ens queda aplicar el classificador per veure si realment aquests descriptors són prou bons per diferenciar els diferents telenotícies de cada cadena.

3.8.2. Classificador de telenotícies: Vídeo

A continuació mostro la gràfica dels errors dels diferents classificadors obtinguts en el cas d'utilitzar només els descriptors de vídeo.

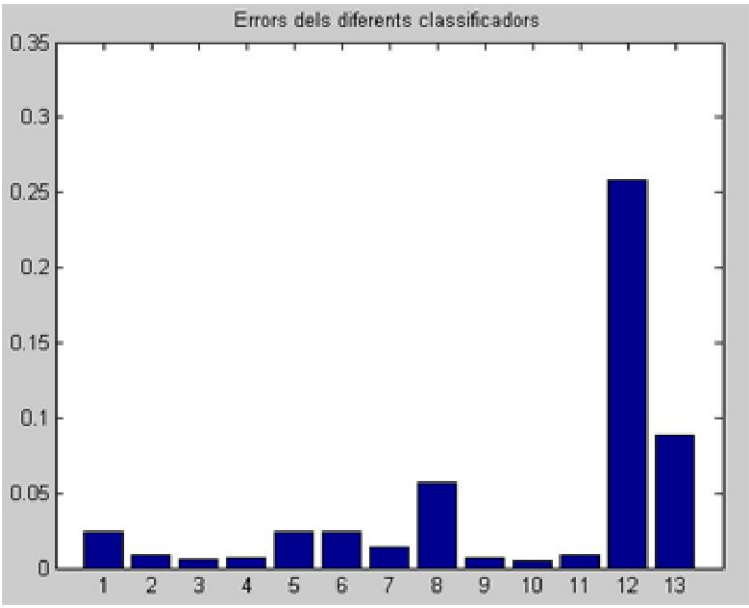


Figura: Gràfica de l'error en els diferents classificadors amb els descriptors de vídeo

Podem veure que funciona molt bé, exceptuant els classificadors *Levenberg-Marquardt* i el *radial basis* que passen del 10% d'error, però la resta són força baixos, on el que ha obtingut un error més petit ha estat el de *K veïns propers* amb un error del 0.0047, que prou baix com per poder dir que és un classificador excel·lent.

Si mirem quines són les característiques que tenen més pes a l'hora de classificar amb els descriptors de vídeo, podem veure que són els següents, ordenats de més rellevants a menys:

1. Color Vermell
2. Escala
3. Desplaçament vertical
4. Color Blau
5. Intensitat
6. Desplaçament horitzontal
7. Color Verd
8. Contrast.

Ens mostra que el descriptor el qual ens permet classificar més és el color vermell, si mirem els diferents telenotícies podem veure que efectivament tenen un color prou diferent. Seguidament veiem que és l'escala, en la gràfica de l'escala mitjana podem veure que és força semblant, però no s'hi veuen reflexades les variacions que sens dubte és un factor que ajuda a classificar-los. D'altra banda, el desplaçament horitzontal que segons les gràfiques del desplaçament mitjà semblava que seria un factor força important, ha quedat dels últims en el ranking.

3.8.3. Classificador de telenotícies: Vídeo i so

Finalment, ajuntant els descriptors de vídeo i de so, podem veure que el resultat és un èxit de nou, amb els descriptors de vídeo teníem un error molt baix, però ara, juntament amb els de so l'obtenim encara més baix i en més classificadors, exceptuant el de la *xarxa neural amb entrenament radial basis* que es dispara.

Aquest errors varien una mica cada vegada perquè fem l'entrenament amb 250 mostres aleatòries cada vegada. En la següent gràfica podem una prova la qual l'error mínim és de **0.0023** amb el classificador *Levenberg-Marquardt* que paradoxalment és dels que funcionava més malament amb les dades de vídeo soles.

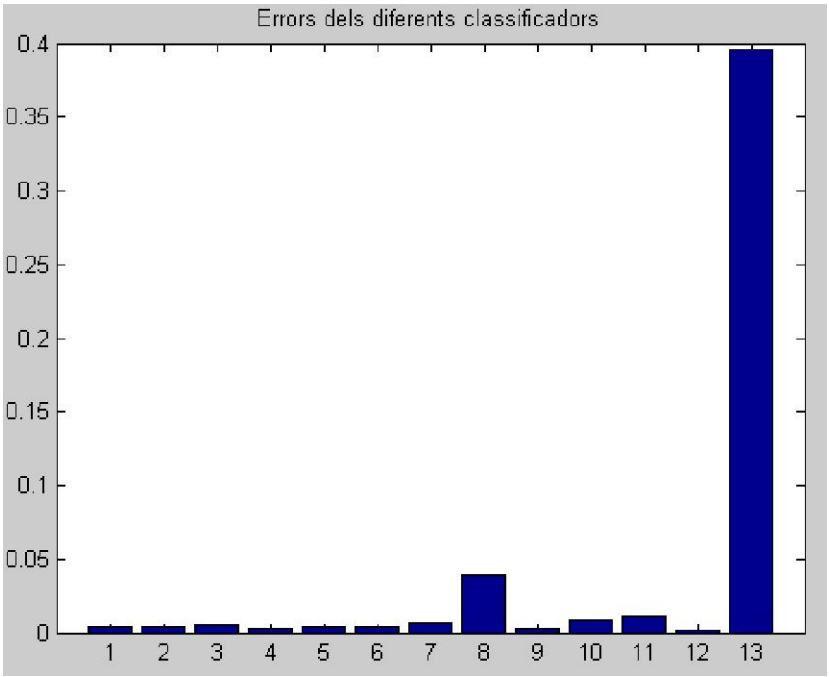


Figura: Gràfica de l'error en els diferents classificadors amb els descriptors de vídeo i so

La majoria de classificadors no arriben a l'1%, per tant, podem que els classificadors seran capaços de diferenciar correctament els diferents telenotícies amb els descriptors que hem extret.

Les sis primeres característiques, que són les que bàsicament influencien en la classificació són:

1. Sonoritat (o volum)
2. Color vermell
3. To
4. Escala
5. Desplaçament vertical

Veiem que estan força repartides entre els descriptors de vídeo i àudio la qual cosa fa que augmenti aquesta eficàcia dels diferents classificadors.

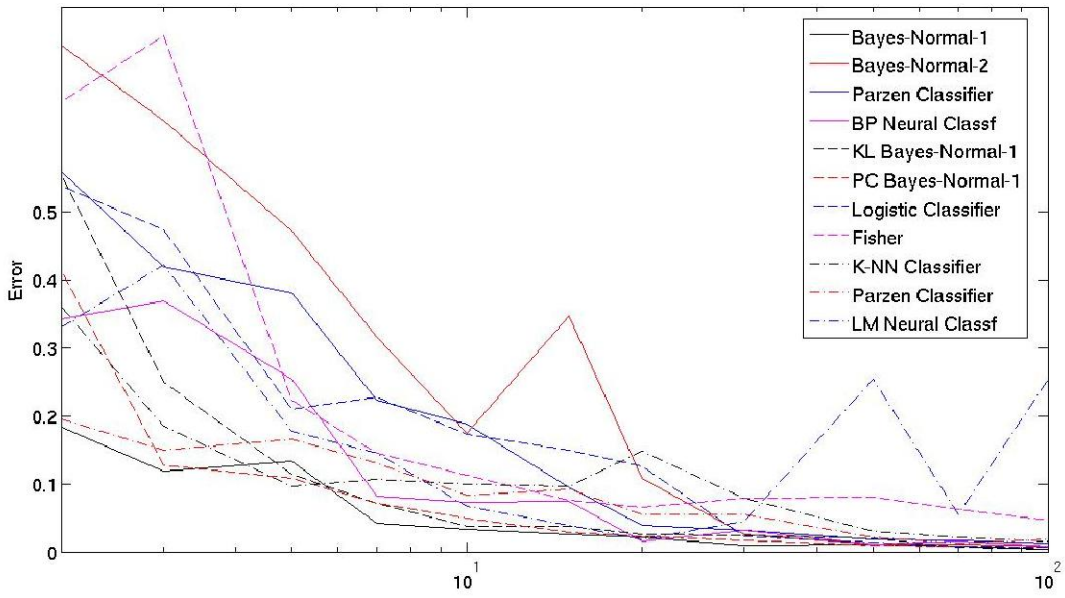


Figura: Corba d'errors dels diferents classificadors

Si mirem la corba d'error segons el número de mostres, podem veure que la majoria de classificadors, a partir de les 150 mostres ja tenen un error molt baix, i que a les 250 l'error és pràcticament nul.

Provant amb més mostres, podem observar que en alguns classificadors, l'error augmenta considerablement en comptes de baixar que seria lo lògic, això es degut a que es produeix memorització, la qual cosa significa que les mostres de test que estiguin als límits entre les categories poden acabar sent classificades malament.

CONCLUSIONS

4. Conclusions

4.1. Objectius assolits

Podem dir que els principals objectius d'aquest projecte han estat assolits molt satisfactòriament. La indexació de continguts automàtica de programes televisius, tant per vídeo com per àudio és possible, i a més ràpida i eficaç.
Hem aconseguit diferenciar els telenotícies de les diferents cadenes de televisió a partir dels descriptors que hem cregut adequats amb un error gairebé nul, la qual cosa ens confirma aquesta possible indexació, i també que els descriptors triats són totalment correctes.

A més, hem pogut extreure aquests descriptors dels vídeos, a una velocitat una mica més ràpida que temps real, tot i que en la extracció dels descriptors del so no ha pogut ser així i requereix força més temps.

Poem concloure també, que la suposició inicial de que hi ha diferències importants entre les cadenes que tenen més pressupost i amb les que en tenen menys és totalment certa. Sent una d'elles la poca uniformitat en alguns descriptors en les diferents emissions dels telenotícies.

4.2. Principals aportacions

En un futur molt proper, el que hem començat en aquest projecte, pot servir per tirar endavant tot un seguit de noves idees referents a la nova televisió.
Hem creat un sistema que ens extreu les característiques de so i de vídeo, i que les guarda en el format XML, el qual pot ser utilitzat molt fàcilment per qualsevol programa per acabar-ne traient les seves conclusions, per classificar els programes segons aquestes característiques.

Creiem que l'aportació més important, són aquestes noves dades, que ens permeten diferenciar els diferents programes a partir solament d'elles. Això sens dubte, és una propietat de la que se'n pot treure molt profit, i que es poden crear noves aplicacions que la utilitzin per noves finalitats que ara mateix no se'ns poden ni acudir.

Nosaltres tan sols ens hem basat en els telenotícies, però aquesta extracció de descriptors por ser duta a terme en qualsevol programa de televisió, ja sigui pel·lícula, sèrie, anuncis etc... Gràcies a totes aquestes dades, es podrà començar una investigació sobre les sensacions que ens transmeten i com ens afecten els esmentats programes televisius.

4.3. Línies de continuació

La primera línia de continuació, i la més important de cara a aquest projecte, seria l'extracció de més descriptors rellevants tan al vídeo com en el so. Com que ens hem basat en els telenotícies, només ens hem centrat en les cares, però en un entorn més divers, com poden ser la resta de programes, caldria tenir en compte més aspectes referents a les persones, com ara si surt sencer, de mig cos, o només la cara. Un altre descriptor que ajudaria molt a fer una molt bona indexació, seria reconèixer les persones que surten. En el cas dels telenotícies seria força fàcil, però en altres programes on surt molta més gent, i molts cops desconeguda per la majoria, ja no seria una tasca tan fàcil.

Dins d'aquests nous descriptors a extreure, també s'haurien de tenir en compte les lletres, així com les frases, subtítols, noms de marques etc... també la bellesa general de la imatge, o la bellesa particular de cada persona que apareix en el programa.

Tots aquest nous descriptors, juntament amb els que ja tenim, ens proporcionarien una base de dades realment completa i de la qual podríem indexar els continguts d'una manera molt eficaç.

Finalment la següent línia de continuació, seria l'anàlisi de les emocions que ens transmeten aquests programes televisius a partir dels descriptors extrets tan de vídeo com se so. Aquesta però, seria una tasca complexa ja que, les emocions de l'esser humà ja ho son i encara no estan massa ben definides.
Aquestes emocions, també les podríem afegir com a possibles descriptors, i així, tenir un sistema encara més complet per la indexació de continguts. Tot això ajuntat amb les noves tecnologies, segurament seria una nova forma d'entendre la televisió, i fer així, una revolució que estem a punt de viure.

5. Bibliografia

[1] Llibreria OpenCV
http://sourceforge.net/projects/opencvlibrary/

[2] Wikipedia OpenCV
http://opencvlibrary.sourceforge.net/

[3] Microsoft DirectShow API Help
http://msdn2.microsoft.com/en-us/library/ms783323.aspx

[4] Michele Covell, Shumeet Baluja, Michael Fink. Detecting Ads in Video Streams Using Acoustic and Visual Cues.

[4] Wikipedia DirectShow
http://en.wikipedia.org/wiki/DirectShow

[5] PrTools (Classifier Matlab ToolBox)
R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, *PRTools4, A Matlab Toolbox for Pattern Recognition, Delft University of Technology*.

[6] Viquipèdia
http://ca.wikipedia.org/

[8] XML
http://www.w3.org/TR/REC-xml/

[9] OpenCV Google Groups
http://groups.google.co.uk/group/OpenCV/topics

[10] Comparison of Automatic Shot Boundary Detection Algorithms
http://www.lienhart.de/Source_Code/source_code.html

[11] Alan Hanjalic. *Extracting Moods from Pictures and Sounds*

[12] Joost

http://joost.com/

[13] YouTube

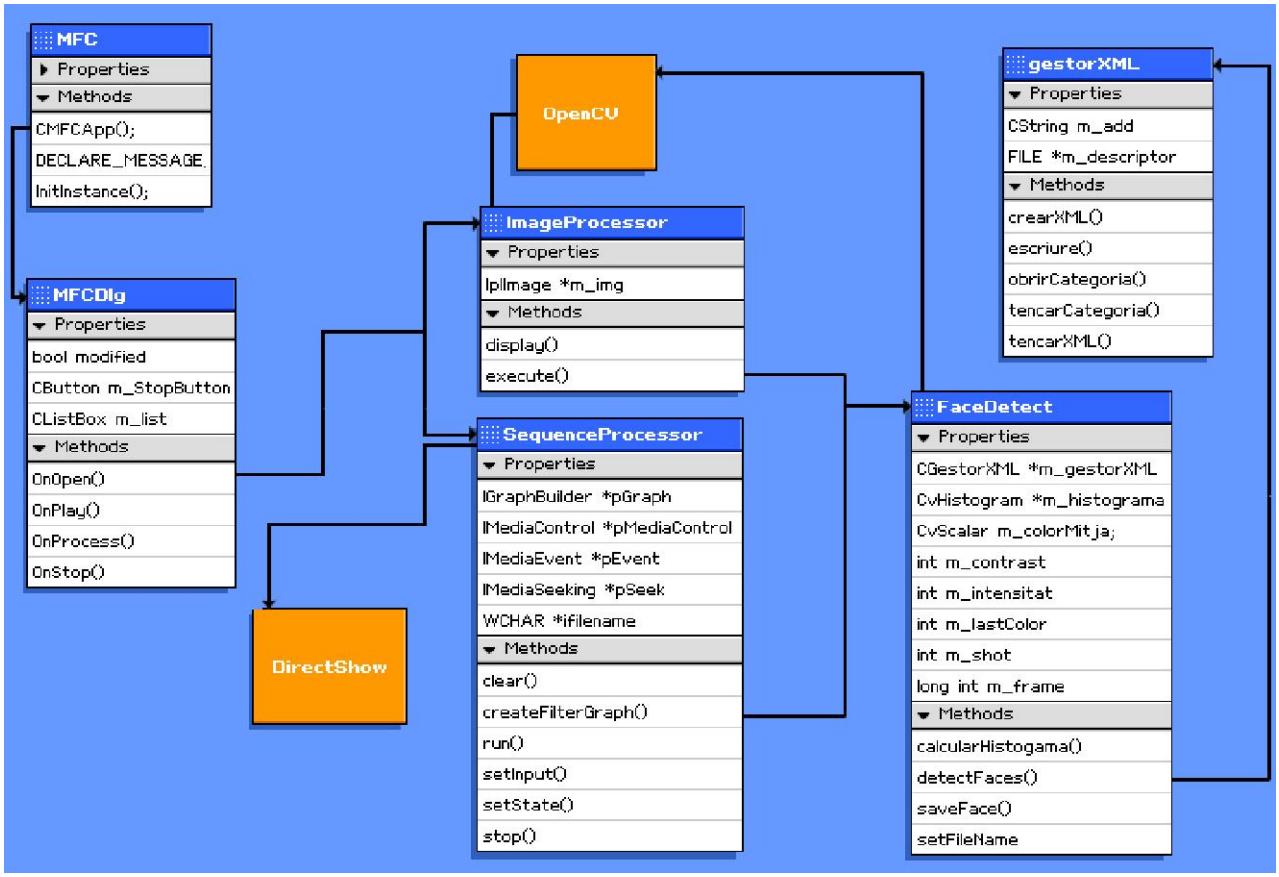
http://www.youtube.com

DOCUMENTS ANNEXES

6. Documents annexes

6.1. Diagrama de classes

Seguidament mostro el diagrama de classes de l'aplicació, juntament amb una breu explicació de cada una i com estan relacionades entre elles.



Les classes *MFC MFCDlg* no són de massa importància en el nostre cas, ja que tan sols són la interfície gràfica, la qual conté un boto per obrir els vídeos o imatges i un per començar a processar.

Quan s'obre una imatge o vídeo, es crea un objecte de la classe *ImageProcessor* o *SequenceProcessor* respectivament, que serà l'encarregat de després crear l'objecte *FaceDetect* que s'encarregarà de extreure els descriptors de la imatge.
A continuació faig una breu descripció de les propietats i mètodes més importants:

ImagePorcessor

Propietats

```
IplImage * m_img;
```

Imatge en el format de OpenCV per poder-la tractar còmodament.

Mètodes

```
void  display ( void );
```
Funció que ens mostra la imatge per pantalla.
```
void  execute ( void );
```
Funció que cridara el FaceDetect per extreure els descriptors de la imatge.

SequenceProcessor

Propietats

```
IGraphBuilder     *pGraph;
```
Variable on es creara el graf de filtres de DirectShow.
```
IMediaControl     *pMediaControl;
```
Variable per controlar la reproducció del clip multimèdia.
```
IMediaEvent       *pEvent;
```
Variable que ens recollira els events que passin en el fitxer multimèdia.
```
IMediaSeeking     *pSeek;
```
Variable per situar el clip multimèdia al segon desitjat.
```
WCHAR             *ifilename;
```
Variable amb el nom del fitxer que utilitzarem per a guardar més enevant els XMLs amb els descriptors extrets.

Mètodes

```
bool  createFilterGraph ( void );
```
Funció que ens creara el graf de filtres depenentment de l'extensio del fitxer.
```
void  setInput          ( CString inputfilename );
```
Funció que cridarem per establir un nou fitxer font per al qual crear el graf de filtres i seguentment reproduir.
```
void  run               ( void ) ;
```
Funció que ens reprodueix clip multimèdia segons on esta el pSeek.
```
void  stop              ( void );
```
Funció que atura la reproducció del clip multimèdia.
```
void  clear             ( void );
```
Funció que ens elimina tots els filtrs del graf per poder tornar a començar amb un altre fitxer.
```
void  setState          ( CString state );
```
Ens guarda l'estat per saber si esta reproduint-se o aturat.

FaceDetect

Propietats

```
CString           m_fileName;
```
Nom del fitxer obert per poder guardar el xmls amb el mateix nom.
```
long int          m_frame;
```
Número de frame que estem calculant.
```
CvHistogram       *m_histograma;
```

Histograma de la imatge.
CGestorXML *m_gestorXML;
Gestor xml per escriure els fitxers amb els descriptors.

Mètodes

void detectFaces (IplImage * img);
Funció que ens extreu els descriptors de la imatge i els guarda en fitxers xml.
void saveFace (IplImage * img , CvRect*, int ncara);
Funció que ens retalla les cares de la imatge i les guarda com a altres imatges per si més endevant es volen analitzar les cares en particular.
void setFileName (CString);
Funció que ens crea el nom a guardar per cada XML i imatge amb la cara retallada depenent del frame i el número de cara.
CvHistogram* calcularHistogama (IplImage * img);
Ens calcula l'histograma de la imatge, com també el valor mig i el contrast.

gestorXML

Propietats

FILE* m_descriptor;
Descriptor del fitxer XML que crearem.
CString m_add;
Variable amb els tabuladors necessaris per mantenir una certa estètica en el fitxer XML.

Mètodes

void crearXML (CString fileName, int frame, int nCara);
Funció que ens crea el fitxer i afageix el primer node.
void tencarXML (void);
Funció que tenca el fitxer i tenca el primer node.
void escriure (CString atribut, CString valor);
Funció que escriu un atribuit i el seu valor (String)
void escriure (CString atribut, int valor);
Funció que escriu un atribuit i el seu valor (Enter)
void obrirCategoria (CString categoria);
Ens crea un nou node *categoria*.
void tencarCategoria (CString categoria);
Funció que ens tanca el node *categoria.*

6.2. Classificadors

Durant la realització dels nostres experiments hem utilitzat una sèrie de classificadors per a mesurar els nostres resultats. A continuació exposem els trets característics més bàsics sobre el funcionament d'aquest tipus d'aplicacions:

Aprenentatge

És la part bàsica que tenen en comú els diferents tipus de classificadors que existeixen

La idea bàsica del aprenentatge consisteix en utilitzar les percepcions no només per actuar, sinó també per a millorar la habilitat d'un agent per actuar en el futur [17] .

Existeixen diversos tipus d'aprenentatge:

- Aprenentatge supervisat: Consisteix en aprendre una funció a partir d'exemples de les seves entrades i les seves sortides. No sempre és possible fer aquest tipus d'entrenament ja que hem de disposar de la sortida esperada en funció de la entrada.
- Aprenentatge no supervisat: Consisteix en aprendre a partir de patrons d'entrades pels quals no s'especifiquen els valors de les seves sortides.
- Aprenentatge per reforçament: L'agent ha d'aprendre a partir d'una funció de reforç, que li serveix de realimentació per a valorar la entrada. Un exemple d'aquest tipus d'aprenentatge seria el cas d'un agent que faci la funció de cambrer. Pot servir-li com a reforç la quantitat de propina que el client li ha donat.

La idea de l'aprenentatge consisteix en construir una funció que tingui el comportament observat en les seves dades d'entrada i de sortida. Els mètodes d'aprenentatge es poden entendre com la cerca d'un espai d'hipòtesis per a trobar la funció adequada. Partint d'una assumpció molt bàsica respecte a la funció.

6.2.1. Classificador Bayesià

Un classificador Bayesià és un classificador de patrons basat en teories estadístiques d'aprenentatge. A continuació mostrem el seu funcionament.

6.2.2. Aprenentatge Bayesià

L'aprenentatge bayesià calcula la probabilitat de cada hipòtesi de les dades, i realitza prediccions sobre aquestes bases. Es realitzaran prediccions fent servir totes les hipòtesis, ponderades amb les seves probabilitats.

Si D és el conjunt de dades i d el valor observat, la probabilitat de cada hipòtesis s'obté aplicant la regla de Bayes:

$$P(h_i \mid d) = \alpha P(d \mid h_i) P(h_i)$$

Suposem que volem fer una predicció sobre una quantitat desconeguda X:

$$P(X \mid d) == \sum_i P(X \mid d, h_i) P(h_i \mid d) = \sum_i P(X \mid h_i) P(h_i \mid d)$$

S'ha fet la assumpció que cada hipòtesi determina una distribució de probabilitats sobre X. La equació mostra que les prediccions son el resultat de ponderar les prediccions sobre les hipòtesis individuals.

6.2.3. Hipòtesi MAP

L'aprenentatge Bayesià és gairebé òptim, però requereix grans quantitats de càlcul degut a que l'espai d'hipòtesis és normalment molt gran , o inclús pot ser infinit. En la majoria dels casos, el càlcul del sumatori és intractable. Això ens obligarà a recórrer a mètodes aproximats, o bé mètodes simplificats.

Una aproximació molt habitual consisteix en fer les prediccions basant-nos en la hipòtesi més probable, una h_i tal que maximitzi $P(h_i \mid d)$.

Aquesta simplificació se la anomena màxim a posteriori, o bé hipòtesi MAP. Les prediccions que realitzen aquestes aproximacions són aproximadament Bayesianes, fins al punt que $P(X \mid d) \approx P(X \mid h_{MAP})$. Això és degut al següent: A mesura que arriben més dades, la predicció MAP i la Bayesiana s'acosten, perquè les competidores de la hipòtesi MAP es van fent menys probables. Trobar la hipòtesi MAP és habitualment més senzill que l'aprenentatge Bayesià, ja que només requereix resoldre un problema d'optimització. En l'aprenentatge Bayesià es requeria resoldre un problema d'integració, o bé resoldre un gran sumatori.

6.2.4. Classificador Parzen

Aquest classificador està basat en el histograma de les dades [22] . Estima les densitats de cada classe. Aquest és el seu algoritme:

Entrada: Conjunt de mostres d'entrenament T_s i conjunt de mostres de test T.

1. Seleccionar σ_h tal que: $\sum_{k=1}^{K}\sum_{j=1}^{N_k} \ln\left(\hat{p}(z_{k,j} \mid \omega_k)\right)$ sigui màxim.
2. Estimar la densitat: per a cada mostra z del conjunt de test calcular la densitat de cada classe: $\hat{p}(z \mid \omega_k) = \frac{1}{N_k} \sum_{z_j \in T_k} \frac{1}{\sigma_h^N \sqrt{(2\pi)^N}} \exp\left(-\frac{\|z - z_j\|^2}{2\sigma_h^2}\right)$
3. Classificació: Assignar les mostres de T a la classe amb una probabilitat màxima a posteriori: $\hat{\omega} = \omega_k \text{ amb } k = \arg\max_{i=1,\ldots,K}\left\{\hat{p}(z \mid \omega_i)\hat{P}(\omega_i)\right\}$

6.2.5. Classificador Backpropagation

El model de xarxa neural habitual que utilitza aquest algoritme consisteix en una xarxa neural amb una capa d'entrada amb tants nodes com entrades tinguem, una capa oculta amb un nombre de nodes variable que dependrà de les característiques del problema, i una capa de sortida amb tants nodes com possibles sortides tinguem. En el nostre cas, com que tenim quatre possibles classes (els quatre telenotícies) tindrà quatre nodes a la última capa. L'algoritme d'entrenament s'anomena Back Propagation (propagació cap a endarrere). Té aquest nom degut al seu funcionament. Inicialment, les entrades es propaguen cap a endavant fins a arribar a la sortida. Després, començant per la última capa, es calcula l'error i la contribució a l'error del sistema per a cada node o neurona. Aquesta contribució es calcula amb propagació cap a endarrere fins als nodes de la primera capa (per això el nom de Back Propagation). A partir de la contribució a l'error global de cada node individualment es modificarà el seu valor amb més o menys quantitat segons aquest valor.

6.2.6. Classificador amb expansió Karhunen-Loève

La expansió Karhunen-Loève és la representació d'un procés estocàstic com a combinació de diverses funcions ortogonals. És un concepte semblant al de les sèries de Fourier, però aquí els coeficients són variables aleatòries, i la expansió base depèn del procés, enlloc de basar-se en senyals sinusoïdals. Les funcions base depenen de la funció de covariància del procés.

Aquesta transformació, anomenada KLT, determinarà una sèrie de coeficients que resultaran útils per a la classificació basant-nos en aquestes dades, alhora que reduiran la dimensionalitat de les dades.

En el cas d'un procés estocàstic centrat {Xt}t [a,b] (on centrat significa que les expectatives E(Xt) estan definides a 0 per tot t) satisfaci una condició de continuïtat tècnica, admetrà una descomposició de la forma:

$$\mathbf{X}_t = \sum_{k=1}^{\infty} \mathbf{Z}_k e_k(t).$$

On $\mathbf{Z}_k$ són parelles correlatives de variables aleatòries, i les funcions e_k són continues amb valors reals a [a,b] el qual es una parella ortogonal a $L^2[a, b]$.

El cas general del procés que no està centrat, pot ser representat expandint la funció d'expectació (la qual es una funció no aleatòria) en la base e_k .

6.2.7. Classificador amb PCA

PCA és definit com a una transformació lineal ortogonal que transforma les dades en un nou sistema de coordenades on la variància màxima per qualsevol projecció de les dades s'estableix en la primera coordenada, anomenada el primer PCA (*Principal Component Analysis*), la segona màxima variància s'estableix en la segona coordenada, etc.

La transformació PCA equival a la transformació discreta de Karhunen-Loève.

6.2.8. Classificador logístic

Són classificadors basats en el model de regressió estadística per variables dependents amb distribució de Bernoulli.

La distribució logística està definida de la següent manera:

$$P(x) = \frac{e^{(x-m)/b}}{|b|[1+e^{(x-m)/b}]^2}$$

6.2.9. Classificador Least Squared Error

Aquest classificador sorgeix com a alternativa al perceptró, ja que aquest darrer no funciona bé en els casos separables. Si el conjunt d'entrenament no és separable, aleshores el procediment iteratiu tendirà a fluctuar al voltant de cert valor [22].

Aquest classificador fa servir una sèrie de vectors K-dimensionals, cadascun associat a una mostra de dades. Calcula els pesos d'aquests vectors mitjançant el criteri de least squares. La solució serà el valor que minimitzi el Least Squared Error.

6.2.10. Classificador amb mixtura de Gaussianes

Aquest classificador [22] , basat en el K veí més proper, assumeix que els objectes en cadascun dels K conjunts existents estan distribuïts amb una distribució gaussiana. Cada conjunt està caracteritzat per la seva mitjana aritmètica (μ_k) i per la seva matriu de covariància (C_k):

$$p(z) = \sum_{k=1}^{K} \pi_k N(z \mid \mu_k, C_k)$$

6.2.11. Classificador K-veí més proper

Aquests classificadors es basen en els models de veïns més propers [17] . Es basen en el següent: és probable que les propietats d'un punt d'entrada particular x siguin similars a les dels punts propers a x.

En aquests mètodes fa falta especificar exactament què és el que s'entén per veí. Si el veïnatge és massa petit, aleshores no contindrà cap punt. En canvi, si és massa gran, pot ser que inclogui tots els punts de dades. La solució a aquest problema consisteix en definir un veïnatge suficientment gran com per a incloure k punts, on k és prou gran com per a assegurar una estimació amb significat. Per a un k fix, la mida del veïnatge varia segons la distribució de les dades. Si les dades estan disperses, el veïnatge serà gran; en canvi, si les dades estan molt properes el veïnatge serà petit.
És evident que ens farà falta una mètrica per a mesurar la distància entre dos punts. La distància euclidiana no és sempre la millor opció. Quan cada dimensió de l'espai es mesura de forma diferent la distància euclidiana no és el més adequat, perquè canviar una escala d'una dimensió podria canviar el conjunt de veïns més propers. Un exemple seria la alçada i el pes.

La solució a aquest problema consisteix a estandarditzar la escala de cada dimensió. Això es fa de la següent manera: mesurem la desviació estàndard de cada característica sobre el conjunt de les dades i expressem els valors com a múltiples de la desviació estàndard d'aquesta característica.
L'aprenentatge supervisat mitjançant aquesta tècnica es fa de la següent manera:
Donat un exemple de test amb entrada *x*, la sortida $y = h(x)$ s'obtindrà a partir dels valors y dels k veïns més propers a x.

Hem de diferenciar el cas simple i el cas continu:
En el cas simple, es pot obtenir la predicció mitjançant el vot de la majoria.
En el cas continu, calculem la mitjana dels k valors, o bé calculem la regressió lineal, ajustant un hiperplà a k punts i predient el valor de x a partir d'aquest hiperplà.
El classificador és simple [22]. La classe assignada a un vector z és la classe amb el màxim nombre de vots de k mostres properes a z.
Si k_k denota el nombre de mostres trobades a la classe w_k, aleshores:

$$\hat{\omega}(z) = \omega_k \qquad \text{amb} \qquad k = \arg\max_{i=1,\ldots,K}\{k_i\}$$

6.2.12. Xarxa neural Levenberg-Marquardt

Aquesta xarxa utilitza un algoritme basat en el mètode de Newton. És un algoritme iteratiu d'optimització en el que el mètode d'iteració està lleugerament modificat respecte l'original. Se'n obté un bon rendiment en l'entrenament de xarxes neurals on el rendiment de la xarxa estigui determinat per l'error mig quadràtic

Interpola entre l'algorisme de Gauss-Newton i el mètode del gradient. LMA és més robuts que no pas el GNA, el que significa que en molts casos troba una solució tot i que comenci força lluny del mínim final. Per altra banda, per funcions que tenen un punt de començament raonable, és un pel més lent que el GNA.

La principal aplicació és en el problema de l'encaix de la curva quadràtica. Donat un conjunt de parelles de dades empíriques (t_i, y_i), optimitzar els paràmetres **p** del model de la curva $f(t|\boldsymbol{p})$ fent Aixa que la suma dels quadrats de les derivades esdevingui mínim.

$$S(\mathbf{p}) = \sum_{i=1}^{m} [y_i - f(t_i \mid \mathbf{p})]^2$$

6.2.13. Xarxa neural Radial Basis

Aquesta xarxa fa servir funcions radials com a funcions d'activació, és a dir, funcions amb valors reals que depenen únicament de la distància a l'origen.

L'estructura típica consisteix en tres capes. Una primera capa d'entrada, una capa oculta amb funcions d'activació radials no lineals i una capa de sortida.

Un vector X d'entrada és usat com a entrada a les funcions radials, cadascuna amb paràmetres diferents. La sortida de la xarxa consisteix en una combinació lineal de les sortides de les funcions radials.

Qualsevol funció φ qeu satisfaci la propietat $\varphi(\boldsymbol{x})=\varphi(||\boldsymbol{x}||)$, es una funció radial. La distancia és normalment la distancia euclidiana.

Les funcions Radial Basis són típicament usades per a construir aproximacions de funcions de la forma

$$y(\mathbf{x}) = \sum_{i=1}^{N} w_i \, \phi(||\mathbf{x} - \mathbf{c}_i||),$$

On la funció y(x) aproximada és representada com la suma de N funcions radial basis, cada una associada amb centres diferents $\boldsymbol{c}_i$, i balancejats per un coeficient w_i. Aquest tipus de funció han estat particularment usats en prediccions de series de temps i en el control de sistemes no lineals que mostren un comportament caòtic simple.

.

Jordi Hernàndez Puigdellívol

Manresa, 10 de Juny de 2007

Indexació de continguts televisius: Anàlisis de vídeo

Aquest és un projecte que tracta sobre la indexació automàtica de continguts televisius. És una tasca que guanyarà importància amb els imminents canvis que hi haurà en la televisió que coneixem. L'entrada de la nova televisió digital, farà que hi hagi una interacció molt més fluida entre l'espectador i la cadena, a més de grans quantitats de canals, cada un amb programes de tipus totalment diferents. Tot això farà que tenir mètodes de cerca basats en els continguts d'aquests programes sigui del tot imprescindible. Així doncs, el nostre projecte està basat plenament en poder extreure alguns d'aquests descriptors que faran possible la categorització dels diferents programes televisius.

Este es un proyecto que trata sobre la indexación automática de contenidos televisivos. Es una tarea que ganará terreno gracias a los inminentes cambios que sufrirá la televisión que conocemos. La llegada de la nueva televisión digital, hará que haya una interacción mucho más fluida entre el espectador i la cadena, además de gran cuantidades de canales, cada uno con programas de tipos completamente distintos. Todo esto hará que tener métodos de búsqueda basados en los contenidos de estos sea del todo imprescindible. Teniendo esto en cuenta, nuestro proyecto está basado en poder extraer algunos de estos descriptores que harán posible la categorización de los diferentes programas televisivos.

This is a project about automatic television content indexing. It is a task that will become very important due to the imminent changes expected in television as we know it. The arrival of the new digital television will bring about a higher interaction between spectators and TV channels, as well as a great quantity of new channels, and also revolutionary new sorts of TV programmes. All of this will make content based search engines absolutely necessary. Our project is entirely based in being able to extract some of these descriptors which will make an automatic categorization of different television programmes through artificial intelligence possible.

Universitat
Autònoma
de Barcelona