Reconeixement d'accions mitjançant visió artificial per Teleassistència

Samuel Sánchez Pérez
Màster Universitari en Enginyeria Informàtica
Intel·ligència Artificial

Samir Kanaan Izquierdo
Carles Ventura Royo

01/06/2016

FITXA DEL TREBALL FINAL

Títol del treball:	*Reconeixement d'accions mitjançant visió artificial per Teleassistència.*
Nom de l'autor:	*Samuel Sánchez Pérez*
Nom del consultor/a:	*Samir Kanaan Izquierdo*
Nom del PRA:	*Carles Ventura Royo*
Data de lliurament (mm/aaaa):	*06/2016*
Titulació o programa:	*Màster Universitari en Enginyeria Informàtica*
Àrea del Treball Final:	*Intel·ligència Artificial*
Idioma del treball:	*Català*
Paraules clau	*Visió artificial, teleassistència, aprenentatge automàtic*

Resum del Treball

L'envelliment continu de la població ha fet que en els últims anys susciti especial interès la recerca i el desenvolupament de sistemes que ajudin a augmentar la qualitat de vida de la gent gran sense que això suposi una despesa insostenible pel sistema de la seguretat social. En aquest context, la teleassistència ha experimentat certes millores incorporant sensors que permeten detectar possibles problemes de salut o accidents, allargant així l'estada al propi domicili i evitant o retardant un ingrés en residència. Aquest treball explora la possibilitat d'utilitzar la visió artificial per monitoritzar accions de la vida quotidiana mitjançant el mètode de generació d'Imatges de la història del moviment (MHI) i la classificació d'aquestes imatges mitjançant Xarxes Neuronals i Màquines de Vectors de Suport. Els resultats que s'han obtingut son insuficients per traslladar la solució construïda a un sistema real però mostren les possibilitats que aquest mètode pot aportar. Una possible millora dels resultats es podria donar ampliant el conjunt de dades etiquetades o millorant el tractament de les imatges. Altra possibilitat de millora podria venir per la combinació del mètode amb altres de visió artificial com ara la detecció i seguiment de parts del cos.

Abstract

The increasingly aging population has led in recent years to special interest on research and development of systems that help increase the quality of life of older people without involving an unsustainable spending for Social Security. In this context, telecare has undergone some improvements related to the use of sensors to detect potential health problems or accidents, thus extending the stay in their own homes and avoiding or delaying entry into a residence. This work explores the possibility of using computer vision to monitor activities of daily life using the method of generating Motion History Images (MHI) and the classification of these images using Neural Networks and Support Vector Machines. The results obtained are insufficient to move the solution to a real system but shows the possibilities this method can bring. A possible improvement of results could be obtained with a wider set of labelled data or improving the image processing. Another possibility for improvement could be the combination with other methods such as body parts detection and tracking.

Índex

1 Plantejament i objectius

En aquest treball es vol construir una aplicació de visió artificial que sigui capaç de reconèixer un conjunt determinat d'activitats humanes que siguin útils per integrar en un sistema de teleassistència basat en la monitorització de l'activitat.

Els sistemes de teleassistència estan concebuts per proporcionar un entorn segur a persones grans o amb algun tipus de discapacitat, que viuen soles al seu domicili. El doble objectiu és, per una banda, evitar o retardar l'ingrés en residències, el que permet que gaudeixin d'una millor qualitat de vida mantenint-se en el seu entorn i sent independents i, per l'altra, obtenir un estalvi, ja que el cost d'atendre aquestes persones en un centre especialitzat és molt més elevat que el d'un servei de teleassistència.

En aquest treball es pretén assolir els següents objectius:

- Triar conjunts de dades d'activitats humanes ja etiquetades, o bé generar-ne un de propi. En aquest cas es tractaria de conjunts d'imatges de vídeo.
- Seleccionar postures o activitats que seria interessant reconèixer per aquest tipus d'aplicació.
- Obtenir un sistema de reconeixement d'activitats humanes que pugui distingir les activitats fent servir Y fotogrames de vídeo.
- Avaluar els resultats.

2 Estat de la qüestió

2.1 Sistemes de teleassistència

Actualment, al mercat hi han molts sistemes de teleassistència basats en la monitorització d'activitat. Alguns d'aquests sistemes estan orientats a un servei professional i d'altres a què siguin els propis familiars els que tinguin cura de les persones grans. Algunes de les característiques dels sistemes de teleassistència basats en la monitorització d'activitat són:

- Botó de pànic: dispositiu amb un botó que permet que l'usuari pugui enviar un avís d'alarma de forma ràpida a través del sistema.
- Presència de sensors instal·lats per tot l'habitatge. Es fan servir diferents tipus de sensors per a capturar senyals que permeten deduir l'activitat dels usuaris. Alguns dels tipus de sensors més habituals són: sensors de moviment PIR i sensors d'obertura de portes.
- Utilització de dispositius portàtils amb sensors integrats i amb forma de penjoll o polsera. Poden portar un sensor de moviment basat en acceleròmetre o giroscopi que permet detectar l'activitat i fins i tot caigudes.
- Aplicació web o per a mòbil per permetre que un cuidador conegui de forma remota l'activitat de la persona monitoritzada. Una de les funcions més comuns de les que disposen aquestes aplicacions és el mapa d'activitat, que permet que un cuidador pugui veure els moviments al llarg del dia per totes les estances de la casa.

- Generació d'alarmes en diferents casos com per exemple: si la persona no s'aixeca abans de certa hora, si no hi ha anat al bany en tot el dia o si ha sortit de casa i no ha tornat abans de certa hora.

Les tecnologies en que es basen tots aquests sensors estan molt provades, ja que són tecnologies d'ús comú, i permeten construir sistemes molt robustos a un cost relativament baix. Malgrat això la informació que proporcionen aquests sensors no poden donar una informació molt acurada de l'activitat que està realitzant l'usuari més enllà de deduir-la en funció de l'estança on es troba (per exemple, deduir que està descansant si està al dormitori). En els següents enllaços es poden veure un parell d'exemples de sistemes d'aquest tipus:

http://www.justchecking.co.uk/

http://www.sensovida.com/

Davant d'aquesta limitació, una possibilitat de millora d'aquests tipus de sistema és la incorporació de càmeres i visió artificial per a poder reconèixer diferents activitats, per exemple el sistema podria reconèixer si l'usuari està caminant, assegut o estirat al sofà, i d'aquesta manera millorar l'eficàcia dels sistemes de teleassistència.

2.2 Reconeixement d'activitat mitjançant vídeo

En quant a la detecció d'activitats mitjançant imatges de vídeo hi han principalment quatre aproximacions (Forsyth, 2012; Bobick, 2016):

1. *Model basat en el reconeixement d'accions a partir de la postura*: Aquí es tractaria de reconèixer primer el cos humà i la postura a través del reconeixement de parts del cos. i caracteritzar els canvis posturals i relacionar-los amb les accions.

2. *Model basat en el reconeixement d'activitats*: Aquí suposem que les activitats es componen d'un conjunt d'accions, de manera que partim del reconeixement d'aquestes accions per reconèixer diferents activitats. És necessari també fer el seguiment de la persona (*tracking*), per a obtenir les accions en forma d'esdeveniments i poder classificar-los com a activitats.

3. *Considerar l'activitat com a moviment*: Aquí no cal ni tan sols reconèixer el cos humà, es tracta de caracteritzar el moviment i relacionar-lo directament amb accions determinades, i així entrenar un classificador. Una forma de caracteritzar el moviment és fent servir una plantilla temporal que es coneix com a "imatges d'història de moviment" (*Motion History Image*) i "imatges de l'energia del moviment" (MEI).

4. *Reconeixement d'activitat a partir d'una sola imatge*: Aquí es tracta de reconèixer el cos i la postura en una imatge fixa i a partir d'aquí relacionar-la amb una activitat.

Les aproximacions que resulten més interessants per a aquest treball són les 1, 3 i 4, ja que es refereixen a la detecció d'activitats de més baix nivell. De les tres opcions candidates, s'optarà per la tercera, ja que els algoritmes utilitzats són ràpids, el que podria permetre la detecció en temps real, fins i tot en dispositius de mida reduïda. Si s'aconsegueix dissenyar un sistema de

detecció d'accions eficient, un següent pas podria ser traslladar-lo a un sistema encastat, obtenint un sensor que indiqués en temps real la presència d'una persona i les accions que està portant a terme sense necessitat de transmetre la imatge de vídeo i preservant, per tant, la seva intimitat.

3 Eines

Per a l'elaboració del treball, s'han considerat les següents eines, que podrien resultar adequades per les característiques que es descriuen a continuació:

- Llenguatge Python: disposa de multitud de llibreries per l'aprenentatge automàtic. Es tracta d'un llenguatge tipus script orientat a objectes. L'inconvenient és que no inclou de forma nativa cap llibreria per tractament de vídeo, tot i que es possible fer servir la coneguda llibreria per visió artificial Open CV.
- Matlab: Proporciona un entorn de desenvolupament (IDE) i un llenguatge que disposa de multitud de llibreries tant per aprenentatge com per anàlisi de vídeo.
- Octave: És l'alternativa gratuïta a Matlab. El llenguatge és molt similar i disposa també pràcticament de les mateixes llibreries que Matlab. No disposa però d'entorn integrat de desenvolupament de manera que s'ha d'escriure el codi fent servir editors de text convencionals i executar-lo a través de la línea de comandes.
- OpenCV: Llibreria d'us gratuït que incorpora multitud d'algoritmes per visió artificial. Disposa de multitud de ports per molts llenguatges de programació, entre d'altres Python i C++.
- C++: Llenguatge orientat a objecte per construir aplicacions natives. Es possible fer servir OpenCV que també està feta amb C++. Aquesta solució permetria construir una solució òptima a nivell d'eficiència, ideal per implementar un producte final però molt poc àgil per fer proves i prototips ràpids.

Ja que aquest treball es basa sobretot en provar diferents tècniques i algoritmes, la millor solució passa per l' eina més àgil, que per la meva experiència és Matlab. Una opció intermitja seria Python, que sobretot tindria sentit si es volgués donar el pas a fer una implementació en C++, ja que només s'hauria de reescriure la part de Python i es podrien mantenir les crides a la llibreria OpenCV.

Matlab disposa d'un paquet específic per visió artificial, anomenat Computer Vision System Toolbox, les característiques principals del qual són les següents:

- Reproducció, generació i manipulació d'imatges i de vídeo.
- Detecció d'objectes i seguiment, incloent-hi mètodes Viola-Jones, Kanade-Lucas-Tomasi(KLT), i Kalman.
- Entrenament per detecció d'objectes, reconeixement d'objectes, consulta d'imatges per contingut, incloent els mètodes de detecció d'objectes en cascada i bossa de característiques.
- Calibratge de mono i estèreo càmeres, incloent detecció automàtica de taula d'escacs.
- Estèreo visió, incloent rectificació, càlcul de disparitat i reconstrucció 3D.
- Detecció, extracció i correspondència de característiques.

- Suport per generació de codi C.

Per altra banda, hem utilitzat Avidemux, una aplicació per conversió de formats de vídeo. També permet reproduir un vídeo fotograma per fotograma de forma fàcil, el que resulta molt útil per prendre nota dels fragments d'interès d'un vídeo, en aquest treball s'ha fet servir per a obtenir el fotograma en que s'inicia i acaba una acció, per a poder etiquetar el conjunt de dades.

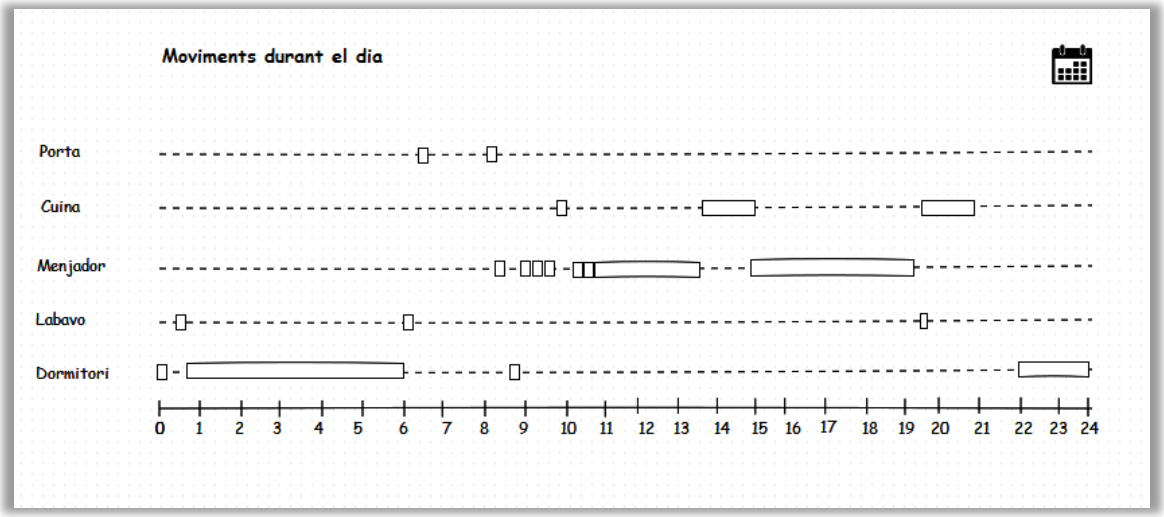
4 Planificació

El treball realitzat ha seguit la següent planificació:

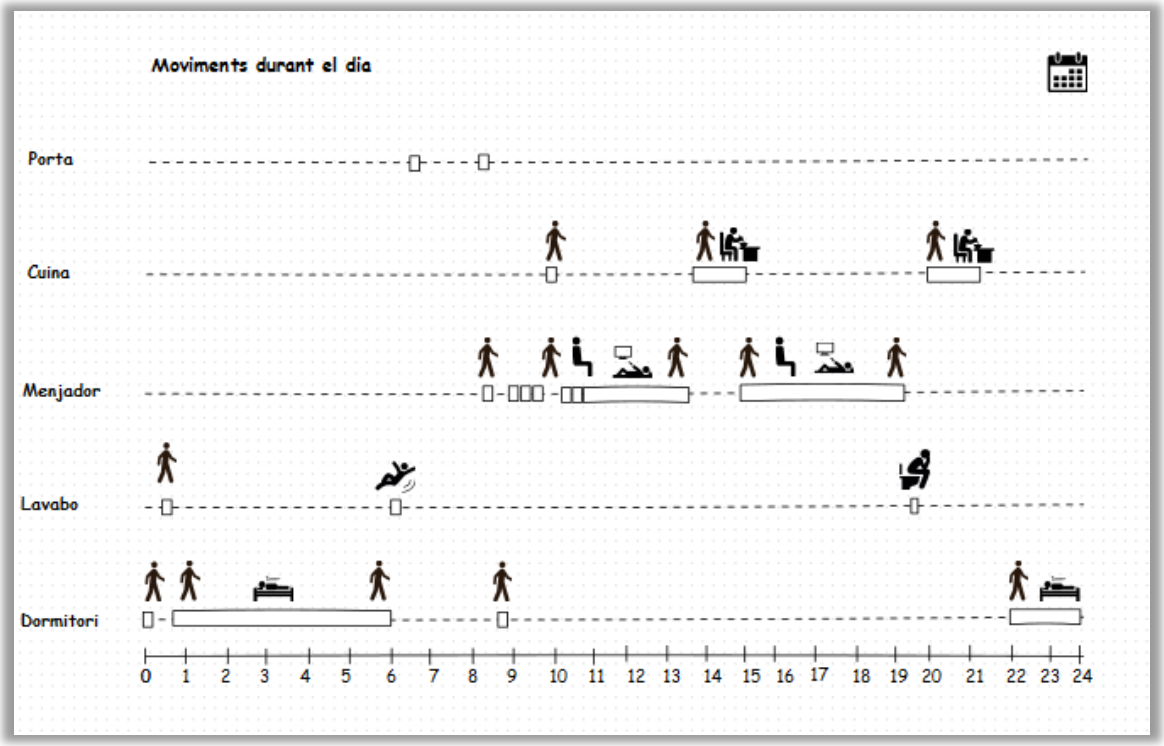
Tasca	**Temporalització**
Proposta inicial amb objectius i estat de l'art.	Setmana 1 - 3
Determinar el conjunt d'activitats i elecció del conjunt de dades.	Setmana 4
Tractament de les dades, construcció d'eines per etiquetar e interpretació de les etiquetes dels exemples.	Setmana 5 i 6
Obtenció de les característiques associades a les seqüències d'imatges que representen les accions etiquetades.	Setmana 7
El mòdul classificador: elecció, entrenament i proves.	Setmana 8 i 9
Sistema de detecció: disseny i implementació.	Setmana 10 i 11
Finalització de la memòria i presentació	Setmana 11 - 15

5 Activitats a reconèixer i cerca del conjunt de dades

Per a l'objectiu del treball ens interessa reconèixer un petit conjunt d'accions que ens permetin completar la informació que aporta un sistema de monitorització d'activitat amb sensors de moviment. Aquest és un exemple de gràfic que mostra un sistema que només disposa de sensors de moviment:



Si a la informació de moviment afegim la informació d'accions senzilles com ara: seure, caminar i jeure, podem obtenir molta més informació sobre l'activitat que es du a terme:



Amb aquesta finalitat hem cercat un conjunt de dades que disposés de les següents característiques:

- Recollir escenes normals de la vida quotidiana a l'interior d'un domicili.
- Les càmeres havien d'estar disposades a prop del sostre, ja que, en el cas d'un sistema final, interessa que la imatge cobreixi la major part de l'estança on es troba i que hi puguin haver el mínim d'oclusions provocades per la presència d'objectes entre la càmera i les persones.

Consultant la bibliografia sobre la detecció d'accions i activitats, per la qual cosa ha resultat de molta ajuda la revisió sobre aquest tema concret que fan Chaquet, Carmona i Fernández-Caballero (2013), veiem que hi han una gran quantitat de datasets però la majoria recullen moviments que no són habituals en el dia a dia o bé capten la persona en un pla massa curt.

S'ha optat finalment per un conjunt de dades elaborat per Auvinet, Rougier, Meunier, St-Arnaud i Rousseau (2010), de la universitat de Montreal. Aquest conjunt de dades consisteix en una col·lecció de vídeos on es veuen accions del dia a dia i caigudes. Concretament compta amb 24 escenes gravades simultàniament amb 8 càmeres. El fet de disposar de les mateixes escenes amb diferents càmeres dóna la possibilitat de tenir diferents perspectives d'un mateix moviment. Com que una mateixa acció capturada des de perspectives diferents pot resultar visualment molt diferent, és possible que la classificació sigui més eficient si dividim una mateixa acció en diferents classes en funció de la perspectiva, per exemple: caminar - frontal, caminar - perfil i caminar - esquena.

Partint d'aquests vídeos, en un principi s'han considerat les següents accions:

1. Caminar
2. Caure
3. Estirat a terra
4. Ajupit
5. Ajupint-se
6. Aixecant-se
7. Assegut
8. Estirat al sofà
9. Parat dret
10. Seient

6 Disseny del sistema

El sistema consisteix en tres blocs principals:

- Mòdul de tractament de dades: responsable de l'obtenció dels conjunts d'exemples a partir del tractament de les imatges.
- Mòdul classificador: mòduls encarregats de la gestió del classificador.
- Mòdul de detecció.

6.1 Mòdul de tractament de dades

6.1.1 Etiquetatge i obtenció de fragments de vídeo

El primer pas per obtenir un conjunt d'exemples etiquetats per poder entrenar un classificador és visualitzar els vídeos i anotar a cada moment l'acció que es du a terme de les que s'han definit a l'apartat 5. Per això es defineixen dos fitxers de text: un descriptor de les accions que

es duen a terme i un fitxer on es defineixen els retards de cada càmera a partir d'una de referència.

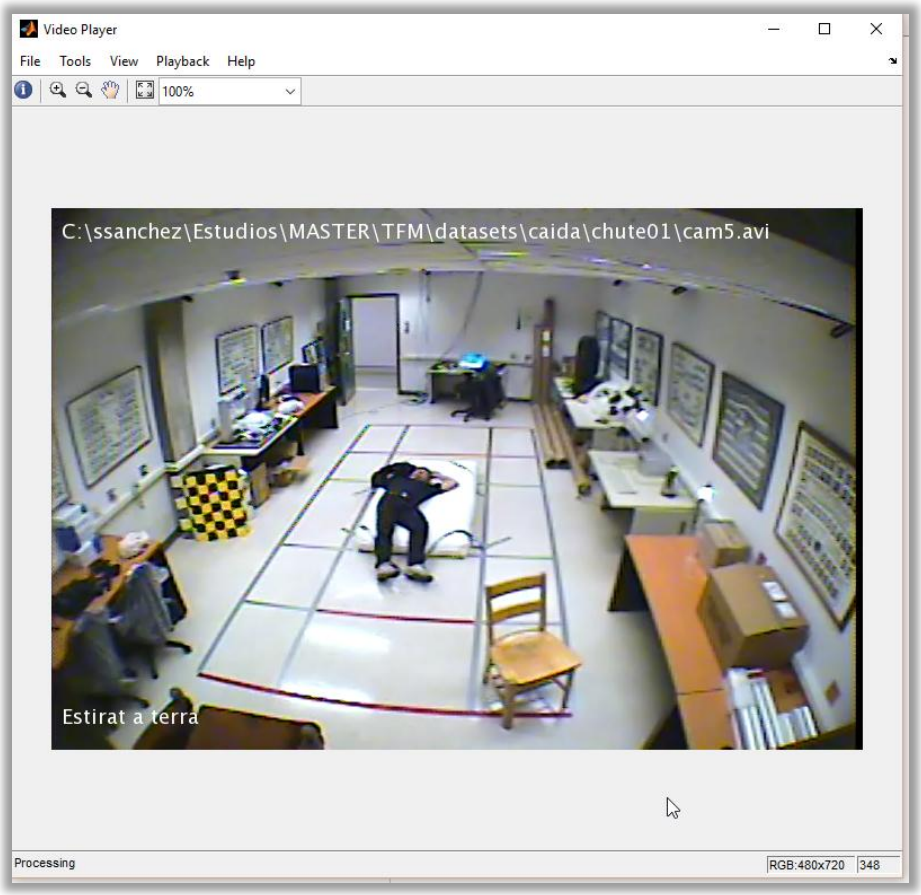
Descriptor de les accions: conté informació de les accions que es donen a cada escena: Escena, frame inici, frame final i identificador de l'acció. S'ha partit de les dades originals i s'han modificat les etiquetes a conveniència.

Retards: Conté els retards de cada càmera respecte d'una referència per cada una de les escenes: Escena, Cam1, Cam2, Cam3, Cam4, Cam5, Cam6 Cam7, Cam8. Inicialment es va fer servir la taula del report adjunt al dataset però es va veure que els retards eren incorrectes i les etiquetes no quedaven sincronitzades amb totes les càmeres, de manera que s'han calculat manualment prenent com a referència un fotograma concret del començament de cada escena.

A partir dels dos fitxers descrits anteriorment es poden reproduir les escenes verificant que l'etiquetatge és correcte executant el mòdul PlayDatasetWithLabels que es descriu a continuació.

PlayDatasetWithLabels.m:

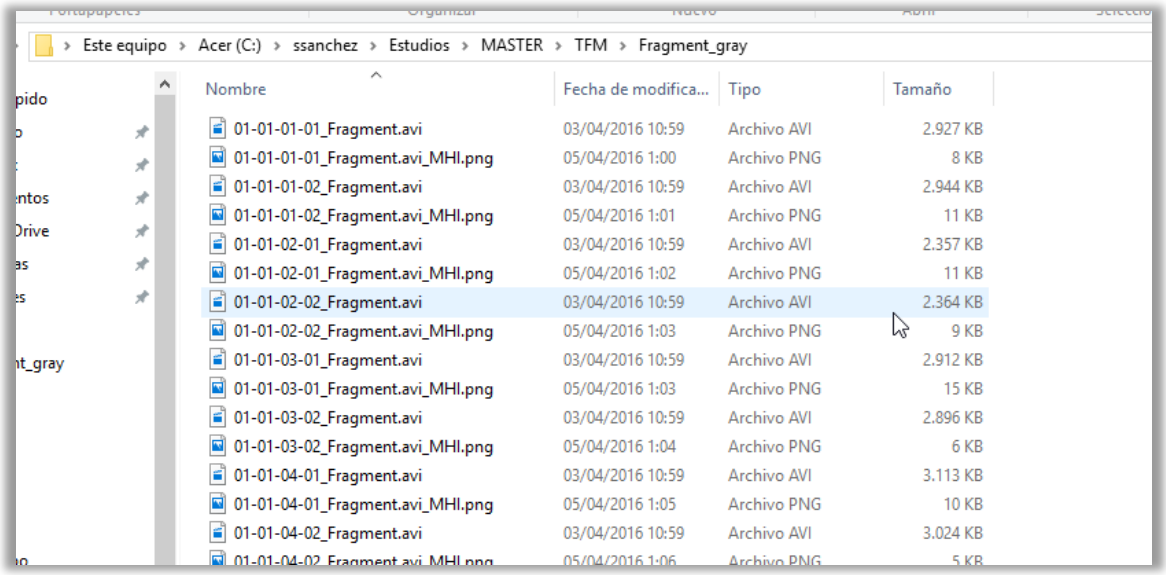
A partir de les dades de les accions i els retards de les càmeres mostra els vídeos amb les etiquetes de les accions sincronitzades amb la imatge.



Un cop tenim els vídeos etiquetats els hem trossejat en fragments d'aproximadament un segon. Posteriorment hem extret un vector de característiques de cadascun d'aquests fragments i els hem fet servir per entrenar un classificador. El mòdul encarregat d'això es el que es descriu a continuació:

GenerateFragments.m:

A partir dels vídeos i les dades tant de les accions com dels retards es generen fragments de vídeo que representen les diferents accions. La durada màxima i mínima de cada fragment són parametritzables. Si establim una durada màxima d'un segon, en el cas que una acció, com per exemple caminar duri més d'1 segon, es divideix en diversos fragments. El mòdul també genera un fitxer de text amb la següent informació: Filename, Scene, Cam, Ini, Fin, Action.



Llistat de fragments obtinguts amb GenerateFragments.m

Per poder tractar posteriorment cadascun dels fragments hem necessitat extreure la part mòbil de la imatge, que en el nostre cas és la persona que surt a l'escena realitzant alguna acció. De la part estàtica de la imatge se'n diu background i la part mòbil foreground. Aquí en podem veure un exemple:

Imatge Background Foreground

Hi han molts algoritmes per obtenir el foreground d'una escena de vídeo. El més senzill es basa en considerar que el fotograma anterior és el background i la diferència ens donarà la imatge de foreground. També és la menys exigent a nivell computacional. Nosaltres obtindrem el background a partir de la mitjana dels n fragments anteriors, el que es coneix com "median filtering". Ja que per obtenir el background fent servir aquest mètode necessitem una escena d'uns 5 segons i els fragments són de al voltant d'un segon, no podem obtenir el background de cada fragment de forma independent. Per resoldre això obtindrem un vídeo amb el background complert per cada escena i cada càmera. El mòdul encarregat d'aquest tractament és GenerateBackground.

GenerateBackGround.m:

Aquest mòdul recorre totes les escenes i totes les càmeres de cada escena. Per cada vídeo genera un altre amb el background calculat fent servir la mitjana dels últims quatre segons.

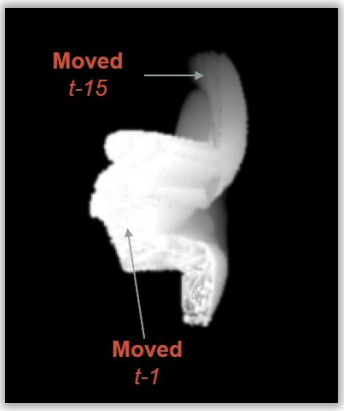
L'ultima part d'aquesta fase consisteix en generar el MHI (Motion History Image) de cada fragment, el mòdul encarregat d'això serà GenerateMHIPerFragment.m, i es descriu a continuació:

GenerateMHIPerFragment.m:

A partir de cada un dels fragments, extreu el foreground a partir de la diferència amb el background, aplica un llindar per obtenir una imatge binària per cada fotograma i genera una MHI del moviment de la persona que surt a l'escena. Una 'Motion History Image' representa l'estela del moviment en escala de grisos, corresponent les tonalitats més clares als instants més recents:

6.1.2 Obtenció de característiques dels moviments a partir de les accions etiquetades

Hi han moltes formes de caracteritzar un moviment però una de les més eficients a nivell computacional és obtenir una MHI o una MEI (Motion Energy Image). Com ja s'ha introduït a l'apartat anterior, per generar una MHI d'un fragment de vídeo cal obtenir la diferència de cada fotograma amb el background i aplicar un llindar per obtenir una imatge binaria del foreground. A partir d'aquestes imatges és fa un processament senzill que dóna com a resultat una sola imatge que aporta informació sobre el moviment i el sentit d'aquest moviment, ja que els píxels amb una tonalitat més clara son els que indiquen que l'objecte ha estat en aquella posició més recentment:



Imatges extretes de Davis i Bobick (1997)

El MEI (Motion Energy Image), parteix de la mateixa idea però no dóna la informació del sentit del moviment ja que és una imatge binària que indica la regió on s'ha produït moviment però no dona una idea de temporalitat. L'equivalent de MEI pel moviment de l'anterior il·lustració és el següent:

El nostre vector de característiques però, no serà cap de les dues imatges, si no uns coeficients calculats a partir d'aquestes, que tenen les propietats de ser invariants a la translació, rotació i escala. Es tracta dels *Hu moments* (Hu, 1962). Els *Hu moments* es calculen a la funció calcHuMoments. El mòdul encarregat de generar els *Hu moments* és GenerateHuMoments, que es descriu a continuació:

GenerateHuMoments.m:

Genera un vector de 14 elements per cada fragment, que serà el vector de característiques de cada exemple. Per cada un dels fragments, carrega la MHI i calcula els *Hu moments* que guarda a les set primers posicions del vector, després obté la MEI aplicant un llindar a la MHI i calcula els *Hu moments* per aquesta, carregant les següents 7 posicions del vector. El mateix mòdul guarda en un fitxer de text els vectors de característiques junt amb la acció que representa. La estructura del fitxer és: Id fragment, hu moments, acció. Amb això tindrem els exemples per entrenar el classificador.

6.2 Mòdul Classificador

El conjunt d'exemples obtingut a la fase de tractament de dades s'ha utilitzat per entrenar un classificador. Es realitzen diferents experiments entrenant una xarxa neuronal amb diferents configuracions i un classificador SVM també amb diferents configuracions. L'objectiu d'aquests experiments es comprovar si la idea de fer servir una caracterització de MHI i MEI a partir dels invariants *Hu moments* pot ser viable a l'hora d'obtenir un classificador que permeti reconèixer un reduït nombre d'accions de la vida quotidiana a l'interior d'una vivenda.

Un cop seleccionat el tipus de classificador més adient i la configuració més adequada s'ha guardat el model del classificador entrenat en un fitxer per poder-lo fer servir en el mòdul de detecció. El mòdul de detecció que es descriu més endavant ha fet servir el classificador per obtenir una predicció en forma de vector de probabilitats i és el propi mòdul de detecció qui aplica els llindars de discriminació òptims per realitzar la predicció.

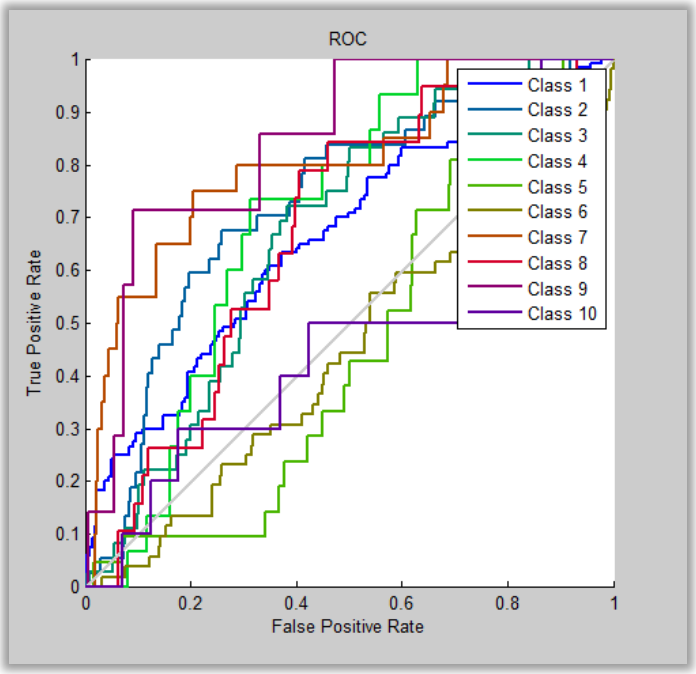
6.2.1 Descripció dels experiments

Inicialment s'han considerat tots els fragments etiquetats amb les deu accions que s'han obtingut de les vuit càmeres. A continuació es mostren vuit fragments de vídeo, que

corresponen a un mateix interval de temps vist des de cadascuna de les vuit càmeres, cada un correspon a un exemple etiquetat amb l'acció "Camina":

Amb els 2249 exemples s'han entrenat els classificadors basats en xarxa neuronal i SVM, provant diferents configuracions de cada un d'ells. Pel que fa al conjunt de dades, s'ha dividit el conjunt d'entrenament, conjunt de validació i conjunt de test. Als següents punts *6.2.2 Xarxes Neuronals* i *6.2.3 Màquines de vectors de suport* es descriuen amb més detall els diferents experiments que es fan amb els dos tipus de classificador.

La mesura de la bondat del classificador ve a partir de l'obtenció de la corba ROC en cadascun dels experiments sobre el conjunt de test. La corba ROC (Receiver Operating Characteristic), ens dona una idea de la qualitat del classificador mostrant la relació entre la taxa de positius reals i la taxa de falsos positius de cada classe en funció del llindar de discriminació. A continuació es mostra un exemple:



La corba del classificador ideal recorreria l'eix vertical per l'esquerra i seguiria l'eix horitzontal per la part superior de la gràfica. La recta amb pendent 1 que es mostra en color gris representa la línea de no discriminació. Un classificador que oferís una corba ROC sobre

aquesta línea donaria una predicció completament aleatòria. En el nostre cas descartarem els classificadors de les accions que mostrin una ROC que no estigui suficientment separada de la línea de no discriminació.

En base a aquests experiments s'han obtingut dotze models de classificador i se n'ha escollit un perquè formi part del mòdul de detecció. Els resultats dels experiments s'exposen amb detall a l'apartat *7 Resultats d'experiments i avaluació del sistema*.

Altres experiments que resultarien interessants però que no s'han realitzat per falta de temps son els que es detallen a continuació:

- Entrenar el classificador basat en xarxa neuronal amb la configuració que ha donat millors resultats però amb un número de classes més reduït, seleccionant només les accions que queden clarament per sobre de la línia de discriminació de les corbes ROC en el primer experiment. Aquest experiment no es realitzaria amb el classificador basat en màquines de vectors de suport perquè en aquest cas els resultats serien els mateixos ja que entrenem n classificadors binaris, un per cada classe amb el procediment d'un contra tots.

- Modificar l'etiquetat del conjunt de dades dividint cada acció en funció de la perspectiva, ja que visualment és molt diferent l'aparença d'una mateixa acció en funció de la perspectiva. D'aquesta manera, per exemple, tindríem de l'acció caminar tres classes diferents si considerem perspectiva frontal, lateral i posterior. Això pot tenir dos inconvenients, per un costat es multiplica el número de classes mentre que es divideix el número d'exemples de cada classe i per l'altre, s'han d'etiquetar novament cada fragment anotant una per una la perspectiva de l'acció representada per cada exemple.

6.2.2 Xarxa neuronal

Per la construcció del classificador hem fet ús de la llibreria "Neural Network Toolbox" de Matlab. Els exemples obtinguts a partir del tractament descrit a l'apartat anterior es divideix en: conjunt d'entrenament, validació creuada i test:

- Conjunt d'entrenament: 70%
- Conjunt de validació creuada: 15%
- Conjunt de test: 15%

S'han fet sis experiments, variant el número d'unitats per capa i el número de capes ocultes. En tots els experiments s'ha fet servir la funció d'entrenament SCG, que és la que proporciona per defecte la llibreria. Com s'ha indicat al punt anterior s'ha obtingut la corba ROC en cadascun dels experiments.

6.2.3 Màquines de vectors de suport

Per a la construcció del classificador de màquines de vectors de suport hem utilitzat la llibreria "LIBSVM" en la seva versió per Matlab. Els exemples obtinguts a partir del tractament descrit a l'apartat anterior es divideix en conjunt d'entrenament i test:

- Conjunt d'entrenament: 85%
- Conjunt de test: 15%

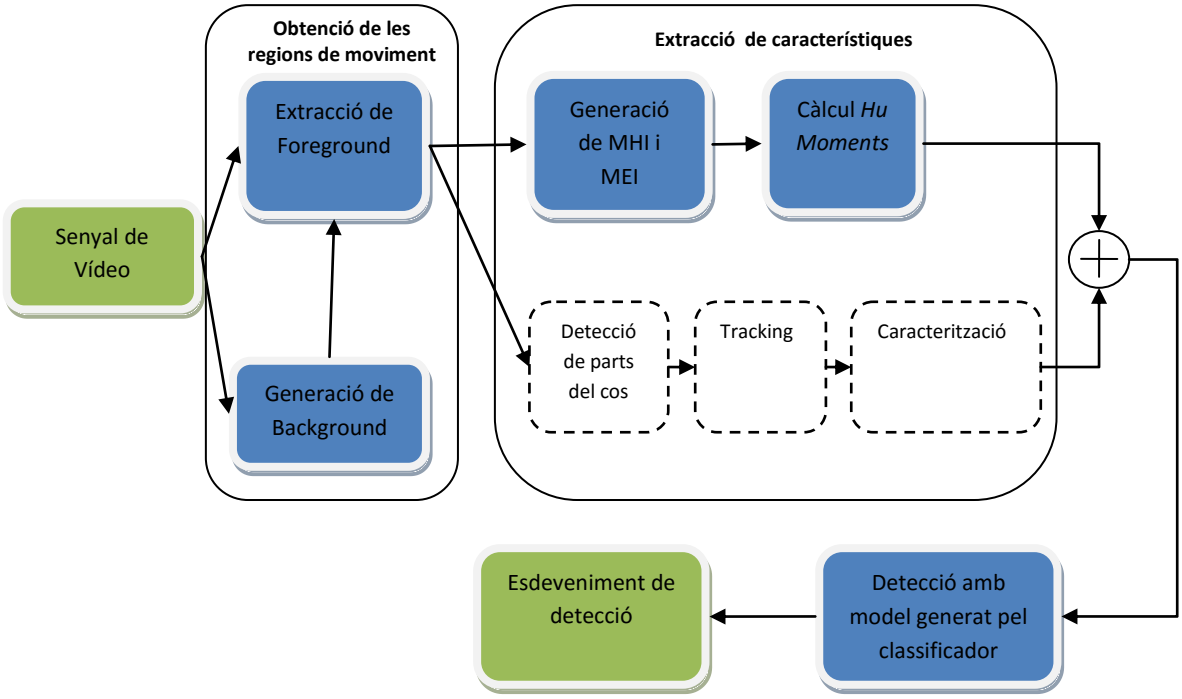
S'han fet proves amb kernel RBF i lineal i l'ajust dels paràmetres s'ha fet de forma automàtica mitjançant un algoritme que prova diferents configuracions i escull la configuració que dóna una millor exactitud sobre el conjunt de validació. Un cop fet l'ajust es realitza l'entrenament de tot el conjunt d'entrenament, incloent el conjunt de validació, es fa el test contra el conjunt de test.

El conjunt de validació que es fa servir per ajustar els paràmetres s'obté dividint el propi conjunt d'entrenament en entrenament i validació, amb una proporció de 70% per entrenament i 30% per validació durant l'ajust dels paràmetres. Aquesta divisió es fa de forma aleatòria per cada iteració de l'algoritme d'ajust. L'entrenament es fa seguint una estratègia de u contra tots de 10 classificadors binaris, un per cada acció.

Per tenir una primera mesura de l'efectivitat del classificador s'ha obtingut la corba ROC per cada classe, al igual que s'ha fet amb el classificador de tipus xarxa neuronal. Això ens ha permès comparar els dos tipus de classificador, escollir una configuració determinada d'un tipus de classificador i descartar les classes que no podem predir amb un mínim de precisió.

6.3 Mòdul de detecció

El sistema de detecció rep com a entrada una senyal de vídeo digital i dona com a sortida esdeveniments de detecció d'accions. A continuació es mostra un diagrama amb els mòduls principals:



Per a la construcció d'aquest mòdul es reutilitzen els components descrits a l'apartat del mòdul de tractament de dades i del mòdul classificador.

6.3.1 Obtenció de les regions de moviment

Aquest mòdul és el responsable d'extreure les regions on es produeix moviment de cadascun dels fotogrames de la senyal de vídeo digital. Internament es subdivideix en una part que s'encarrega d'obtenir la part estàtica o backgorund i una altra que s'ocupa d'extreure el background de la imatge original i obtenir les regions de moviment o foreground. També realitza altres processaments de la imatge com ara la transformació a escala de grisos o l'aplicació de filtres. No entrem en més detall perquè el procés d'extracció de foreground es descriu més àmpliament en l'apartat 6.1.2.

6.3.2 Extracció de característiques

Aquest mòdul té la responsabilitat d'extreure els vectors de característiques. En el nostre cas consisteix en els mòduls de generació de MEI i MHI i el de càlcul des invariants coneguts com - *Hu moments* que hem introduït a l'apartat 6.1.2. La sortida d'aquest mòdul és un vector de catorze elements que caracteritza el fragment de l'escena que s'està processant.

A la llum dels resultats obtinguts i que es presenten a l'apartat 7, és possible que per obtenir un millor classificador necessitem afegir més dimensions als vectors de característiques, el que es podria aconseguir mitjançant la utilització d'alguna altra metodologia, com la detecció de parts del cos, el seguiment d'aquestes i la caracterització d'aquest moviment. Aquests blocs

s'indiquen a l'esquema en línees discontinues per donar la idea de que el mòdul d'extracció de característiques es pot ampliar tot i que queda fora de l'abast del treball.

6.3.3 Detecció d'accions

El mòdul de detecció rebrà com a entrada el vector de característiques del fragment de vídeo i, en cas de que identifiqui que correspon a alguna de les accions que es capaç de detectar, emetrà un esdeveniment de detecció indicant l'acció. Si es volgués implementar una detecció en temps real hauríem de tenir en compte que, per la naturalesa del problema i del mètode utilitzat, hi ha un retard en la detecció que serà al menys la durada del fragment de vídeo tractat en cada pas i que estarà al voltant d'un segon.

El mòdul de detecció farà servir el model del classificador prèviament entrenat, li passarà el vector de característiques i obtindrà un vector amb els valors de la probabilitat per cada classe. El mòdul de detecció aplicarà el llindar de discriminació òptim per cada una de les classes i en cas d'obtenir més d'un positiu seleccionarà la classe amb la probabilitat més alta de les que estiguin en disputa.

El criteri per la selecció dels llindars de detecció es farà en base a obtenir una bona precisió, ja que prioritzem el fet de que la major part de les deteccions siguin positius reals al fet de que es detectin la major part dels positius de cada classe (sensibilitat o recall). S'ha pres aquesta decisió perquè de cara a un sistema de detecció en temps real on s'activaria un pas de detecció cada 100 - 500ms, es tolerable rebre un esdeveniment de detecció cada cinc o deu segons. Això voldrà dir que podem fer un seguiment de les accions d'una persona amb una resolució de 5 o 10 segons, i per tant una sensibilitat d'un 10% resultaria més que suficient per l'aplicació concreta, sempre que la precisió sigui superior al 50%.

6.3.3.1 Elecció dels llindars de detecció i descart de les classes no predictibles

Seguint el criteri de prioritzar una alta precisió a costa de perdre sensibilitat que ja hem apuntat a l'apartat anterior, hem establert un procediment per la selecció dels llindars de detecció per cada classe fent servir el classificador que ofereixi millors resultats.

El procediment ha consistit en seleccionar manualment el llindar de detecció que proporcionés la màxima precisió, sempre superior o igual al 50% i que impliqués una sensibilitat no inferior al 10%. Aquest ajust s'ha fet sobre el conjunt de validació. Per això obtenim una gràfica per cada classe on es mostren la tasa de falsos positius, la tasa de positius reals o sensibilitat i la precisió, tot en funció del llindar de discriminació.

Aquest anàlisi ens ha permés establir els llindars de discriminació per cada classe i descartar les classes que no resulten predictibles segons els criteris que hem imposat. Finalment s'ha obtingut la matriu de confusió on tenim les classes predictibles i les classes no predictibles agrupades en una sola classe. Els resultats d'aquest anàlisi i la prova de detecció es mostra amb detall a l'apartat *7.2 Llindars de discriminació i classes predictibles.*

7 Resultats dels experiments i avaluació del sistema

En aquest apartat s'exposen els resultats obtinguts en els diferents experiments. Com veurem més endavant, els resultats ens permeten veure que el mètode escollit pot ser un bon candidat a l'hora de predir determinades accions, però no totes les que s'han considerat originalment.

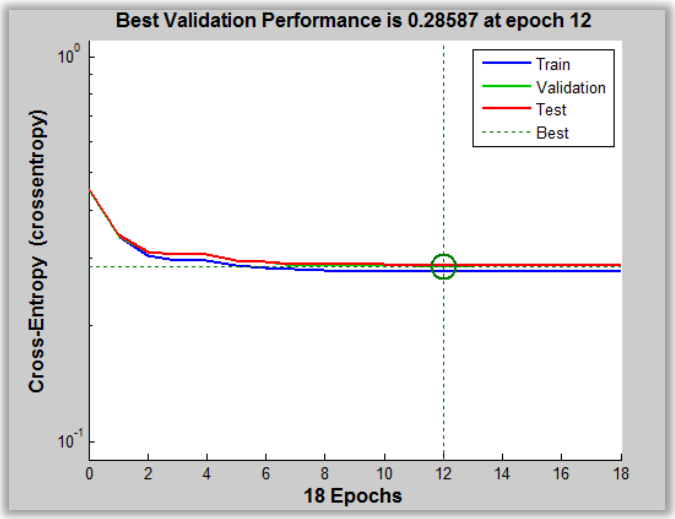
Al primer apartat es presenten els resultats dels diferents experiments i estan enfocats a l'elecció de tipus de classificador tenir una primera idea de les classes que resulten més fàcilment predictibles. El segon apartat està dedicat a l'elecció dels llindars de discriminació que ens permetrà obtenir unes prediccions on l'objectiu es obtenir la màxima precisió, tal i com s'explica al punt *6.3.3 Detecció d'accions*.

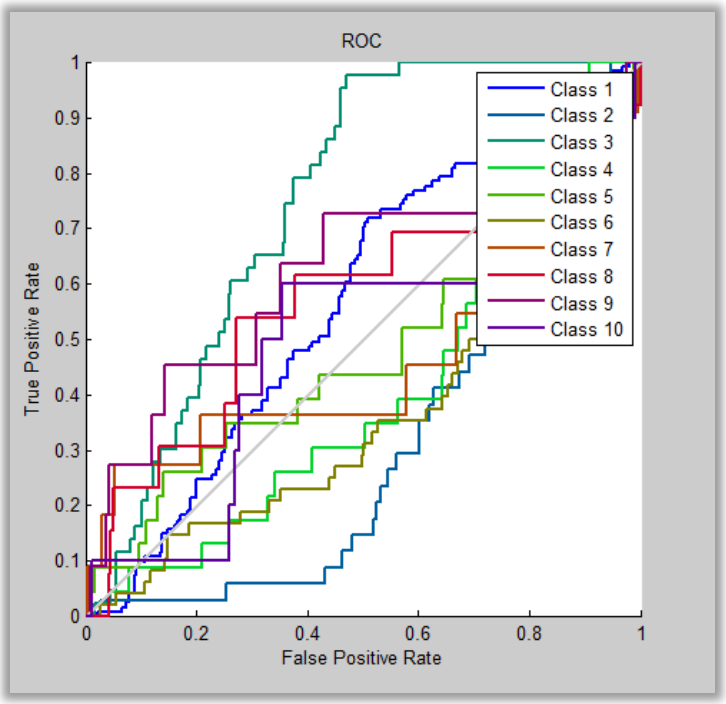
7.1 Elecció de tipus de classificador i classes predictibles

7.1.1 Experiments amb Xarxa Neuronal

A continuació es presenten els resultats amb diferents configuracions de la Xarxa Neuronal pel conjunt d'exemples dividit en conjunt d'entrenament, validació i test.

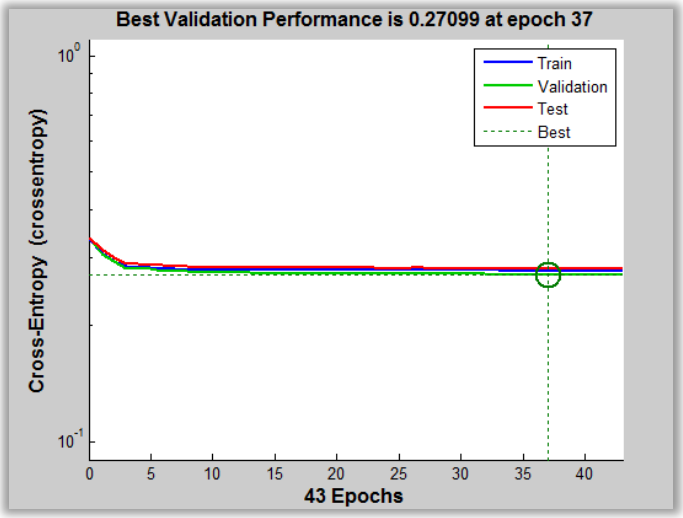
Nº d'experiment	1
Capes ocultes	1
Número d'unitats per capa	10
Funció d'entrenament	Scaled Conjugate Gradient
Funció de rendiment	crossentropy

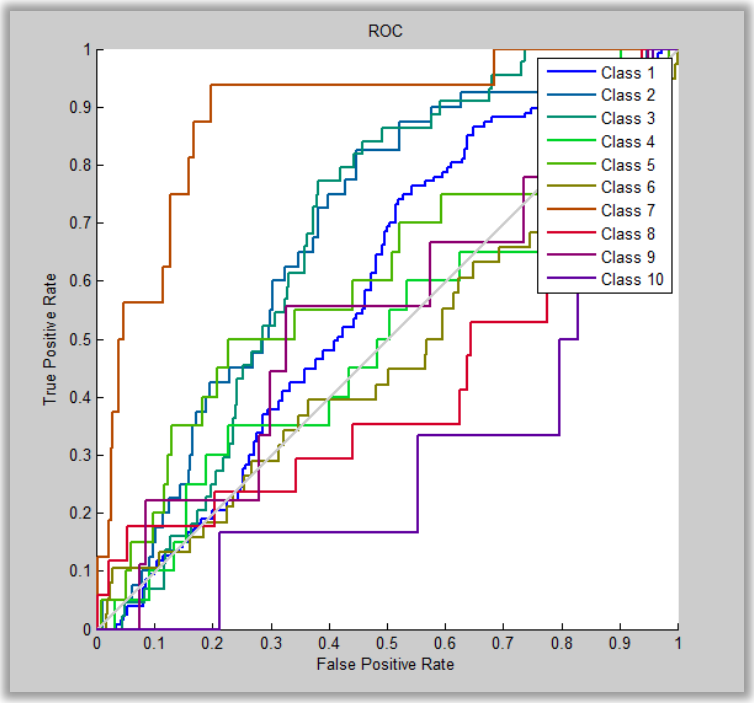




Aquí ja veiem que per sobre de la línea de no discriminació queden les següents classes: 1, 3, 7, 8 i 10.

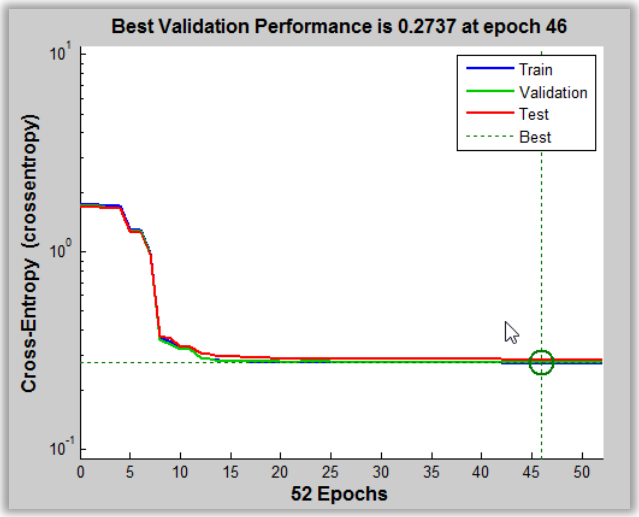
Nº experiment	2
Capes ocultes	3
Número d'unitats per capa	10
Funció d'entrenament	Scaled Conjugate Gradient
Funció de rendiment	crossentropy

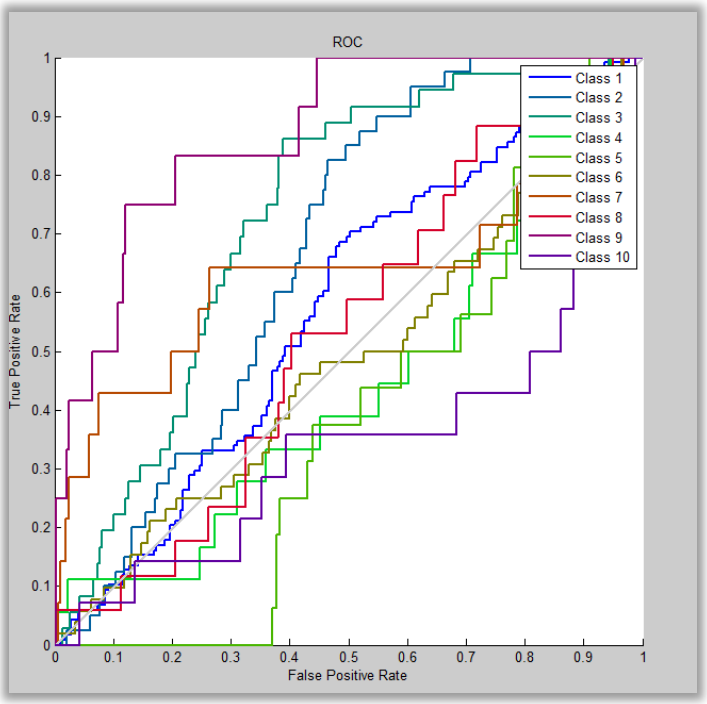




En aquest cas, queden per sobre de la línea de no discriminació les classes: 1, 3, 5, 7, 9 i 10.

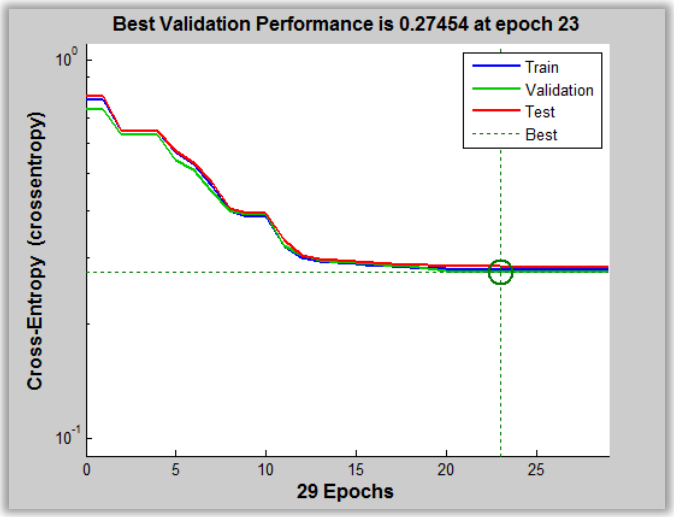
Nº experiment	3
Capes ocultes	1
Número d'unitats per capa	100
Funció d'entrenament	Scaled Conjugate Gradient
Funció de rendiment	crossentropy

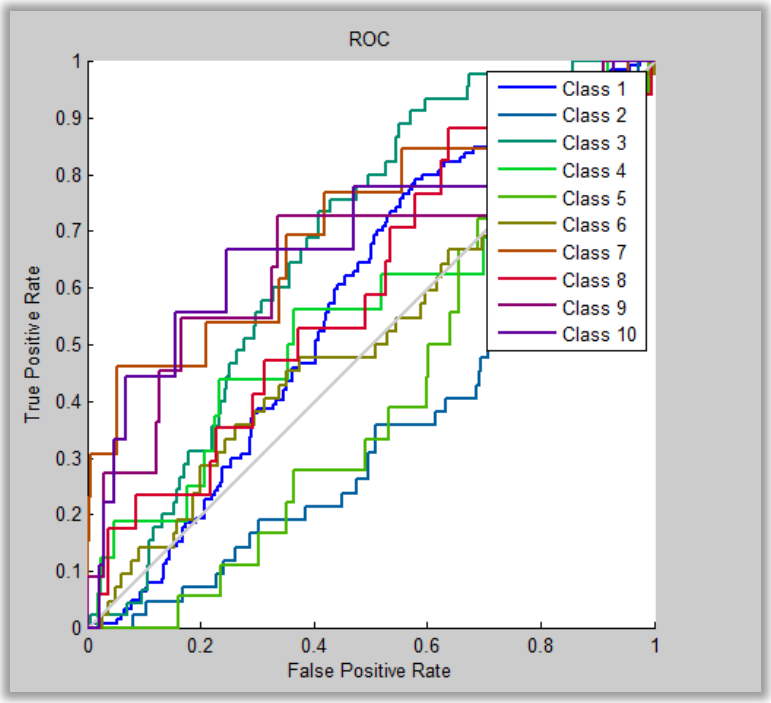




En aquest cas, per sobre de la línia de no discriminació trobem les accions: 1, 2, 3, 7 i 10.

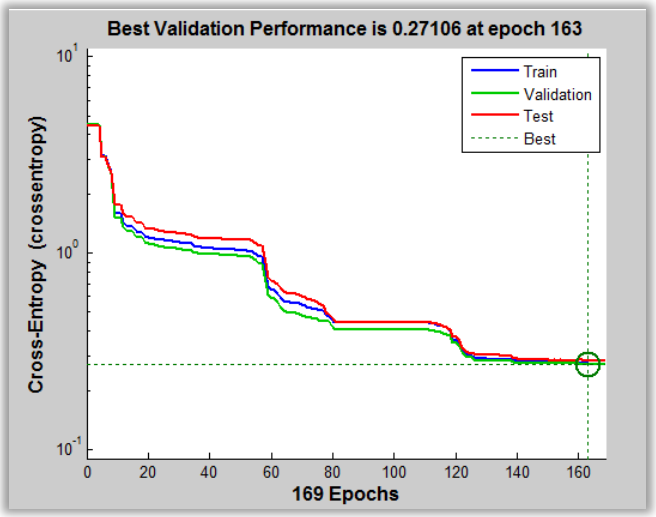
Nº experiment	4
Capes ocultes	3
Número d'unitats per capa	100
Funció d'entrenament	Scaled Conjugate Gradient
Funció de rendiment	crossentropy

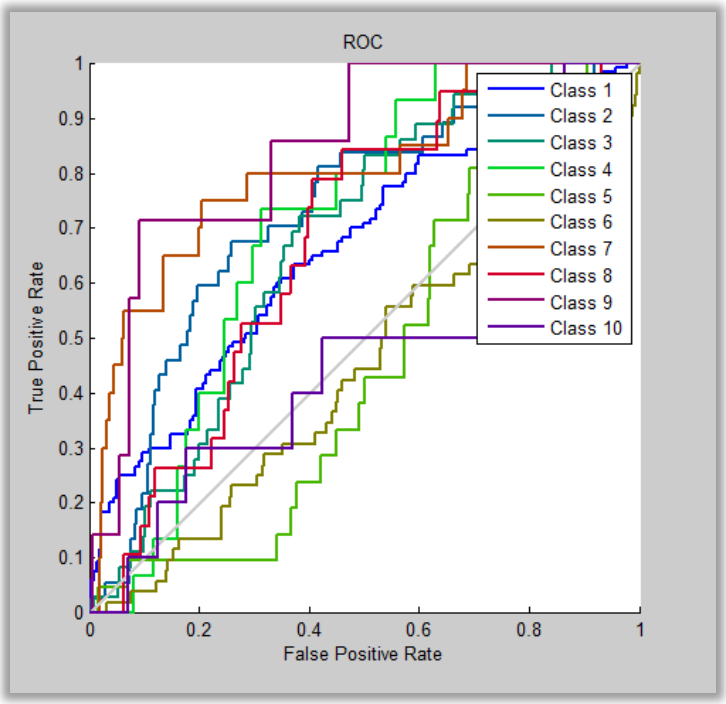




En aquest cas, per sobre de la línia de no discriminació trobem les accions: 1, 3, 7, 9 i 10.

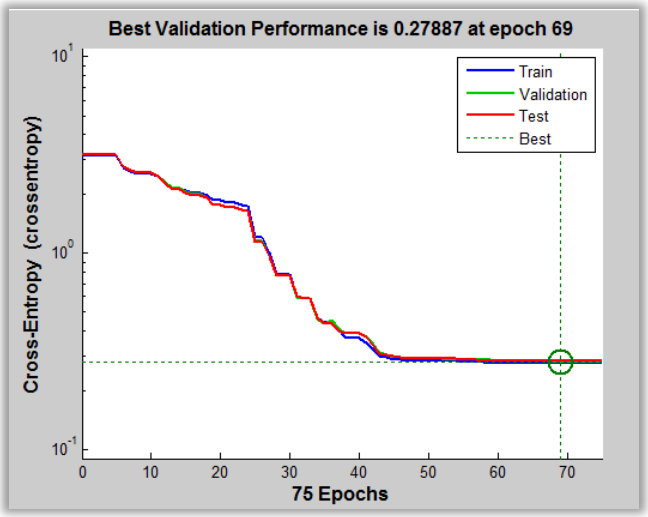
Nº experiment	5
Capes ocultes	1
Número d'unitats per capa	1000
Funció d'entrenament	Scaled Conjugate Gradient
Funció de rendiment	crossentropy

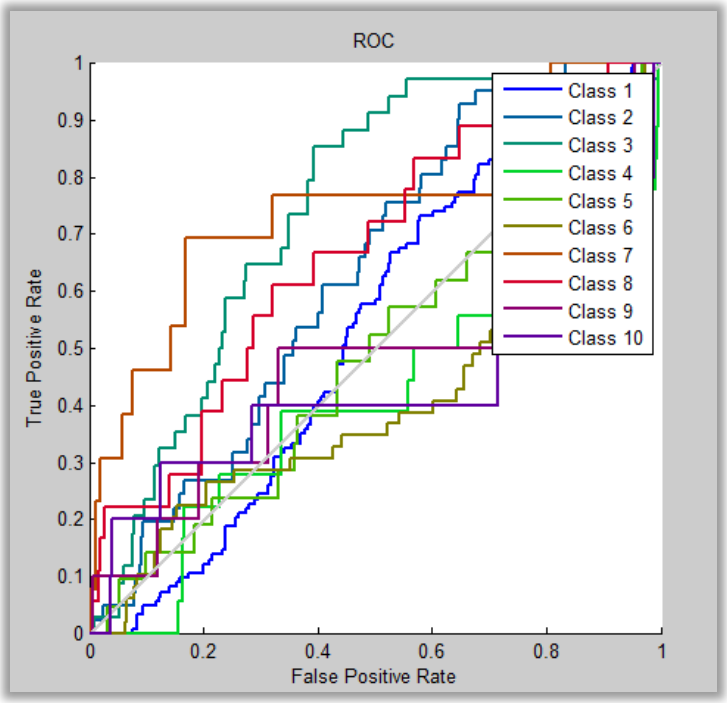




En aquest cas, per sobre de la línia de no discriminació trobem les accions: 1, 2, 3, 4, 7 i 9.

Nº experiment	6
Capes ocultes	3
Número d'unitats per capa	1000
Funció d'entrenament	Scaled Conjugate Gradient
Funció de rendiment	crossentropy



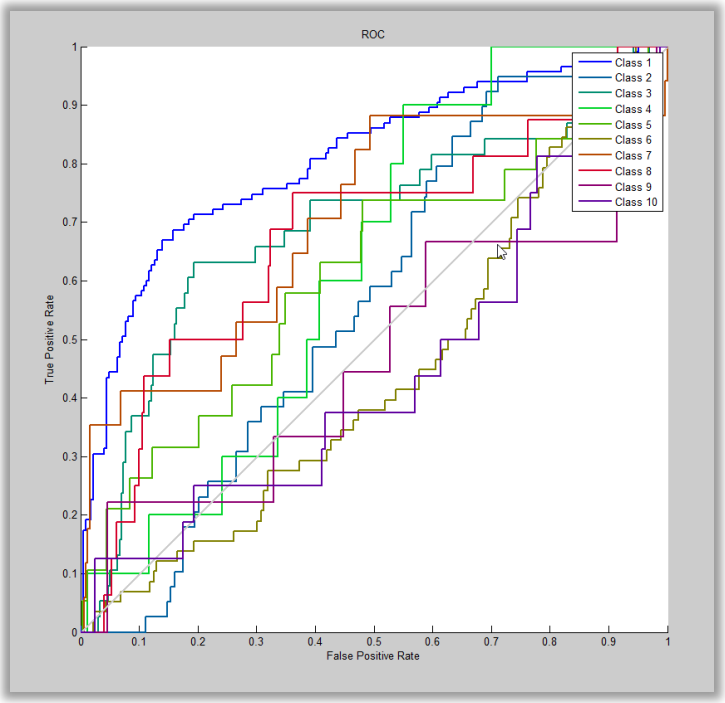


En aquest cas, clarament per sobre de la línia de no discriminació trobem: 1, 2, 3, 7 i 8.

7.1.2 Experiments amb màquines de vectors de suport

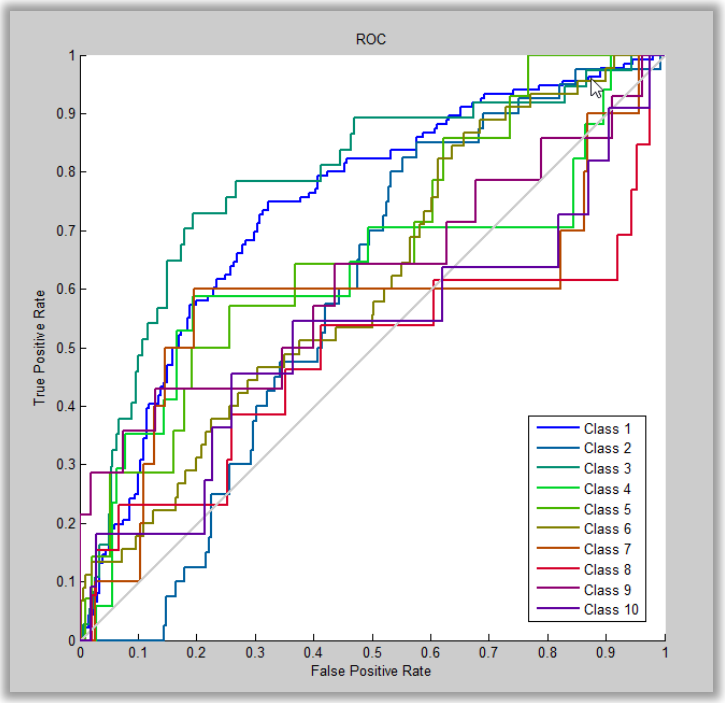
En aquest apartat s'exposen els resultats obtinguts amb el classificador basat en màquines de vectors de suport. S'han realitzat tres proves amb un kernel RBF amb diferents configuracions i tres més amb kernel lineal.

Nº Experiment	1
Tipus	C-SVC
Tipus de kernel	RBF: exp(-gamma*|u-v|^2)
gamma	4
cost	2



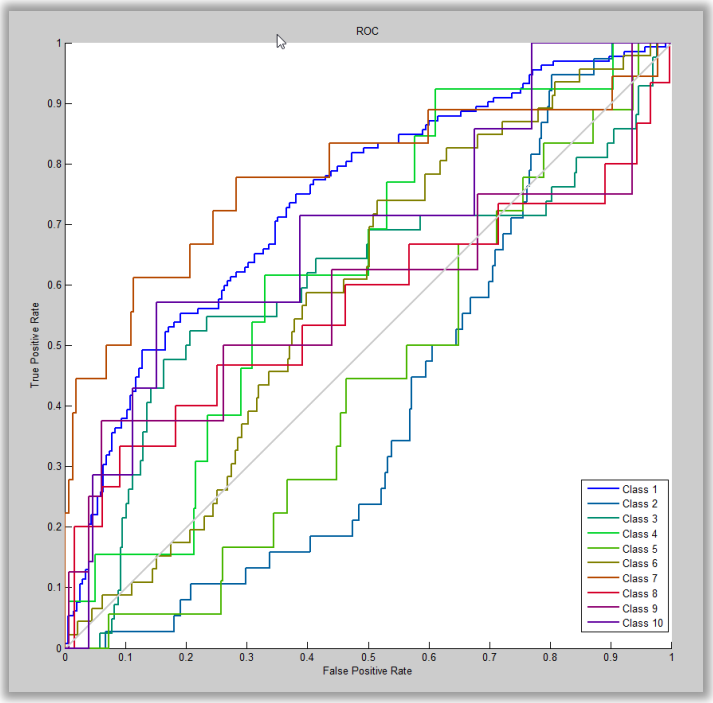
En aquest cas veiem clarament per sobre de la línia de no discriminació les classes: 1, 3, 7 i 8.

Nº Experiment	2
Tipo	C-SVC
Tipo de kernel	RBF: exp(-gamma*|u-v|^2)
gamma	4
cost	8



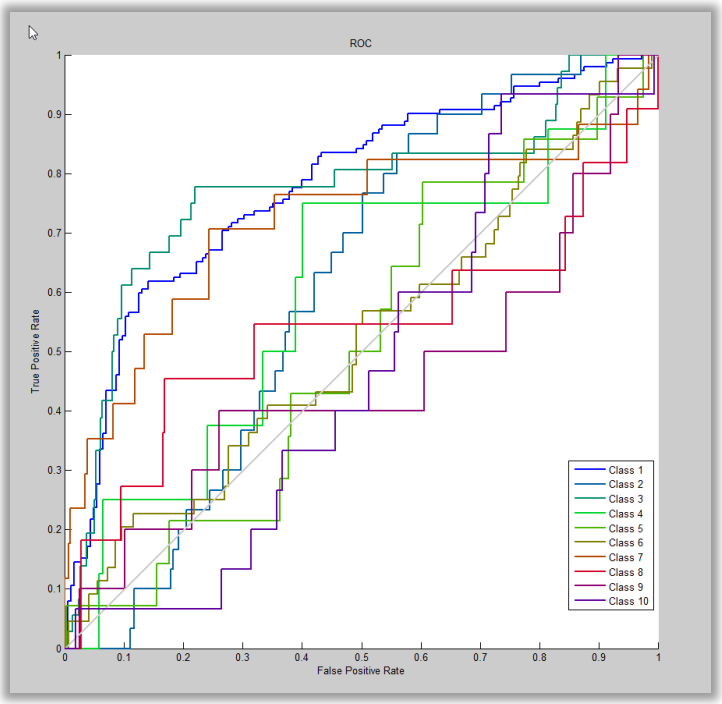
En aquest cas, clarament per sobre de la línia de no discriminació trobem: 1, 4, 5, 3, 7, 9.

Nº Experiment	3
Tipo	C-SVC
Tipo de kernel	RBF: exp(-gamma*|u-v|^2)
gamma	2
cost	8



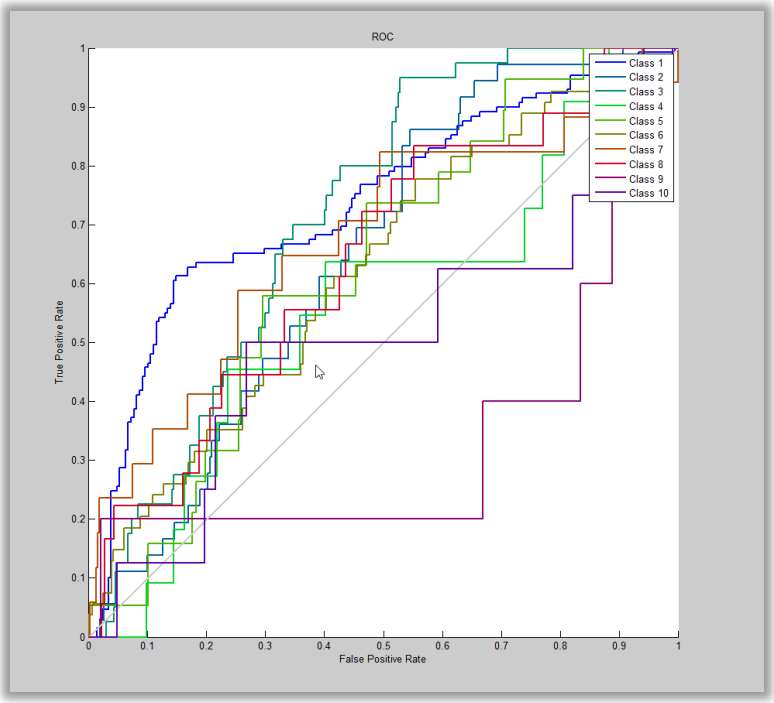
Aquí veiem clarament per sobre de la línia de no discriminació, les accions: 1, 3, 4, 7, 8, 9, 10.

Nº Experiment	4
Tipo	C-SVC
Tipo de kernel	RBF: exp(-gamma*|u-v|^2)
gamma	2
cost	8



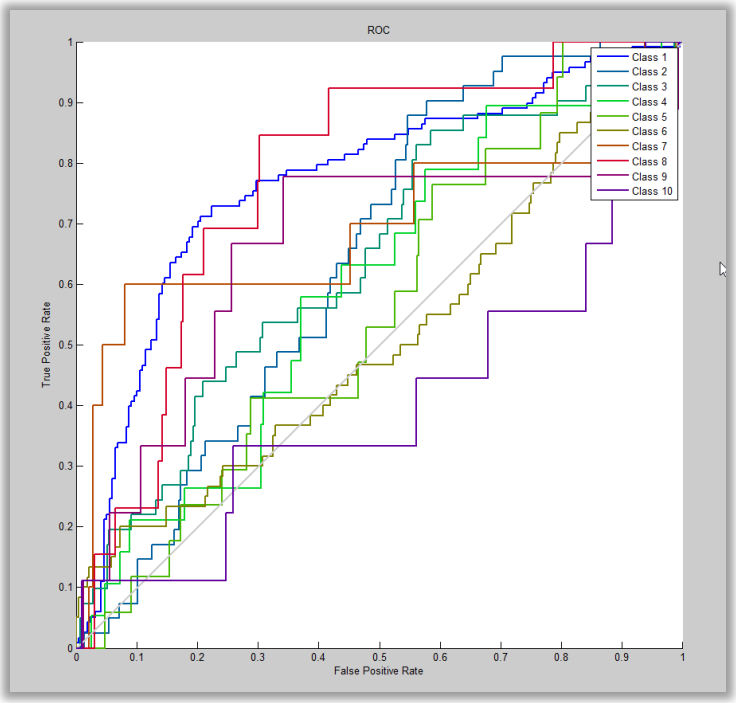
Aquí trobem clarament per sobre de la línia de no discriminació les classes: 1, 3 i 7.

Nº Experiment	5
Tipus	C-SVC
Tipus de kernel	Lineal: u'*v
cost	4



Aquí per sobre de la línea de no discriminació trobem totes les classes: 1, 2, 3, 4, 5, 6, 7 i 8.

Nº Experiment	6
Tipus	C-SVC
Tipus de kernel	Lineal: u'*v
cost	2



Aquí veiem per sobre de la línea de no discriminació, les classes: 1, 2, 3, 7, 8, 9.

7.1.3 Elecció de classificador

Comparant les corbes ROC de les diferents configuracions del classificador basat en xarxa neuronal veiem que la configuració que proporciona unes corbes més allunyades de la línea de no discriminació és la configuració amb una sola capa oculta i 1000 unitats (Experiment nº 5). Veiem doncs que en el nostre cas el fet de tenir més capes ocultes no millora el resultat però sí que ho fa el fet de tenir més unitats per capa.

En quant a les classes predictibles, si observem les classes que queden per sobre de la línea de no discriminació en trobem que les classes 1, 3 i 7 son les úniques que apareixen en tots els casos. A continuació es mostra una taula on es marquen les classes que queden per sobre de la línea de no discriminació a cada experiment:

Tipus de classificador: Xarxa Neuronal

	Nº Experiment					
Accions	1	2	3	4	5	6
1 - Camina	X	X	X	X	X	X
2 - Caient			X		X	X

3 - Estirat a terra	X	X	X	X	X	X
4 - Ajupit					X	
5- Ajupint-se		X				
6-Aixecant-se						
7-Assegut	X	X	X	X	X	X
8-Estirat al sofà	X					X
9-Parat dret		X		X	X	
10-Seient	X	X	X	X		

En quant al classificador basat en màquines de vectors de suport, tenim uns resultats molt similars si només mirem les corbes ROC, de manera que n'escollim una configuració a l'atzar amb kernel RBF i una altra amb kernel lineal per analitzar-les més en profunditat. També observem, com en els experiments amb xarxa neuronal, que les classes que sempre queden per sobre de la corba de no discriminació son les 1, 3 i 7:

Tipus de classificador: SVM

	Nº Experiment					
Accions	1	2	3	4	5	6
1 - Camina	X	X	X	X	X	X
2 - Caient					X	X
3 - Estirat a terra	X	X	X	X	X	X
4 - Ajupit		X	X		X	
5- Ajupint-se		X			X	
6-Aixecant-se					X	
7-Assegut	X	X	X	X	X	X
8-Estirat al sofà	X		X		X	X
9-Parat Dret		X	X			X
10-Seient			X			

Fins ara tenim que de les deu classes que hem considerat inicialment veiem que hi ha tres que en tots els experiments ofereixen unes corbes ROC per sobre de la línea de no discriminació. Ens enfocarem doncs en les classes 1, 3 i 7 per escollir entre un d'aquests tres models per utilitzar-lo al mòdul de detecció:

- Model 1: obtingut de l'experiment nº 5 de classificador basat en xarxa neuronal. Aquest és el que ofereix unes Corbes ROC més separades de la línea de no discriminació.
- Model 2: obtingut de l'experiment nº6 de classificador SVM amb kernel lineal. L'hem escollit a l'atzar entre els tres experiments realitzats amb kernel lineal.
- Model 3: obtingut de l'experiment nº1 de classificador SVM amb kernel RBF. També escollit a l'atzar entre els models obtinguts dels tres experiments amb kernel RBF.

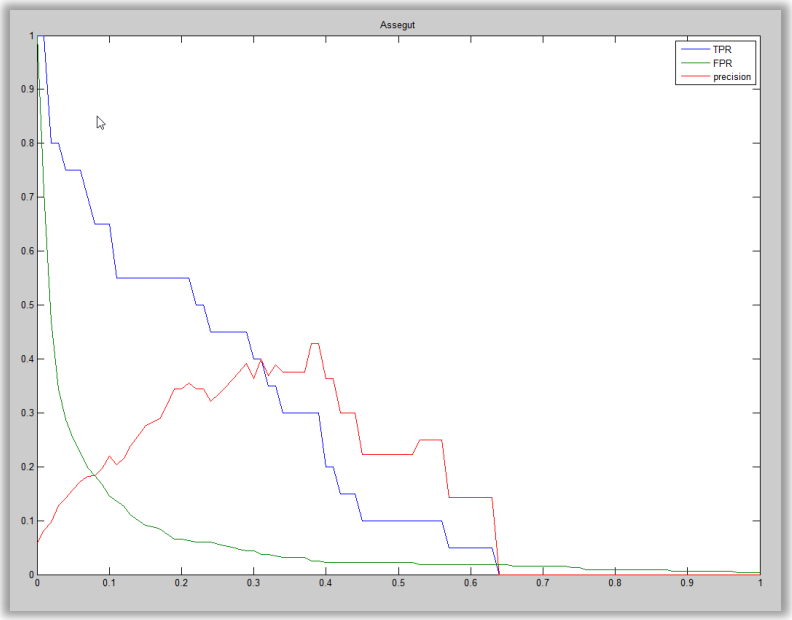
Per comparar entre els tres models hem seguit el criteri definit al punt *6.3.3 Detecció d'accions*, i hem buscat la màxima precisió, sempre que la sensibilitat no fos inferior al 10%.

Obtenim doncs les gràfiques mostrant TPR, TFP i precisió pels models seleccionats i per les classes 1, 3 i 7. A continuació es mostren els resultats obtinguts:

		Classe 1	Classe 3	Classe 7
Model 1	Precisió	70%	18%	43%
	Sensibilitat	23%	22%	30%
Model 2	Precisió	71%	28%	4%
	Sensibilitat	21%	10%	70%
Model 3	Precisió	95%	12%	54%
	Sensibilitat	16%	100%	35%

Ja que amb el model 3 obtenim la major precisió per les classes 1 i 7, hem seleccionat aquest model pel mòdul de detecció.

Exemple de gràfica obtinguda amb el model 2 per la classe 3:

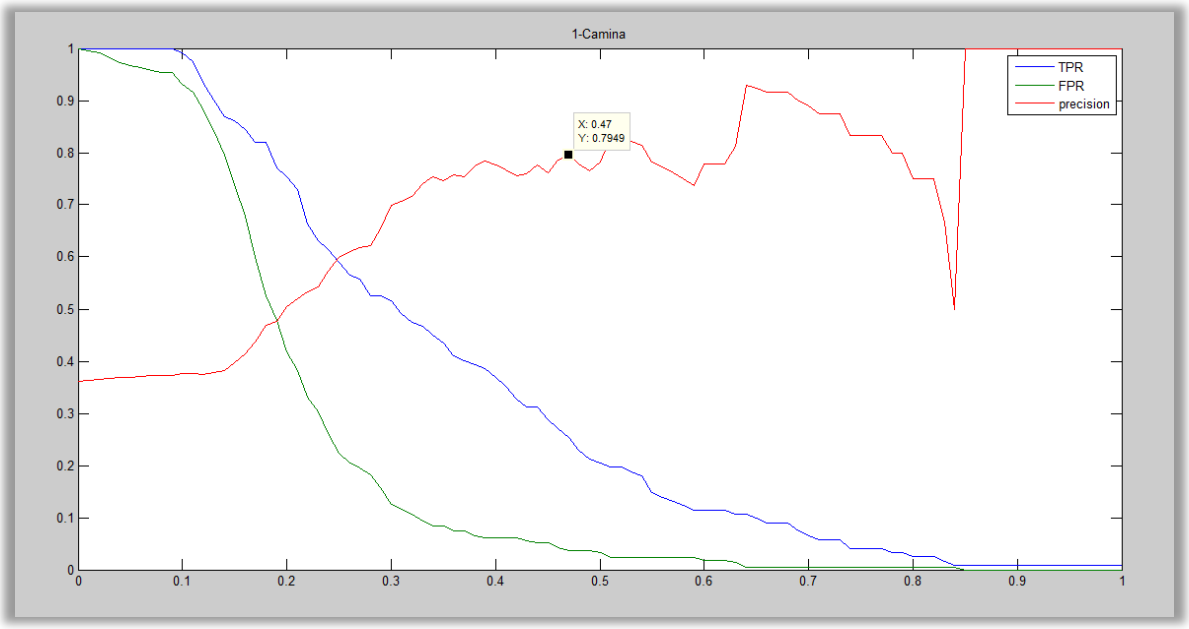


La gràfica mostra la TPR o sensibilitat, la FPR o fall-out (1 - especificitat) i la precisió (TP / (TP + FP)), en funció del llindar de discriminació representat a les abscisses.

7.2 Llindars de discriminació i classes predictibles

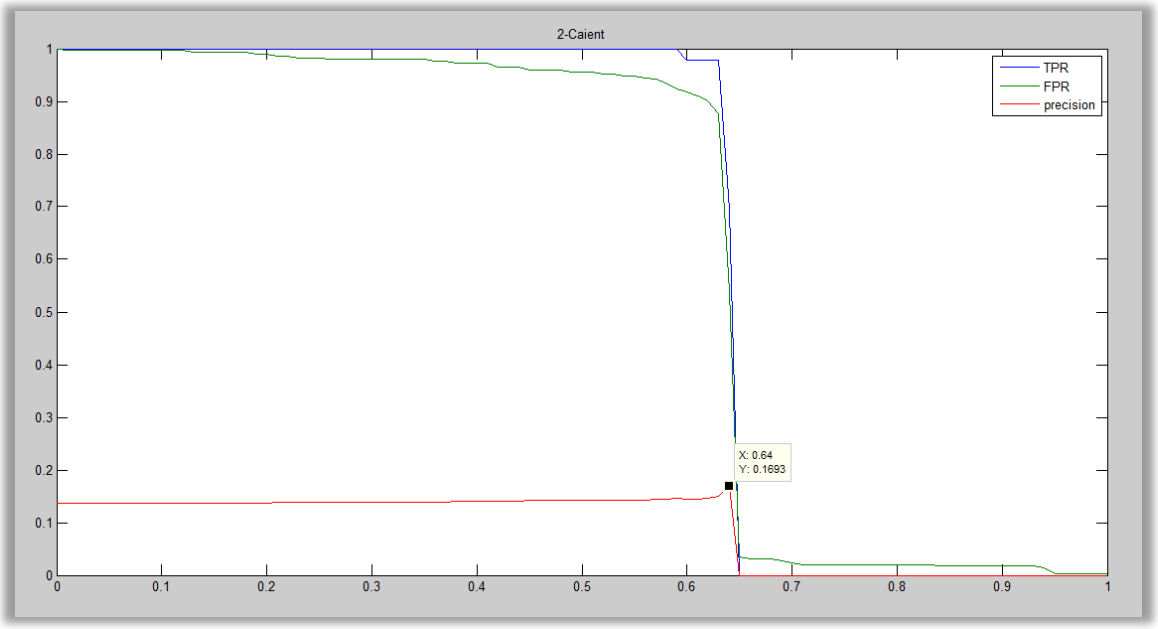
Amb el model seleccionat a l'apartat anterior, que correspon a un classificador de tipus SVM amb kernel RBF, hem cercat manualment els llindars de discriminació amb el criteri definit a l'apartat *6.3.3 Detecció d'accions*. Les classes que no poden complir el criteri han quedat descartades de forma que el mòdul de detecció tan sols emetrà esdeveniments de detecció de les classes que definim com predictibles.

Acció "Caminar"



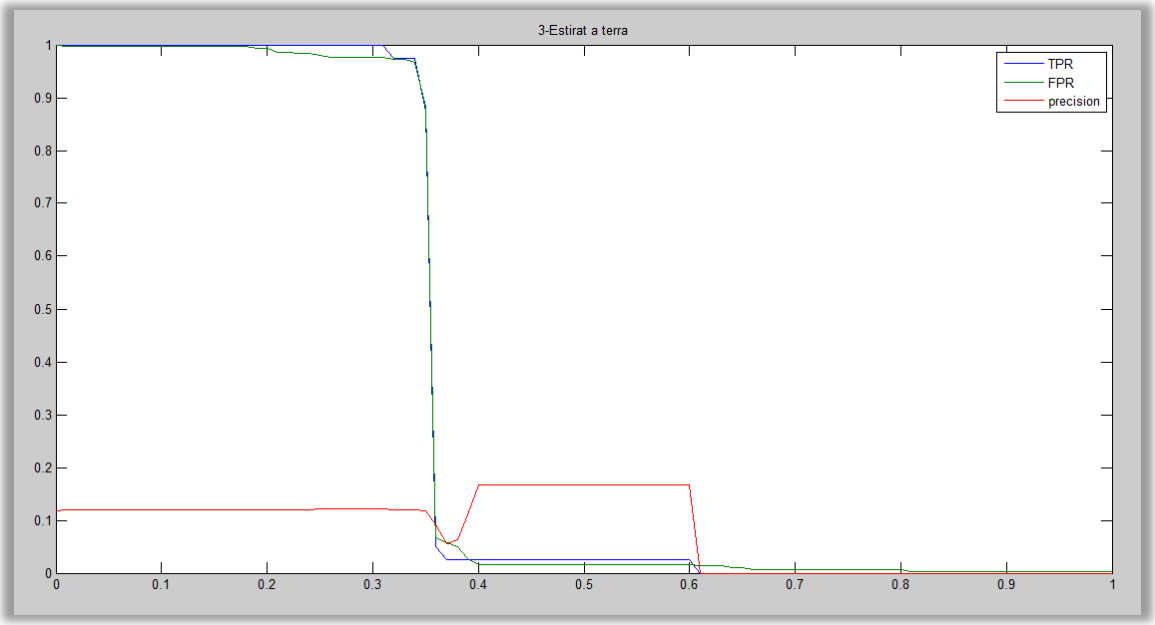
Seleccionem el llindar de probabilitat 0.47, que aportaria una precisió del 80% pel conjunt de validació i una sensibilitat del 25%.

Acció "Caient"



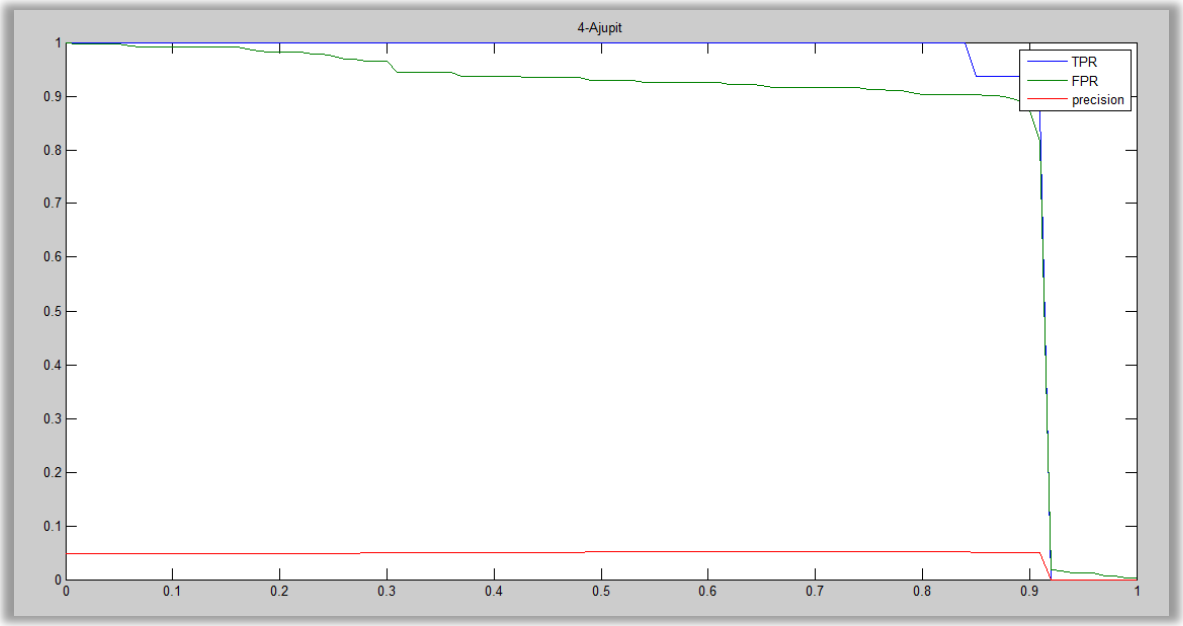
Podem veure en aquest cas que la màxima precisió que podem obtenir al conjunt de validació és d'aproximadament del 17%, clarament insuficient pel nostre propòsit. Descartem doncs l'us d'aquest clasificador ja que no compleix les mínimes condicions de precisió.

Acció "Estirat a terra"



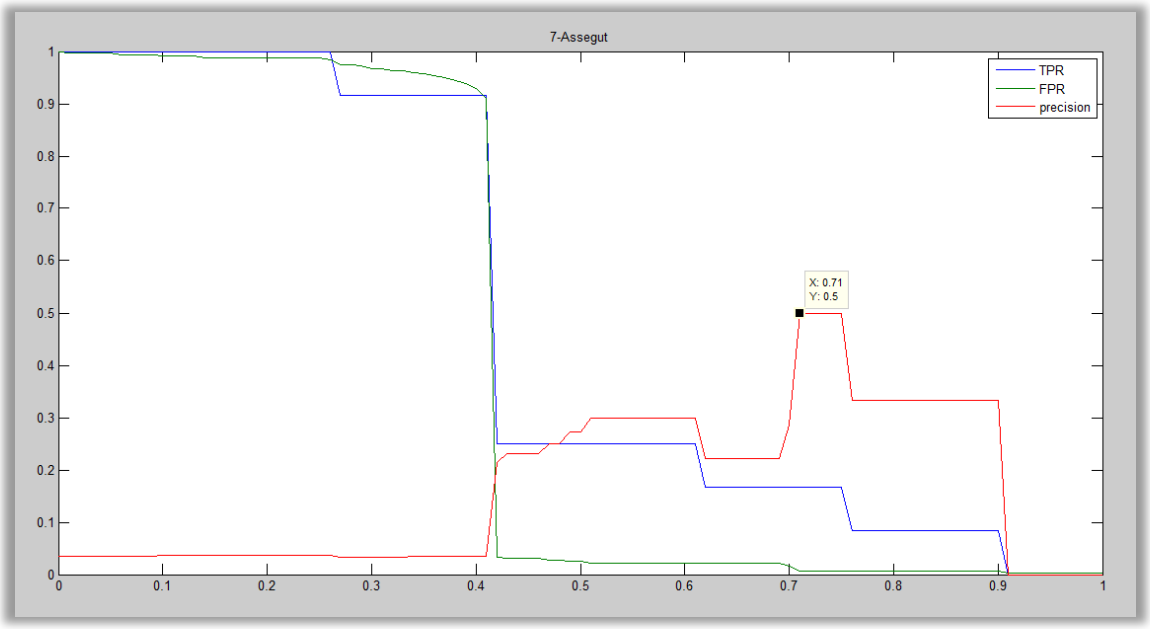
Segons aquesta gràfica també hauriem de descartar aquest classificador. Per un costat veiem una precissió insuficient, una mica inferior al 20% i per altra banda veiem els índex de falsos positius i la sensibilitat estan molt properes. En aquest cas el classificador està fent una predicció gairebè aleatoria.

Acció "Ajupit"



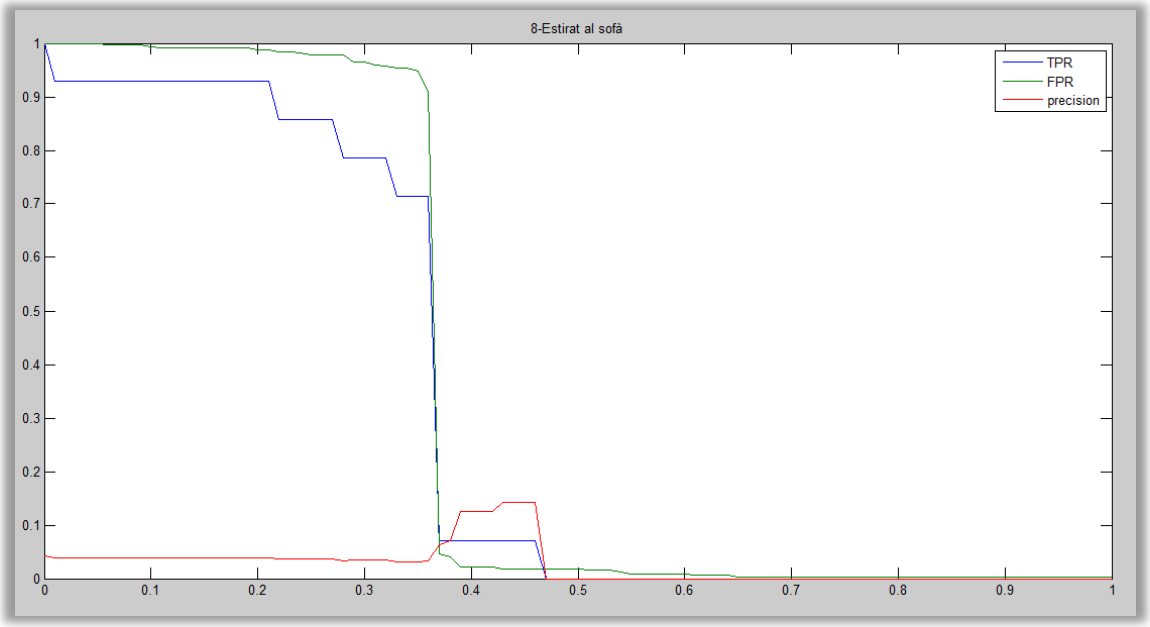
Aquí també podem observar, com en el cas anterior, que els resultats son pràcticament aleatoris, per tant descartem el classificador com a viable per proporcionar prediccions mínimament precises.

Acció "Assegut"



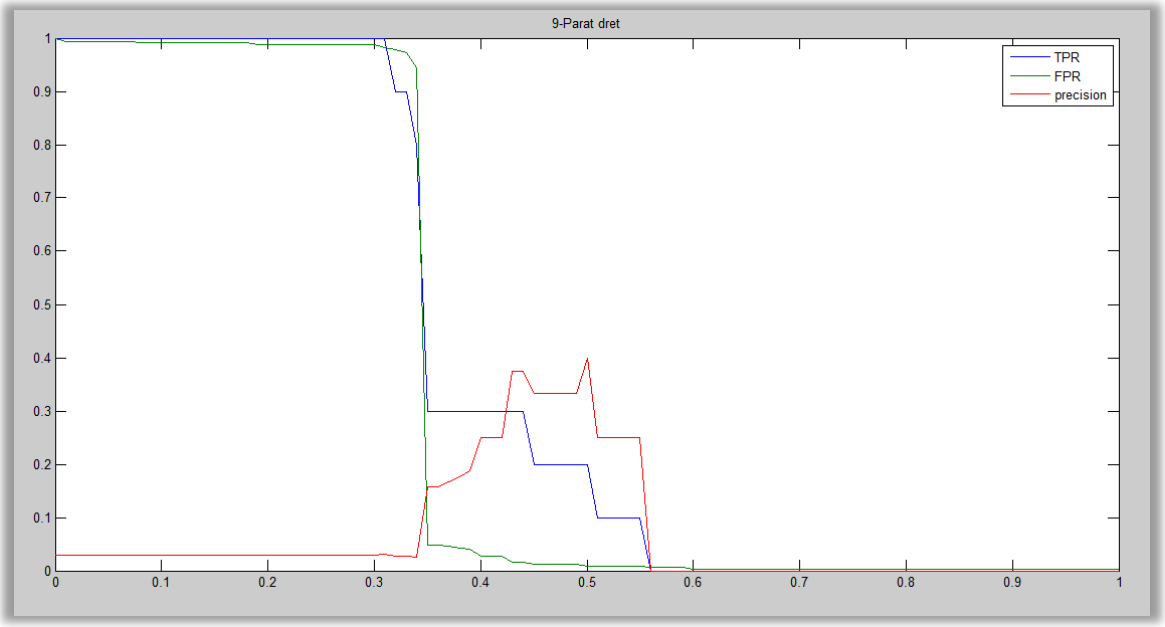
Aquí observem clarament com a partir de 0.4 millora enormement la relació entre la sensibilitat i la tasa de falsos positius alhora que augmenta la precisió. Seleccionarem el llindar de probabiliat de 0.71 i tot i que ens dona una precisió molt ajustada la considerarem predictible.

Acció "Estirat al sofà"



Aquí podem observar que tot i que ni la precisió ni la sensibilitat arriben als mínims requerits, es veu com hi ha un llindar a partir del qual baixa la taxa de falsos positius bastant per sota de la sensibilitat i augmenta notablement la sensibilitat. No obstant descartarem aquest classificador perquè formi part del mòdul de detecció per no assolir els requisits de sensibilitat i precisió.

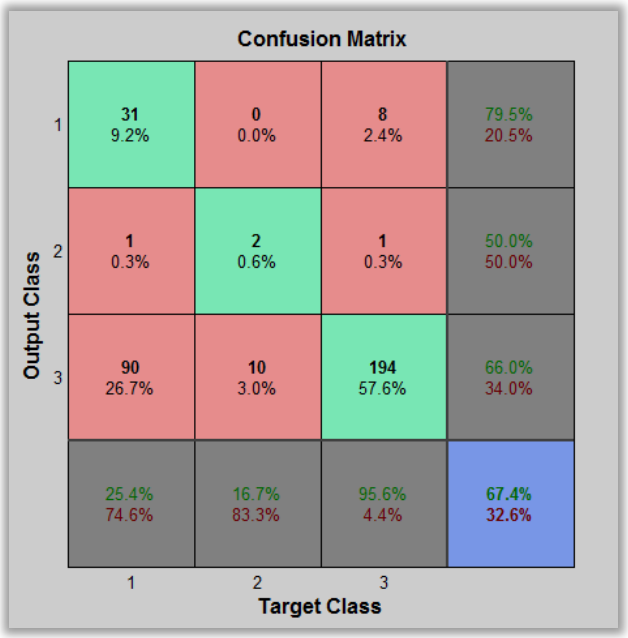
Acció "Parat dret"



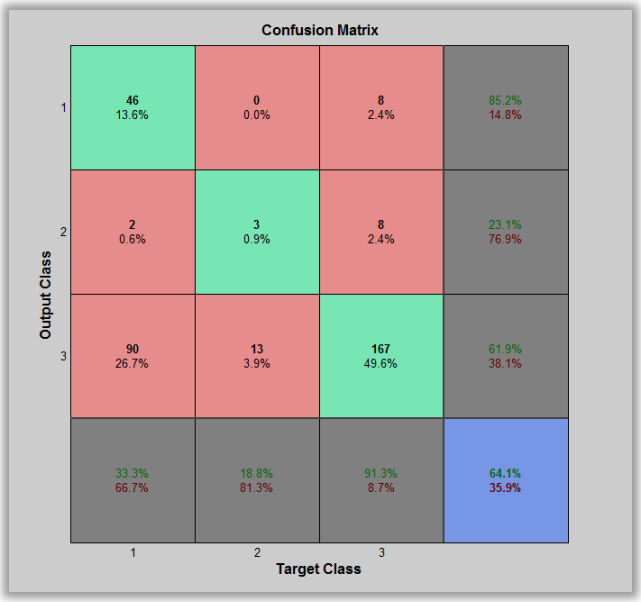
Aquest classificador tampoc assoleix els mínims requetits però veiem un llindar clar a partir del qual la sensibilitat i la precisió estan molt per sobre de la taxa de falsos positius.

Com a resultat de les analisis realitzades s'han descartat 8 de les 10 clases que es consideraven originalment, de manera que tan sols podrem construir un detector per les accions *1 - Camina* i *7- Asegut*. A continuació mostrem la matriu de confusió pel conjunt de validació. Aquí hem renombrat les clases:

- Classe 1 = "Camina"
- Classe 2 = "Asegut"
- Classe 3 = La resta d'accions que hem descartat.



A continuació mostrem la matriu de confusió per al conjunt de test:



8 Conclusions

Finalment, de les deu accions humanes que es volien detectar, tan sols una ofereix un baix nivell d'aleatorietat, l'acció caminar, i una altra, la d'estar assegut apunta a que es podria millorar però encara no asoleix el nivell de precisió mínim que ens haviem plantejat.

Cal destacar que el fet d'escollir un conjunt de dades que emula un escenari real a l'interior d'una vivenda afegeix certes dificultats: condicions de llum no ideals, objectes que provoquen oclusions, accions realitzades de forma natural, sense marcar les poses en un primer pla davant d'una càmera, presència de moviments secundaris als característics de l'acció com caminar mentre es mou una cadira o es transporta una caixes. Aquests tipus de dificultats són habituals quan es tracta de la detecció d'accions en un entorn real.

Als resultats dels experiments que es presenten a l'apartat 7 es pot veure que de mitjana hi han més de sis de les deu classes que queden clarament per sobre de la línea de no discriminació de la corba ROC. Tot i que finalment només una compleixi els mínims de precisió i sensibilitat exigits, hi han signes de no aleatorietat a la resta de classes.

Malgrat que els resultats no son suficientment bons com per traslladar la solució a un sistema real, sobretot per detectar el nombre d'accions amb què vam etiquetar el conjunt de dades, sí que es demostra que el mètode pot ser vàlid per detecar un reduit nombre d'accions. De fet, amb la solució actual podriem disposar d'un modest detector de persones caminant. Només amb això ja podriem millorar un sistema de teleasisténcia amb monitorització d'activitat basat en sensors de moviment i saber, per exemple, quanta estona camina una persona durant el dia sense necessitat de que porti cap dispositiu a sobre.

Hem pogut veure que els dos tipus de classificador que hem tingut oportunitat de provar proporcionen resultats molt similars, el que apunta a que una millora del sistema no depen tant del tipus de classificador i la seva configuració com dels propis exemples, es a dir, el número de dades, el tractament de les imatges o el métode en que ens hem basat per generar

els vectors de característiques. Respecte de la millora dels resultats que podria venir donada per l'ampliació del conjunt de dades, ho veiem en el fet que la classe amb la què obtenim millors resultats és aquella de la que disposem de més exemples, concretament 880, unes tres vegades més que la segona en número d'exemples. En quant al tractament de les dades, seria imprescindible millorar l'extracció de la regió mòvil o foreground en els següents aspectes:

- Eliminar la sensibilitat de l'actual mètode a qualsevol tipus de moviment i centrar-se només en el moviment de persones.
- Eliminar la sensibilitat a moviments secundaris, que no caracteritzen les accions, com ara el moviment d'un braç quan es camina o el fet de portar un objecte a la mà com ara una cadira. Per fer això es fa necessari afegir un mòdul de detecció de persones i de parts del cos.
- Quan la persona que surt a a la imatge porta roba amb una tonalitat similar al color de fons, el foreground resultant es una figura fraccionada. Adressar aquest problema milloraria la qualitat de les MHI.

Centrar-nos en optimitzar l'efciència del reconeixement de les accions amb les que hem tingut un millor resultat: caminar, seure i jeure, abans d'intentar ampliar el conjunt d'accions.

A més, amb un número d'accions més reduït i més exemples per cada acció podriem fer l'experiment que es sugeria a l'apartat *6.2.1 Descripció d'experiments* i que es referia a dividir les classes en funció de la perspectiva en que es veu l'acció ja que com indicàvem la imatge MHI de moviments enregistrats amb la mateixa perspectiva son més semblants entre ells i es de preveure que més fàcils de classificar.

Finalment, una altra possible millora seria ampliar el vector de característiques amb informació sobre la posició del cos. Per això es podria detectar les diferents parts del cos i fer el seguiment, el que té un cost computacional important, o bé fer servir altres tipus de tècniques basades en *poselets* com descriuen Meji et al, (2011, citat a Forsyth i Ponce, 2012).

9 Bibliografia

Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2010). Multiple cameras fall dataset. *DIRO-Université de Montréal, Tech. Rep*, *1350*.

Bobick, A. (2016). *Introduction to Computer Vision* [PowerPoint slides]. Retrieved from https://www.udacity.com/course/introduction-to-computer-vision--ud810

Chaquet, J. M., Carmona, E. J., & Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, *117*(6), 633-659.

Davis, J. W., & Bobick, A. E. (1997, June). The representation and recognition of human movement using temporal templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on* (pp. 928-934). IEEE.

Forsyth, D. & Ponce, J. (2012). *Computer Vision. A Modern Approach.* Person Education Inc.

Hu, M. K. (1962). Visual pattern recognition by moment invariants. *information Theory, IRE Transactions on*, *8*(2), 179-187.