IMPROVED DENSE TRAJECTORIES FOR GESTURE RECOGNITION

Memòria
curso 2017-18 Q2

Roger Pujol Torramorell
Facultat d'Informàtica de Barcelona

Índex de continguts

1 Resum **4**
1.1 Resum . . . 4
1.2 Resumen . . . 4
1.3 Abstract . . . 5

2 Introducció **6**
2.1 Context . . . 6
2.2 Formulació del problema . . . 7
2.3 Actors implicats . . . 7
2.4 Estat de l'art . . . 8
2.5 Abast . . . 9

3 Planificació **10**
3.1 Metodologia i rigor . . . 10
3.2 Descripció de les tasques . . . 11
3.3 Possibles obstacles i solucions . . . 13
3.4 Duració aproximada . . . 14
3.5 Diagrama de Gantt . . . 15
3.6 Recursos . . . 16
3.7 Lleis i regulacions . . . 16
3.8 Modificacions a la planificació . . . 16
3.9 Duració aproximada (definitiva) . . . 17
3.10 Diagrama de Gantt definitiu (Nova planificació) . . . 19

4 Descripció de la tècnica utilitzada **20**
4.1 Extracció de característiques . . . 20
4.2 Codificació de les característiques . . . 23
4.3 Classificació . . . 23

5 Implementació **26**
5.1 *Improved Dense Trajectories* . . . 26
5.2 *Feature encoding* . . . 26
5.3 Classificador . . . 27

6 Sostenibilitat **29**
6.1 Autoavaluació del domini actual 29
6.2 Matriu de sostenibilitat 29
6.3 Àmbit ambiental 30
6.4 Pressupost inicial 30
6.5 Modificacions en els costos 35
6.6 Àmbit econòmic 35
6.7 Àmbit social 36

7 Resultats **38**
7.1 *Dataset* 38
7.2 Selecció i parametrització del classificador 41
7.3 Testeig amb diferents paràmetres per *IDT* 44
7.4 Testeig amb vídeos externs 46

8 Conclusions **50**

Referències **51**

Índex de figures

1 Diagrama de Gantt 15
2 Diagrama de Gantt (Nova planificació) 19
3 Il · lustració del funcionament de *Dense Trajectories* 20
4 Il · lustració del funcionament de *HOG* 21
5 Il · lustració del funcionament d'*Optical Flow* 22
6 Il · lustració del funcionament de *MBH* 22
7 Representació gràfica simplificada de *SVM* 24
8 Exemple de *SVM* amb diferents valors per "C" 25
9 Captura d'*IDT* funcionant amb la càmera i ensenyant les trajectòries . . 26
10 Acció caminar 38
11 Acció córrer 38
12 Acció saltar 38
13 Acció galop lateral 39
14 Acció ajupir-se 39
15 Acció saludar a una mà 39

16 Acció saludar a dues mans 39
17 Acció salt en una posició 40
18 Acció salt de tisora 40
19 Acció salt a peu coix 40
20 Exemples dels vídeos extres 40

Índex de taules

1 Taula de duracions aproximades 14
2 Taula de duracions aproximades 18
3 Matriu de sostenibilitat 30
4 Pressupost *Hardware* 31
5 Pressupost *Software* 31
6 Pressupost de recursos humans 32
7 Dedicació aproximada de cada rol 32
8 Pressupost de possibles costos causats per desviacions 33
9 Costos indirectes 33
10 Pressupost total 34
11 Cost estimat de les tasques 34
12 Dedicació aproximada de cada rol 35
13 Pressupost de recursos humans 35
14 Matriu de confusió del millor model lineal amb PCA 42
15 Matriu de confusió del millor model no-lineal amb PCA 42
16 Matriu de confusió del millor model lineal sense PCA 43
17 Matriu de confusió del millor model no-lineal sense PCA 44
18 Matriu de confusió (modificant la densitat) 45
19 Matriu de confusió (Prova amb *PCA*) 47
20 Matriu de confusió (Prova sense *PCA*) 48

1 Resum

1.1 Resum

El reconeixement de gestos/accions, ha estat sent una àrea d'investigació molt activa durant les ultimes tres dècades. Gràcies a aquesta investigació continuada, a l'actualitat existeixen una gran varietat d'algoritmes amb molts bons resultats, capaços de reconèixer accions en vídeos amb una precisió molt alta.

En aquest projecte utilitzarem eines que actualment tenen resultats al nivell del estat-del-art com *Improved Dense Trajectories*, que és un algoritme per extreure característiques útils per reconeixer accions, per tal d'obtenir els millors resultats possibles. El nostre objectiu en aquest projecte és aconseguir un detector de gestos robust capaç de classificar uns moviments prèviament establerts i a més a més fer que funcioni en una càmera activa amb el menor retràs possible. Per entrenar el classificador i provar l'efectivitat del projecte, utilitzarem el *Dataset* de Weizmann i per tant els moviments que tractarem de detectar seran els 10 moviments definits en aquest conjunt de vídeos.

1.2 Resumen

El reconocimiento de gestos/acciones, ha estado siendo una área de investigación muy activa durante las ultimas tres décadas. Gracias a esta investigación continuada, en la actualidad existen una gran variedad de algoritmos con muy buenos resultados, capaces de reconocer acciones en vídeos con una precisión muy alta.

En este proyecto utilizaremos herramientas que actualmente tienen resultados al nivel del estado-del-arte como *Improved Dense Trajectories*, que es un algoritmo para extraer características útiles para reconocer acciones, para obtener los mejores resultados posibles. Nuestro objetivo en este proyecto es conseguir un detector de gestos robusto capaz de clasificar unos movimientos previamente establecidos y además hacer que funcione en una camera activa con el menor retraso posible. Para entrenar el clasificador y probar la efectividad del proyecto, utilizaremos el *Dataset* de Weizmann y por lo tanto los movimientos que trataremos de reconocer serán los 10 movimientos definidos en este conjunto de vídeos.

1.3 Abstract

The gesture/action recognition has been an active research area for over three decades. Thanks to this continued research, nowadays there are a great variety of algorithms with great results, capable to recognize actions in videos with a high precision.

In this project we will use tools that currently have results similar to the state-of-art like *Improved Dense Trajectories*, which is an algorithm used to extract useful features for action recognition, to obtain the best possible results. Our goal in this project is to achieve a robust gesture detector capable to classify a previously established set of movements and also make it work with an active camera with the minimum possible delay. To train the classifier and test the effectiveness of the project, we will use the Weizmann dataset and therefore the movements we will try to recognize will be the 10 movements defined in this set of videos.

2 Introducció

2.1 Context

A l'actualitat cada vegada més es fa el possible per aconseguir maneres naturals d'interactuar amb els dispositius informàtics. Per exemple, ara ja és habitual poder utilitzar els *smartphones* parlant-hi com si fos una persona gracies a assistents com *Siri*, *Google Assistant*, *Cortana*... que ens ajuden a interactuar de manera molt més natural que els sistemes més tradicionals.

Això ens permet que interactuar amb els dispositius sigui molt més intuïtiu i accessible per gent que no està tant habituada a utilitzar coses com mòbils, ordinadors... Per tant és una manera molt efectiva de fer arribar la tecnologia a molta més gent sense que necessitin aprendre moltes coses noves.

La visió per computador es dedica a aconseguir que un ordinador pugui obtenir informació d'imatges, així que també podria ser una manera que té un ordinador o altre dispositiu per interpretar coses del món real. Un dels temes més interessants en els quals s'està investigant actualment en la visió per computador, és la detecció de gestos. Aquesta seria una manera molt intuïtiva i senzilla d'interactuar amb la tecnologia. Com he mencionat abans la comprensió de veu per dispositius mòbils i ordinadors, però en comptes de parlar el que es faria seria fer algun gest concret que el dispositiu comprendria.

La detecció de gestos, és un tema que es porta treballant des de fa molt temps i al qual s'han aconseguit alguns algoritmes bastant bons al llarg dels anys. Un dels més destacats a l'actualitat és *"Improved Dense Trajectories"*[1] el qual estudiarem a fons per tal d'aprofitar-ne tot el seu potencial.

En aquest projecte procurarem aconseguir un algoritme robust per detectar uns gestos determinats prèviament realitzats davant d'una càmera. El problema és més complex del que podria semblar, ja que per un ordinador no és tant fàcil com als humans adaptar-se a canvis en l'entorn, com podria ser l'il · luminació, el fons, distància a la càmera, angle de visió... Aquest tema ha estat abordat per algunes entitats amb l'objectiu d'aconseguir una interacció més humana amb els ordinadors. Hi ha diversos algoritmes utilitzats en aquesta finalitat, en aquest projecte ens centrarem en utilitzar i adaptar al nostre propòsit l'algoritme *"Improved Dense Trajectories"*.

2.2 Formulació del problema

2.2.1 Motivació

Com ja he mencionat al apartat anterior, la detecció de gestos pot servir com una manera d'interactuar amb els computadors o altres dispositius de forma més natural i humana. A més a més aquest camp encara té un gran marge de millora a la actualitat. Per això tractar amb els algorismes per obtenir suficients característiques de seqüencies d'imatges (com *"Improved Dense Trajectories"*) per poder entrenar un classificador i obtenir així un detector de gestos robust, és una bona manera per avançar en aquest camp.

2.2.2 Objectius

Els objectius del projecte són:

- Implementar un programa capaç de detectar gestos.
- Aquest programa ha de poder detectar i classificar entre 8 i 15 gestos diferents prèviament establerts.
- Tot lo anterior ha d'acabar funcionant en temps real a través d'una càmera.

2.3 Actors implicats

2.3.1 Desenvolupador

El desenvolupador és l'encarregat d'implementar i documentar el software del projecte. També és el responsable de escriure les memòries i altre documentació necessària. Haurà de treballar sempre en acord amb el director, però només el desenvolupador és el responsable de complir les dates limit. En aquest projecte el desenvolupador seré jo, Roger Pujol Torramorell.

2.3.2 Director

El director és el màxim responsable de guiar, donar consell i ajudar en el que sigui necessari al desenvolupador. En aquest projecte el director serà Joan Climent Vilaró.

2.3.3 Beneficiaris

Com que el resultat final d'aquest projecte no és un producte concret, no té beneficiaris finals directes. Però a partir dels resultats d'aquest producte es podrien crear una gran

varietat de productes amb moltes finalitats diferents. Per exemple, es podria utilitzar per controlar un ordinador o un robot amb instruccions bàsiques donades a traves de gestos, també es podria provar d'entrenar amb un dataset molt més gran i intentar fer un petit interpret de llenguatge de signes... Les possibilitats són moltes si els resultats finals són suficientment bons.

2.4 Estat de l'art

En matèria de detecció gestual cal distingir entre els algoritmes que es centren només en el moviment de les mans i els que inclouen part o el cos complet.

En la detecció de gestos de les mans acostuma a ser per escenaris més estèrils, ja que tendeixen a segmentar les mans de la imatge marcar punts clau (*keypoints*) que serien les puntes dels dits i altres punts d'interès. Aquests *keypoints* després es segueixen en el temps i es pot saber que està fent la mà. El problema d'aquest sistemes es que son molt sensibles a oclusions i altres possibles problemes de soroll de la imatge.

En la detecció de gestos tenint en compte una imatge més natural i tenint en compte tot el cos, els primers algoritmes que s'utilitzaven eren *KLT Tracker* o *SIFT* (que són algoritmes que lliguen punts similars entre 2 imatges) per veure la trajectòria en que es desplacen els punts d'interès. El problema és que aquests no proporcionaven ni suficients punts ni suficient qualitat de punts.

"Dense Trajectories"[3] el que feia era simplement buscar punts densos per seguir-ne el moviment entre fotogrames i això va resultar ser molt més robust. Uns anys més tard van millorar l'algoritme utilitzant *SURF* [5] per eliminar soroll de fons com podria ser moviments de la càmera o moviment en el fons, també van utilitzar un detector d'humans per eliminar inconsistències en les connexions dels punts. Aquest últim el van anomenar *"Improved Dense Trajectories"* i és el que utilitzaré en el projecte, ja que en la actualitat continua sent dels algoritmes més robustos per la detecció de gestos.

Per tant en el nostre projecte en principi utilitzarem alguns dels algoritmes més punters en el sector amb els quals treballarem per obtenir uns resultats similars als millors que es poden aconseguir a l'actualitat.

2.5 Abast

Primer es començara per compilar instal · lar i comprovar que funcionen de manera adequada algunes de les eines principals que utilitzarem com a fonaments del projecte. L'eina principal serà *OpenCV* (la llibreria de visió per computador més gran actualment).

Caldrà també obtenir un dataset de gestos gravats en vídeo, a ser possible de conjunts classificats d'una llista d'entre 10 i 15 gestos diferents amb gran varietat de vídeos per cada gest. Seria desitjable que la varietat dels vídeos inclogues diferents il · luminacions, angles de càmera, persones realitzant el gest... Aquest pot ser un punt complicat de complir, ja que aconseguir un dataset tant concret no és trivial i és possible que aquest no existeixi i s'hagi de crear a base de gravacions.

Seguidament s'estudiarà el codi principal de *"Improved Dense Trajectories"*[1] per veure com està implementat concretament l'algoritme per així poder-ne aprofitar el màxim potencial possible en el nostre objectiu. Un cop estudiat, es modificarà de tal manera que obtingui suficients *features* de seqüencies d'imatges per poder passar al següent punt.

Després de trobar com extraure *features* rellevants caldrà buscar un classificador que pugui treballar bé amb aquesta informació. Aquest s'haurà d'ajustar per tal d'obtenir uns millors resultats. A partir d'aquí ja tindríem un programa capaç de detectar i distingir uns gestos preestablerts (entre 8 i 15 gestos) amb certa precisió en seqüencies d'imatges. Finalment l'objectiu seria aconseguir que tot això funcioni en temps real a través d'una càmera.

Amb tot això caldria passar-hi els test adients com provar que la precisió es manté amb diferents actors fent els gestos i altres casos per assegurar que funciona correctament en tots (o la majoria) els casos.

Tot i que el programa resultant està pensat per acabar funcionant com a forma d'interacció entre un humà i un robot, en aquest projecte ens limitarem a fer-lo funcionar en un ordinador i deixarem la aplicació a altres dispositius fora dels nostres objectius.

3 Planificació

La duració estimada d'aquest projecte és de 4 mesos. El projecte va començar el 19 de febrer i la data límit és el 18 de juny, una setmana abans del període de presentacions orals.

3.1 Metodologia i rigor

A causa del ajustat calendari que tenim per desenvolupar el projecte, el millor serà utilitzar una metodologia àgil. Aquest tipus de metodologies proporcionen més rapidesa i flexibilitat en el desenvolupament, obtenint així resultats en menor temps que utilitzant altres mètodes habituals. Les metodologies àgils acostumen a ser bastant orientades al treball en grup, tot i això, encara ens podrem beneficiar dels seus conceptes bàsics per un projecte individual. Concretament el mètode que utilitzaré en aquest projecte serà *"Extreme Programming"*[6] abreviat com a XP.

3.1.1 Cicles curts de desenvolupament

Mitjançant l'ús de cicles d'entre una o dues setmanes, s'aconsegueix mantenir més fàcilment al dia el treball i ser conscient en cas de endarrerir-se en el desenvolupament.

3.1.2 Feedback intensiu del client

Tot i no haver-hi un client real, podem considerar el director com a client del projecte. Així el director podrà donar la seva opinió de manera més continuada, això evitaria possibles malentesos de conceptes o solucionar-los abans de que sigui difícil tornar enrere.

3.1.3 Eines de seguiment

GitHub serà l'eina principal per tenir un seguiment òptim del treball, ja que el sistema de *commits* permetrà tenir documentats tots els canvis en el codi fets per cada iteració dels cicles de desenvolupament. A més a més també té altres opcions pel seguiment de problemes o marcar-se fites que ens seran molt útils per tenir sempre en compte el que s'ha de fer.

3.1.4 Mètodes de validació

Per evitar que els petits errors vagin a més causant problemes greus al llarg del temps, realitzarem petites proves pel codi a cada iteració en els cicles de desenvolupament. Això

es complementarà amb el feedback intensiu del director com ja he mencionat en apartats anteriors.

Al seguir una metodologia àgil és possible que hi hagi modificacions a la planificació inicial al trobar nous requisits o contratemps. Per evitar que aquestes desviacions de la planificació original provoquin un retràs massa gran que impossibilitin acabar a temps, s'han deixat unes setmanes de marge al final del procés en la planificació.

3.2 Descripció de les tasques

3.2.1 Configuració del entorn

Per poder començar el projecte és necessari tenir apunt les llibreries i programes que utilitzarem. Les instal · lacions que ens consumiran més temps són *OpenCV* que s'ha de compilar amb els paràmetres adequats pels nostres propòsits. També caldrà compilar i fer funcionar l'implementació dels propis creadors de *"Improved Dense Trajectories"*[1] (IDT).

3.2.2 Fita inicial

Aquesta part és independent de la implementació del projecte i per això es pot fer en paral · lel amb la configuració del entorn. La fita inicial està orientada a planificar i començar a escriure la memòria del projecte. Consta dels següents apartats:

- Definició de l'abast i contextualització
- Planificació temporal
- Gestió econòmica i sostenibilitat
- Presentació preliminar
- Plec de condicions
- Documentació final

3.2.3 Obtenció i anàlisi del dataset

Per tal de poder provar tot el que s'implementi, és necessari tenir un dataset que s'ajusti al nostre objectiu. Per això una de les primeres coses que haurem de fer després d'aconseguir

fer funcionar *IDT* serà obtenir un dataset per poder comprovar el seu funcionament. En el cas de no poder obtenir un dataset ja creat, s'hauria de generar un a mida pel nostre propòsit.

Un cop obtingut el dataset, s'haurà d'analitzar tota la informació que inclou per poder aprofitar-la al màxim.

3.2.4 Anàlisi i adaptació del codi d'*IDT*

Com ja hem mencionat anteriorment, utilitzarem l'algoritme *IDT* de base pel nostre projecte. Per assolir uns bons resultats, analitzarem en profunditat l'implementació de del algoritme per tal de fer-ne un bon ús.

Un cop entenem i comprenguem el funcionament del algoritme a la perfecció, podrem modificar la implementació per ajustar-la correctament. Amb aquesta adaptació pretenem tenir un algoritme que ens doni una informació amb la qual pugui treballar un classificador per classificar els gestos de diferents seqüencies d'imatges.

3.2.5 Classificador

En aquest punt ja podrem obtenir informació de seqüencies d'imatges i necessitarem un classificador que tracti amb aquesta informació. Primer serà important triar un bon classificador.

Seguidament s'han de trobar els paràmetres amb els quals s'aprofiti al màxim el potencial del classificador. Quan és tingui la parametrització podrem començar a entrenar-lo i veure els primers possibles resultats. Utilitzarem els resultats obtinguts per repetir el procés anterior i afinar encara més els paràmetres.

3.2.6 Adaptació a càmera

En el moment que tenim un programa que classifica gestos en seqüencies d'imatges, podem prosseguir a implementar el codi per aconseguir que el classificador funcioni amb una càmera que estigui gravant. Quan aconseguim que funcioni tot amb una càmera, s'hauran de fer les proves adients per veure amb quina precisió funciona en diferents situacions realistes.

3.2.7 Fita final

En aquest punt s'acabarà de polir el codi del projecte i finalment s'escriurà tota la part restant de les memòries per acabar tota la documentació.

3.3 Possibles obstacles i solucions

Com que utilitzarem una metodologia àgil, en principi ens permetrà adaptar-nos sobre la marxa i així facilitar la correcció d'errors o altres imprevistos. Tot i això també com ja he mencionat abans en la planificació hem intentat deixar unes setmanes extres abans de l'entrega per marge en casos de problemes inesperats.

3.3.1 Dificultats per trobar el dataset

Com ja he mencionat al "Abast", trobar un dataset que s'ajusti a les necessitats del projecte pot ser molt complicat. El problema és que difícilment hi haurà un dataset que compleixi tot el que volem i tal com ho volem. Com a solució es podria crear un dataset ajuntant datasets diferents orientats a altres qüestions per ajustar-los a les nostres necessitats, o en cas de no trobar res així sempre es podria gravar la nostre pròpia base de dades amb diferents persones i diferents situacions.

3.3.2 Bugs

En codis tant complexes com el d'aquest projecte és inevitable tenir alguns errors o *bugs*. Per tal d'evitar-ho intentarem provar sempre el codi en cada canvi que s'hi faci en cada iteració (de la metodologia àgil) i així intentar no arrossegar errors a mesura que avancem en la implementació.

3.3.3 Calendari

El calendari és bastant ajustat així que és molt fàcil endarrerir-se i arribar a un punt on no sigui possible la finalització del projecte a temps. Per això serà important des del primer dia seguir amb la planificació i posant més esforç en el cas de veure un retràs respecte els plans.

3.3.4 Fracassar en compilar i fer funcionar *IDT*

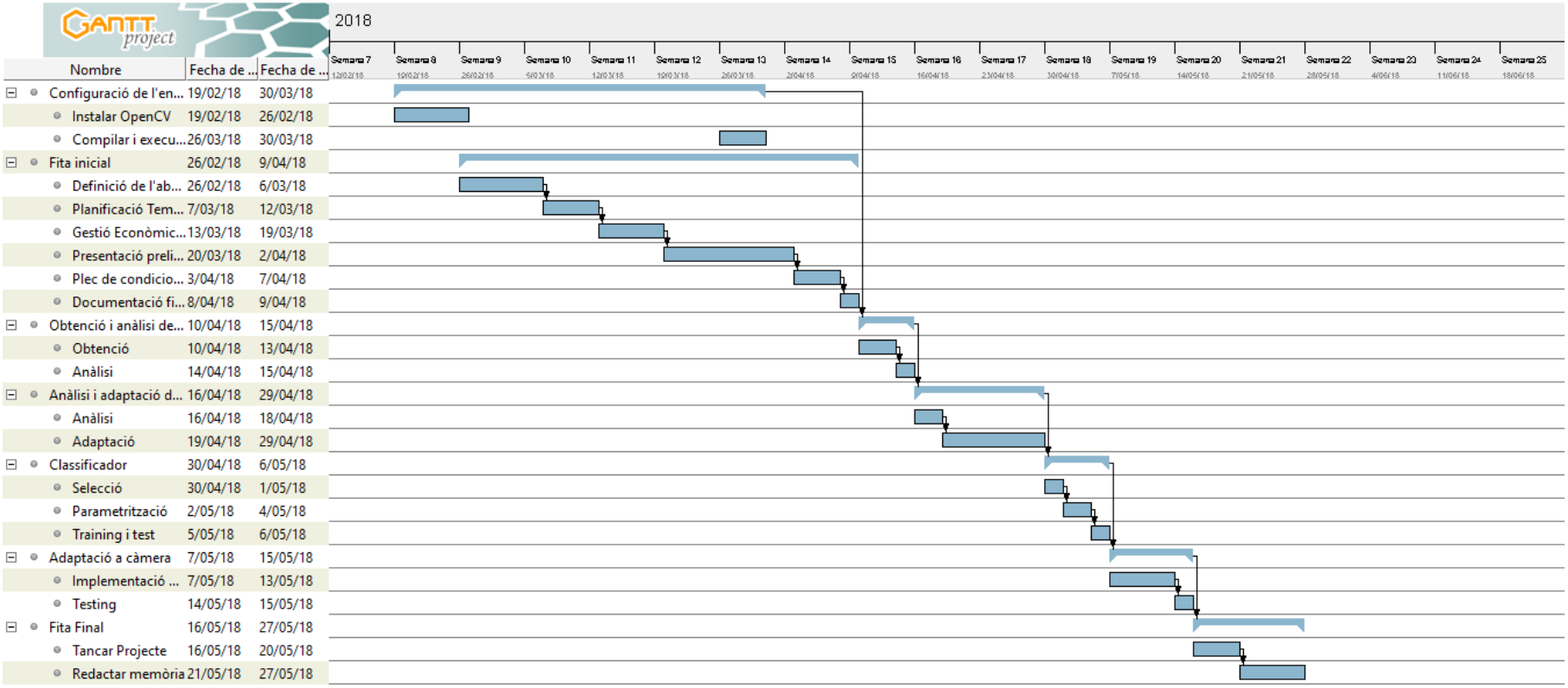
És possible que la implementació que pretenem utilitzar de *IDT* no compili o no funcioni correctament degut a variacions en les llibreries, actualitzacions de funcions... En aquest cas es podrien tenir tres possibles solucions depenent de la gravetat del problema. Primer si el codi falla per petits problemes deguts a diferents versions, es podria adaptar directament el codi per fer que funcionés. Si amb això seguis sense funcionar, es podrien buscar altres implementacions d'internet. Com que és un algoritme bastant conegut és possible trobar-ne un altre que funcioni correctament. Si finalment no haguéssim trobat cap implementació, s'hauria de fer una implementació pròpia. Aquest seria el pitjor cas possible ja que provocaria un retràs difícil de recuperar, però també és una possibilitat molt remota la qual és molt improbable que acabi passant.

3.4 Duració aproximada

Tasca	Duració aproximada
Configuració del entorn	20
Fita inicial	90
Obtenció i anàlisi del dataset	25
Anàlisi i adaptació del codi d'IDT	100
Classificador	70
Adaptació a càmera	55
Fita final	60
Total	420

Taula 1: Taula de duracions aproximades

3.5 Diagrama de Gantt



Nombre	Fecha de ...	Fecha de ...
Configuració de l'en...	19/02/18	30/03/18
Instalar OpenCV	19/02/18	26/02/18
Compilar i execu...	26/03/18	30/03/18
Fita inicial	26/02/18	9/04/18
Definició de l'ab...	26/02/18	6/03/18
Planificació Tem...	7/03/18	12/03/18
Gestió Econòmic...	13/03/18	19/03/18
Presentació preli...	20/03/18	2/04/18
Plec de condicio...	3/04/18	7/04/18
Documentació fi...	8/04/18	9/04/18
Obtenció i anàlisi de...	10/04/18	15/04/18
Obtenció	10/04/18	13/04/18
Anàlisi	14/04/18	15/04/18
Anàlisi i adaptació d...	16/04/18	29/04/18
Anàlisi	16/04/18	18/04/18
Adaptació	19/04/18	29/04/18
Classificador	30/04/18	6/05/18
Selecció	30/04/18	1/05/18
Parametrització	2/05/18	4/05/18
Training i test	5/05/18	6/05/18
Adaptació a càmera	7/05/18	15/05/18
Implementació ...	7/05/18	13/05/18
Testing	14/05/18	15/05/18
Fita Final	16/05/18	27/05/18
Tancar Projecte	16/05/18	20/05/18
Redactar memòria	21/05/18	27/05/18

Figura 1: Diagrama de Gantt

3.6 Recursos

3.6.1 *Hardware*

En aquest projecte l'únic recurs hardware que utilitzaré serà el meu ordinador personal: Lenovo Yoga 510-14IKB. Aquest ordinador té la potencia necessària per executar el codi que farem, a més a més d'una càmera per poder fer les proves.

3.6.2 *Software*

De software utilitzarem:

- Atom: un editor de text amb moltes utilitats orientades a una programació més còmode.
- OpenCV: la llibreria més gran i open source de visió per computador.
- LaTeX: un sistema de composició de textos d'alta qualitat.
- ShareLatex: editor online per crear documents amb LaTeX.

3.7 Lleis i regulacions

Per aquest projecte tot el codi que utilitzem és *open source* amb l'excepció de l'algoritme *SIFT*[4] que fa servir *IDT*. *SIFT* és un algoritme patentat que limita el seu ús comercial, però com que en el nostre cas en farem ús acadèmic (no comercial), no ens afecta. Només ens afectaria en el cas de que aquest projecte s'acabés comercialitzant i actualment no és l'objectiu.

3.8 Modificacions a la planificació

La planificació original s'ha hagut de modificar degut principalment a 2 imprevistos, els quals també han fet incrementar lleugerament els costos esperats inicialment.

3.8.1 Configuració del entorn

Primer la configuració del entorn, va causar un retràs degut a problemes a l'instal · lació d'*OpenCV*. Els problemes van ser de primer instal · lar versions incorrectes d'*OpenCV*, després les versions correctes donaven problemes de compatibilitat en el meu terminal i finalment vaig instal · lar la ultima versió *release*, la qual és compatible amb el meu

ordinador però no amb la implementació d'*IDT*. Per aquest motiu vaig haver d'adaptar la implementació d'*IDT* per fer-la compatible a la ultima versió d'*OpenCV*.

3.8.2 Nova tasca: Feature Encoding

El segon imprevist va ser el *feature encoding* o codificació de les característiques (explicat en detall al apartat "Descripció de la tècnica") d'*IDT*, ja que en un principi estaria inclòs a la tasca de "Anàlisi i adaptació del codi d'IDT ", però al no estar en la implementació del algoritme, la he hagut d'implementar des de zero. Per aquest motiu l'he considerat una tasca nova i ha causat un increment en el temps de desenvolupament final.

3.8.3 Anàlisi d'alternatives

Pels problemes de compatibilitat d'*OpenCV* la millor alternativa per solucionar el problema ha sigut utilitzar la ultima versió estable de la llibreria i actualitzar el codi d'*IDT* per tal de que funcionés correctament.

Per la nova tasca *Feature Encoding* es necessari escollir com codificarem la informació que retorna *IDT* per poder-la utilitzar posteriorment al classificador, ja que *IDT* retorna les característiques (*features*) del vídeo en mides irregulars depenent en el nombre de trajectòries que trobi a cada frame i el classificador necessita que cada element tingui una mida estàndard. Les alternatives per solucionar aquest problema són principalment 2, utilitzar *Bag of Words* o utilitzar *Fisher Vetors* per estandarditzar la informació. D'aquestes possibles solucions he decidit utilitzar *Fisher Vectors* perquè aprofita més informació de la entrada que *BoW*.

3.9 Duració aproximada (definitiva)

Tenint en compte les modificacions a la planificació original esmentades al apartat anterior la duració aproximada del projecte quedaria de la següent manera:

Tasca	**Duració aproximada**
Configuració del entorn	50
Fita inicial	90
Obtenció i anàlisi del dataset	25
Anàlisi i adaptació del codi d'IDT	80
Feature Encoding	40
Classificador	70
Adaptació a càmera	55
Fita final	60
Total	470

Taula 2: Taula de duracions aproximades

3.10 Diagrama de Gantt definitiu (Nova planificació)

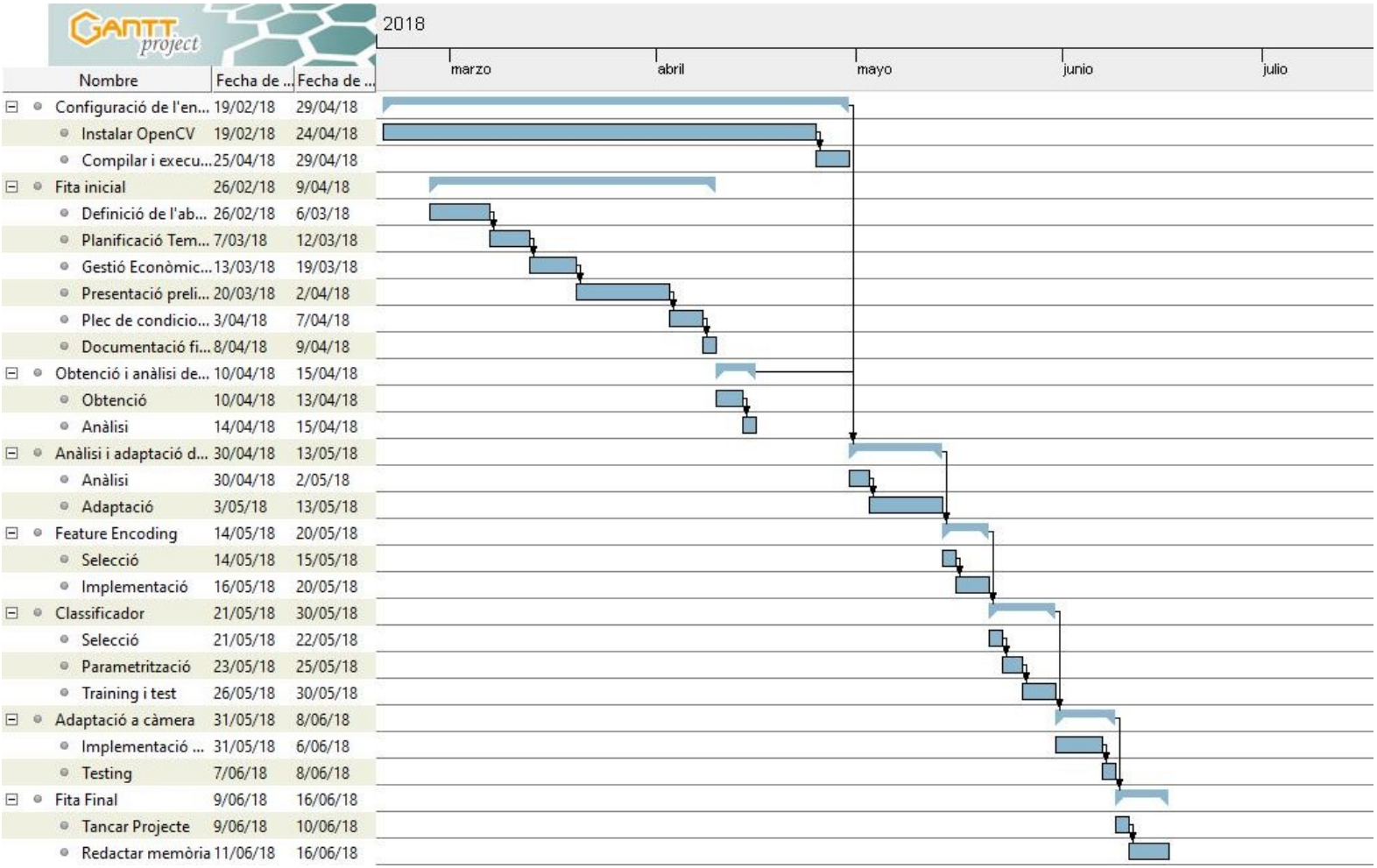


Figura 2: Diagrama de Gantt (Nova planificació)

4 Descripció de la tècnica utilitzada

Aquest projecte es pot separar en tres grans apartats: extracció de característiques, codificació de les característiques i classificació.

4.1 Extracció de característiques

L'extracció de característiques o *feature extraction*, en visió per computador, és l'acció d'obtenir la informació rellevant d'una imatge, vídeo o model complex de dades. En el nostre cas utilitzarem l'algoritme *Improved Dense Trajectories* [1] el qual extreu informació rellevant per descriure moviment en un vídeo.

4.1.1 *Improved Dense Trajectories*

L'algoritme *IDT* bàsicament funciona primer detectant trajectòries de moviment i després obtenint descriptors de cada una.

Per detectar una trajectòria primer es marquen diferents punts en l'espai, després aquests punts s'aparellen a cada *frame* utilitzant *SIFT* [4] de tal manera que es pot veure com s'han desplaçat en el temps i si s'han desplaçat suficient es consideren com a trajectòria. Un cop detectada, s'extreuen un conjunt de descriptors de la zona per tenir més informació sobre el que es mou.

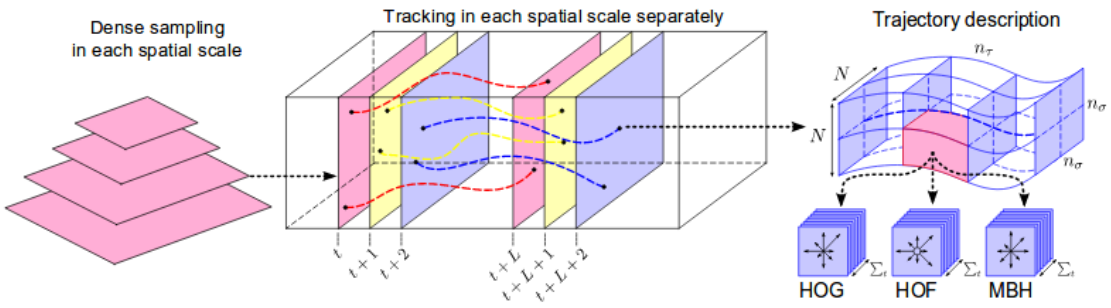


Figura 3: Il · lustració del funcionament de *Dense Trajectories*. Esquerra: Els punts característics són densament mostrejats en multiples escales espacials. Centre: Es fa un seguiment dels punts a la corresponent escala espacial a traves de L *frames*. Dreta: Es creen els descriptors de la trajectòria basats en la forma representada per les coordenades relatives del punt a més de l'aparença i informació del moviment de la zona al seu voltant de N x N pixels.

Els descriptors que s'extreuen són *HOG*[7] (*Histograms of Oriented Gradients*), *HOF*[8] (*Histograms of Optical Flow*) i *MBH*[9] (*Motion Boundary Histograms*).

Improved Dense Trajectories té uns quants paràmetres que poden afectar al rendiment i als resultats. Llargada de les trajectòries, que és el nombre de *frames* en que s'ha de mantenir la trajectòria per ser considerada com a tal. Densitat de mostreig, que marca cada quants píxels pot haver-hi un punt d'interès, per tant amb valors grans hi haurà poca densitat que significarà més eficiència però menys precisió i amb valors petit hi haurà molta densitat que suposarà menys eficiència però més precisió. La distància de veïnat, determina el nombre de píxels d'amplada i alçada (NxN) que s'agafen per calcular els 3 descriptors mencionats anteriorment.

4.1.2 *Histogram of Oriented Gradients*

HOG és un descriptor de la forma d'una imatge el qual descriu els gradients d'aquesta. Aquests gradients es guarden en forma de histogrames que descriuen la direcció i intensitat del gradient per cada zona.

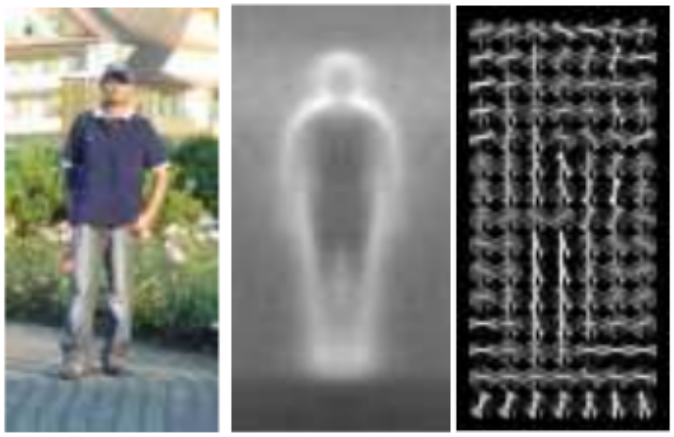


Figura 4: Il · lustració del funcionament de *HOG*. Esquerra: Imatge de test. Centre: Mitjana dels gradients d'imatges d'una persona. Dreta: Resultat HOG de la imatge de test.

4.1.3 *Histogram of Optical Flow*

HOF és l'histograma que descriu l'*Optical Flow*[10] o Flux Òptic el qual és el patro de moviment aparent d'una imatge entre dos *frames* consecutius causats pel moviment d'un objecte o el moviment de la càmera.

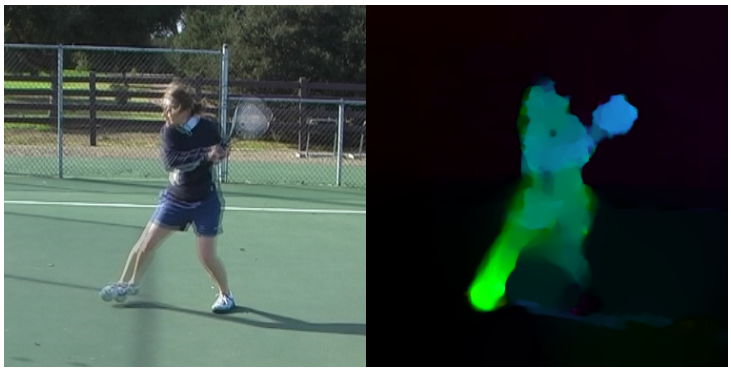


Figura 5: Il · lustració del funcionament d'*Optical Flow*. Esquerra: Imatges d'un moviment superposades. Dreta: Resultat d'*Optical Flow*.

Els mètodes que s'utilitzen per calcular l'*Optical Flow* són diferencials que fan servir aproximacions de la senyal de la imatge basant-se en series locals de *Taylor*, per tant fan servir derivades parcials respecte les coordenades espacials i temporals.

4.1.4 *Motion Boundary Histograms*

MBH també es basa en *Optical Flow*, el qual ja he explicat al apartat anterior, en aquest cas es fan les derivades de les components horitzontals i verticals del *Optical Flow*. Així es codifica el moviment relatiu de píxels i s'elimina gran part del moviment causat per la càmera.

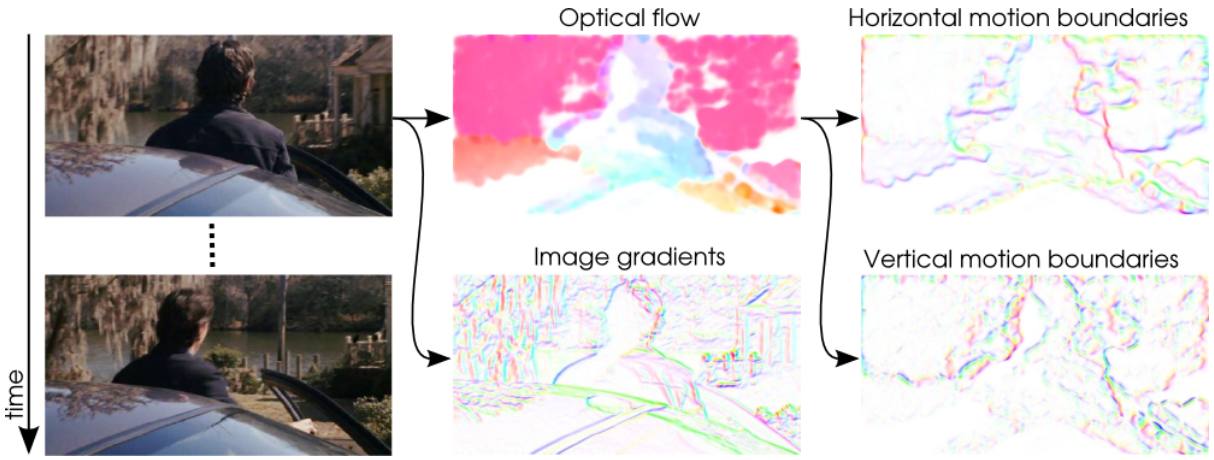


Figura 6: Il · lustració del funcionament de *MBH*. Esquerra: Imatges d'un moviment. Centre: Resultats d'*Optical Flow* i gradients de la imatge. Dreta: Resultats de *MBH* horitzontal i vertical.

4.2 Codificació de les característiques

Degut a que les característiques extretes amb l'algoritme *IDT* no tenen una mida estandarditzada, ja que retorna la informació de totes les trajectòries importants del vídeo i per tant el nombre de trajectòries es variable. Necessitem que el resultat sigui d'una mida estàndard perquè els classificadors només accepten que tots els exemples d'entrada tinguin la mateixa mida. Per aconseguir una mida única per les característiques de tots els vídeos s'han de codificar les característiques resultants de la extracció. Els mètodes més utilitzats per aquest objectiu son *Bag of words* i *Fisher Vectors encoding*[11]. Per aquest projecte utilitzarem *Fisher Vectors*, ja que aprofita millor la informació de les característiques.

Fisher Vectors[11] (*FV*) utilitza *Gaussian Mixture Model* (*GMM*) per modelar la distribució de característiques extretes de la imatge. Llavors *FV* codifica els gradients de la *log-likelihood* de les característiques sota *GMM* respecte els paràmetres de *GMM*. Per tant els paràmetres de *GMM* representen els moments de primer ordre de les característiques (en el nostre cas les trajectòries) i *FV* codifica els moments de segon ordre (en el nostre cas els vídeos). En el nostre cas a més a més fem un *Principal Component Analysis* (*PCA*) de cada descriptor per reduir la dimensionalitat a la meitat abans de crear el *Fisher Vector*.

Principal Component Analysis és una tècnica d'estadística que a través de transformacions ortogonals converteix un conjunt d'observacions de variables possiblement correlacionades a un conjunt de valors de variables linealment no correlacionades anomenades components principals. Aquestes components principals estan ordenades de major a menor variància i amb major variància podríem dir que aporten més informació. Per això normalment aquesta tècnica s'utilitza per eliminar variables irrellevants o que aporten poca informació, per fer-ho simplement s'agafa el nombre desitjat de components (després de la transformació) i aquestes aportaran el màxim d'informació sobre les mostres amb aquest nombre de components. En aquest procés d'eliminació es perd informació, però s'aconsegueix reduir la mida de les dades considerablement sense perdre gaire informació útil.

4.3 Classificació

Un cop extreta la informació útil dels vídeos necessitem entrenar un classificador per tal de que el programa sigui capaç de classificar els diferents possibles gestos. El classificador que hem escollit per aquest projecte és *Suport Vector Machines*[12] (*SVM*), ja que actualment

és dels més potents juntament amb les Xarxes Neuronals, però aquestes segones necessiten un *Dataset* molt gran per funcionar correctament.

Una tasca de classificació normalment implica separar les dades en conjunt d'entrenament i conjunt de prova. Cada instancia del conjunt d'entrenament conte el *Target Value* o valor objectiu (les etiquetes de classe) que en el nostre cas seria el moviment que es realitza en el vídeo i els atributs que serien les característiques extretes anteriorment. L'objectiu de *SVM* és produir un model (basat en les dades d'entrenament) capaç de predir els *Target Values* del conjunt de proba. Per aconseguir-ho, *SVM* mapeja els vectors d'entrenament a un espai dimensional superior utilitzant les funcions *kernel* i finalment troba un hiperpla que separi les classes amb el màxim marge dins d'aquest espai dimensional superior. Depenent de la funció *kernel* que s'utilitzi es trobarà una solució més o menys ajustada al nostre *Dataset*.

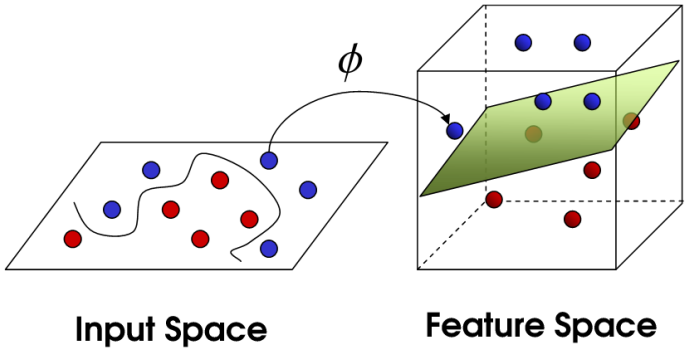


Figura 7: Representació gràfica simplificada de *SVM*. La funció *kernel* (Φ) mapeja a un espai dimensional superior i en aquest espai es busca un hiperpla capaç de separar les classes.

A més a més de la funció *kernel* també hi ha 2 altres paràmetres importants que afecten al resultat del model. El paràmetre "C" (cost d'error en classificació) el qual equilibra la simplicitat de la solució respecte la perfecte classificació dels exemples d'entrenament. Això significa que una "C" gran farà que el classificador intenti classificar correctament tots els exemples i per tant pot induir a *overfitting*, i una "C" petita buscarà una solució més simple que potser classificarà malament alguns exemples de *training*.

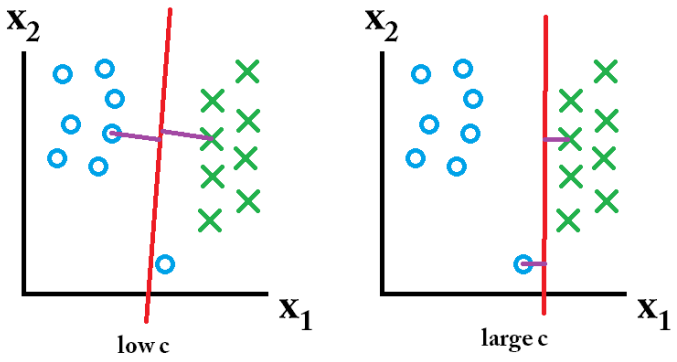


Figura 8: Exemple de *SVM* amb diferents valors per "C"

Un altre paràmetre important és la *Gamma* (γ) que afecta només si s'utilitza un *kernel* no-lineal. Aquest valor fa que a l'hora de mapejar, es mapegin de manera més suau o més brusca, ja que els models no-lineals es basen en Gaussianes i la γ modifica la forma de "campana". Això provoca que si tenim una γ petita resultarà en un biaix petit i una variància alta, mentre que una γ gran resultarà en un gran biaix i poca variància.

SVM va ser originalment dissenyat per separar classes binaries, per això, en el nostre cas hem d'aplicar-lo d'una manera especial per tal de utilitzar-lo per separar classes múltiples (més de 2). La solució que utilitzem en el projecte és *One Vs Rest*, el qual crea un classificador per cada classe que decideix si és o no és d'aquella classe. De cada classificador s'obté el valor de confiança de si pertany a aquella classe i s'escolleix el valor més alt com a resultat de la classificació.

Per tal de que *SVM* funcioni més ràpid, abans de entrenar-lo podem tornar a reduir la dimensionalitat de les característiques (ja codificades amb *FV*) utilitzant *PCA* sense que la precisió de *SVM* canvii gaire.

5 Implementació

Per implementar el projecte hem seguit la planificació així que també podem dividir-ho en tasques.

5.1 *Improved Dense Trajectories*

Per la implementació d'*IDT* hem utilitzat de base el codi font en *C++* dels propis creadors del algoritme disponible a la seva web [13]. A aquest codi font l'hem modificat primer per actualitzar-lo, ja que l'hem adaptat per fer-lo funcionar a la ultima versió d'OpenCV que és la 3.4.1 (el codi original era per la version 2.4). A més a més hem afegit algunes funcionalitats com la opció de escollir si es vol o no veure les trajectòries en imatges i la possibilitat de funcionar amb la càmera com a entrada.

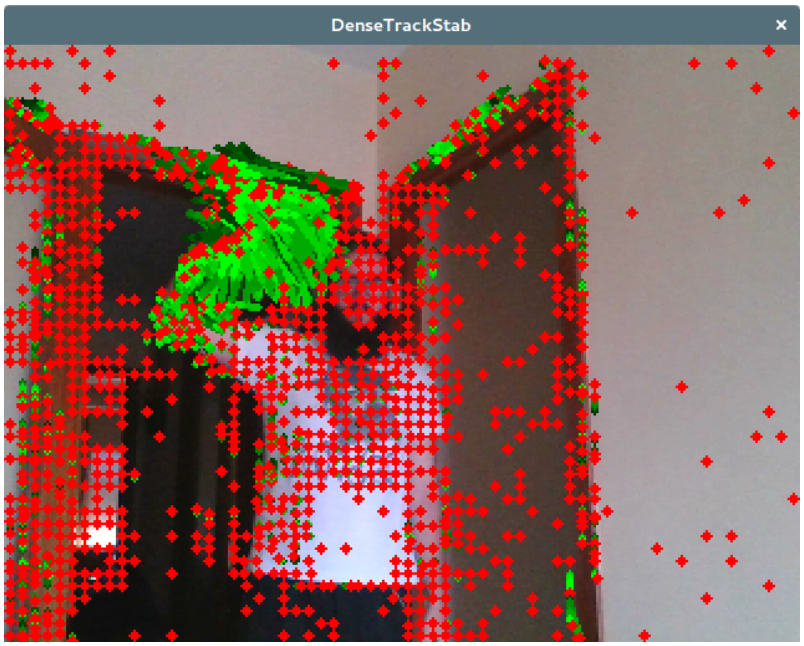


Figura 9: Captura d'*IDT* funcionant amb la càmera i ensenyant les trajectòries. Els punts vermells són els punts d'interès que detecta i les línies verdes les trajectòries.

Amb el executable resultant extraurem les característiques dels vídeos.

5.2 *Feature encoding*

Per codificar les característiques primer necessitem extreure uns valors amb *Gaussian Mixture Model* (*GMM*). Per poder aconseguir aquests valors hem creat un *script* el qual

crida al executable d'*IDT* per extreure les característiques de tots els vídeos que farem servir per entrenar. Aquest *script* a diferencia d'*IDT* és en *Python* (com tot el codi a partir d'aquest punt) perquè és un llenguatge molt més còmode per llegir i escriure arxius, a més a més té llibreries més avançades i actualitzades per *Machine Learning* que ens seran útils. El funcionament és simple, primer llegeix les característiques de cada vídeo (que suposadament estan guardades amb el mateix nom que el vídeo però amb extensió ".features") i les guarda a una matriu, després amb aquesta matriu s'entrena un model *GMM* utilitzant la llibreria *Yael*[14] i finalment es guarda els resultats a un arxiu. Inicialment en aquest *script* primer cridàvem a un altre que ens creava els arxius amb les característiques, però que per accelerar el procés ho he eliminat ja que només fa falta extreure les característiques una vegada i és bastant costos en temps.

Un cop calculats els valors que ens interessen de *GMM*, hem creat un altre *script* que s'encarrega d'utilitzar les *features* i els valors de *GMM* per calcular la codificació de *Fisher Vector*. Per calcular *FV* simplement tornem a fer servir la llibreria de *Yael* la qual té una funció per això i simplement se l'hi ha de passar els valors retornats per *GMM* i la informació que es vol codificar. Els resultats els normalitzem per estandarditzar una mica més les característiques. Finalment guardem les característiques codificades de cada vídeo per separat amb el nom del vídeo i una extensió especifica (".fisher.npz").

5.3 Classificador

Com a classificador, com ja he mencionat anteriorment he escollit *SVM*. En el aquest projecte utilitzarem la llibreria *scikit-learn* (*sklearn*) que és una llibreria de *Machine Learning* molt gran i possiblement la més important de *python*. Primer llegim els arxius amb les característiques codificades dels vídeos que volem utilitzar per entrenar i les guardem a una matriu. Després per generar el classificador creem un model amb els paràmetres que volem i l'entrenem amb la matriu i un vector amb els resultats esperats que han de donar. Amb això ja tenim el model el qual guardem per després poder utilitzar-lo per classificar els vídeos que desitgem. A més a més també hi ha la opció de reduir la dimensionalitat de les característiques codificades utilitzant *PCA* abans d'entrenar el model i així reduir considerablement el temps que tarda. Per trobar els paràmetres més adients per *SVM* utilitzarem un altre *script* el qual descriuré en detall als resultats.

Pels *scripts* per classificar funcionen primer carregant el model, obtenint les *features*,

codificant-les utilitzant alguns dels arxius calculats pel entrenament i utilitzant el model per obtenir una classificació. En el cas del *script* per classificar en temps real amb la informació obtinguda a traves de la càmera, com que no és un vídeo de temps limitat amb només l'acció, cada vegada agafem un nombre limitat de *frames*, els codifiquem i després classifiquem aquell fragment. Aquest ultim *script* té d'entrada un corrent de característiques *IDT* el qual utilitzem executant l'algoritme d'extracció de *features* i passar-les a través d'un *pipe* al *script*.

6 Sostenibilitat

6.1 Autoavaluació del domini actual

Crec que actualment el meu nivell de coneixement relacionat amb la sostenibilitat és bastant baix, perquè realment no conec massa els mecanismes necessaris per analitzar la sostenibilitat d'un projecte. Aquest és un tema que no hem treballat gaire i que personalment no hi he posat especial interès a informar-men en cap moment.

Al apartat econòmic conec les bases per realitzar anàlisis i estimar costos, però no en tinc els coneixements gaire treballats i això provoca que possiblement falli en alguns casos que amb més experiència no fallaria.

Al apartat ambiental és on tinc el nivell més baix, ja que al haver fet una especialitat orientada al software, l'impacte ambiental d'aquests aspectes em sembla mínim. Si que és veritat que a través de eines software és podria millorar la situació ambiental i amb el seu bon ús es podria millorar. Però al no ser aquest un tema gaire del meu interès, no he investigat gaire al respecte i per tant en els meus projectes no es té pràcticament en compte aquest aspecte.

A l'apartat social és on crec que tinc millor nivell, degut a que en la meva opinió un enginyer és dedica a crear solucions a problemes de la societat i per tant a aportar millores socials, com per exemple a través de software que millori la qualitat de vida a algunes persones. Tot i així encara tinc un gran marge de millora en aquest aspecte per la meva falta de coneixements en alguns apartats més concrets d'aquesta categoria, com podrien ser els riscos indirectes que poden tenir alguns projectes.

6.2 Matriu de sostenibilitat

Crec que en aquest projecte podríem dir que tenim una sostenibilitat bastant alta tenint en compte el que es valora (que podem veure-ho a la matriu). Per la part ambiental el nostre projecte no es veu afectat ja que no es basa en software que no té cap efecte positiu ni negatiu sobre el medi ambient. L'apartat econòmic és el que més ens afecta però pràcticament tots els costos són deguts a recursos humans els quals realitzaré jo personalment, per tant el cost és simbòlic (o en temps) i no real (en diners). Finalment respecte L'àmbit social personalment tindrà un efecte molt positiu pels coneixements que

guanyaré i a la societat no tindrà un impacte directe però podria tenir-lo depenent de les aplicacions que se l'hi doni al projecte en un futur.

	PPP	Vida Útil	Riscos
Ambiental	Consum del disseny	Empremta ecològica	Riscos ambientals
Econòmic	Factura	Pla de viabilitat	Riscos econòmics
Social	Impacte personal	Impacte social	Riscos socials

Taula 3: Matriu de sostenibilitat

6.3 Àmbit ambiental

Degut a que el projecte serà realitzat amb el meu propi ordinador personal, l'impacte ambiental serà mínim perquè la part hardware no requerirà cap extra i per tant no tindrà impacte. L'únic consum que podríem tenir en compte es l'energètic que seria la electricitat que consumira l'ordinador que, a part de no ser gaire gran, probablement s'hauria consumit igualment en l'ús quotidià del mateix.

A la vida útil del "producte" com és software no te un impacte directe ambientalment. En quant a impacte indirecte com a molt es podria considerar que altres formes d'interactuar amb dispositius no requereixen de càmera, i la seva fabricació si que te certa empremta ecològica. Però també cal tenir en compte que la majoria de dispositius en els quals pot acabar utilitzant-se aquest projecte ja acostumen a utilitzar càmera així que tampoc suposaria cap despesa ecològica significativa. Globalment aquest projecte no afectarà a l'empremta ecològica actual, ni de manera positiva ni de manera negativa.

6.4 Pressupost inicial

Aquest projecte requerirà de certs recursos per tal de poder ser desenvolupat. En aquest apartat procurarem proporcionar una estimació acurada dels costos del projecte causats pels recursos humans, el hardware, el software i altres elements.

També intentarem lligar aquests costos a les diverses tasques esmentades anteriorment en el diagrama de Gantt per així poder tenir un seguiment en les despeses i tenir millor control per no sortir-se del pressupost.

6.4.1 Recursos Hardware

Per tal de poder fer el projecte tant per la part de implementació com la de documentació, serà necessari l'ús d'un ordinador. Aquest ordinador en considerarem les amortitzacions pertinents.

Producte	Preu	Unitats	Vida Útil	Amortització
Lenovo Yoga 510-14IKB	649 €	1	5 anys	48,68 €
Total				48,68 €

Taula 4: Pressupost *Hardware*

6.4.2 Recursos *Software*

També per realitzar el projecte seran necessàries diverses eines de software, però totes les que utilitzarem són gratuïtes així que no afectaran al pressupost.

Producte	Preu	Unitats	Vida Útil	Amortització
Atom	0 €	1	–	0 €
GanttProject	0 €	1	–	0 €
Git	0 €	1	–	0 €
GitHub	0 €	1	–	0 €
LaTeX	0 €	1	–	0 €
OpenCV	0 €	1	–	0 €
ShareLatex	0 €	1	–	0 €
Total				0 €

Taula 5: Pressupost *Software*

6.4.3 Recursos humans

Aquest projecte serà realitzat només per una persona, tot i això aquesta haurà de tenir diferents rols. Els rols a tenir en compte seran: Cap de projecte [15], desenvolupador de *Software* [16] i *Tester* [17]. El projecte, com ja vam veure en la planificació, tindrà una duració d'un total de 420 h. Aquestes hores estaran distribuïdes entre els diferents rols i a la següent taula podem veure els costos d'aquests (els preus han estat obtinguts d'ofertes reals i es poden veure a les referencies).

Rol	Duració (h)	€/Hora	Preu Total
Cap de projecte	85	50 €/h	4.250 €
Desenvolupador de Software	260	35 €/h	9.100 €
Tester	75	30 €/h	2.250 €
Total	420		15.600 €

Taula 6: Pressupost de recursos humans

A la següent taula (taula 4) podem veure com estan repartits els rols entre les diferents tasques del Gantt.

Tasca	Duració (h)	Dedicació (h)		
		Cap de projecte	Desenvolupador de Software	Tester
Configuració del entorn	20	0	20	0
Fita inicial	90	30	60	0
Obtenció i anàlisi del dataset	25	5	20	0
Anàlisi i adaptació del codi d'IDT	100	15	60	25
Classificador	70	10	35	25
Adaptació a càmera	55	5	30	20
Fita final	60	20	35	5
Total	420	85	260	75

Taula 7: Dedicació aproximada de cada rol

6.4.4 Possibles costos de desviacions

Com ja havíem explicats al apartat de planificació, hi ha possibles apartats on podríem tenir alguns contratemps. Tenint en compte el pitjor cas en el qual passes tot els possibles problemes esmentats anteriorment, els costos que suposarien en increments de temps serien els següents.

Rol	Duració (h)	€/Hora	Preu Total
Cap de projecte	10	50 €/h	500 €
Desenvolupador de Software	30	35 €/h	1.050 €
Tester	5	30 €/h	150 €
Total	45		1.700 €

Taula 8: Pressupost de possibles costos causats per desviacions

6.4.5 Costos indirectes

A part dels costos directes del projecte, també hi han costos indirectes que també afecten al pressupost del projecte. Aquests estan comptabilitzats a la taula 6.

Producte	Preu	Unitats	Cost estimat
Electricitat	0,14 €/kWh	750kWh	105 €
Fibra òptica	37 €/mes	4 mesos	148 €
Total			253 €

Taula 9: Costos indirectes

6.4.6 Pressupost total

Tenint en compte totes les despeses esmentades, obtenim el pressupost total. Com que hi ha riscos que podríem no haver contemplat, fixarem un marge de contingència del 5 %. Tot això està detallat a la taula 7.

A la taula 8 podem veure més detalladament els costos estimats de cada tasca de la planificació.

Motiu	Cost estimat
Hardware	48,68 €
Software	0 €
Recursos Humans	15.600 €
Desviacions	1.700 €
Costos indirectes	253 €
Subtotal	17.601,68 €
Contingències (5%)	880,09 €
Total	18.481,77 €

Taula 10: Pressupost total

Tasca	Cost estimat
Configuració del entorn	955,59 €
Fita inicial	3.643,10 €
Obtenció i anàlisi del dataset	1.205,59 €
Anàlisi i adaptació del codi d'IDT	4.068,10 €
Classificador	2.943,10 €
Adaptació a càmera	2.368,10 €
Fita final	2.418,10 €
Total	17.601,68 €

Taula 11: Cost estimat de les tasques

6.4.7 Control de gestió

Per controlar els costos i no sortir-nos del pressupost, al final de cada tasca revisarem les hores reals invertides i el cost real que ha suposat. Això ho compararem amb les estimacions que hem fet i comprovar si hi han hagut desviacions. En cas d'haver-nos excedit procurarem reajustar els temps i pressupostos de les tasques que quedin per fer.

Els costs derivats de desviacions en principi estan coberts en els possibles costos de desviacions i amb l'extra de la contingència, així que no haurien de suposar un problema que ens fes sobrepassar el pressupost.

6.5 Modificacions en els costos

A causa dels canvis vistos a l'apartat de "Modificacions a la planificació", els costos originals descrits al pressupost inicial s'han vist lleugerament afectats:

Tasca	**Duració (h)**	**Dedicació (h)**		
		Cap de projecte	Desenvolupador de Software	Tester
Configuració del entorn	50	0	50	0
Fita inicial	90	30	60	0
Obtenció i anàlisi del dataset	25	5	20	0
Anàlisi i adaptació del codi d'IDT	80	15	40	25
Feature Encoding	40	5	30	5
Classificador	70	10	35	25
Adaptació a càmera	55	5	30	20
Fita final	60	20	35	5
Total	470	90	300	80

Taula 12: Dedicació aproximada de cada rol

Rol	**Duració (h)**	**€/Hora**	**Preu Total**
Cap de projecte	90	50 €/h	4.500 €
Desenvolupador de Software	300	35 €/h	10.500 €
Tester	80	30 €/h	2.400 €
Total	420		17.400 €

Taula 13: Pressupost de recursos humans

Això significa que tenim un increment del cost de recursos humans d'uns 1.800 € respecte la planificació original ,però si tenim en compte els 1.700 € que havíem deixat per possibles imprevistos només significa 100 € extres al pressupost els quals podrien agafar-se de les contingències.

6.6 Àmbit econòmic

El cost econòmic directe de la realització projecte, ja l'hem vist en detall al apartat del pressupost. Allà podem veure que pràcticament l'únic cost a tenir en compte es el de

recursos humans, ja que és el 98,3 % dels costos totals.

Al ser una eina de software, a la vida útil del producte l'únic cost és el manteniment, en el cas de que es tracti d'actualitzar, millorar o solucionar errors del producte final. Si volguéssim actualitzar de manera periòdica el programa per tal d'afegir millores, els costos serien similars als costos de producció però de manera continuada. En el nostre cas l'única despesa del projecte és la seva implementació i la realitzarem amb els mínims recursos necessaris i sempre utilitzant eines gratuïtes. Comparat amb altres projectes relacionats amb aquest, els costos probablement seran similars amb l'única possible diferencia en les eines utilitzades que poden augmentar una mica els costos finals.

6.7 Àmbit social

A nivell personal, la realització d'aquest projecte m'aportarà una millora considerable en la gestió de projectes, ja que al ser un projecte més gran del que estic acostumat a fer, és necessària una bona gestió. També servirà per aconseguir més experiència i coneixements.

6.7.1 Integració de coneixements

En aquest projecte s'integren principalment 2 grans disciplines: la visió per computador [2] i el *Machine Learning*.
La part de visió per computador és al apartat d'extracció de les característiques o *features*, que en el nostre cas és l'algoritme de *IDT* el qual fa ús de varies tècniques per tal de extreure informació rellevant dels vídeos. Algunes de les tècniques que utilitza són per exemple: *SIFT*[4], *HOG*, detecció d'humans, *SURF*[5]...
Per la part de *Machine Learning* tenim *Fisher Vector* per tal de estandarditzar la entrada al classificador i el propi classificador que en el nostre cas utilitzarem *SVM*. Tot això és necessari per tal de que el programa final primer pugui ser entrenat i després sigui capaç de classificar segons el que ha après en la fase d'entrenament.

6.7.2 Impacte Social

A la actualitat hi ha moltes maneres d'interactuar amb els dispositius tecnològics, com per exemple, escrivint a traves d'un teclat o amb comandes de veu. La detecció de gestos permetria una altre forma d'interactuar amb aquests dispositiu que és útil en algunes situacions a més a més de ser molt natural i intuïtiva. En el nostre cas el projecte no té

un objectiu social directe perquè el resultat no és un producte final dirigit a un public, però segons l'ús final que es vulgui donar al projecte pot aportar moltes coses a la societat com per exemple un robot guiat per gestos en situacions que ho requereixin o fins i tot es podria adaptar a un traductor de llenguatge de signes.

7 Resultats

7.1 *Dataset*

El funcionament del projecte es basa en uns algoritmes orientats al *Machine Learning*, això significa que necessitem un *Dataset* per entrenar i probar el classificador.

He elegit el *Dataset* de *Action as Space-Time Shapes* de Weizmann[18], el qual te un total de 90 vídeos de baixa resolució (180x144) i 25 fps, dividits en 10 classes diferents i per tant 9 exemples de cada classe. Tots els vídeos estan gravats amb una càmera fixe i un fons uniforme. Els diferents moviments a classificar són els següents:

- **Caminar** (*"walk"*): Acció de caminar travessant d'una punta a l'altre del quadre.

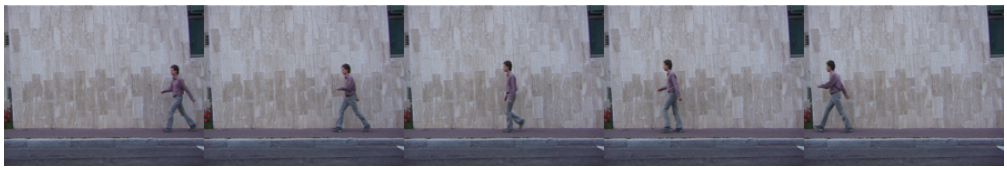


Figura 10: Acció caminar

- **Córrer** (*"run"*): Acció de córrer travessant d'una punta a l'altre del quadre.

Figura 11: Acció córrer

- **Saltar** (*"jump"*): Acció de saltar amb els peus junts travessant d'una punta a l'altre del quadre.

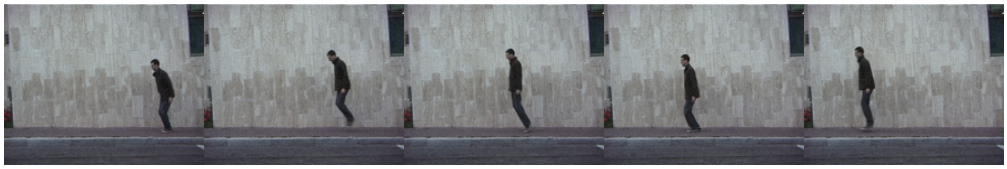


Figura 12: Acció saltar

- **Galop de costat** (*"side"*): Acció de galopar anant de costat (obrint i tancant les cames) travessant d'una punta a l'altre del quadre.

Figura 13: Acció galop lateral

- **Ajupir-se** (*"bend"*): Acció de ajupir-se simulant recollir quelcom del terra.

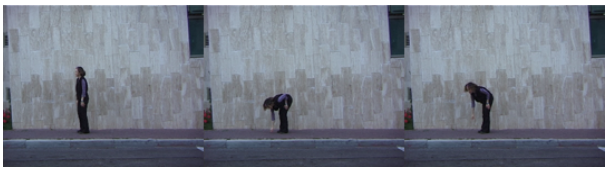


Figura 14: Acció ajupir-se

- **Saludar a una mà** (*"wave1"*): Acció d'aixecar i sacsejar una mà.

Figura 15: Acció saludar a una mà

- **Saludar a dues mans** (*"wave2"*): Acció d'aixecar i sacsejar les dues mans.

Figura 16: Acció saludar a dues mans

- **Salt en una posició** (*"pjump"*): Acció de saltar amb els peus junts sense canviar de posició.

Figura 17: Acció salt en una posició

- **Salt de tisora** (*"jack"*): Acció de saltar obrint i tancant les cames, a la vegada que es pugen i baixen ambdós braços.

Figura 18: Acció salt de tisora

- **Salt a peu coix** (*"skip"*): Acció de saltar a peu coix travessant d'una punta a l'altre del quadre.

Figura 19: Acció salt a peu coix

Per poder fer probes en un entorn més realista, a més del *Dataset* he gravat 30 vídeos dels 10 moviments realitzats per 3 persones diferents.

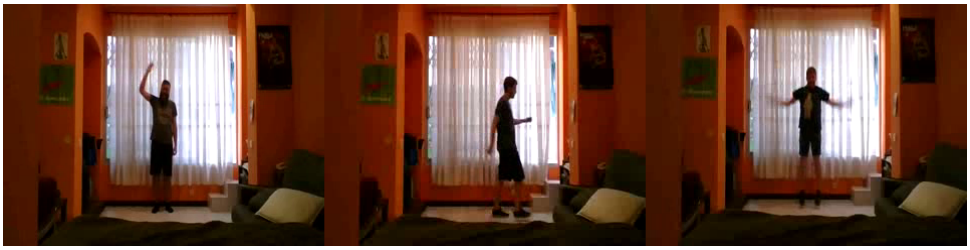


Figura 20: Exemples dels vídeos extres

Com podem veure aquest entorn té molt més soroll que els dels vídeos del *Dataset* i tenen una limitació en l'espai disponible per desplaçar-se horitzontalment degut a dues parets.

7.2 Selecció i parametrització del classificador

Primer cal analitzar els resultats de *SVM* amb diferents paràmetres per trobar el millor model pel nostre objectiu. Per aconseguir-ho hem realitzat un anàlisis exhaustiu provant diferents paràmetres, els paràmetres que hem provat son els següents:

- **C**: per aquest paràmetre hem provat els valors 1, 10, 50, 100 i 1000, com més gran sigui aquest valor hi haurà menys error de *training* però aixo pot causar *Overfitting*.
- **Funció de pèrdua**(només afecta als *SVM* lineals): per aquest paràmetre hem provat les funcions *hinge* (articulació) i *squared-hinge* (articulació quadràtica), aquesta funció és la encarregada de penalitzar els valors mal classificats.
- ***Kernel***(només afecta als *SVM* no-lineals): per aquest paràmetre hem provat els *kernels poly* (*kernel* polinòmic), *RBF* (*Radial Basis Function*) i *sigmoid* (sigmoide), aquesta funció serà la responsable de mapejar les característiques a un espai dimensional superior on es pugui trobar un hiperpla capaç de dividir les classes.
- **Gamma** γ (només afecta als *SVM* no-lineals): per aquest paràmetre hem provat els valors 0.01, 0.001 i 0.0001, aquest valor marca la importància de cada exemple individualment.

Aquests valors els hem testejat utilitzant totes les possibles combinacions. Com a mètode de validació hem utilitzat *Cross Validation* (*CV*) amb 9 particions, això significa que hem partit el *Dataset* en 9 parts i hem utilitzat 8 per entrenar i 1 per testejar, repetint 9 cops aquest procediment utilitzant totes les particions per testejar. Hem fet servir aquest mètode perquè és bastant efectiu per *Datasets* petits i aprofita al màxim els exemples. També hem fet dues versions diferents, en una hem reduït la dimensionalitat de la entrada a 1000 components utilitzant *Principal Component Analysis* i en l'altre no.

Utilitzant *PCA* els millors models que hem obtingut són:

- **Lineal**: Amb C=1 i *Squared Hinge* de funció de pèrdua hem aconseguit un 95.55 % d'encert de mitjana de totes les iteracions de *CV*. Tots els resultats de *Cross Valitadion* (*CV*) són 0.9, 0.9, 1, 0.9, 0.9, 1, 1, 1 i 1, així que com a molt en cada test ha fallat només un exemple.

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	9	0	0	0	0	0	0	0	0	0
	bend	0	9	0	0	0	0	0	0	0	0
	jack	0	0	9	0	0	0	0	0	0	0
	jump	0	0	0	7	0	0	0	2	0	0
	pjump	0	0	0	0	9	0	0	0	0	0
	run	0	0	0	0	0	9	0	0	0	0
	side	0	0	0	0	0	0	9	0	0	0
	skip	0	0	0	1	0	1	0	7	0	0
	wave1	0	0	0	0	0	0	0	0	9	0
	wave2	0	0	0	0	0	0	0	0	0	9

Taula 14: Matriu de confusió del millor model lineal amb PCA

- **No-Lineal**: Amb C=1000, *kernel* sigmoide i γ=0.01 hem obtingut un 93.33 % d'encert de mitjana de totes les iteracions de *CV*. Tots els resultats de *CV* són 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 1, 1 i 1, que també com a molt fallen en un exemple de cada test però falla lleugerament més que la versió lineal.

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	9	0	0	0	0	0	0	0	0	0
	bend	0	9	0	0	0	0	0	0	0	0
	jack	0	0	9	0	0	0	0	0	0	0
	jump	0	0	0	6	0	0	1	2	0	0
	pjump	0	0	0	0	9	0	0	0	0	0
	run	0	0	0	0	0	8	0	1	0	0
	side	0	0	0	0	0	0	9	0	0	0
	skip	0	0	0	1	0	1	0	7	0	0
	wave1	0	0	0	0	0	0	0	0	9	0
	wave2	0	0	0	0	0	0	0	0	0	9

Taula 15: Matriu de confusió del millor model no-lineal amb PCA

Per tant el millor model si reduïm la dimensionalitat del problema amb *PCA* és un

SVM lineal amb C=1 i *Squared Hinge* de parametres.

Utilitzant la totalitat de les característiques (sense reduir la dimensionalitat amb *PCA*) els millors models que hem obtingut són:

- **Lineal**: Amb C=10 i *Hinge* de funció de pèrdua hem aconseguit un 95.55 % d'encert de mitjana de totes les iteracions de *CV*. Tots els resultats de *CV* són 0.8, 1, 0.9, 0.9, 1, 1, 1, 1 i 1, en aquest cas algun test falla 2 exemples però el total d'encerts continua sent el mateix que reduint amb *PCA*.

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	9	0	0	0	0	0	0	0	0	0
	bend	0	9	0	0	0	0	0	0	0	0
	jack	0	0	9	0	0	0	0	0	0	0
	jump	0	0	0	8	0	0	0	1	0	0
	pjump	0	0	0	0	9	0	0	0	0	0
	run	0	0	0	0	0	8	0	1	0	0
	side	0	0	0	0	0	0	9	0	0	0
	skip	0	0	0	1	0	1	0	7	0	0
	wave1	0	0	0	0	0	0	0	0	9	0
	wave2	0	0	0	0	0	0	0	0	0	9

Taula 16: Matriu de confusió del millor model lineal sense PCA

- **No-Lineal**: Amb C=1000, *kernel RBF* i γ=0.01 hem obtingut un 95.55 % d'encert de mitjana de totes les iteracions de *CV*. Tots els resultats de *CV* són 0.8, 1, 0.9, 0.9, 1, 1, 1, 1 i 1, que és exactament el mateix resultat que utilitzant la solució lineal.

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	9	0	0	0	0	0	0	0	0	0
	bend	0	9	0	0	0	0	0	0	0	0
	jack	0	0	9	0	0	0	0	0	0	0
	jump	0	0	0	8	0	0	0	1	0	0
	pjump	0	0	0	0	9	0	0	0	0	0
	run	0	0	0	0	0	8	0	1	0	0
	side	0	0	0	0	0	0	9	0	0	0
	skip	0	0	0	1	0	1	0	7	0	0
	wave1	0	0	0	0	0	0	0	0	9	0
	wave2	0	0	0	0	0	0	0	0	0	9

Taula 17: Matriu de confusió del millor model no-lineal sense PCA

Com podem veure a les matrius de confusió, els moviments més problemàtics solen ser els que impliquen desplaçament horitzontal, especialment saltar, saltar a peu coix i alguna vegada córrer. Aquestes confusions es poden entendre, ja que almenys saltar i saltar a peu coix són molt similars i depenent de com es realitzi el moviment poden ser pràcticament igual.

Amb aquests resultats podem arribar a la conclusió que reduir la dimensionalitat amb *PCA* pràcticament no afecta al resultat final, ja que utilitzant una versió lineal els resultats són els mateixos i en el cas dels no-lineals només varia un 2 % d'encert. Els beneficis que obtenim de *PCA* és que especialment l'entrenament és molt més ràpid, una demostració és els temps que hem necessitat per fer les probes ja que les que redueixen la dimensionalitat tarden aproximadament 1 minut mentre que les que utilitzen la totalitat de les característiques tarden aproximadament 30 minuts.

7.3 Testeig amb diferents paràmetres per *IDT*

Com ja hem mencionat a la descripció de la tècnica, l'algoritme d'*IDT* té uns quants paràmetres que es poden modificar. Per això provarem si modificant aquests paràmetres podem millorar els resultats o el rendiment.

Primer provarem d'augmentar el valor de separació entre trajectòries, per tant reduirem la densitat de punts d'interès per tal de reduir el nombre total de trajectòries detectades. Això pot ser útil ja que podríem aconseguir reduir el temps d'extracció i el temps de codificació i així funcionaria de manera més fluida en temps real. El valor predeterminat que hem utilitzat fins ara ha sigut de 5, com que la precisió que hem obtingut fins ara ja es bastant alta intentarem millorar el rendiment (per això provarem només amb valors més alts). Hem provat amb 6, 7 i 10, i els resultats han sigut pràcticament idèntics, ja que amb *Cross Validation* i utilitzant *SVM* lineal amb la parametrització trobada en l'apartat anterior, hem obtingut un 96.66 % de precisió en els tres casos. Això sembla millorar els resultats anteriors per molt poc i en els tres casos fallen en els mateixos punts tenint els tres una matriu de confusió idèntica.

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	9	0	0	0	0	0	0	0	0	0
	bend	0	9	0	0	0	0	0	0	0	0
	jack	0	0	9	0	0	0	0	0	0	0
	jump	0	0	0	9	0	0	0	0	0	0
	pjump	0	0	0	0	9	0	0	0	0	0
	run	0	0	0	0	0	8	0	1	0	0
	side	0	0	0	0	0	0	9	0	0	0
	skip	0	0	0	1	0	1	0	7	0	0
	wave1	0	0	0	0	0	0	0	0	9	0
	wave2	0	0	0	0	0	0	0	0	0	9

Taula 18: Matriu de confusió (modificant la densitat)

Com que baixa la densitat, té sentit augmentar la finestra que s'agafa de descriptors per així agafar suficient informació tot i tenir menys trajectòries. Hem provat modificant la finestra de 32x32 píxels a 40x40 tenint la densitat d'abans. Els resultats tornen a ser exactament els mateixos que hem obtingut modificant només la densitat, així que per saber si realment millora o empitjora, necessitarem fer probes amb vídeos diferents. Això ho veurem al apartat següent.

7.4 Testeig amb vídeos externs

Per fer un anàlisis en un entorn més similar als que tindrem quan utilitzem la càmera en temps real, farem servir els vídeos gravats per mi mateix (els 30 vídeos extra explicats a l'apartat del *Dataset*). Al ser vídeos gravats en un entorn bastant diferent al *Dataset* original els resultats seran probablement pitjors que els obtinguts anteriorment. Per provar-los utilitzarem els millors models de *SVM* obtinguts anteriorment entrenats amb la totalitat del *Dataset*.

Com quan hem buscat els paràmetres de *SVM* també provarem dos versions: reduint la dimensionalitat de les *features* amb *PCA* i amb les *features* sense tractar.

7.4.1 Proves utilitzant *PCA*

Per aquest test simplement hem extret les característiques dels vídeos, les hem codificat amb *Fisher Vector*, reduït amb *PCA* la dimensionalitat a 1000 (com havíem fet per entrenar el model) i finalment hem demanat al model millor model que hem obtingut anteriorment (*SVM* lineal amb C=1 i *Squared Hinge* de funció de pèrdua) que classifiqui els vídeos. Després comparant els resultats de la classificació amb els resultats reals podem calcular la precisió. Amb aquesta proba hem obtingut una precisió del 60 % que esta molt per sota del que esperàvem, vistes les precisions que havíem obtingut anteriorment.

Si fem un cop d'ull a la matriu de confusió podem veure on falla més:

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	3	0	0	0	0	0	0	0	0	0
	bend	0	3	0	0	0	0	0	0	0	0
	jack	0	0	3	0	0	0	0	0	0	0
	jump	2	0	0	0	1	0	0	0	0	0
	pjump	0	0	0	0	3	0	0	0	0	0
	run	3	0	0	0	0	0	0	0	0	0
	side	2	0	0	0	0	0	1	0	0	0
	skip	3	0	0	0	0	0	0	0	0	0
	wave1	0	0	0	0	0	0	0	0	2	1
	wave2	0	0	0	0	0	0	0	0	0	3

Taula 19: Matriu de confusió (Prova amb *PCA*)

Aquí podem veure que on falla casi sempre és a les accions que impliquen desplaçament que són mal classificades totes com a *"walk"*. Això és probablement degut a que l'entorn on vam gravar els vídeos per fer aquestes probes no era gaire gran, això provoca que en els casos que ens desplaçàvem no ho fèiem en gaire espai i en alguns casos en obligava a fer la acció de manera irregular o més lenta del que hauria de ser. Si només tenim en compte els 5 gestos que no impliquen desplaçament (*bend, jack, pjump, wave1 i wave2*) obtenim una precisió del 93.33 % que és molt millor i a l'altura de les probes amb els vídeos del *Dataset*.

7.4.2 Proves amb característiques sense tractar

Les proves són exactament les mateixes que al apartat anterior però aquesta vegada sense reduir la dimensionalitat amb *PCA* i òbviament utilitzant el millor model entrenat també amb els *Fisher Vectors* directament sense tractar (*SVM* no-lineal amb C=1000, *kernel RBF* i γ=0.01). I hem obtingut una precisió del 63.33 % que és lleugerament millor que les proves anteriors però continua sent insuficient.

Si observem la matriu de confusió veiem que el problema continua sent el mateix:

		Valor predit									
		walk	bend	jack	jump	pjump	run	side	skip	wave1	wave2
Valor real	walk	3	0	0	0	0	0	0	0	0	0
	bend	0	3	0	0	0	0	0	0	0	0
	jack	0	0	3	0	0	0	0	0	0	0
	jump	1	0	0	1	1	0	0	0	0	0
	pjump	0	0	0	0	3	0	0	0	0	0
	run	3	0	0	0	0	0	0	0	0	0
	side	2	0	0	0	0	0	1	0	0	0
	skip	3	0	0	0	0	0	0	0	0	0
	wave1	0	0	0	0	0	0	0	0	2	1
	wave2	0	0	0	0	0	0	0	0	0	3

Taula 20: Matriu de confusió (Prova sense *PCA*)

Els resultats són casi idèntics a les proves amb reducció, nomes hi ha un exemple que canvia de ser mal classificat com a *"walk"* a ser correctament classificat com a *"jump"*. També si com abans tenim en compte només els valors de gestos sense desplaçament obtenim una precisió del 93.33 %.

7.4.3 Utilitzant paràmetres d'*IDT* diferents

Fent servir uns paràmetres diferents per *IDT*, els quals al apartat anterior havíem vist que amb els vídeos del *Dataset* podia ser igual o més efectiu que amb els predeterminats. Al provar en vídeos externs els paràmetres que anteriorment semblaven els més eficients, ens trobem la sorpresa de que fallen estrepitosament. Utilitzant una densitat baixa amb el valor d'espai a 10 i una finestra de descriptor més gran de l'habitual de 40 obtenim una precisió de només 23.33 % que sembla confondre la majoria de vídeos classificant-los com a salt de tisora. Aquí és probable que falli perquè la baixa densitat fa que no detecti tots els punts importants, que al tenir un entorn més sorollós pot ser que agafi punts que no és mouen i passi per alt els que si que ho fan. També pot ser que al fer la finestra dels descriptors més gran, especialment el *HOG* detecti erròniament més soroll.
He repetit l'experiment però sense modificar la finestra i reduint lleugerament la densitat a 6 de separació (predeterminat a 5)i torna a fallar més o menys igual que l'anterior amb només un 30 % de precisió. Així que per les proves en viu ja no tindrem en compte aquests paràmetres i només utilitzarem els predeterminats.

7.4.4 Proves en viu

Per fer les proves en viu simplement hem activat el programa classificador que funciona a través de la pròpia càmera del portàtil i hem realitzat els gestos establerts i comprovat quants encertava. Com que aquesta manera no és tant precisa com amb vídeos amb només una acció agafats amb les condicions més òptimes possibles, és esperable uns resultats menys encertats.

Al realitzar els gestos (en un entorn similar al dels vídeos de prova) hem pogut comprovar que els resultats també fallen en els gestos amb desplaçament. També hem pogut veure que els gestos sense desplaçament són lleugerament menys precisos que en els vídeos, ja que pot no agafar el moviment a la perfecció per algun petit retràs en la càmera o per no seleccionar correctament una part important del moviment.

8 Conclusions

Finalment hem aconseguit que el projecte compleixi els objectius que ens havíem marcat, ja que hem creat un programa que pot reconèixer 10 gestos concrets i que pot funcionar en temps real en una càmera. Aquest programa, com hem vist en els resultats, és bastant precís quan el vídeo esta gravat en un entorn ideal i quan l'entorn no és tant ideal comença a fallar més, especialment en alguns moviments concrets que són més sensibles a entorns limitats. En quant a la classificació en temps real, hem pogut observar que és menys encertat degut a que a l'hora de classificar el vídeo d'entrada té més imperfeccions i no te les accions delimitades a la perfecció, a més a més l'entorn també en general no és el més ideal.

Per millorar el programa i obtenir millors resultats, es podria aconseguir un *Dataset* més gran i variat que probablement solucionaria els problemes que tenim al canviar l'entorn. També amb un *Dataset* més gran es podria afegir una categoria que representes les accions que no es pretenen classificar, així especialment al funcionar en temps real quan no s'estigues realitzant cap acció no detectaria gestos aleatoris. A partir d'aquí aquest programa podria utilitzar-se com a base per fer programes amb diferents utilitats, per exemple aplicar-ho en un robot per poder donar-li ordres de manera gestual. Les possibles aplicacions són moltes i realment depenen només de la imaginació de qui vulgui donar-hi un ús.

Referències

[1] H. Wang & C.Schmid (2013). Action Recognition with Improved Trajectories. Obtingut de <https://www.cv-foundation.org/openaccess/content_iccv_2013/papers/Wang_Action_Recognition_with_2013_ICCV_paper.pdf>

[2] R. Szeliski (setembre, 2010). Computer Vision: Algorithms and Applications. Obtingut de <http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf>

[3] H. Wang, A. Kläser, C. Schmid & L. Cheng-Lin (juny, 2011). Action Recognition by Dense Trajectories. Obtingut de <https://hal.inria.fr/inria-00583818/document>

[4] Yan Ke & R. Sukthankar (novembre, 2003). PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. Obtingut de <http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/user/rahuls/www/pub/irp-tr-03-15-rahuls.pdf>

[5] H. Bay, T. Tuytelaars & Luc Van Gool (2006). SURF: Speeded Up Robust Features. Obtingut de <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>

[6] K. Beck & C. Andres (1999). Extreme Programming Explained (Free sample chapter). Obtingut de <http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>

[7] . Dalal and B. Triggs (2005). Histograms of oriented gradients for human detection. Obtingut de <https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf>

[8] J. Perš, V. Sulić, M. Kristan, M. Perše, K. Polanec & S Kovačič (2010). Histograms of Optical Flow for Efficient Representation of Body Motion. Obtingut de <https://vision.fe.uni-lj.si/docs/janezp/PersPRL-HOFpreprint.pdf>

[9] H. Wang, A. Kläser, C. Schmid & L. Cheng-Lin (gener, 2013). Dense trajectories and motion boundary descriptors for action recognition. Obtingut de <https://hal.inria.fr/hal-00725627v2/document>

[10] OpenCV: Optical Flow. Obtingut de <https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html>

[11] J. Sánchez, F. Perronnin, T. Mensink & J. Verbeek (maig, 2013). Image Classification with the Fisher Vector: Theory and Practice. Obtingut de <https://hal.inria.fr/hal-00779493/document>

[12] Chih-Wei Hsu, Chih-Chung Chang & Chih-Jen Lin (maig, 2016). A Practical Guide to Support Vector Classification. Obtingut de <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

[13] THOTH - Improved Trajectories Video Description. Obtingut de <https://lear.inrialpes.fr/people/wang/improved_trajectories>

[14] Yael's documentation. Obingut de <http://yael.gforge.inria.fr/>

[15] Michael Page: Oferta Project Manager. Obtingut de <https://www.michaelpage.es/job-detail/project-manager/ref/267847?source=search>

[16] Michael Page: Oferta Programador (Desenvolupador Software). Obtingut de <https://www.michaelpage.es/job-detail/programador-plc-senior/ref/265595?source=search>

[17] Page Personnel: Software Tester. Obtingut de <https://www.pagepersonnel.es/job-detail/software-tester/ref/266113?source=search>

[18] L. Gorelick, M. Blank, E. Shechtman, M. Irani & R. Basri (2005). Actions as Space-Time Shapes. Obtingut de <http://www.wisdom.weizmann.ac.il/\%7Evision/SpaceTimeActions.html>