

# Reconeixement d'emocions de persones en una imatge mitjançant Visió per Computadors

Treball de Fi de Grau

*Autor:* Ávila Parra, Rafael

*Data de defensa:* 28/01/2019

*Curs:* 2018-2019/Q1

*Especialitat:* Computació

*Titulació:* Grau en Enginyeria Informàtica

*Director:* Climent Vilaró, Joan

*Departament:* Eng.Sistemes, Automàtica i Inf.Ind.

*Centre:* Facultat d'Informàtica de Barcelona (FIB)

*Universitat:* Universitat Politècnica de Catalunya (UPC) – BarcelonaTech

---

# Índex

<b>1. Resum/Resumen/Abstract</b>	<b>3</b>
<b>2. Introducció</b>	<b>5</b>
2.1. Contextualització	5
2.2. Formalització del problema	5
2.3. Actors implicats	6
2.4. Objectius generals	6
2.5. Estat de l'art	8
2.6. Abast	11
<b>3. Planificació</b>	<b>12</b>
3.1. Programa general	12
3.2. Mètodes de desenvolupament	12
3.3. Tasques a realitzar	13
3.4. Recursos	15
3.5. Entorn de treball	16
3.6. Diagrama de GANTT	17
3.7. Conclusió	19
<b>4. Detecció facial</b>	<b>20</b>
4.1. Fonaments teòrics	20
4.2. Limitacions i restriccions	22
4.3. Detecció en imatges	22
4.4. Detecció en vídeos	23
4.5. Conclusions	24
<b>5. Reconeixement d'emocions</b>	<b>25</b>
5.1. Fonaments teòrics	25
5.2. Format dels resultats	27
5.3. Tractament dels punts clau	30
5.3.1. Punts	30
5.3.2. Vectors	32
5.3.3. Angles	33
5.4. Histogrames de gradients orientats	36
5.4.1. Fonaments teòrics	36
5.4.2. Aplicació general	37
5.4.3. Punts clau	38
5.5. Combinació de característiques	39
5.6. Filtres	40

---

5.7. Mètodes alternatius	42
5.7.1. k-nearest-neighbours	42
5.7.2. Anàlisi discriminant	43
5.7.3. Paral·lelització del TreeBagger	45
5.8. Conclusions	46
<b>6. Resultats</b>	<b>47</b>
6.1. Base de dades	47
6.2. Resum dels resultats	49
6.3. Comparació amb alternatives al mercat	50
<b>7. Sostenibilitat i compromís social</b>	<b>53</b>
7.1. Autoavaluació sobre sostenibilitat	53
7.2. Dimensió econòmica	54
7.2.1. Gestió Econòmica	54
7.2.2. Revisió de la planificació i la gestió econòmica	57
7.2.3. Reflexió sobre la sostenibilitat	61
7.3. Dimensió ambiental	63
7.4. Dimensió social	65
<b>8. Conclusions</b>	<b>67</b>
8.1. Treball futur	68
<b>9. Annex</b>	<b>70</b>
9.1. Documentació del software	70
<b>10. Bibliografia</b>	<b>73</b>
10.1. Referències	73
10.2. Figures	75

# 1. Resum/Resumen/Abstract

## Resum

La capacitat de poder reproduir parts del coneixement humà en una màquina ha estat una qüestió molt tractada per enginyers informàtics durant els darrers anys. El fet que un programa sigui capaç de donar una resposta similar a la d'un humà de forma automàtica té un gran ventall de possibilitats en la societat moderna. Aquest treball es dedicarà a explorar aquesta disciplina, més concretament, el reconeixement de les emocions humanes. Les persones utilitzem diferents tècniques de forma subconscient que hem après de forma gradual per fer aquest reconeixement. L'objectiu principal del projecte serà simular aquest procés d'aprenentatge i ser capaç de posar-lo a prova amb imatges de persones mostrant diferents emocions.

La primera part del treball es centrarà en la detecció de cares de persones en una imatge. Aquest pas previ és essencial ja que localitzar a la persona en la imatge és un pas lògicament necessari abans de poder realitzar un anàlisi més profund. Mitjançant una tècnica d'ajust de models deformables es procedirà a fer la detecció en imatges, i posteriorment, en vídeos. Una taxa d'encert superior al 90% en aquesta secció va suposar que el projecte pogues continuar amb la segona part.

Aquesta segona part ja té com a objectiu realitzar un software capaç de reconèixer expressions facials i, per tant, les emocions. Tota aquesta part del treball s'ha realitzat a partir d'una constant experimentació amb una important diversitat de tècniques sobre diferents factors que tenen afectació en els resultats. Aquests són els tipus de classificadors utilitzats en el procés d'aprenentatge o el càlcul de característiques que diferencien una cara d'una altra quan expressen una determinada emoció. Els resultats obtinguts en aquesta segona part superen unes taxes d'encert superiors al 90%.

El resultat del treball és un software capaç de distingir entre 8 emocions diferents tant en imatges com en vídeos.

## Resumen

La capacidad de poder reproducir partes del conocimiento humano en una máquina ha sido una cuestión muy tratada por ingenieros informáticos durante los últimos años. El hecho de que un programa sea capaz de dar una respuesta similar a la de un humano de forma automática tiene un gran abanico de posibilidades en la sociedad moderna. Este trabajo se dedicará a explorar esta disciplina, más concretamente, el reconocimiento de las emociones humanas. Las personas utilizamos diferentes técnicas de forma subconsciente que hemos aprendido de forma gradual para hacer este reconocimiento. El objetivo principal del proyecto será simular este proceso de aprendizaje y ser capaz de ponerlo a prueba con imágenes de personas mostrando diferentes emociones.

La primera parte del trabajo se centrará en la detección de caras de personas en una imagen. Este paso previo es esencial ya que localizar a la persona en la imagen es un paso lógicamente necesario antes de poder realizar un análisis más profundo. Mediante una técnica de ajuste de modelos deformables se procederá a hacer la detección en imágenes, y posteriormente, en videos. Una tasa de acierto superior al 90% en esta sección supuso que el proyecto pudiera continuar con la segunda parte.

Esta segunda parte ya tiene como objetivo realizar un software capaz de reconocer expresiones faciales y, por tanto, las emociones. Toda esta parte del trabajo se ha realizado a partir de una constante experimentación con una importante diversidad de técnicas sobre diferentes factores que tienen afectación en los resultados. Estos son los tipos de clasificadores utilizados en el proceso de aprendizaje o el cálculo de características que diferencian una cara de otra cuando expresan una determinada emoción. Los resultados obtenidos en esta segunda parte superan unas tasas de acierto superiores al 90%.

El resultado del trabajo es un software capaz de distinguir entre 8 emociones diferentes tanto en imágenes como en videos.

## Abstract

The ability to reproduce parts of human knowledge on a machine has been an issue that has been addressed by computer engineers in recent years. The fact that a program is capable of giving a response similar to that of a human automatically has a wide range of possibilities in modern society. This work will be devoted to exploring this discipline, more specifically, the recognition of human emotions. People use different techniques subconsciously that we have learned gradually to do this recognition. The main objective of the project will be to simulate this learning process and be able to the test it with images of people showing different emotions.

The first part of the work will focus on the detection of people's faces in an image. This previous step is essential since locating the person in the image is a logically necessary step before being able to perform a deeper analysis. By means of a technique of deformable model fitting, it will proceed to make the detection in images, and later, in videos. A success rate higher than 90% in this section meant that the project could continue with the second part.

This second part is already aimed at making software capable of recognizing facial expressions and, therefore, emotions. All this part of the work has been made from a constant experimentation with an important diversity of techniques on different factors that affect the results. These are the types of classifiers used in the learning process or the calculation of characteristics that differentiate one face from another when they express a certain emotion. The results obtained in this second part are success rates of over 90%.

The result of the work is a software capable of distinguishing between 8 different emotions in both images and videos.

## 2. Introducció

### 2.1. Contextualització

Durant la nostra vida, ens dediquem a obtenir coneixement de moltes formes diferents. Moltes no s'aprenen a l'escola ni en altres centres educatius. Una d'elles és la capacitat de poder distingir com es troba psicològicament una persona pel simple fet d'observar-la. Com podem saber si està contenta o si està enfadada? També pot estar trista a la vegada que mostra senyals de simpatia amb la persona que està parlant. No és una ciència exacta, però les persones som capaces de donar un resultat extremadament encertat en un espai molt reduït de temps.

Llavors, si els humans som capaços de dur a terme aquest procés de forma tan efectiva, perquè no ho codifiquem de forma senzilla per automatitzar-ho? La resposta és simple: no existeix una forma fàcil de fer-ho. La clau per entendre la superioritat d'una persona davant d'una màquina a l'hora d'entendre les emocions resideix en la experiència. Una persona de 50 anys ha pogut veure milers de cares diferents expressant centenars d'emocions diferents. Pot ser la primera vegada que les va veure, no va poder entendre quina emoció amagava cadascuna, però la desena vegada potser sí. És aquí on es pot veure la problemàtica aparent a l'hora de codificar tot això. Com podem incorporar una vida de coneixements tan precisos en un programa per poder decidir amb una efectivitat similar a la d'una persona?

### 2.2. Formalització del problema

Ara que ja hem pogut entendre la situació que vol afrontar aquest treball, podem passar a formalitzar quin serà el problema a resoldre. El projecte estarà dedicat a crear un software capaç de reconèixer les emocions d'una persona en una imatge.

Durant la realització del treball ens dedicarem a analitzar com crear aquest software de la forma més efectiva possible mitjançant un gran ventall de tècniques de la disciplina de la Visió per Computadors. S'observarà quins efectes tindran en els resultats utilitzar diferents algoritmes o variacions dels mateixos. Més endavant, podrem veure com petites modificacions poden tenir un enorme impacte en els resultats.

Per dur a terme tot això, ens caldrà un software que sigui capaç de detectar la cara d'una persona en una imatge. A partir d'aquí hauria de poder trobar on són les parts més importants de la cara, com poden ser els ulls, la boca, el nas, entre d'altres. A continuació, caldrà codificar el procés d'aprenentatge, el qual s'encarregarà de simular la situació explicada anteriorment sobre la persona de 50 anys i la seva experiència. Finalment, aquest software haurà de decidir i donar un resultat per la imatge que s'ha utilitzat com a entrada.

Per descomptat, caldrà un dataset amb imatges que continguin cares de persones, per poder afrontar aquest problema. Aquestes serviran tant pel procés d'aprenentatge com per posar a prova el nostre software.

## 2.3. Actors implicats

Aquests seran tots els actors que s'implicaran en el desenvolupament o en altres parts del projecte

- **Desenvolupador:** La tasca del desenvolupament del projecte serà realitzada per mi ja que serè el participant més actiu d'aquest projecte. També seré l'encarregat de tasques de testejar i d'investigació de possibles optimitzacions.
- **Director del projecte:** El director del projecte és el professor de la UPC, Joan Climent Vilaró. Serà l'encarregat d'anar supervisant el projecte i la seva evolució. També podrà aportar idees i suggerències per tal de millorar el desenvolupament del treball.
- **Institucions:** Els principals clients que pot tenir aquest tipus de software són institucions que necessitin conèixer millor als seus clients o que simplement vulguin detectar les seves reaccions emocionals. Per tant, aquests serien els hipotètics clients en potència principals del software desenvolupat en el treball.
- **Altres usuaris:** També és possible que altres tipus d'usuaris, com ara persones normals, vulguin tenir accés al software. Per tant, el seu ús privat també s'ha de tenir en consideració.

## 2.4. Objectius generals

Els objectius generals que tindrem en aquest treball estan establerts de forma seqüencial. És una manera de tenir punts intermitjos que s'hauran d'anar complint per tal de poder progressar a passos més complexos.

### Primer objectiu

Ser capaç de detectar la cara d'una persona en una imatge i una sèrie de punts rellevants. Encara que sembli senzill, aquesta serà la base de tot el treball, per tant, cal tenir un software fiable que pugui dur a terme aquesta tasca. Com veurem més endavant, aquesta és una part de la branca de Visió per Computadors molt desenvolupada en l'actualitat.

### Segon objectiu

Aconseguir desenvolupar un software que doni un resultat al problema amb un mínim de coherència. Això significa haver implementat un software capaç de reconèixer les expressions de les cares per poder obtenir una emoció com a resultat.

Aquest segon pas, ja significarà que tenim un software funcional que resol el problema proposat. Però aquest no és l'objectiu final d'aquest projecte.

### Tercer objectiu

Investigar i estudiar possibles maneres d'augmentar el percentatge d'encert del nostre programa.

Aquest darrer objectiu és sens dubte el més complex de tots. Com podem veure més endavant, pel tipus d'algoritmes i lògica que s'utilitzen, els resultat mai tindran una precisió del 100% de forma consistent. Per tant, el darrer i principal objectiu del treball serà arribar a la màxima efectivitat possible utilitzant totes les tècniques que puguem trobar o inventar.

## 2.5. Estat de l'art

### Detecció facial en una imatge

El primer que cal dir sobre la detecció facial és definir en què consisteix realment. Aquesta tecnologia permet localitzar amb un rectangle la posició d'una cara en una imatge. És ben diferent al que es coneix com a reconeixement facial. En aquest cas, vol dir que el dispositiu que contingui aquesta tecnologia, pot identificar a una persona a partir d'una fotografia o un frame d'un vídeo. Per tant, no s'ha de confondre amb, per exemple, la propietat coneguda com a "Face ID" implementada en els iPhone X al 2017. [1][2]

La detecció facial és un algoritme que ja existeix fa molts anys. Però no ha sigut fins a l'aparició dels *smartphones* que s'ha convertit en una eina a disposició de tothom. El seu funcionament es basa en, com és habitual en Visió per Computadors, modificar la imatge i buscar elements clau. El més habitual és convertir la imatge a escala de grisos (eliminar el color) i buscar les formes i ombres que podem considerar com a cares. El més habitual és buscar una "T" formada pel front i el nas, ja que és la part més semblant entre qualsevol persona. [3]

Aquest és un exemple de detecció facial amb un software desenvolupat per Apple:

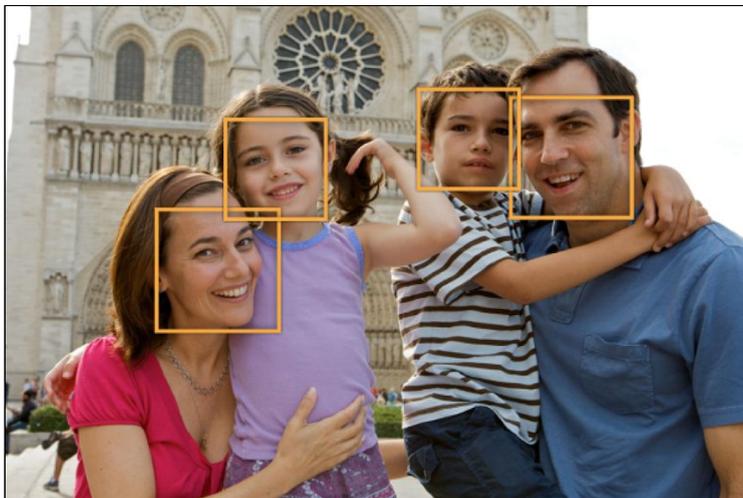


Figura 1: Imatge amb rectangles que mostren les cares detectades

### Lògica difusa

La lògica difusa és un element clau en la detecció d'emocions. La gran diferència d'aquest tipus de lògica és que no assigna un resultat unitari a un enunciat. Per exemple, si utilitzem lògica proposicional, obtindrem dos possibles respostes: cert o fals. En canvi, la lògica difusa, dirà quin serà "el resultat més possible".

Concretament, quan volem resoldre un problema amb aquest tipus de lògica, necessitem un conjunt de possibles resultats possibles. A partir d'aquí, a cadascun d'aquests resultats s'assigna un *truth value*, és a dir, en quina mesura és cert un possible resultat (aquest valor oscil·la entre 0 i 1). [4]

A continuació tenim un exemple de com decidir si l'aigua està freda, temperada o calenta. Com es pot veure, hi ha franges de temperatura on no hi ha una veritat absoluta i on més d'una resposta té opció de ser possible.

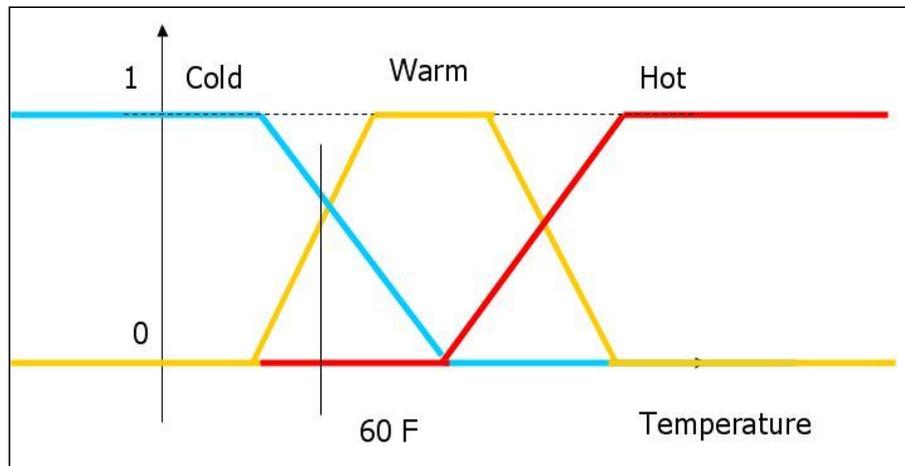


Figura 2: Exemple de *fuzzy logic*

## Detecció d'emocions

Durant els últims anys, hi ha hagut una sèrie d'antecedents respecte al tema com ara treballs d'investigació, informes universitari, entre d'altres. S'ha utilitzat per analitzar situacions ben diferenciades. Aquests són uns quants exemples:

- “Emotion Detection in the Loop from Brain Signals and Facial Images”: Informe universitari que tracta la detecció d'emocions amb una sèrie de mètodes. Entre ells, utilitza imatges facials. [5]
- “Spontaneous Emotional Facial Expression Detection”: Informe que tracta principalment les emocions espontànies de les persones a partir de les expressions facials. [6]
- “Emotion Detection From Infant Facial Expressions And Cries”: Presenta un sistema que tradueix el significat dels plors d'un nen petit a partir d'imatges facials i altres elements (En aquesta pàgina es poden trobar diversos documents similars). [7]

El seu funcionament el podem descriure a partir del que s'ha vist fins ara, la detecció facial en una imatge i la lògica difusa. Primer, cal detectar la cara a analitzar en la imatge. A partir d'aquí es troben punts rellevants de la cara com són els ulls, el nas, la boca, les celles, entre altres. Llavors, s'obtenen una sèrie de valors a partir de la seva posició relativa entre ells.

---

Amb aquests valors, es pot fer una comparació amb resultats d'imatges ja conegudes (procés d'aprenentatge, mencionat anteriorment) i s'obté un resultat mitjançant lògica difusa.

Actualment existeixen diferents aplicacions que simulen aquest procés. Un exemple prou rellevant és l'aplicació desenvolupada per Microsoft anomenada "Emotion Recognition", que permet fer proves online i rebre resultats de forma immediata. [8]

Un altre exemple és l'aplicació "Emotient", adquirida fa uns anys per Apple. Amb ella també podem veure l'efecte d'utilitzar lògica difusa, com es pot apreciar en la següent imatge. Cada emoció té assignat un *truth value* en cada moment del vídeo que s'hagi utilitzat:

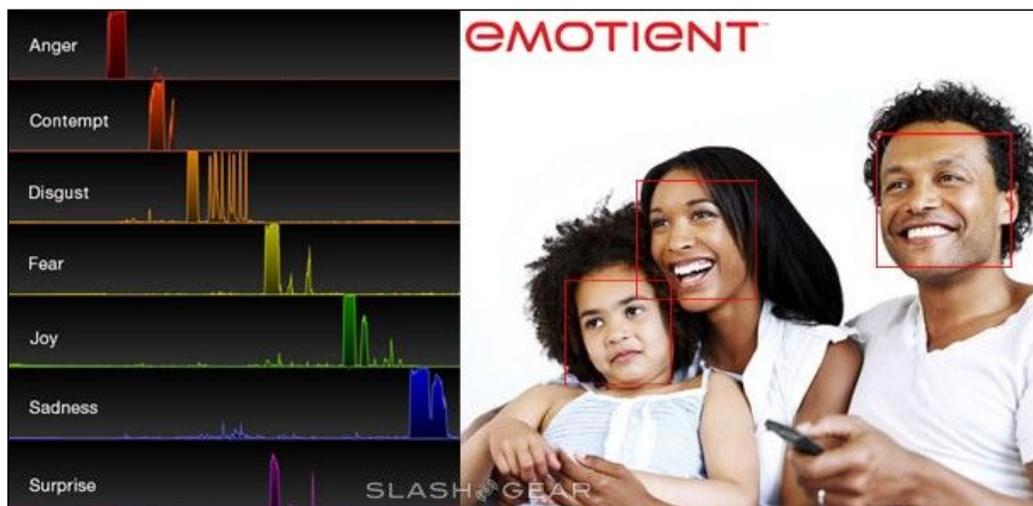


Figura 3: Resultats d'aplicar lògica difusa en el reconeixement d'emocions (App: Emotient)

## 2.6. Abast

### Abast del projecte

La meta final del projecte seria poder aconseguir optimitzar la detecció d'emocions fins arribar a obtenir un ratio d'efectivitat similar, o si pogués ser, superior al d'algunes aplicacions que podem trobar en el mercat.

També es valora la possibilitat d'afegir funcionalitats extras al software com poden ser:

- Mesura del grau en què trobem una emoció en particular: Si el client està interessat en detectar una certa reacció per part d'una persona davant d'un esdeveniment concret, només li interessarà centrar-se en una única emoció.
- Afegir emocions secundàries o una mescla de les ja conegudes: Aquest pot ser un element realment distintiu que permetria tenir en compte altres situacions que softwares existents no tracten.

### Principals obstacles

Aquests podran ser els principals obstacles que hi haurà durant el desenvolupament del projecte:

- La detecció i el reconeixement poden ser imprecisos i/o complicats: Buscar patrons determinats en imatges en Visió per Computadors no és una ciència exacte. A l'hora de detectar cares en imatges, no podem esperar que totes siguin iguals. Per tant, obtenir bons resultats pot ser difícil. El mateix fet s'aplica al pas posterior del reconeixement d'emocions.
- Dificultat en detectar errors o buscar optimitzacions: Moltes vegades, la única manera de detectar errors en el codi serà en un descens de l'efectivitat del programa. Per tant, descobrir la font de l'error significaria trobar quina característica o grup d'elles estan afectant negativament tot el procés de reconeixement, pot ser complicat.
- Possibilitat de trobar-se amb llargs temps d'execució: Amb anteriors experiències d'utilització del llenguatge Matlab, necessari pel projecte, s'ha de destacar el considerable temps d'execució que hi pot haver a l'utilitzar grans bases de dades.
- Temps limitat de desenvolupament: El temps de dedicació al projecte no és superior als 4 mesos, per tant, caldrà ser eficient en la cerca de mètodes i en el seu anàlisi per tal d'augmentar la efectivitat sobre l'objectiu principal.

## 3. Planificació

### 3.1. Programa general

El desenvolupament d'aquest projecte tindrà lloc en un espai de temps aproximat de 4 mesos. El seu inici va tenir lloc el 17 de setembre del 2018 amb l'inici del treball de la fita inicial, i finalitzarà en una data estimada a finals del mes de gener del 2019 amb la fita final i la lectura de la memòria.

Com a conseqüència de la metodologia escollida pel desenvolupament del projecte, el model en cascada, les dates de termini intermitges s'hauran de complir de forma estricta per no acumular retards innecessaris. Tot això estarà ben controlat gràcies al mètode de validació mencionat anteriorment.

### 3.2. Mètodes de desenvolupament

#### Mètodes de treball

Com ja s'ha especificat en diversos apartats, els objectius d'aquest projecte estan clars, però és difícil veure en quin punt s'hauran complert, especialment l'últim. Com que l'objectiu és arribar a la màxima efectivitat, es podria dir que mai serà suficient, per això cal dissenyar una estratègia d'acord amb la situació.

L'estratègia de treball a seguir serà d'emprar un temps constant al desenvolupament. Dedicant un nombre similar d'hores a la setmana obliga a seguir investigant i treballant per assolir millors resultats, però sense la pressió que genera la necessitat d'arribar a un objectiu molt marcat. Això sí, els dos primers objectius s'haurien de complir en un temps raonable per deixar temps a la part més important del treball. S'introduiran més detalls en l'apartat de la planificació del projecte.

Una altra estratègia rellevant en aquest projecte serà anar avançant en funció dels resultats obtinguts. Tenint en compte que l'objectiu és maximitzar l'eficàcia, la direcció de les investigacions i el desenvolupament del codi anirà d'acord als resultats obtinguts amb el treball fet fins al moment.

Per tant, pels motius explicats fins ara, i la meta de complir els tres objectius principals, el mètode escollit serà el model en cascada. Com que tots 3 objectius, i les subtasques corresponents que veurem a la planificació, depenen els uns dels altres, aquest és el mètode de treball més indicat pel projecte.

## Eines de seguiment

El seguiment del projecte es farà deixant constància de diferents versions del software de forma periòdica, juntament amb reunions amb el director per posar en comú tot el que s'ha fet fins ara i quins seran els pròxims passos.

## Mètode de validació

Per validar que efectivament s'estan complint amb els mètodes de treball i els objectius, es dedicaran uns minuts de forma semanal a revisar el treball fet durant els darrers 7 dies. D'aquesta manera, es podrà determinar si s'està seguint el pla de forma satisfactòria. També serà un moment per validar la feina feta durant les diverses reunions amb el director del projecte.

## Eines de treball

Per aquest treball s'utilitzarà:

- Ordinador portatil ASUS com a eina pel desenvolupament de codi
- Software amb llicència Matlab versió R2017b
- Google Drive per l'emmagatzegament de versions

## 3.3. Tasques a realitzar

### Fita Inicial

La primera tasca, que ja és l'objectiu actual de desenvolupament, és la fita inicial del projecte. Consisteix en la realització de 5 lliuraments que serviran com a punt de partida del projecte i l'inici de la redacció de la memòria. Aquests són:

- Definició de l'abast i contextualització, a lliurar el 25/09/2018 (25h)
- Planificació temporal, a lliurar el 01/10/2018 (10h)
- Gestió econòmica i sostenibilitat, a lliurar el 08/10/2018 (20h)
- Plec de condicions, a lliurar el 15/10/2018 (5h)
- Document i presentació final, a lliurar també el 15/10/2018 (20h)

El temps de dedicació per aquesta tasca serà aproximadament de 80 hores a repartir en les 4 setmanes de la seva duració. Tenint en compte que cada lliurament té una càrrega de treball diferent, la dedicació també ho serà. Al costat de cada sub-tasca queda constància de les hores de dedicació aproximada.

## Model funcional bàsic

Un cop acabada la fase inicial i tota la planificació necessària, ja es pot començar a treballar en el software. Complir amb aquesta part serà necessari per assolir els dos primers objectius descrits anteriorment. Al igual que l'anterior, aquesta tasca es pot dividir en diverses sub-tasques més petites. Seran aquestes:

- Investigació i disseny de la detecció facial (5h)
- Programar i/o adaptar la detecció facial (35h)
- Fase de testing de la detecció facial (10h)
- Investigació i disseny de la detecció d'emocions (5h)
- Programar i/o adaptar la detecció d'emocions (35h)
- Fase de testing de la detecció d'emocions (10h)

Caldrà tenir en consideració que les 3 darreres sub-tasques no poden començar fins a la finalització de les 3 primeres. També assenyalar que les tasques d'investigació seran prèvies a la programació i el testing. Finalment, també dir que les fases de programar i testing es solapen, ja que per la naturalesa del projecte, s'han de provar totes les implementacions que es fan per detectar millores o errors de forma eficaç.

La duració de les tasques d'investigació i de disseny es limitaran a 2 dies, mentre que les de programar i testing s'allargaran fins als 12 dies. En total, la càrrega de treball serà d'unes 100 hores a dividir en 4 setmanes.

## Investigació i optimitzacions

A continuació, un cop finalitzats els 2 primers objectius del projecte, cal centrar-se en el tercer i últim, que també és el més important. La metodologia de treball serà similar a la de la tasca anterior, però no idèntica. Per això, cal adaptar la planificació de les tasques de forma adequada. En aquest cas, les sub-tasques seran:

- Investigació i disseny de nous mètodes d'optimització (30h)
- Programar i/o adaptar els nous mètodes d'optimització (180h)
- Fase de testing dels nous mètodes d'optimització (30h)

A diferència de la planificació anterior, totes aquestes subtasques tindran lloc de forma simultània. Ja que el procediment a seguir per aquesta part és cíclic (investigar, dissenyar, implementar, testejar, repetir) i no té un punt final clarament marcat, les enmarcarem en el mateix interval de temps.

La duració en aquest cas s'estendrà fins a dues setmanes abans de la finalització del projecte, per tant estem parlant de 8 setmanes, amb una càrrega aproximada de treball de 240 hores distribuïdes entre les 3 sub-tasques.

---

## Fita Final

Un cop implementat i testejat el software, cal centrar-se en la darrera part del projecte. Aquesta consisteix bàsicament en la redacció completa de la memòria, la seva entrega i la presentació final davant del tribunal. També hi haurà una part subratllant totes les reunions i actes de seguiment en relació a la fita de seguiment. Finalment, també s'entregarà el codi del resultat (o resultats) conseqüència de tot el treball fins al moment.

Aquesta tasca requerirà de tot el temps restant fins a la data d'entrega en una data a finals del mes de Gener del 2019. Per tant, hi haurà una dedicació de 50 hores en un període aproximat de 2 setmanes.

## Distribució provisional del treball

Un cop explicades totes les tasques i la dedicació corresponent a cadascuna, aquesta serà la distribució provisional del projecte:

<b>Tasca a realitzar</b>	<b>Hores</b>
Fita Inicial	80
Model funcional bàsic	100
Investigació i optimitzacions	240
Fita final	50
<b>Treball de Fi de Grau</b>	<b>470</b>

## 3.4. Recursos

### Recursos Humans

La part essencial dels recursos en qualsevol projecte són els recursos humans, ja que és necessiten els serveis de diferents tipus de persones especialitzades per tirar-lo endavant. Aquests són els 4 rols que s'han de completar per la realització del treball:

- Cap de projecte: Encarregat de dirigir el treball, fer la planificació i redactar els informes corresponents.
- Dissenyador: Té la funció de dissenyar estratègies per poder resoldre el problema plantejat en el projecte.
- Programador: Encarregat de desenvolupar tot el codi del projecte.

- Tester: Persona que posarà a prova tot el treball realitzat pel programador. S'ha d'assegurar que el seu treball sigui prou bo com per seguir endavant.

## Hardware

Com que tot l'entorn de desenvolupament està centrat en el sistema operatiu Windows, l'únic element de hardware necessari pel treball serà un ordinador portàtil ASUS.

## Software

Els elements de software necessaris pel desenvolupament del projecte són:

- Microsoft Office 2010
- Google Drive
- Matlab R2017b
- Google Chrome (executar Javascript)

No es descarta augmentar aquesta llista a conseqüència de les tasques d'investigació. Aquestes podrien obligar a utilitzar elements de software nous per adaptar possibles optimitzacions.

## Espai de treball

L'espai de treball pel desenvolupament d'aquest projecte serà essencialment al meu domicili o la biblioteca de la UPC.

## 3.5. Entorn de treball

El desenvolupament d'aquest projecte es realitzarà en un ordinador portàtil ASUS que disposa del sistema operatiu Windows 7. L'elecció d'aquest sistema operatiu es justifica simplement amb el fet que tinc una major experiència en la seva utilització i no hi ha una raó aparent per utilitzar-ne un altre. Aquesta serà la única part a comentar referent al hardware, ja que com s'ha dit abans no s'utilitzaren altre eines en aquest sentit.

L'elecció del software mereix una mica més d'atenció. Per la primera part del desenvolupament del projecte, la detecció de cares, s'utilitzaran els llenguatges HTML i Javascript. Les raons per fer-ho són diverses, però la més important és la necessitat d'incorporar una llibreria de Javascript que serà essencial pel seu desenvolupament. Més endavant s'aprofundirà sobre les seves característiques i el que ens pot oferir. Una altra raó per la utilització d'aquests llenguatges són la seva facilitat per dissenyar entorns interactius amb l'usuari on pugui veure fàcilment la informació que es pot obtenir amb el software. El disseny de la pàgina no cal que sigui molt complex, el que implica que els coneixements de HTML necessaris pel treball són baixos. Javascript és un llenguatge molt similar a Java i C++, el que suposa una garantia per la meua habilitat amb aquests llenguatges.

També cal destacar el fet que per fer servir la llibreria mencionada, cal executar els nostres scripts en un servidor. En el cas del projecte es farà servir Python per crear un servidor local que serà més que suficient per poder realitzar totes les funcionalitats necessàries. Els servidors locals es poden montar de moltes formes diferents, però s'ha escollit aquesta pel fet que ja tenia experiència amb ella i ja tenia Python instal·lat.

Per la segona part del projecte, el reconeixement d'emocions, s'ha decidit utilitzar Matlab com a llenguatge de programació. S'ha utilitzat aquest i no altre per molts motius. Matlab és un llenguatge de programació excel·lent per tractar amb imatges i amb qualsevol problema relacionat amb la Visió per Computadors. Conté un gran ventall de funcions especialitzades de manipulació d'elements visuals i permet obtenir característiques molt fàcilment. També és un llenguatge òptim per altres disciplines com les matemàtiques o la estadística, que també estan presents en el projecte. Per aquesta part es podrien haver considerat altres alternatives, però potser la única de raonable seria Python, que comparteix moltes de les característiques mencionades anteriorment. Però el fet de tenir una major experiència i facilitat per programar amb Matlab, tractant-se d'un projecte de magnituds considerables, es va decidir ràpidament eliminar aquesta alternativa.

Aquestes decisions tenen el pes suficient com per generar un problema bastant evident. S'estan realitzant les dues grans parts del projecte amb llenguatges molt diferents i semblarà difícil unificar-ho tot. Més endavant, durant l'explicació del desenvolupament del treball, es veuren les mesures que s'han pres per pal·liar aquesta problemàtica. La conclusió principal de totes aquestes decisions és la prioritització d'utilitzar un llenguatge adequat al problema presentat davant de qualsevol altre motiu considerat secundari.

### 3.6. Diagrama de GANTT

Per la realització del diagrama de GANTT del projecte cal tenir en consideració el mètode de treball escollit, el mètode en cascada. Això vol dir que totes les tasques tenen un ordre molt rigorós com s'ha explicat anteriorment.

Per simplificar la visualització del diagrama, s'ha separat el calendari en les 18 setmanes que el componen. La setmana 1 serà la que comença el dia 17/09/2018, l'inici del projecte, i la darrera, la setmana 18, finalitza el 20/01/2019. Aquesta darrera data és provisional i es podria veure afectada.

A continuació es pot veure el diagrama de Gantt amb la planificació inicial. Les tasques més generals es poden veure en color **blau** i les sub-tasques en color **verd**. Les dependències de les tasques principals estan marcades amb línies **negres**. Les altres dependències, entre les sub-tasques, ja s'han explicat apartats anteriors.

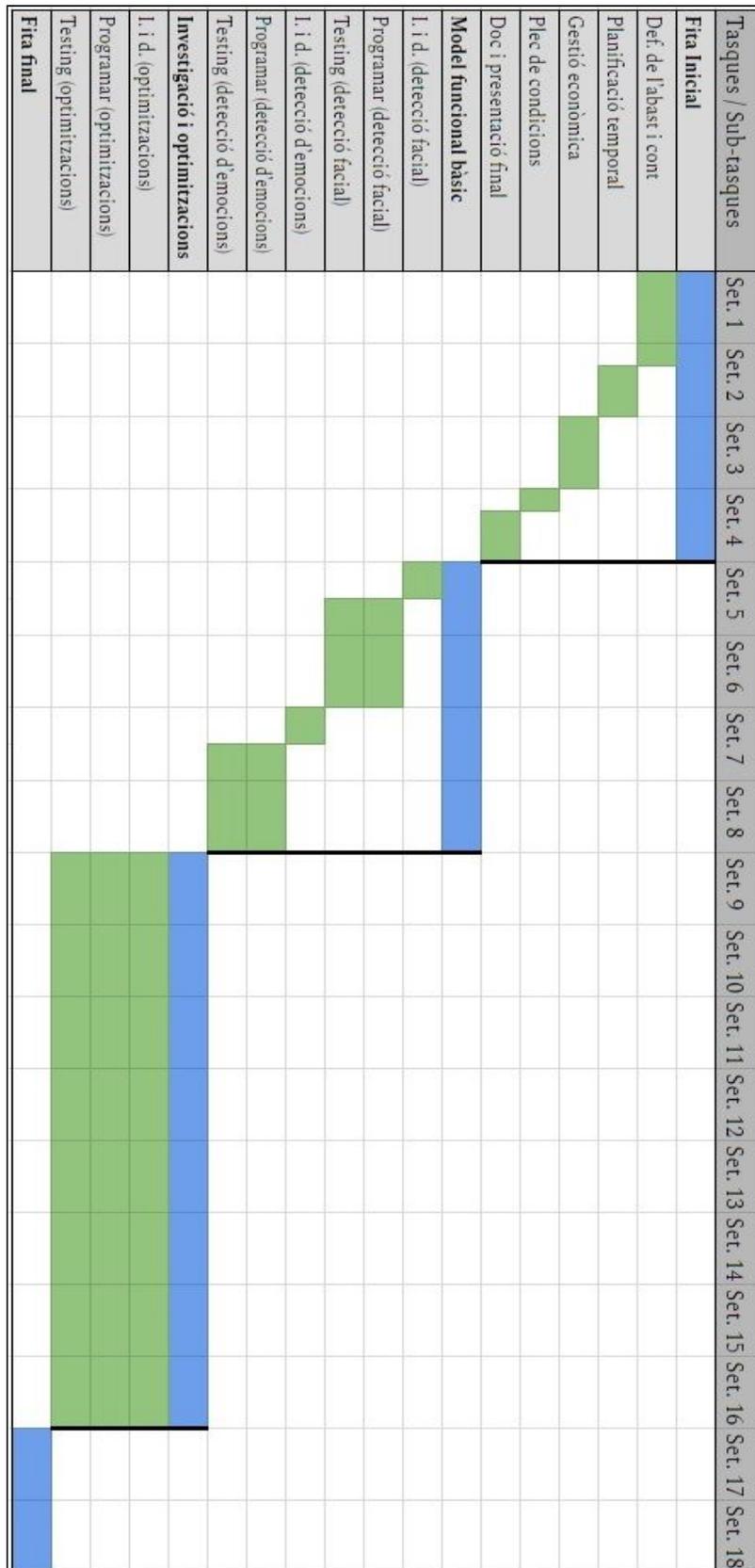


Figura 4: Diagrama de Gantt inicial

### 3.7. Conclusió

Tenint en compte el calendari clarament marcat mostrat en els apartats anteriors i les solucions mencionades als principals problemes que poden aparèixer, juntament amb els mètodes de validació, es pot garantir que el projecte estarà finalitzat en el temps establert.

## 4. Detecció facial

La detecció facial és la primera part que cal abordar en aquest treball ja que d'ella depèn el primer objectiu del projecte. Tenint en compte que l'objectiu final és obtenir un software que permeti reconèixer emocions, aquesta primera part no és la més important, però si és necessari que sigui precisa. Si no aconseguim fer una bona detecció de les cares en les imatges, no podrem continuar amb un anàlisi més profund.

Per aquest motiu, s'ha decidit utilitzar un software extern open-source per agilitzar aquesta part a l'hora de garantir una bona eficàcia. En aquest treball es farà servir una llibreria anomenada "clmtrackr". [9]

### 4.1. Fonaments teòrics

Tota la teoria per aquesta part es basa en un concepte proposat per Jason M. Saragih, Simon Lucey i Jeffrey F. Cohn en el seu article anomenat "Deformable Model Fitting by Regularized Landmark Mean-Shift". [10] Traduït al català significa aproximadament, l'ajust de models deformables per desplaçaments promitjats de punts claus regularitzats.

Començem per la primera part, què és un model deformable? En la disciplina en la que estem treballant, un model serà un conjunt de punts que representen la estructura d'una cara estàndard. A continuació es pot veure un exemple proposat per la llibreria clmtrackr que serà el que s'utilitzarà en el treball.

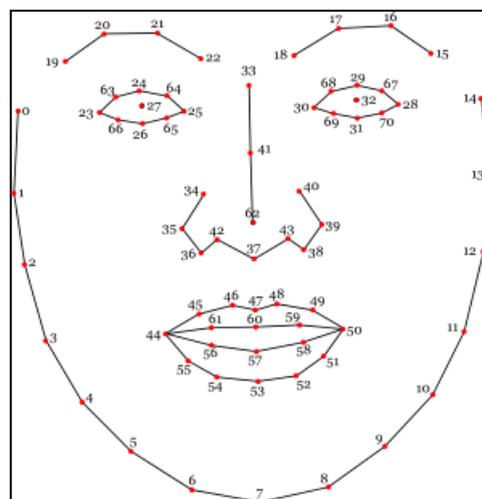


Figura 5: Model d'una cara

En ell podem observar un total de 71 punts diferents que simbolitzen un dels múltiples punts claus que podem trobar a la cara d'una persona.

La següent part serà col·locar-lo i adaptar-lo de forma apropiada perquè representi realment la cara de la persona en la imatge. Per fer-ho, es segueix un procés iteratiu que no s'aturarà fins a una possible convergència. Aquesta és una aproximació dels passos a seguir: [11]

- **Primer pas:** trobar una posició aproximada de la cara en la imatge. Clmtrackr pot realitzar aquest procés de forma automàtica o manual. En el cas del treball, fa una petita cerca automàtica per tal de trobar-la. Com s'explicarà més endavant, degut a limitacions establertes del treball, aquest pas no serà un problema de cara al seu encert.
- **Segon pas:** Per cadascun dels 71 punts, buscar en la regió local corresponent per obtenir una imatge resposta. La regió local d'un punt representa una petita porció de la imatge on es troba aquest punt. La imatge resposta és el resultat de tractar la imatge en una regió local amb un preprocessament determinat que servirà per deduir on és més probable que sigui el punt clau en qüestió. En el nostre cas, clmtrackr no farà cap preprocessament de les imatges respostes per agilitzar la detecció. A continuació, un exemple proposat a l'article de Jason M. Saragih de com són aquestes imatges i com varien en funció del mètode de preprocessament escollit.

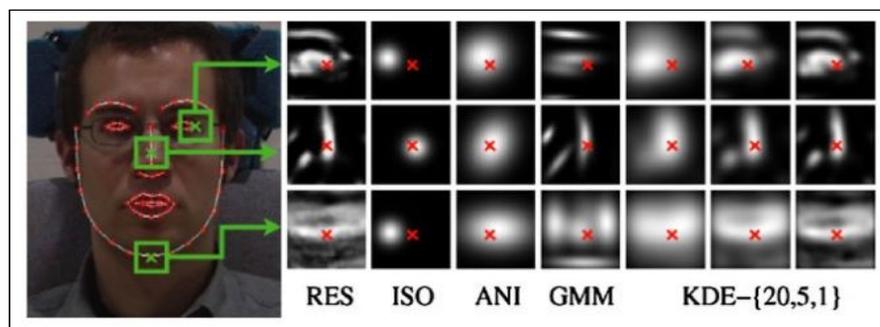


Figura 6: Mapa d'imatges resposta de 3 punts utilitzant diferents mètodes

- **Tercer pas:** Adaptar cada imatge resposta a una funció quadràtica. Tant aquest pas com el següent són estrictament matemàtics. Més detalls es poden trobar en l'article original de Jason M. Saragih a la bibliografia.
- **Quart pas:** Trobar la millor posició de cada punt clau optimitzant les funcions quadràtiques respectant les limitacions de la forma del model. Aquest és el moment on es desplaça la posició original de cadascun dels punts del model per una certa variació. Tot això depèn, evidentment, de la imatge resposta obtinguda.
- **Cinquè pas:** Si hi ha convergència, finalitzar, si no, tornar al pas número dos. El grau de convergència està marcat per la mitjana del moviment dels punts en les darreres iteracions. Quan un model ja és "estable" (té un grau de convergència suficient), vol dir que s'ha adaptat el model satisfactòriament. Clmtrackr permet manipular el grau de convergència (de 0 a 1) necessari per a considerar que una imatge a convergit, que en el nostre cas serà de 0,5.

---

Com ja s'ha esmentat en algun moment, aquest és un procés bastant tècnic, sobretot per la part matemàtica. Tot està explicat de forma molt més detallada en l'article: “Deformable Model Fitting by Regularized Landmark Mean-Shift”, adjuntat a la bibliografia.

## 4.2. Limitacions i restriccions

Molts softwares de detecció i reconeixement de cares destaquen per la seva versatilitat. És a dir, poden complir el seu propòsit en condicions no òptimes, com ara una pobre il·luminació, alineaments complexos de la cara o altres factors. En canvi, com que l'objectiu final del treball no és realment aquest, es faran unes certes restriccions en les imatges que el nostre software podrà tractar correctament. D'aquesta manera, el primer objectiu serà més senzill de complir i ens podrem centrar més en els dos següents.

Els requisits de les imatges amb les que treballarà el nostre software són:

- Condicions òptimes d'il·luminació (il·luminació frontal o total preferentment)
- Enfocament de la cara directe a la càmera o amb una molt lleugera inclinació
- Totalitat de la cara (punts claus) en l'interior de la imatge
- Cara situada en una posició molt propera al centre de la imatge

Tot això tindrà una afectació clara en l'elecció d'una base de dades pel projecte. Més endavant es dedicarà una explicació en aquest sentit quan es parli dels resultats.

## 4.3. Detecció en imatges

Un cop explicat tot el que era necessari, ja es pot passar a la part més pràctica i a experimentar amb el nostre dataset dedicat al testing.

Gràcies a l'ús de diverses funcions de les que disposa la llibreria “clmtrackr”, no ha estat problema programar un simple disseny en html amb el seu corresponent script en Javascript. És el moment de passar la prova i veure el seu comportament. Per fer-ho, s'ha fet un recompte manual de les imatges que han convergit de forma raonable i que per tant, es pot considerar que s'ha realitzat un bon ajust del model. Aquests són els resultats obtinguts, utilitzant dos subjectes de test composts de 800 imatges cadascun:

	Test 1 (home)	Test 2 (dona)
Acceptable	674/800 (84,25%)	622/800 (77,75%)
No acceptable	126/800 (15,75%)	178/800 (22,25%)

La taula anterior és el resultat de proves variant diferents paràmetres en les diverses funcions del *tracker*, però els resultats no han sobrepassat mai les xifres anteriors. Considerant el número de restriccions establert anteriorment, aquestes no semblen unes taxes d'error raonables per tirar endavant. Tirar endavant un error d'aquesta magnitud pot afectar

greument al reconeixement d'emocions posterior. Aquest és un exemple de distinció entre un ajust acceptable i un que no ho és.



Figures 7 i 8: Exemples d'ajust acceptable (esquerra) i no acceptable (dreta)

El problema que s'observa és que aquests errors sorgeixen molt cops de manera puntual. Això té sentit pel fet que cada cop que hi ha una imatge nova, no es guarda informació de l'anterior, el procés descrit als fonaments teòrics comença de zero amb cada imatge.

És per això que s'ha pres la decisió d'aprofitar el fet que clmtrackr té suport per a vídeos. Com que la diferència de frame a frame en un vídeo és mínima, si el model a convergit en un frame  $i$ , el frame  $i+1$  requerirà només un petit ajust per tornar a convergir. D'aquesta manera és més que assumible que tots aquests errors puntuals desapareixeran.

#### 4.4. Detecció en vídeos

Seguint una pauta similar al codi prèviament utilitzat per a la detecció en imatges, s'ha adaptat l'entrada a format de vídeo. Evidentment, la base de dades també haurà de ser en aquest format. Després d'haver sincronitzat aquesta nova base de dades amb la d'imatges, ja es pot provar aquesta nova versió. Els experiments realitzats han sigut idèntics als de la secció anterior, seguint els mateixos criteris. Aquests són els resultats obtinguts per dos vídeos de 800 frames cadascun:

	Test 1 (home)	Test 2 (dona)
Acceptable	771/800 (96,38%)	758/800 (94,75%)
No acceptable	29/800 (3,62%)	42/800 (5,25%)

Com es pot veure, ara sí podem veure uns resultats més positius. Probablement es podria perfilar una mica més afegint tècniques de preprocessament a les imatges originals i les imatges resultat, però afectarà a la eficiència, la qual és molt important per poder funcionar en un vídeo.

Un cop s'ha finalitzat tota aquesta part, ara cal transferir tota la informació obtinguda a la següent part del nostre software. Per això, s'ha afegit l'opció de descarregar tota la informació en fitxers. Aquesta informació són les coordenades de tots els punts en el moment corresponent del vídeo. També s'ha afegit aquesta opció a la detecció en imatges per si es vol fer servir per altres fonts que no siguin vídeos.

## 4.5. Conclusions

Aquesta primera part ja es pot donar com a finalitzada ja que la taxa d'encert (en vídeos) és més que acceptable: 96,38% i 94,75%. Hi han hagut més contratemps de l'esperat pel fet que la primera aproximació a la solució del problema no ha estat suficientment bona. Però realment calia que aquest primer objectiu s'acomplís de forma sòlida ja que la resta del treball depèn de la seva eficàcia.

El fet de modificar el tipus d'entrada original (imatges) a vídeo ha sigut una decisió molt rellevant. Al cap i a la fi, el títol del treball indica que el reconeixement es farà en imatges. Però realment, el problema a resoldre no ha variat. És més, un vídeo és una seqüència d'imatges, per tant, el seu anàlisi serà el mateix que com si fos una imatge. Això s'ha pogut apreciar en aquesta part i també es podrà fer en les següents fases del treball.

## 5. Reconeixement d'emocions

Un cop acabat el primer objectiu del treball, cal centrar-se en el què realment centra l'atenció del projecte: el reconeixement d'emocions. Fins ara el que tenim és un dataset de format per frames de vídeo amb els seus corresponents fitxers obtinguts amb el software anterior. Aquests contenen la posició dels 71 punts claus definits pel model utilitzat vist a la figura 4. Ara cal saber que farem amb aquesta informació per traduir-la en una resposta apropiada, és a dir, una emoció.

### 5.1. Fonaments teòrics

La tècnica fonamental que s'utilitzarà en aquesta part és la classificació. Part molt important en la disciplina del *deep learning* en Visió per Computadors, és una tècnica que permet deduir la classe a la que pertany un objecte mitjançant el seu anàlisi.

El primer pas és realitzar la fase de *learning* o aprenentatge. El procés és molt mecànic i es pot veure resumit a la figura 8. Primer tenim una imatge, com per exemple un gos. A partir d'ella hem d'extreure una sèrie de característiques que ens donin informació que demostri que hi ha un gos a la imatge. La qualitat d'aquestes característiques serà una de les claus per millorar el rendiment de qualsevol software de reconeixement. Un cop les tenim, cal assignar-li un nom a aquesta informació de la que disposem, més concretament a quina classe pertany. En l'exemple, el grup de característiques seran considerades com un objecte de la classe "gos". Ara ja tenim la imatge classificada i només cal repetir aquest procediment amb tot el dataset del que disposem.

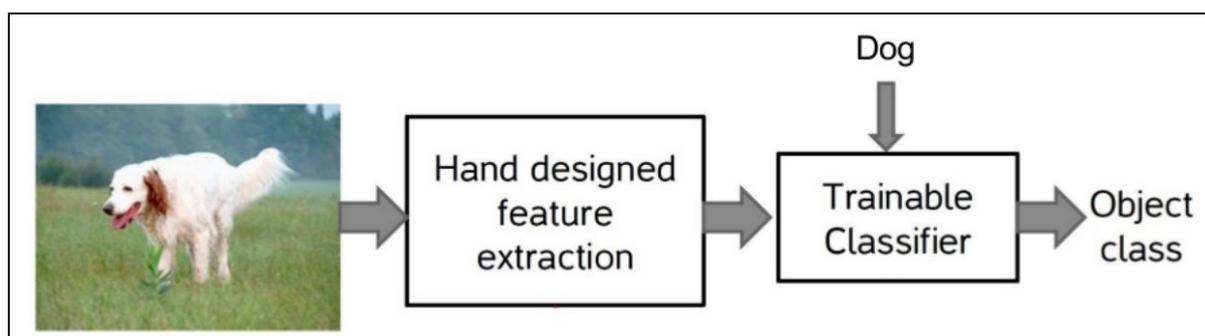


Figura 9: Exemple de classificació d'una imatge

Ara que ja s'ha finalitzat la fase de *learning*, cal posar a prova el nostre classificador mitjançant una fase de *testing*. L'objectiu d'aquesta fase és repetir un procediment similar al de la fase d'aprenentatge, però ara no li assignarem una classe al set de característiques. La responsabilitat de respondre a quina classe pertany serà del classificador.

A continuació podem veure un esquema del procediment complet en qüestió:

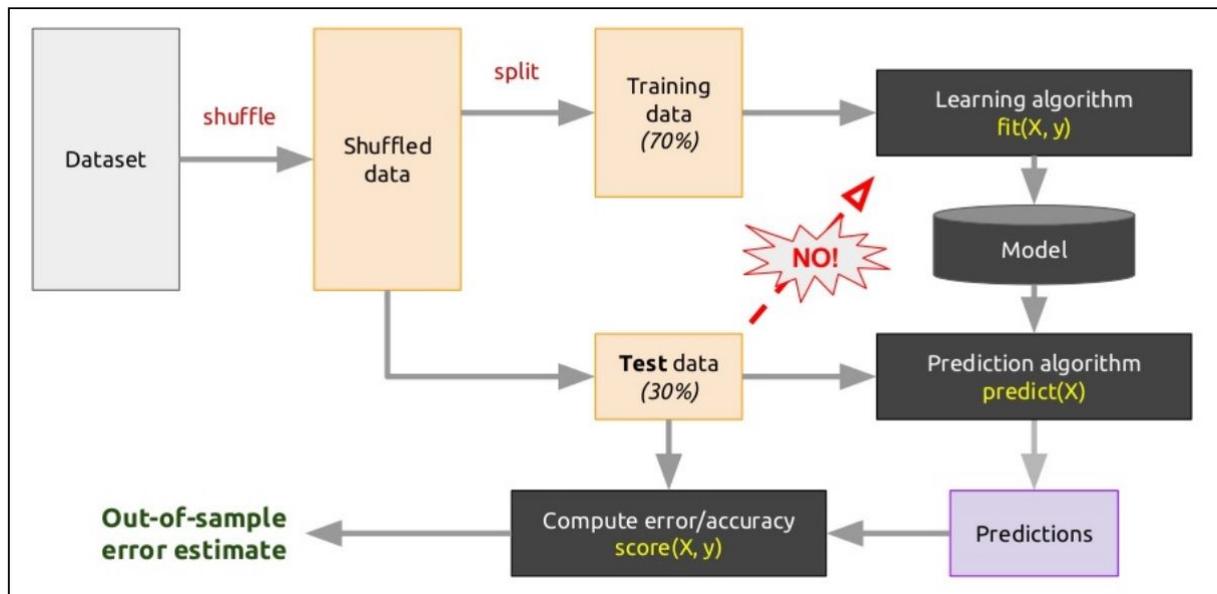


Figura 10: Procediment d'entrenament i testing

Elements a destacar de l'esquema:

- Per tal de posar a prova el software cal dividir el nostre dataset en dos parts. La primera servirà per entrenar i generar el classificador. La segona servirà per posar-lo a prova al *testing*. La proporció ha de ser generalment més gran a favor de l'aprenentatge. Com ja s'ha explicat en la secció de la base de dades, en aquest projecte s'ha dividit en un ~86% destinat al *learning* i un ~14% al *testing*.
- En cap moment es pot utilitzar una part del dataset de *testing* en el procés d'aprenentatge. Això faria que aquesta segona part no tingués sentit, ja que el classificador ja coneix la imatge tal qual.
- La capsula anomenada "Prediction algorithm" fa al·lusió al pas on a partir d'un set de característiques es fa la predicció corresponent.
- Com que nosaltres coneixem a quina classe pertanyen tots els elements del dataset, podem fer una comparació amb l'obtingut gràcies al predictor. D'aquí sortirà la taxa d'encert que representarà la eficàcia (o no eficàcia) del nostre programa.

Finalment, exposar el classificador que s'utilitzarà inicialment en aquest projecte: el *TreeBagger*, traduït al català com a bossa d'arbres de decisió. Un arbre de decisió és una eina molt utilitzada en el camp de la intel·ligència artificial. Permet obtenir una resposta mitjançant un nombre de decisions representats en forma d'arbre. Per tant, una bossa d'arbres serà un conjunt d'aquest tipus d'estructura. El resultat serà el resultat més repetit entre tot el conjunt d'arbres.

Inicialment, en el treball, el nombre d'arbres utilitzat serà de 25. Aquest nombre ja es una garantia d'evitar errors puntuals no desitjats i no suposarà un temps d'execució massa gran generar-lo ni posar-lo a prova.

A continuació es pot veure la estructura d'un arbre de decisió en Matlab:

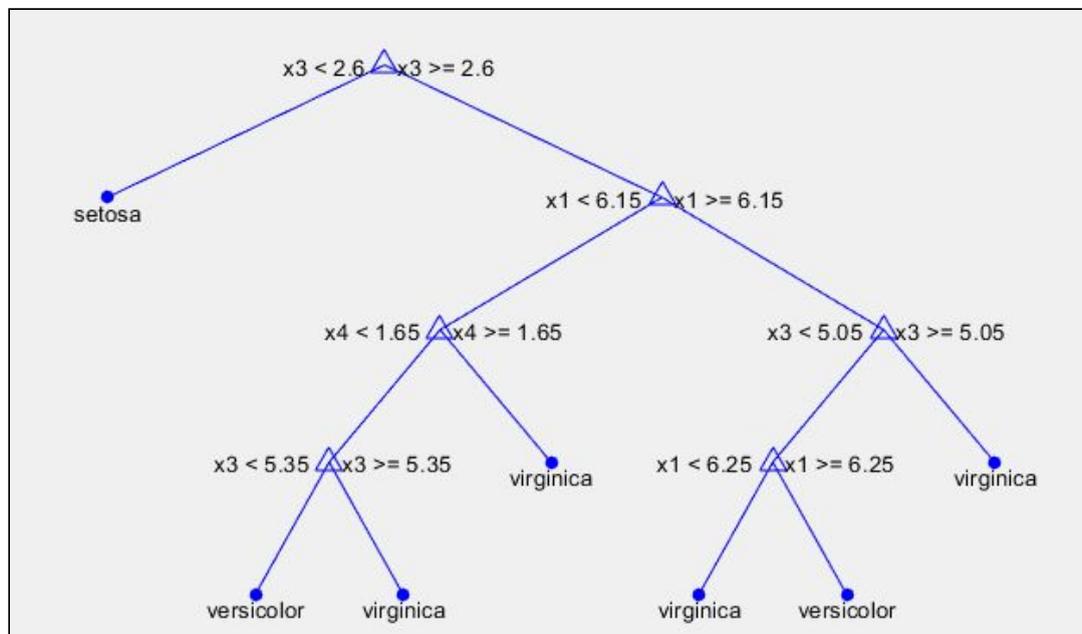


Figura 11: Arbre de decisió

## 5.2. Format dels resultats

Un cop que ja entenem els conceptes que s'utilitzaran, cal aplicar-los i veure el seu funcionament real. Per això, caldrà passar per una fase de *testing* per cada mètode nou que s'implementi, però cal entendre el que obtindrem d'aquesta fase, els resultats.

Primer cal recordar les 8 emocions diferents que el nostre programa ha de saber distingir. Aquestes són: neutral, calma, alegria, tristesa, enuig, por, fastic i sorpresa. Per obtenir resultats convincents, caldrà fer proves amb totes elles. Al igual que per la detecció de cares, disposem de 1600 imatges (o frames) per aquest procediment. Aquestes es divideixen en 800 per cada actor, un home (l'anomenarem "test 1") i una dona ("test 2"). Aquestes 800 imatges també es poden dividir en 8 grups de 100 que fan referència a cadascuna de les emocions a tractar. Per tant, el que tindrem al final seran 2 vídeos (o seqüències d'imatges) que mostraran totes les emocions en l'ordre en què s'han mencionat anteriorment. Més detalls a l'apartat de base de dades a la secció de resultats.

Per tant, com que coneixem a la perfecció el nostre dataset de *testing*, podem ara comparar el nostre coneixement amb el que obtindrem amb el classificador en tot moment. La primera forma en que veurem els resultats és la més evident i eficaç, la taxa d'encert. Aquesta es calcula simplement dividint els resultats obtinguts correctament a través del programa entre el nombre total d'imatges, en aquest cas 800 per element de test. Aquesta serà la base principal per treure conclusions sobre la validesa, o no, de qualsevol mètode arbitrari que dissenyem.

La segona forma de la que disposarem per veure el resultat ja és bastant més visual i menys numèrica. Serà l'element que permetrà a usuaris menys especialitzats amb el tema entendre els resultats obtinguts. Primer es mostrarà la imatge analitzada en qüestió amb un requadre emmarcant la cara de la persona analitzada. El color i la etiqueta representen la emoció detectada amb una major intensitat. En la part inferior s'adjunta un gràfic de barres amb els valors obtinguts pel predictor per cadascuna de les 8 classes del problema (les emocions). Les barres representen la probabilitat de que aquesta persona estigui expressant una certa emoció. Més endavant aprofundirem amb aquest tema. A continuació apareix un exemple de resultat amb aquest sistema:

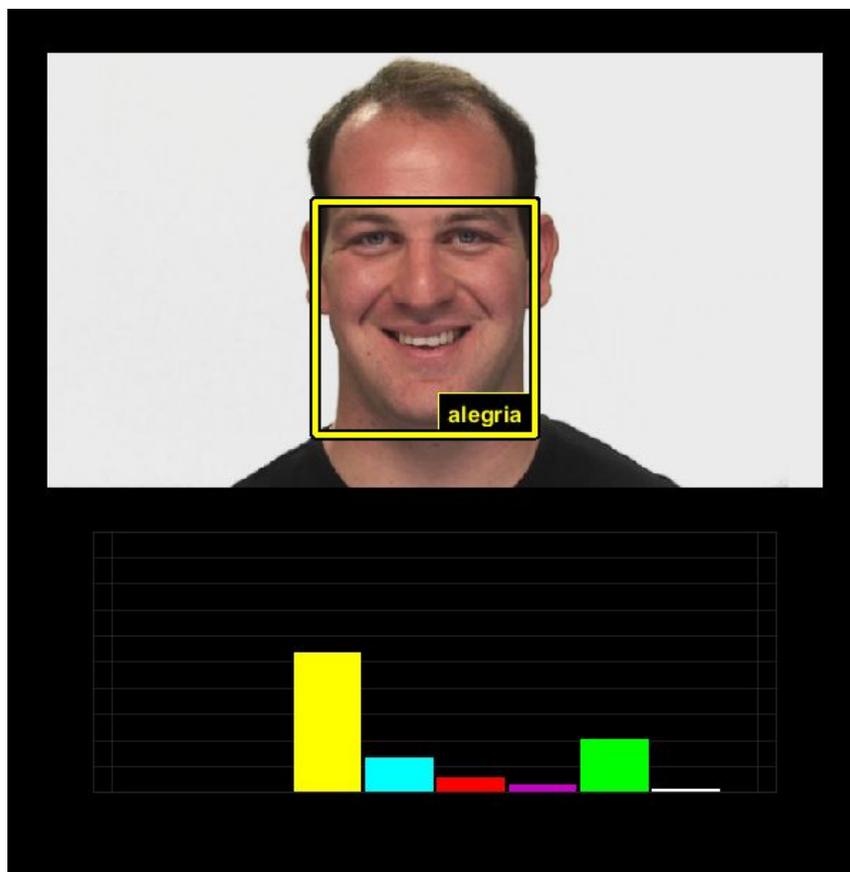


Figura 12: Exemple de resultat amb imatge i gràfic de barres

La tercera i última forma de visualitzar els resultats també és visual i el seu principal objectiu és veure el comportament general del predictor amb un dels elements del test. Consisteix en un total de 8 gràfiques (una per a cada emoció) que representen el valor del resultat obtingut pel predictor en funció del temps. Aquest temps fa al·lusió a la duració completa del video que representen els 800 fotogrames mencionats anteriorment. Els valors obtinguts oscil·len entre 0 i 1. Mentre més alt és el número en un moment donat, més gran és la probabilitat de que la persona estigui expressant aquella emoció en aquell moment del video.

Per tant, un resultat ideal seria el que està representat en la figura 12 a continuació. En els moments en què la persona està, per exemple, alegre, el valor del seu gràfic corresponent seria 1 mentre que els de la resta seria 0. Aquesta seria la definició d'un reconeixement perfecte de les emocions.

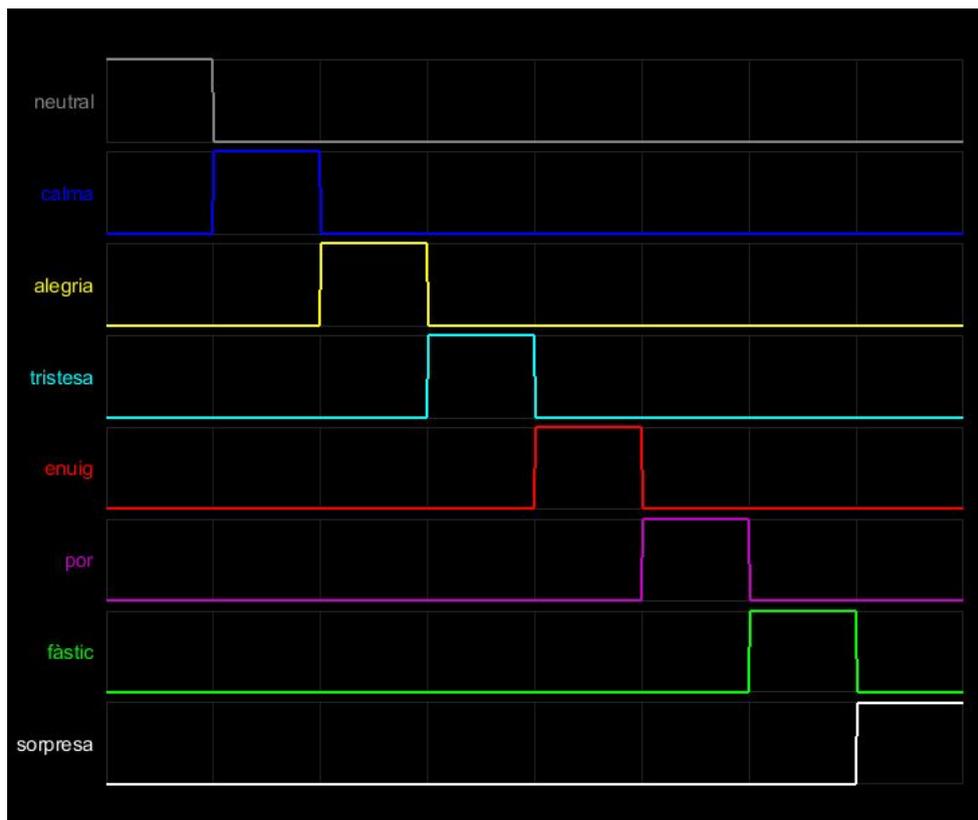


Figura 13: Exemple d'un resultat ideal

Per descomptat, els resultats que s'obtinguin en aquest projecte no seran com el d'aquest exemple. L'objectiu és aproximar-se el màxim possible. Encara que tampoc s'ha de passar per alt el fet que no cal que una emoció rebi un resultat de 1 per part del predictor per ser escollida la emoció predominant. Només es necessita que sigui la que tingui el valor més alt entre les 8. Per això, hipotèticament, podríem obtenir resultats perfectes (100% d'encert) i no obtenir un resultat com el de la figura 12.

Observació: encara que la taxa d'encert sigui 100%, hi ha resultats millors. Obtenir un resultat com el mostrat abans garanteix que el predictor està completament segur que la emoció predominant és correcta. Si la puntuació obtinguda per la emoció predominant és de, per exemple, un 40%, vol dir que hi ha un 60% de probabilitat que la emoció detectada sigui una altra, encara que de forma individual no arribin al 40%. Aquest fet és conegut com la *confiança* del predictor a l'hora de donar resultats.

## 5.3. Tractament dels punts clau

Un cop finalitzades les explicacions més teòriques, podem començar amb els experiments. El primer que cal fer és decidir quin serà el conjunt de característiques que s'utilitzaran primerament al *learning* i després al *testing*. Cal recordar que el classificador que s'utilitzarà per ara és un TreeBagger amb 25 arbres.

Es podria dir que aquesta primera part de la cerca de les característiques serà de pura experimentació. Aquesta consistirà en dissenyar mètodes propis a partir dels punts obtinguts en la detecció de cares com s'ha explicat en el punt anterior. Hi ha moltes maneres de tractar-los i es pot obtenir molta quantitat d'informació si es processen correctament. Ara s'aniran presentant els mètodes de càlcul de característiques que s'han dissenyat i provat respectivament.

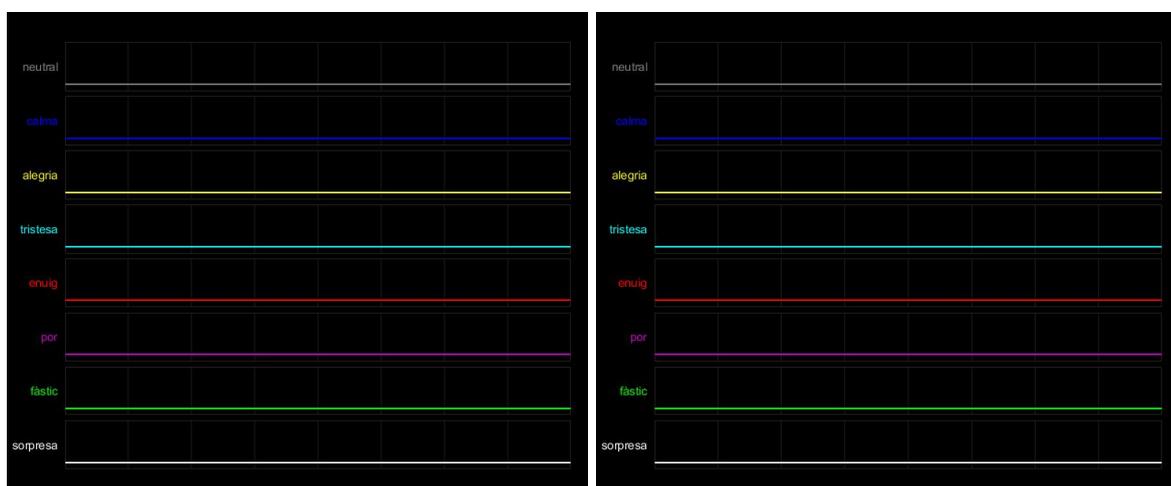
No cal oblidar que encara s'ha de complir el segon gran objectiu del projecte, que consistia en dissenyar un software funcional que permeti fer un reconeixement d'emocions amb resultats mínimament acceptables.

### 5.3.1. Punts

Aquest primer mètode és ben simple, les característiques seran les coordenades dels punts obtinguts directament de la detecció de cares. Aquesta serà una senzilla prova per veure el funcionament del TreeBagger i el predictor.

Per cada mètode es presentaran els resultats amb els percentatges de les taxes d'encert (al final de cada secció) i els gràfics dels dos tests. Això ens permetrà analitzar millor els resultats i veure les possibles fonts d'errors.

Aquests són els resultats del mètode "Punts" (número de característiques: 142):

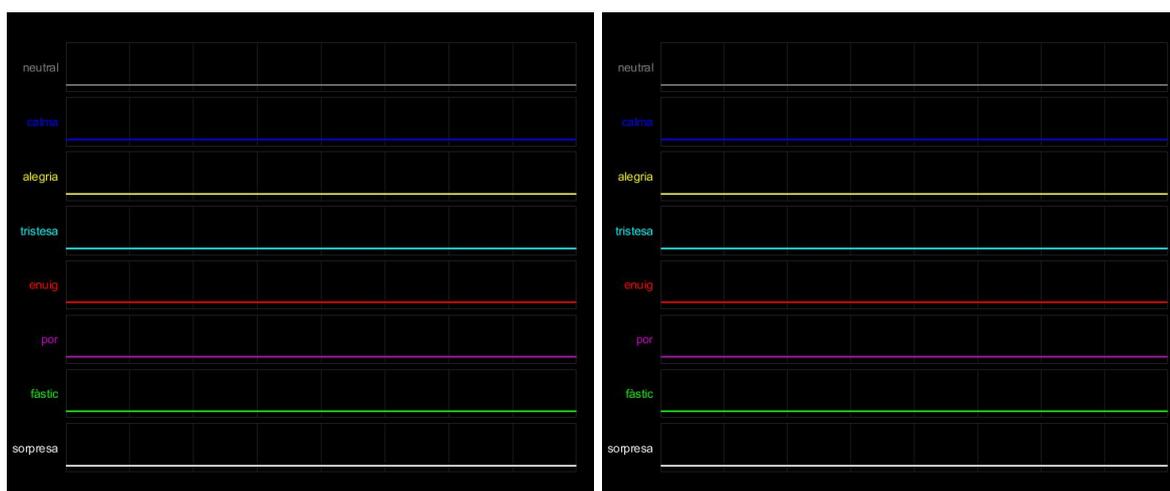


Figures 14 i 15: Resultats amb el mètode "Punts"

Com es pot veure, el predictor no és capaç de fer distincions entre emocions. Això no es problema ni de programació ni del classificador ni del predictor. La causa són les característiques. Es pot veure clarament com assigna una probabilitat idèntica a totes les emocions perquè no té elements suficients per diferenciar-les. Un exemple simplístic del que està succeint seria com intentar distingir entre dos cotxes utilitzant el seu nombre de rodes com a característica.

Hem vist que amb els punts únicament no serveix, per tant, podem provar a normalitzar-los. La posició de la cara en la imatge podria estar alterant molt els resultats en el darrer mètode, per això, transformar les coordenades a un valor unitari només tenint en compte el rang on estan situats els punts seria una bona solució.

Aquests són els resultats del mètode “Punts (normalitzat) (nombre de característiques: 142)”:



Figures 16 i 17: Resultats amb el mètode “Punts (normalitzat)”

Un cop més, els resultats es repeteixen, les conclusions són les mateixes: les característiques no són conclouents.

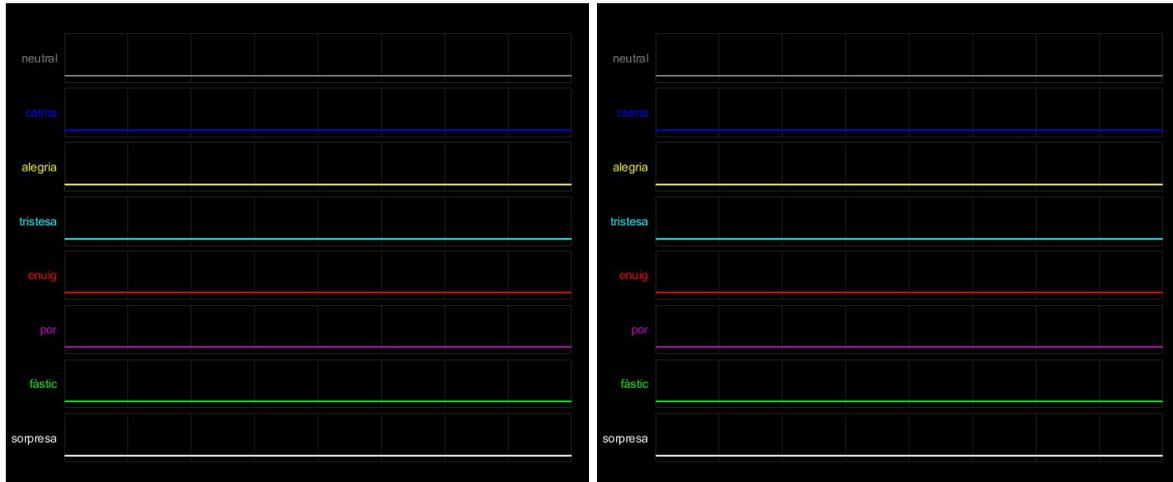
Taxes d'encert de la secció:

Mètode	Test 1 (home)	Test 2 (dona)
Punts	<b>12,50%</b>	<b>12,50%</b>
Punts (normalitzats)	<b>12,75%</b>	<b>12,63%</b>

### 5.3.2. Vectors

Veient que els punts no són suficients, passarem a utilitzar vectors. Aquest mètode consistirà en generar un vector entre un punt i el següent, guardant la seva longitud i angle.

Aquests són els resultats del mètode “Vectors consecutius (número de característiques: 140)”:



Figures 18 i 19: Resultats amb el mètode “Vectors consecutius”

El problema persisteix, per tant, els vectors no són encara l’element distintiu que busquem.

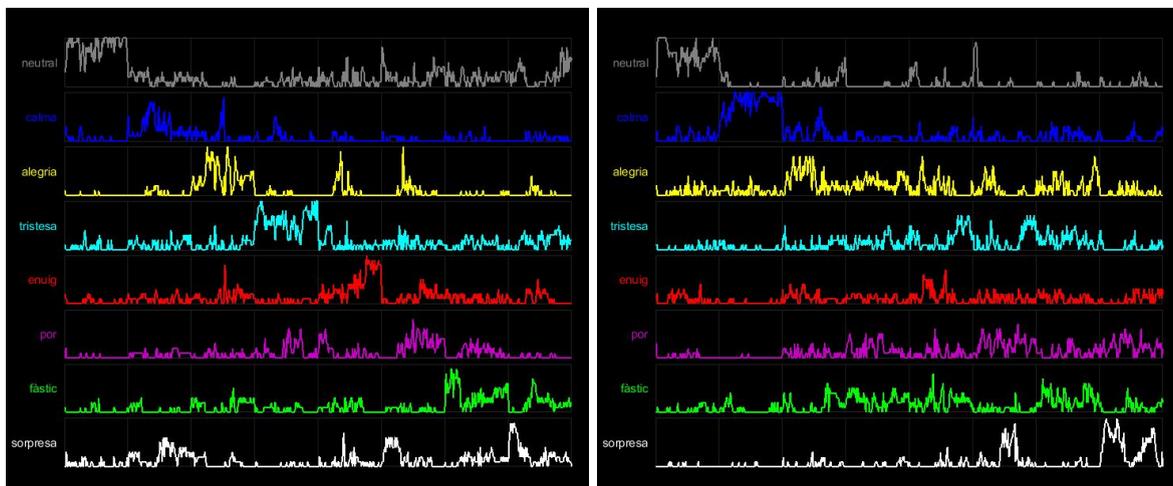
Taxes d’encert de la secció:

Mètode	Test 1 (home)	Test 2 (dona)
Vectors consecutius	<b>13,00%</b>	<b>12,88%</b>

### 5.3.3. Angles

Si ni els punts, ni els vectors han funcionat, el següent pas serà calcular angles entre punts. Per aquesta primera aproximació, seguirem l'algoritme anterior i calcularem l'angle que formen 3 punts consecutius. La utilització d'angles ja és un element que s'utilitza de forma més freqüent en el reconeixement facial i en modelar cares per triangulació. Es d'esperar que aquest mètode ja sigui suficient per poder tenir resultats acceptables.

Aquests són els resultats del mètode “Angles consecutius (número de característiques: 69)”:



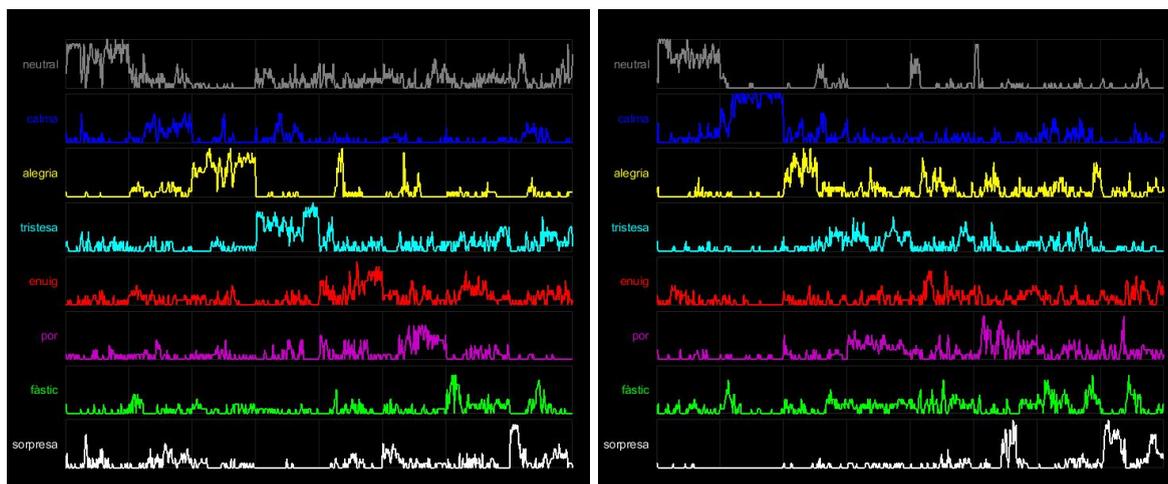
Figures 20 i 21: Resultats amb el mètode “Angles consecutius”

Finalment, podem veure com els resultats ja varien de les línies horitzontals amb les que ens hem trobat fins ara. A més, ja es pot observar com hi ha emocions que tenen uns molt bons resultats, com ara neutral en el test 1 o calma en el test 2. La resta encara està lluny de tenir un encert destacable, però ja hem trobat un camí amb el que podem tirar endavant.

Aquest és el moment en el que l'objectiu número 2 del treball es pot donar per satisfet. Com es pot veure al final de la secció, la taxa d'encert ja està al voltant del 50%. A partir d'ara comença la fase de les optimitzacions i intentar arribar a la màxima taxa d'encert possible en el temps de desenvolupament del projecte.

Seguint amb els experiments, tenint la garantia que els angles són unes característiques adequades per a la distinció entre emocions, aprofundirem una mica més per millorar aquesta taxa d'encert. En aquest pròxim experiment ens dedicarem a augmentar el número de característiques. Aquest fet normalment sol comportar uns millors resultats, ja que tindrem encara més informació sobre la cara de la persona de la que teniem abans. Per això, ara calcularem combinacions d'angles entre un punt i els seus veïns. En aquest cas, seran característiques tots els angles possibles entre un punt i els seus 6 veïns més propers per cada costat (seguint l'ordre d'etiquetatge, per evitar discrepàncies entre diferents cares).

Aquests són els resultats del mètode “Angles combinats (número de característiques: 2556)”:

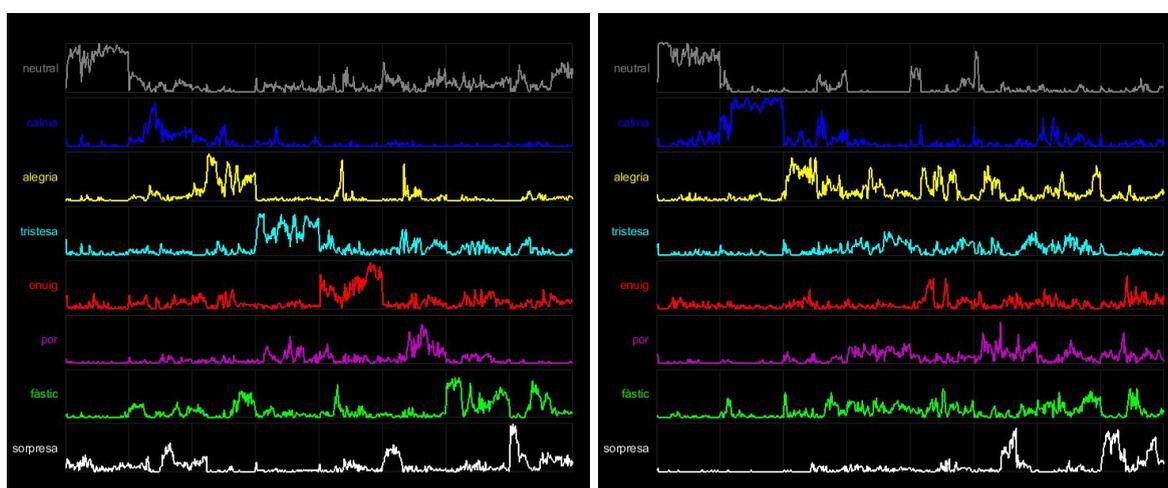


Figures 22 i 23: Resultats amb el mètode “Angles combinats”

Els resultats són bastant similars, però a la taxa d'encert podem veure com realment han millorat una mica. Per tant aquest és el nou millor mètode.

Ara cal pensar en què és el que realment marca la diferència entre les expressions. Les parts més evidents són la boca, els ulls, les celles, o fins i tot una mica el nas. Per això, si ens fixem en els 15 primers punts del model, aquests fan referència al contorn de la cara. Tenir aquesta informació pot arribar a ser útil per a reconèixer persones (la forma de la cara), però per detectar emocions difícilment ens serà necessari. Per això, aquest següent mètode farà el mateix que l'anterior, però sense fer servir els punts del contorn de la cara. Però com que ara disposem de menys punts, podem augmentar fins a 7 el número de veïns que consultarem per calcular els angles sense augmentar massa el nombre de característiques.

Aquests són els resultats del mètode “Angles combinats (sense contorn) (número de característiques: 2744)”:



Figures 24 i 25: Resultats amb el mètode “Angles combinats sense contorn”

---

Difícilment es veurà en la gràfica, però un cop més, els resultats han millorat en aquest test, sobre tot en el test 1. Aquest serà el millor mètode que obtindrem a partir dels punts generats a la detecció de cares.

Taxes d'encert de la secció:

<b>Mètode</b>	<b>Test 1 (home)</b>	<b>Test 2 (dona)</b>
Angles consecutius	55,75%	49,63%
Angles combinats	57,63%	53,00%
Angles combinats sense contorn	63,37%	52,38%

## 5.4. Histogrames de gradients orientats

Un cop s'ha experimentat de diverses maneres amb els punts de la detecció de cares, cal provar altres alternatives per obtenir característiques. El mètode alternatiu escollit són els histogrames de gradients orientats, també coneguts com a HOG.

### 5.4.1. Fonaments teòrics

Els histogrames de gradients orientats són uns generadors de característiques locals en una imatge. Local significa que divideix el tractament de la imatge en porcions rectangulars o quadrades anomenades cel·les i les analitza de forma independent. A cada part de la imatge es realitzarà un anàlisi en funció de les formes que hi hagi. Aquesta es representarà mitjançant un conjunt de vectors de diferents gradients que tindran una longitud en funció de les formes que hi hagi amb un gradient similar al seu. Tota aquesta informació es guarda i pot ser utilitzada per classificar imatges. La figura 25 és un exemple d'aplicació de HOG a una imatge on les formes es poden veure de forma evident.

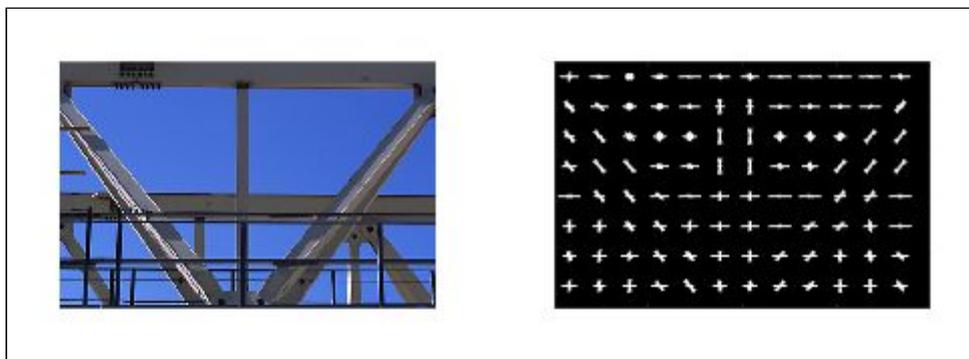


Figura 26: Resultat d'aplicar HOG a una imatge

Matlab incorpora la funció “extractHOGFeatures” que permet fer aquest anàlisi de forma senzilla i modificant una sèrie de paràmetres, com ara el tamany de les cel·les on es fa l'anàlisi local, el qual es farà servir més endavant.

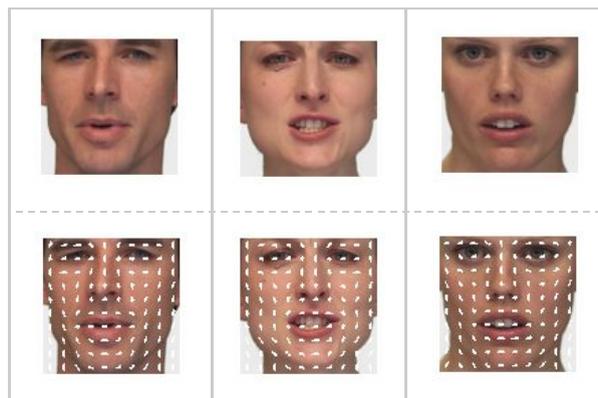


Figura 27: Abans i després de l'aplicació de HOG en 3 imatges de cares

Un element molt important, i el que ha decantat el fet de fer ús dels HOG al projecte, és la seva efectivitat a l'hora d'extreure propietats de les cares de les persones. Com es pot veure en la figura 26, regulant el tamany de les cel·les de forma adient, s'obté una bona aproximació de les formes de les parts més rellevants de la cara, com la boca o els ulls. Ara només cal comprovar la seva efectivitat amb el reconeixement d'emocions.

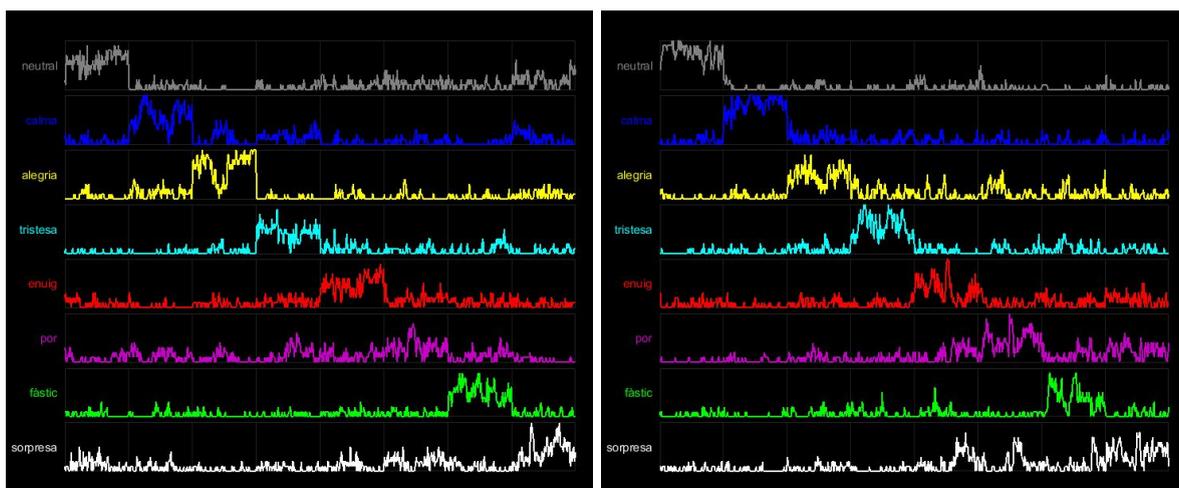
### 5.4.2. Aplicació general

La primera aplicació que es farà en relació als HOG serà una general. Definim general com a fer una única crida a la funció de HOG per a cada imatge. Però abans, cal fer un tractament previ per obtenir la informació que realment volem. Aquests són els passos previs a seguir:

- Retallar la cara de la imatge. Per fer-ho farem servir els punts obtinguts a la detecció de cares. Formarem un requadre amb els mínims i màxims de cada coordenada tenint en compte els 71 punts.
- Redimensionar la imatge. Per fer un anàlisi consistent amb els HOG cal que totes les imatges tinguin el mateix tamany per tenir el mateix número de cel·les i, per tant, de característiques. En aquest mètode, aquest *resize* serà de 100x100 píxels.
- Fixar el tamany de les cel·les. Per tenir una informació més detallada cal un número decent de cel·les, per això, s'ha decidit que el tamany serà de 10x10 píxels, amb un total de 100 cel·les per imatge.

Ara que ja tenim fixat el processament previ de les imatges, ja poder fer la crida a la funció de Matlab “extractHOGFeatures” i obtenir les característiques.

Aquests són els resultats del mètode “HOG (número de característiques: 2916)”:



Figures 28 i 29: Resultats amb el mètode “HOG”

Els resultats obtinguts són notablement millors del que hem obtingut fins al moment amb tots els mètodes de punts. La forma de les gràfiques ja comença a recordar lleugerament a l'exemple de resultat òptim presentat anteriorment.

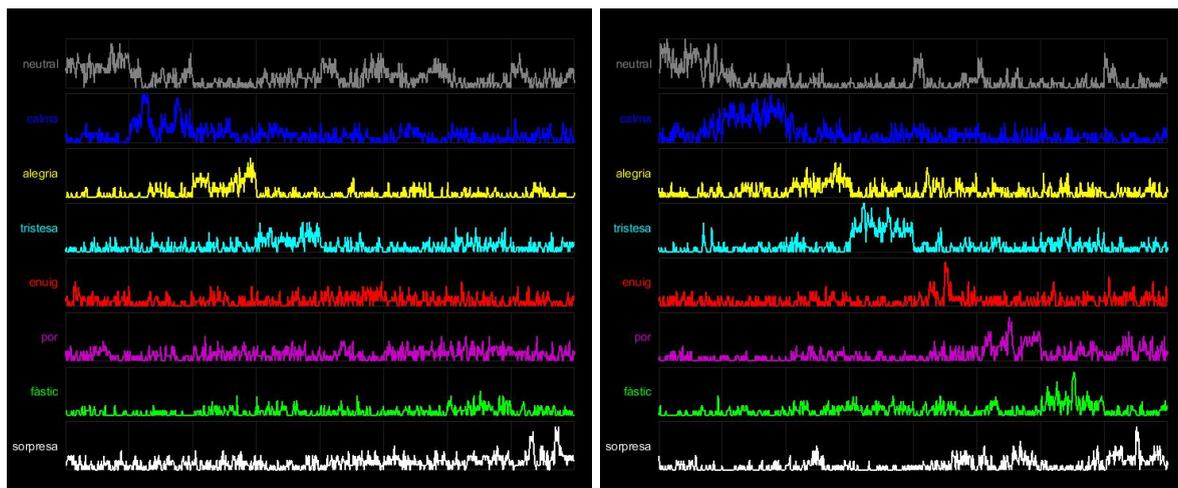
### 5.4.3. Punts clau

Després de l'observació dels bons resultats obtinguts amb el HOG aplicat de forma general, és evident buscar alternatives amb aquesta mateixa tècnica per la seva aparent precisió amb aquest problema. Ara el farem servir de forma més localitzada. Ara, en canvi, el farem servir de forma molt més localitzada. Retallarem una porció de la imatge per a cadascun dels 56 punts claus (sense contorn de la cara) amb un tamany de 7x7 pixels (sent el punt clau el pixel central).

El motiu per provar aquest mètode és el fet que els punts claus representen punts que tindran una gran rellevància a l'hora de determinar la emoció predominant. Per tant, obtenir els gradients al voltant d'aquests punts pot significar obtenir característiques molt rellevants.

El processament previ de les imatges serà el mateix que amb l'aplicació general. L'única diferència és que com les sub-imatges són ara més petites, també ho seran les cel·les. Ara tindran un tamany de 3x3 pixels.

Aquests són els resultats del mètode "HOG dels punts clau (número de característiques: 2016)":



Figures 30 i 31: Resultats amb el mètode "HOG dels punts clau"

Doncs resulta que posant a la pràctica aquest mètode, els resultats no han sigut els esperats, que han sigut inclus inferiors als obtinguts amb molts dels mètodes de la secció anterior "Punts".

Observant els gràfics podem veure com els resultats no són estables. Això pot ser conseqüència de tractar amb sub-imatges tan petites. Al buscar una informació tan localitzada, un petit error en la posició del punt en el reconeixement de cares pot fer variar enormement els resultats dels HOG. Per tant, el HOG aplicat de forma general segueix sent l'opció més encertada.

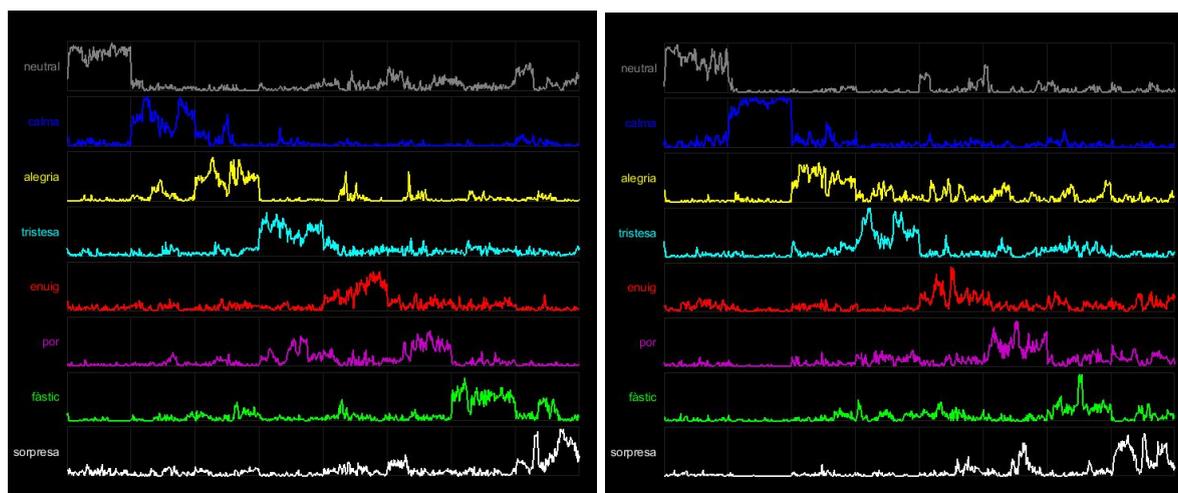
Taxes d'encert de la secció:

Mètode	Test 1 (home)	Test 2 (dona)
HOG	78,63%	76,25%
HOG dels punts clau	41,88%	56,87%

## 5.5. Combinació de característiques

Una tècnica freqüent a l'hora de dissenyar mètodes per obtenir característiques és fer combinacions de les obtingudes amb diferents mètodes. D'aquesta manera, el classificador pot tenir més elements per fer les distincions. Per fer-ho, només cal concatenar les matrius amb les característiques de forma horitzontal i ja podem introduir-ho al classificador.

Aquests són els resultats del mètode "HOG + Angles combinats sense contorn" (número de característiques: 5660)":



Figures 32 i 33: Resultats amb el mètode "HOG + Angles combinats sense contorn"

Els resultats són molt positius. La taxa d'encert és lleugerament superior al mètode "HOG", el que el converteix en el millor mètode de càlcul de característiques. En el gràfic es pot veure com hi ha traces dels gràfics de tots dos mètodes resultant en irregularitats que hauran de ser tractades més endavant.

Taxes d'encert de la secció:

Mètode	Test 1 (home)	Test 2 (dona)
HOG + Angles combinats sense contorn	81,50%	77,00%

## 5.6. Filtres

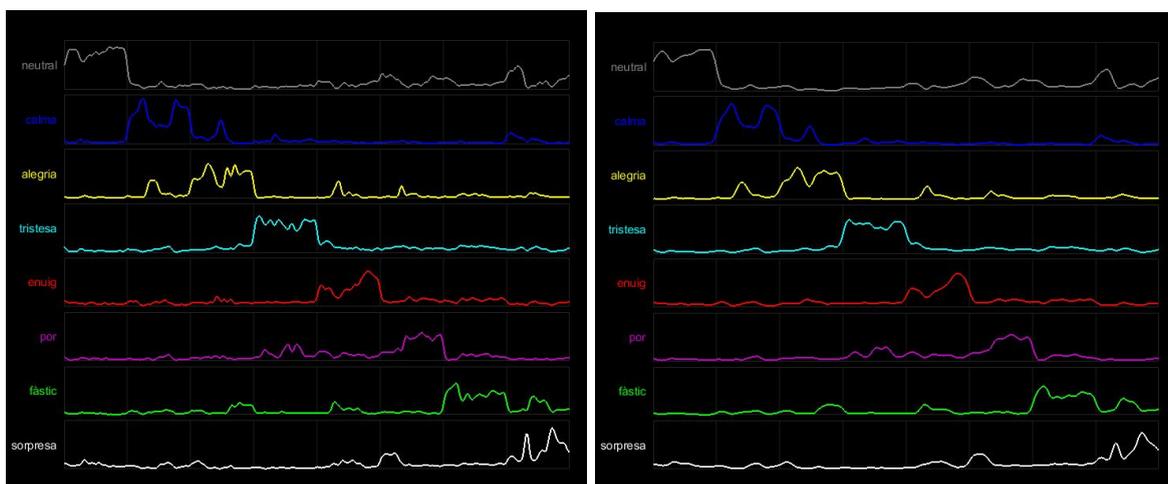
Fins ara s'han provat una sèrie de mètodes per trobar les característiques més adequades pel reconeixement d'emocions. Però totes les gràfiques tenien en comú una certa irregularitat i molts errors puntuals que difícilment es poden evitar. Per tal de suavitzar aquests resultats i eliminar possibles errors puntuals en els mètodes anteriors s'han implementat una sèrie de filtres. El seu objectiu principal serà, per tant, augmentar la taxa d'encert. La seva aplicació consistirà en realitzar una convolució dels resultats prèviament obtinguts amb el filtre escollit (un vector). El mètode utilitzat per aquestes proves és "HOG + Angles combinats sense contorn" ja que ha demostrat ser el més precís, com s'ha vist en la secció anterior

Abans de continuar, cal exposar els diferents tipus de filtres escollits. Són els següents:

- **Piramidal:** Filtre en el qual hi ha un increment lineal dels valors dels extrems cap al centre, on és el màxim, p. e. [1/9 2/9 3/9 2/9 1/9]. Dona un major pes al frame central i un menor als dels extrems.
- **Gaussià:** Filtre modelat a partir de la Campana de Gauss. Dóna un gran pes al frame central i molt menys als extrems. Molt més centralitzat que el Piramidal.
- **Lineal** (o mitjana): Filtre que otorga el mateix valor a tots els frames que abarca, tant si és el central com un dels extrems. Ideal per eliminar desviacions puntuals.

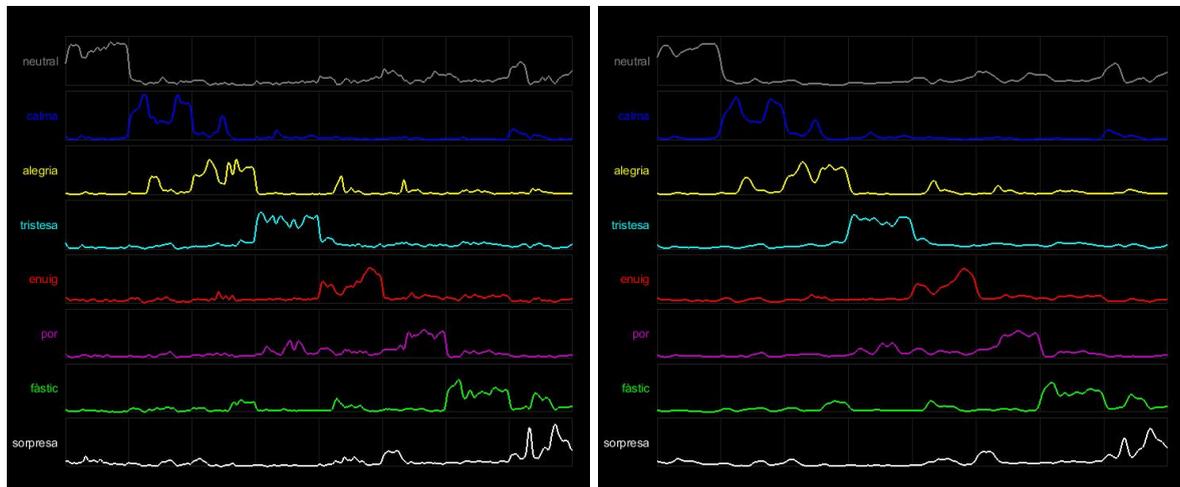
De tots aquests filtres també es farà una distinció segons el tamany. Es considerarà un filtre petit a aquell que abarca el frame actual  $\pm 5$  frames i gran a aquell que considera el frame actual  $\pm 10$  frames. La longitud escollida pels filtres és arbitrària, però tenen en compte un temps raonable en què una persona pot canviar d'estat d'ànim de forma repentina i que el software sigui capaç de reaccionar.

Aquests són els resultats amb el filtre Piramidal:



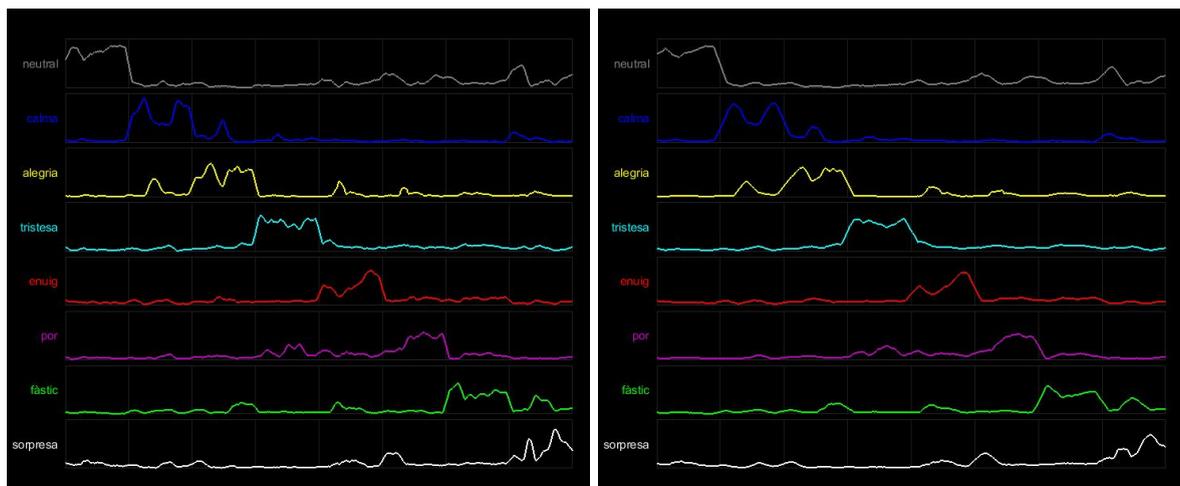
Figures 34 i 35: Resultats amb el filtre piramidal petit (esquerra) i gran (dreta)

Aquests són els resultats amb el filtre Gaussià:



Figures 36 i 37: Resultats amb el filtre gaussià petit (esquerra) i gran (dreta)

Aquests són els resultats amb el filtre Lineal:



Figures 38 i 39: Resultats amb el filtre lineal petit (esquerra) i gran (dreta)

Punts a destacar sobre els gràfics:

- Els filtres “grans” tenen una taxa d’encert superior als “petits”.
- Els filtres lineals eliminen de forma més efectiva els errors puntuals i han aconseguit la millor taxa d’encert que la resta.
- El filtre gaussià és el més sensible a canvis més puntuals.
- El filtre piramidal és una bona opció intermitja entre les característiques que tenen el gaussià i el lineal.
- Resulta evident que filtrar les dades originals és essencial en aquest problema.

Taxes d'encert de la secció:

Filtre	Test 1 (home)	Test 2 (dona)
Sense filtre	81,50%	77,00%
Piramidal petit	85,57%	87,25%
Piramidal gran	87,25%	92,50%
Gaussià petit	84,75%	85,63%
Gaussià gran	87,00%	91,75%
Lineal petit	86,12%	89,38%
Lineal gran	<b>88,25%</b>	<b>94,50%</b>

## 5.7. Mètodes alternatius

Un cop ja s'han decidit les millors característiques i un filtre adequat per optimitzar els resultats i lliurar-los d'errors puntuals, cal tornar al principi. Una de les primeres pautes que es van establir per aquesta part del desenvolupament del projecte és el classificador a utilitzar: el TreeBagger. Però ara cal investigar si altres classificadors de característiques més aviat similars poden ajudar a millorar encara més els resultats. A continuació es presentaran i provaran un parell d'alternatives al TreeBagger.

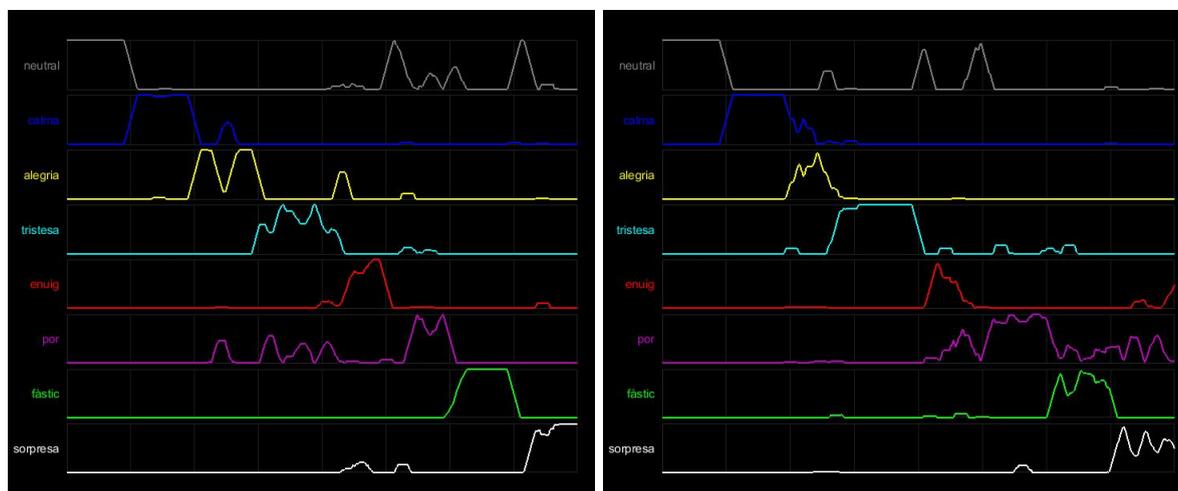
### 5.7.1. *k-nearest-neighbours*

La primera alternativa serà classificar les dades amb l'algoritme *k-nearest-neighbours*. Com el seu nom indica, l'objectiu d'aquest algoritme és trobar els veïns més propers per a cada objecte. El seu procediment és el següent: [12]

- Primer pas: classificar les dades utilitzant aquest algoritme
- Segon pas: per cada element que s'utilitza a la fase de *testing*, el predictor calcularà la seva distància amb cada element classificat a la fase de *learning*.
- Tercer pas: ordenar els elements per distàncies de forma decreixent i quedar-se amb els *k* primers elements de la llista.
- Quart pas: donar un resultat en funció de la classe més repetida entre els *k* elements escollits prèviament.

Per tant, aquest algoritme pot ser molt útil amb el nostre problema buscant les *k* cares més similars (segons la informació obtinguda per les característiques) en cada iteració de la fase de *testing*, i donar un resultat.

Aquests són els resultats amb el classificador *k-nearest-neighbours* amb  $k = 5$ :



Figures 40 i 41: Resultats amb el classificador “knn”

Els resultats són bons, però no arriben a la taxa d'encert que hem obtingut prèviament amb el classificador TreeBagger.

Cal destacar que el valor utilitzat com s'ha exposat anteriorment és de  $k = 5$ . Aquest valor ha estat testejat com a òptim amb una sèrie d'experiments amb aquest mètode de classificació. Uns valors massa grans poden tenir en consideració altres classes d'emocions similars però que no són la actual. Un valor massa petit, com ara  $k = 1$ , té una gran probabilitat de donar un resultat erroni que no es pugui promitjar amb altres “veïns” que es trobin a una distància similar.

Finalment, destacar la forma tan diferent que té el gràfic en comparació amb els que s'han vist fins al moment. En general, els resultats correctes han rebut una puntuació molt propera a 1, el que simbolitza la confiança del predictor en donar aquesta resposta davant d'altres. El problema d'aquest algoritme és la precisió clarament menor, que és el que fa baixar realment la taxa d'encert, el nostre objectiu principal.

Taxes d'encert de la secció:

Classificador	Test 1 (home)	Test 2 (dona)
k-nearest-neighbour ( $k = 5$ )	83,25%	81,37%

### 5.7.2. Anàlisi discriminant

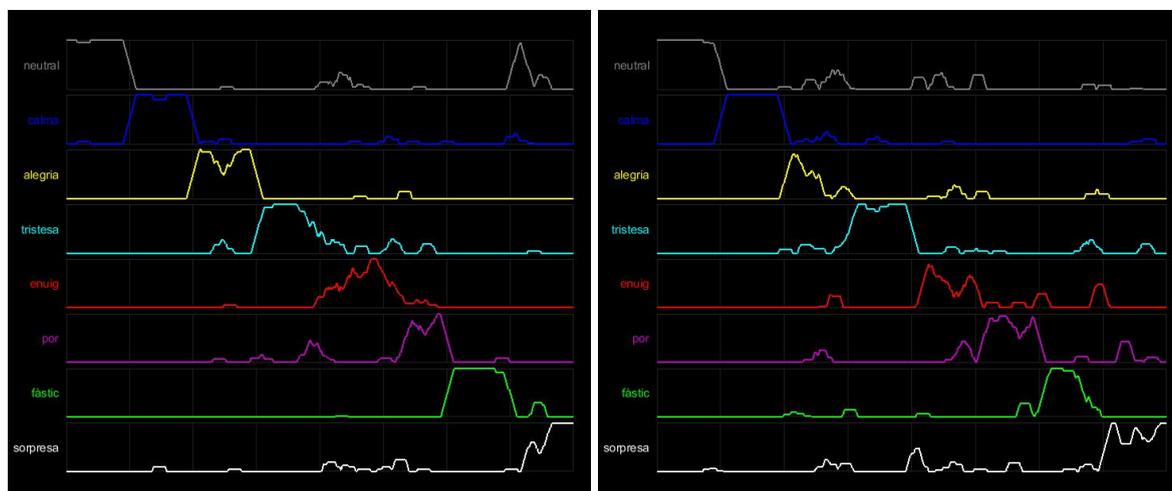
La segona alternativa que es proposa en aquest treball és un classificador per “anàlisi discriminant”. Aquest mètode de classificació es pot dividir en dues parts. [13]

La primera és una anàlisi de regressió. La variable independent serà el conjunt de característiques que determinen a quina classe pertany cada objecte de la fase de *learning*. La variable dependent, en aquest cas, serà categòrica, que vol dir que serà una de las categories (o classes) de les que es tracten en el problema. En el nostre cas, les classes són les 8 emocions que s'han mencionat repetidament en el treball.

La segona és assignar cadascun dels elements nous (test) a una de les classes ja designades amb un grau de risc. Com que ja s'ha fet l'anàlisi de regressió prèviament, aquesta part és senzillament calcular a quina de les classes és més probable que el nou objecte es pugui assignar.

De cara a veure la seva aplicació al nostre problema, és pot veure com una forma d'intentar buscar una solució. En funció dels resultats es veurà si és un algoritme de classificació adequat o no.

Aquests són els resultats amb el classificador d'anàlisi discriminant:



Figures 42 i 43: Resultats amb el classificador “Discriminant”

Efectivament, aquest classificador es pot aplicar pel reconeixement d'emocions. Els resultats són molt similars als obtinguts amb el TreeBagger encara que una mica per sota en general. També destacar, com amb el classificador “knn” el grau elevat de confiança del predictor al donar respostes en general. Però, com l'objectiu del projecte és augmentar la taxa d'encert, el TreeBagger serà encara l'algoritme de classificació utilitzat.

Taxes d'encert de la secció:

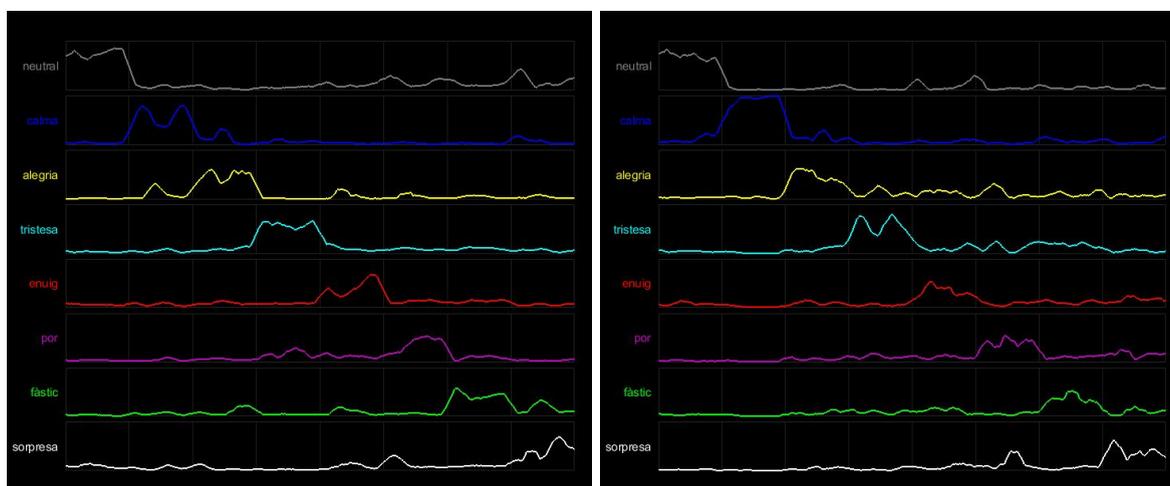
Classificador	Test 1 (home)	Test 2 (dona)
Discriminant	91,50%	87,38%

### 5.7.3. Paral·lelització del TreeBagger

Com s'ha pogut veure en els apartats anteriors, les altres dues alternatives proposades no han resultat ser millors (encara que per poc) que el TreeBagger. Per això, es podria considerar com a alternativa intentar màximitzar la eficiència del classificador actual per veure si encara es pot optimitzar una mica més.

La idea principal seria augmentar el nombre d'arbres significativament, però això també comportarà pèrdues de rendiment i un augment de la necessitat de memòria. Però existeix una solució al problema del rendiment, que és paral·lelitzar la creació del classificador. D'aquesta manera es poden generar els arbres de forma més ràpida i eficient. Per aquest motiu, utilitzant l'opció de Matlab "UseParallel" s'ha decidit augmentar el nombre d'arbres de 25 a 100. Amb això, s'espera augmentar lleugerament la taxa d'encert mantenint un temps d'execució similar o no massa superior. [14]

Aquests són els resultats amb el classificador TreeBagger amb 100 arbres:



Figures 44 i 45: Resultats amb el classificador "TreeBagger (100 arbres)"

Com s'esperava, les taxes d'encert són lleugerament superiors, però tractant-se de valors ja superiors del 90%, qualsevol petita optimització és benvinguda.

Aquests seran els resultats definitius i la taxa d'encert més elevada que s'assolirà en aquest projecte.

Taxes d'encert de la secció:

Classificador	Test 1 (home)	Test 2 (dona)
TreeBagger (100 arbres)	91,00%	96,00%

## 5.8. Conclusions

Finalment, després de totes les setmanes de disseny i experimentació per buscar les millors tècniques de reconeixement d'emocions, aquesta és la combinació que ha donat els millors resultats:

Característiques: “HOG + Angles combinats sense contorn”

Filtres: “Lineal (o mitjana) gran”

Classificador: “TreeBagger paral·lelitzat amb 100 arbres”

Els resultats obtinguts amb aquesta combinació han estat del 91.00% i 96.00% en els dos tests dels que disposem. Aquests resultats han estat realment satisfactoris i fins i tot per sobre de les expectatives inicials. Ara només cal comprovar quin és el grau de precisió del nostre software en comparació amb altres de similars.

---

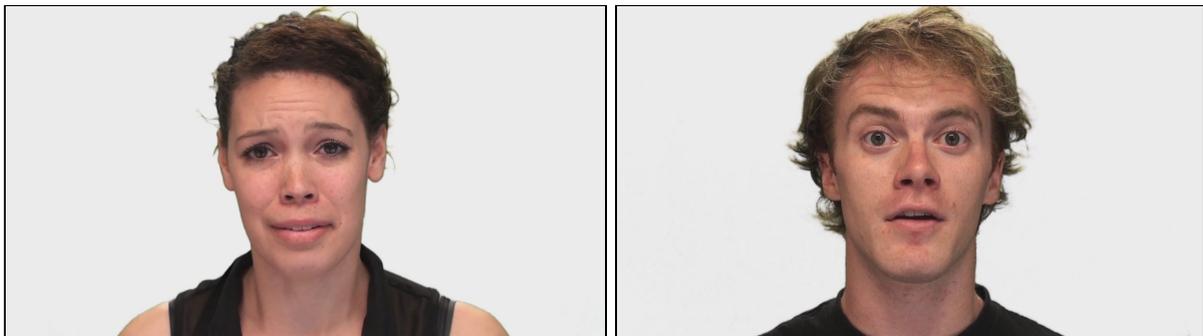
## 6. Resultats

### 6.1. Base de dades

La base de dades utilitzada en el projecte és un element que calia concretar de forma inicial ja que tot el treball estarà centrat en el seu entorn. Aquesta elecció s'havia de fer amb criteri ja que el comportament del software podia variar enormement en funció del que escollim com a base de dades.

Inicialment, la condició número 1 per la cerca d'una base de dades és que tingui les imatges classificades per emoció. Evidentment, si el treball està centrat en el reconeixement d'emocions, la base de dades escollida ha d'estar centrada en aquest aspecte. També és important que hi hagi diversitat i un volum considerable d'imatges. Com s'ha tractat abans amb més profunditat, hi ha una sèrie de característiques que les imatges han de tenir per aquest treball. Aquestes tenen a veure amb la posició de la persona fotografiada a la imatge, la il·luminació, entre d'altres.

Per totes aquestes raons, la base de dades escollida és "The Ryerson Audio-Visual Database of Emotional Speech and Song" o RAVDESS. [15]



Figures 46 i 47: Imatges de la base de dades RAVDESS

Aquesta base de dades consisteix en un total de 7356 arxius de vídeo en format mp4. En ells apareixen un total de 24 actors, 12 homes i 12 dones, recitant una frase o cantant. L'element més important per aquest treball es que tots els vídeos estan repetits amb diferents emocions per part de tots els actors. Podem trobar fins a 8 emocions diferents amb un grau d'intensitat alt o baix. Per simplificar el treball, només s'han utilitzat les emocions amb intensitat alta, per facilitar el seu reconeixement.

Cal destacar el fet que aquesta base de dades no són imatges, si no vídeos. Això no resultarà en un problema ja que es poden convertir fàcilment en una seqüència d'imatges i el problema restarà inalterat pel moment.

Com s'ha dit abans, en la base de dades hi ha un total de 24 actors, meitat homes i meitat dones. Per tal d'afrontar un problema com el del treball cal fer una distinció entre la part de

la base de dades dedicada a la fase de *learning* i la fase de *testing*. Els detalls més tècnics sobre el significat i la utilitat de fer aquesta divisió ja s'han explicat anteriorment, però cal fer-la amb criteri. S'ha decidit que 22 dels 24 actors serviran com a base de dades de la fase de *learning*. Encara que no tots s'han arribat a utilitzar. S'han filtrat una sèrie de vídeos per part d'alguns dels actors perquè excedien alguna de les restriccions explicades al capítol anterior, principalment per no mirar directament a la càmera en determinats moments. D'altra banda, això deixa a 2 dels 24 actors restants com a elements amb una finalitat de *testing*: un home (test 1) i una dona (test 2).

Finalment, el total de frames del que disposarem serà de 9600 per la fase de *learning* (1200 frames per emoció) i 1600 frames per la fase de *testing* (800 frames per actor i 100 per emoció). La proporció per tant ha quedat en aproximadament un 85,7% del dataset per l'aprenentatge i un 14,3% per les proves.

## 6.2. Resum dels resultats

Un cop analitzada la base de dades utilitzada pel projecte, aquests són tots els resultats que s'han obtingut. El color **vermell** representa els mètodes que no són concloents o acceptables, depenent de la situació. En color **verd** es destaca el millor dels mètodes, el que ha obtingut una millor taxa d'encert en els dos tests.

Resultats de la detecció de cares en imatges:

	Test 1 (home)	Test 2 (dona)
Acceptable	<b>674/800 (84,25%)</b>	<b>622/800 (77,75%)</b>
No acceptable	<b>126/800 (15,75%)</b>	<b>178/800 (22,25%)</b>

Resultats de la detecció de cares en videos:

	Test 1 (home)	Test 2 (dona)
Acceptable	<b>771/800 (96,38%)</b>	<b>758/800 (94,75%)</b>
No acceptable	<b>29/800 (3,62%)</b>	<b>42/800 (5,25%)</b>

Resultats de la cerca de característiques:

Mètode	Test 1 (home)	Test 2 (dona)
Punts	<b>12,50%</b>	<b>12,50%</b>
Punts (normalitzats)	<b>12,75%</b>	<b>12,63%</b>
Vectors consecutius	<b>13,00%</b>	<b>12,88%</b>
Angles consecutius	55,75%	49,63%
Angles combinats	57,63%	53,00%
Angles combinats sense contorn	63,37%	52,38%
HOG	78,63%	76,25%
HOG dels punts clau	41,88%	56,87%
HOG + Angles combinats sense contorn	<b>81,50%</b>	<b>77,00%</b>

Resultats amb filtres:

Filtre	Test 1 (home)	Test 2 (dona)
Sense filtre	81,50%	77,00%
Piramidal petit	85,57%	87,25%
Piramidal gran	87,25%	92,50%
Gaussià petit	84,75%	85,63%
Gaussià gran	87,00%	91,75%
Lineal petit	86,12%	89,38%
Lineal gran	<b>88,25%</b>	<b>94,50%</b>

Resultats amb diferents classificadors:

Classificador	Test 1 (home)	Test 2 (dona)
TreeBagger (100 arbres)	<b>91,00%</b>	<b>96,00%</b>
k-nearest-neighbour (k = 5)	83,25%	81,37%
Discriminant	91,50%	87,38%

### 6.3. Comparació amb alternatives al mercat

Un cop finalitzat amb tot el software propi, cal posar-lo a prova comparant-lo amb altres de similars que es poden trobar en el mercat. Aquest anàlisi tindrà en consideració un estudi realitzat amb tres softwares similars: Affectiva (AFFDEX), OpenFace-Cambridge i Emotient (iMotions-FACET). [16]

Primer compararem els resultats obtinguts amb la detecció de cares amb els d'aquests altres softwares. Recordar que la mitjana d'encert dels dos tests que hem realitzat és del 95,56%. A continuació apareix la gràfica amb les dades de la resta.

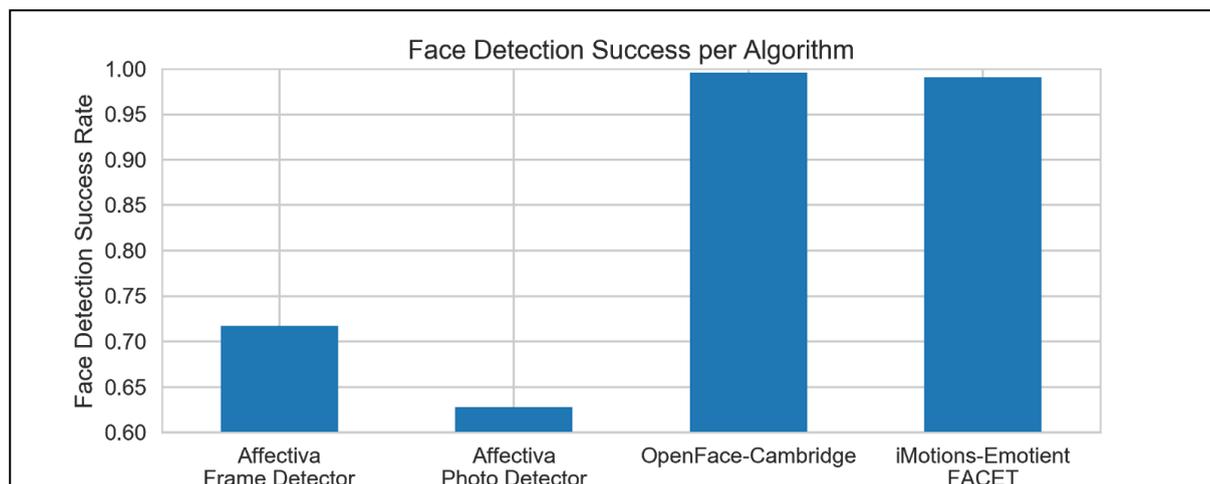


Figura 48: Taxes d'encert en la detecció de cares per part d'Affectiva, OpenFace i Emotient

Com es pot veure, Affectiva té una taxa realment baixa comparada amb la resta, mentre que OpenFace i Emotient estan situats per sobre del 99%. Això significa que el nostre software no estaria a l'alçada dels dos millors softwares de reconeixement d'emocions en la detecció de cares, però tampoc massa lluny.

Seguidament, passarem a comparar els resultats del reconeixement d'emocions. Per aquesta part només farem la comparació amb Affectiva (AFFDEX) i Emotient (FACET), ja que no hi ha resultats numèrics contrastats per part de OpenFace. Com la majoria de softwares de reconeixement d'emocions, disposen d'un ventall de 7 o 8 emocions, que són les mateixes que s'han tractat en aquest treball contant o no amb l'emoció neutral. [17]

Evidentment, recordar que els tests realitzats en el nostre software han donat una taxa d'encert promitjada d'un 93,5%. Els resultats de FACET i AFFDEX s'han provat en 3 bases de dades diferents: ADFES, WSEFEP i RaFD. Els seus resultats han estat del 73%, 66% i 77% respectivament per a AFFDEX i del 99%, 92% i 99% respectivament per a FACET. Això dona uns resultats aproximats del 72% per a Affectiva i un 96.7% per a Emotient.

Remarcar que aquests resultats, encara que no

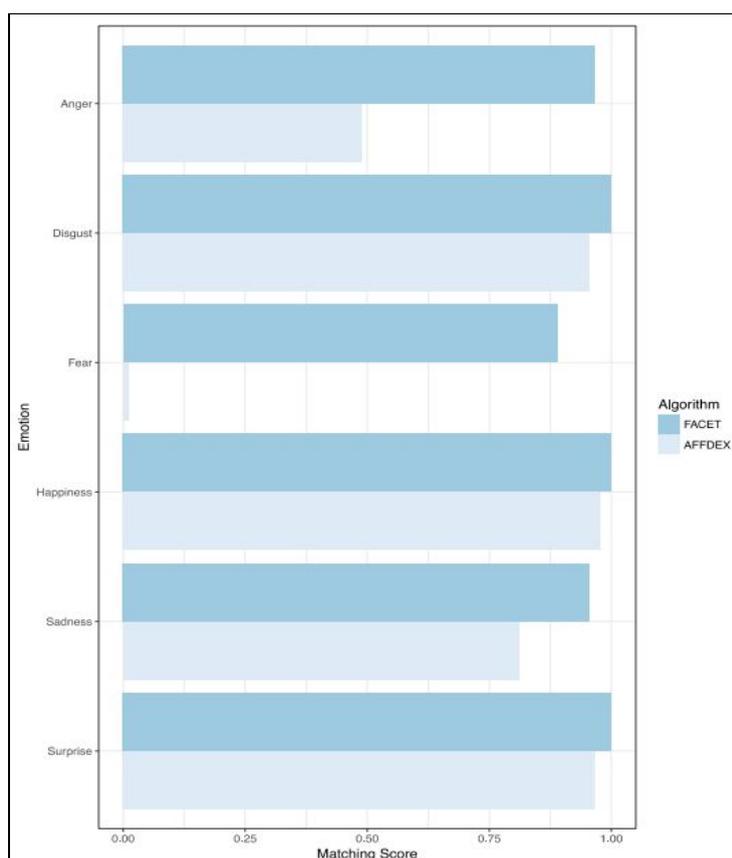


Figura 49: FACET vs. AFFDEX

s'han obtinguts ni en les mateixes condicions, ni amb les mateixes bases de dades, són bastant orientatius d'on està situat el nostre software en relació a ells. Destacar que un cop més la taxa d'encert ha estat superior a Afectiva però no a Emotient, que sembla ser el referent absolut en aquest sector. La figura 48 representa la seva efectivitat en cadascuna de les emocions representades (l'enuig o *contempt* no està representat ja que la base de dades WSEFEP no conté aquesta emoció).

Un cop més, aquesta comparació no és del tot real, però serveix per donar-nos una idea del nivell que ha assolit el nostre software. A les conclusions finals es debatrà aquest tema més en profunditat.

## 7. Sostenibilitat i compromís social

A continuació, es resoldran certes qüestions en relació a la sostenibilitat del projecte en les seves tres dimensions principals. A partir d'aquí, es podran treure les conclusions corresponents. Destacar que aquesta és la primera part de l'anàlisi, a la fita final es tractarà la part restant del tema.

### 7.1. Autoavaluació sobre sostenibilitat

El tema de la sostenibilitat sempre l'he entès com un concepte simple. Si un producte o servei perjudica al medi ambient de qualsevol forma, no es pot considerar sostenible. Per una banda tenia raó, però aquesta és una de les múltiples branques que engloba el concepte de sostenibilitat. Durant l'aprenentatge autònom i l'autoavaluació he pogut veure com hi ha tres grans blocs: ambiental, econòmic i social. A banda de la seva nomenclatura, amb experiències anteriors de projectes i treballs realitzats, he pogut veure com alguns dels elements que cal tractar ja els tenia en compte de forma automàtica, mentre que d'altres ni se m'havien passat pel cap.

Un dels meus punts forts que sempre he tingut és la planificació d'un projecte. Sempre he tingut en compte si era factible o no de realitzar, i aquest és un punt molt important en la sostenibilitat econòmica d'un gran treball. En general, també he sigut capaç de controlar el tema de costos de forma eficient tenint en el cap sempre tots els tipus que existeixen i les amortitzacions. D'altra banda, un punt molt important és també la creativitat, la innovació i la seva afectació en la societat. Aquesta és una part molt important en la sostenibilitat social que també puc considerar com un dels meus punts forts.

En canvi, he pogut veure com molts aspectes com ara l'accessibilitat, l'ergonomia, la seguretat dels productes i projectes TIC, la justícia social, equitat, diversitat i transparència, entre d'altres, els he passat per alt. Això em serveix com a avís i com a manera de posar-me a prova per poder fer un projecte TIC sostenible en tots els aspectes possibles.

## 7.2. Dimensió econòmica

### 7.2.1. Gestió Econòmica

Aquest apartat es dedicarà exclusivament a quantificar el volum de gastos que suposarà la realització del treball de fi de grau. Durant la seva realització caldrà també gestionar de forma òptima el pressupost en els tipus de despeses que s'exposen a continuació.

#### Recursos humans

Seguint les pautes establertes a la planificació, els quatre rols del projecte els farà jo mateix. Aquests són: cap de projecte, dissenyador, programador i tester. A continuació es mostra una taula amb el cost total aproximat juntament amb el corresponent cost per hora i les hores totals que ha de dur a terme cada rol en el projecte.

Rol	Cost/hora	Hores totals	Cost total
Cap de projecte	30€	130h	3.900€
Dissenyador	25€	40h	1.000€
Programador	25€	250h	6.250€
Tester	20€	50h	1.000€
<b>Total</b>	-	<b>470h</b>	<b>12.150€</b>

#### Software i llicències

Gran part del software mencionat a la planificació no comporta cap cost adicional. Els únics que caldria tenir en compte són la llicència de Microsoft Office i de Matlab. La primera té un cost aproximat d'uns 35€ i una vida útil indefinida, però que limitarem a 6 anys per les noves versions que surten i deixen obsoleta a aquesta. En canvi, la llicència de Matlab es proveïda de forma gratuïta pels alumnes de la FIB, per tant no suposa un cost extra. L'amortització té en compte els 4 mesos de duració del TFG. La taula següent ho resumeix.

Producte	Preu	Vida útil	Amortització
Microsoft Office 2010	35€	6 anys	1,94€
Matlab R2017b	0€	-	0€
<b>Total</b>	<b>35€</b>	-	<b>1,94€</b>

## Hardware

L'únic element de hardware que s'utilitzarà és un ordinador portatil ASUS, el qual té una vida útil de 3 anys. L'amortització funciona igual que amb el software.

Producte	Preu	Vida útil	Amortització
Ordinador Portatil ASUS	250€	3 anys	27,78€
<b>Total</b>	<b>250€</b>	-	<b>27,78€</b>

## Altres despeses

Fins ara, hem vist els costos directes del projecte, però també en tenim d'indirectes, com ara són la electricitat a pagar pel funcionament del hardware o la impressió de la memòria. Això queda exposat a continuació.

Producte	Quantitat	Preu per unitat	Preu Total
Impressió memòria	4 unitats	4 €/impressió	16€
Electricitat	200 kWh	0,12 €/kWh	24€
<b>Total</b>	-	-	<b>40€</b>

Per altra banda tenim els costos degut a contingències i imprevistos. Aquesta part cobrirà els possibles riscos explicats anteriorment a la planificació. Com que s'ha assumit que l'afectació dels riscos a la planificació serà mínima, es destinarà un 10% del cost total en aquest sentit.

No hi haurà costos addicionals pels espais de treball explicats a la planificació.

## Pressupost total

La següent taula resumeix totes les despeses mencionades anteriorment.

Tipus de despesa	Cost total
Recursos humans	12.150€
Software	1,94€
Hardware	27,78€

---

Altres despeses	40€
<b>Total sense contingències i imprevistos</b>	<b>12.218,72€</b>
Contingències i imprevistos	1.221,87€
<b>Total</b>	<b>6.440,59€</b>

## Control de gestió

Aquest pressupost inicial pot oscil·lar durant el transcurs del projecte, però en tot cas seria mínim, mai superant el 10% establert anteriorment. Per assegurar-nos de que no ens excedim, el control serà similar a l'explicat al mètode de validació en l'apartat de mètodes de desenvolupament, amb controls setmanals. Si existeixen variacions significatives en un cert punt, es prendran mesures perquè no tornin a succeir.

Cal tenir en compte que l'únic cost que realment pot variar considerablement són els recursos humans. La resta ja són elements que ja tenia en consideració a banda del projecte. Per tant, si hi ha un excés en el número d'hores en la dedicació, quedarà reflexat en els controls setmanals mencionats abans. Si això succeeix, ja tenim els fons destinats a contingències per fer-se càrrec del problema, que com ja s'ha dit abans, hauria de ser més que suficient.

## 7.2.2. Revisió de la planificació i la gestió econòmica

### Progrés, desviacions i conseqüències

Durant el transcurs del desenvolupament del treball han aparegut alguns imprevistos que han obligat a modificar la planificació inicial.

La primera part del projecte, la fita inicial, ha estat completada seguint els terminis previstos. Ja es va deixar clar que el risc era molt baix i que unes dates d'entrega estrictes farien que no s'allargués més del compte. Les hores de dedicació han estat pràcticament les planificades.

En canvi, la segona part del projecte, el desenvolupament del “model funcional bàsic” ha tingut certs problemes. La primera meitat, relacionada amb la detecció facial va anar millor de l'esperat. En una setmana i mitja ja estava enllestit gràcies a l'ús d'una llibreria de Javascript (més detalls en la secció sobre “detecció facial”). Però durant la segona meitat van aparèixer problemes. El desenvolupament del reconeixement d'emocions en sí no va ser el problema ja que va avançar ràpidament. El contratemps va ser que els resultats no eren convincents, i no es podia afirmar que s'havia completat el model funcional bàsic per una manca evident d'encert.

La causa va ser fàcil de detectar, però costosa en temps de solucionar-la. La detecció facial prèviament programada no era consistent per l'ús d'imatges com a base de dades. Una sèrie de problemes explicats amb més deteniment més endavant, van suposar retards inesperats. Gràcies a una bona presa de decisions, aquests retards es van poder controlar minimitzant l'impacte sobre la planificació inicial tot el possible. En total, la tasca “model funcional bàsic” es va allargar una setmana i mitja, però es va poder completar amb èxit.

Tot això va tenir conseqüències evidents en la tercera tasca, encarregada de complir amb el tercer i darrer objectiu del projecte. Com que la data final d'entrega del treball és fixa, calia reduir la dedicació a optimitzar el software. En lloc de tenir 8 setmanes de dedicació, en va tenir 6 i mitja. Durant aquestes setmanes, s'han dissenyat, implementat i testejat una sèrie de mètodes propis per millorar el reconeixement d'emocions. La taxa d'encert va augmentant, el que significa que es progressa de forma adequada. Aquesta ha estat la tònica durant totes aquestes setmanes fins arribar a les darreres dues setmanes de la planificació.

La fita final, la redacció de la memòria, es va mantenir en les dues últimes setmanes de la planificació, el que significa que s'han complert els terminis del treball.

Un resum de tot el que s'ha explicat es pot veure en el diagrama de Gantt que hi ha a continuació. La comparativa amb l'inicial es pot realitzar amb el presentat al capítol 3.

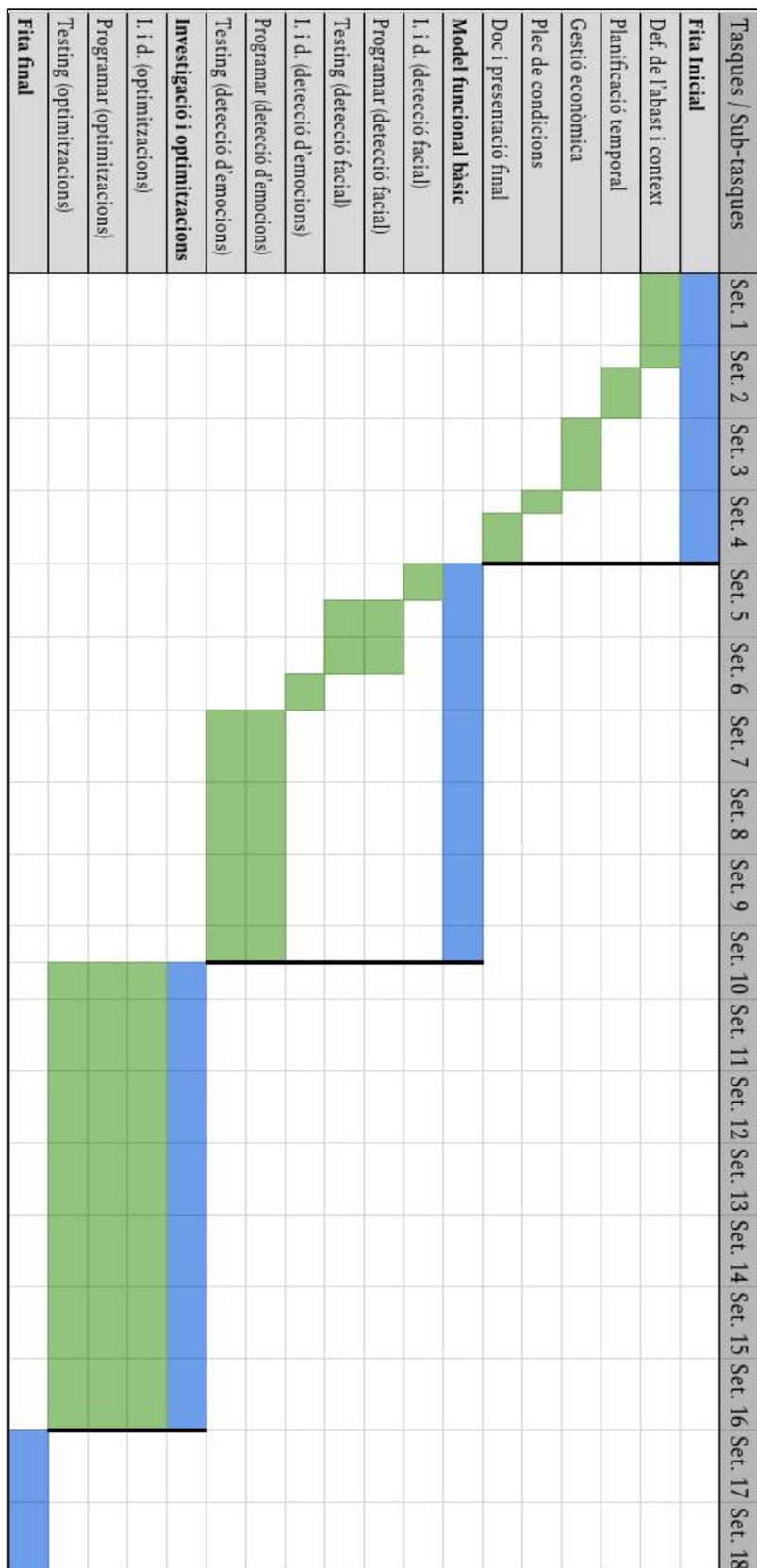


Figura 50: Diagrama de Gantt final

## Abast

L'abast del projecte s'ha limitat al seu objectiu principal, que era la consecució de tots tres objectius tenint un software de reconeixement d'emocions prou optimitzat. Les altres funcionalitats, com ara el reconeixement d'emocions secundàries no s'ha dut a terme. En canvi, el fet de reconèixer una emoció en concret, encara que no sigui una funcionalitat específica del software, sempre es pot consultar aquesta informació en particular en tot moment.

Els principals obstacles que es podien trobar ja s'han destacat en el resum del progrés de l'apartat anterior, encara que s'analitzaran en més detall en el treball més endavant quan correspongui.

## Dedicació i costos

Tot l'explicat anteriorment ha desencadenat una sèrie d'alteracions a les hores de dedicació previstes del projecte. Aquest és l'estat final:

Tasca a realitzar	Hores inicials	Hores definitives
Fita Inicial	80	80
Model funcional bàsic	100	140
Investigació i optimitzacions	240	190
Fita final	50	50
<b>Treball de Fi de Grau</b>	<b>470</b>	<b>460</b>

Evidentment, aquests canvis tenen una afectació en els costos de desenvolupament. Concretament, els afectats és el pressupost dedicat als recursos humans, ja que els diferents rols tindran uns números d'hores de treball diferents. Quedarà així:

Rol	Cost/hora	Hores totals	Cost total
Cap de projecte	30€	130h → 130h	3.900€ → 3.900€
Dissenyador	25€	40h → 30h	1.000€ → 750€
Programador	25€	250h → 250h	6.250€ → 6.250€
Tester	20€	50h → 50h	1.000€ → 1.000€
<b>Total</b>	-	<b>470h → 460h</b>	<b>12.150€ → 11.900€</b>

## Metodologia i rigor

La metodologia que s'ha estat utilitzant és el model de desenvolupament en cascada inicialment proposat. S'adapta perfectament al desenvolupament d'aquest treball, que estableix unes dependències molt clares entre tasques. Això ja s'ha pogut veure amb els inconvenients explicats a la planificació, on problemes amb una tasca han obligat a postpondre l'inici de la següent.

Les eines de seguiment, el mètode de validació i les eines de treball són les mateixes que es van explicar a la planificació inicial i s'han mantingut fins al final del desenvolupament del projecte. Veient que han estat la forma adequada de progressar tot i els inconvenients ja explicats, no han existit raons per canviar-ne cap dels mètodes o eines.

## Anàlisi d'alternatives

L'anàlisi d'alternatives per resoldre el problema és l'essència de la tercera tasca explicada a la planificació. L'objectiu és utilitzar diferents mètodes per obtenir un software més precís, amb una millor taxa d'encerts. Per això, en aquest sentit, el projecte ha tingut en compte un rang d'alternatives per resoldre el problema. Evidentment, aquest rang només conté les opcions que tinguin una possibilitat de donar resultats positius. Les estrictes dates que s'han de complir ho limiten. Les opcions adoptades al final del projecte seran les que donin millors resultats. Tots els detalls s'expliquen a les seccions pròpies del desenvolupament del treball i del software.

### 7.2.3. Reflexió sobre la sostenibilitat

• *Respecte el Projecte posat en producció (PPP): Reflexió sobre el cost que has estimat per la realització del projecte?*

El cost proposat pel projecte és realment baix. Tenint en compte que gran part té el seu origen als recursos humans, part inevitable si es vol desenvolupar un projecte. L'única forma de reduir costos seria baixar el nombre d'hores de treball, cosa realment complicada degut a la magnitud del treball.

• *Respecte el Projecte posat en producció (PPP): Has quantificat el cost (recursos humans i materials) de la realització del projecte?*

Sí, està tot detallat a l'apartat de Gestió econòmica.

• *Quines decisions has pres per reduir el cost?*

Gran part de cost està relacionat amb els Recursos Humans, per tant, els estalvis més significatius estan lligats amb les hores de dedicació del projecte. No hi ha decisions explícites que baixin aquest número d'hores si no les situacions amb què et trobes durant la conducció del projecte. Finalment, es pot veure en la planificació definitiva com el nombre d'hores s'ha reduït lleugerament, per tant hi hagut un estalvi en aquest sentit.

• *Has quantificat aquest estalvi?*

Sí, com es pot veure de forma més detallada a la Gestió econòmica, l'estalvi ha estat de 250€.

• *Respecte el Projecte posat en producció (PPP): S'ha ajustat el cost previ al cost final?*

Sense tenir en compte l'estalvi comentat anteriorment, el cost final s'ha ajustat molt bé al cost inicialment calculat.

• *Has justificat les diferències (leccions apreses)?*

No hi ha hagut diferències notables a banda de l'estalvi mencionat anteriorment.

• *Respecte la Vida Útil: Com es resolen actualment els aspectes de costos del problema que vols abordar (estat de l'art)?*

Els pressupostos d'altres empreses o desenvolupadors de software similars són molt més elevats que el proposat aquí. Per tant és força competitiu en aquest sentit.

• *En què millorarà econòmicament (costos...) la teva solució respecte a les existents?*

No hi haurà una millora aparent.

- *Respecte la Vida Útil: Quin cost estimes que tindrà el projecte durant la seva vida útil?*

És difícil estimar el cost que pot arribar a tenir el projecte durant la seva vida útil. El seu manteniment no serà especialment costós, però si ho podria ser una possible inversió per realitzar noves optimitzacions. En aquest cas s'hauria de calcular un pressupost en funció de les hores de dedicació per aquestes tasques.

- *Es podria reduir aquest cost per fer-lo més viable?*

Donat que el cost de manteniment serà baix i probablement invariable i que les possibles optimitzacions són només hipòtesis, sembla complicat reduir costos de forma aparent.

- *Respecte la Vida Útil: S'ha tingut en compte el cost dels ajustos/actualitzacions/reparacions durant la vida útil del projecte?*

En l'apartat de Gestió econòmica només s'han considerat els gastos relacionats amb el desenvolupament del projecte. Les actualitzacions o reparacions que puguin tenir lloc més endavant no s'han quantificat, però si s'han tingut en consideració en les preguntes anteriors.

- *Respecte els Riscos: Podrien produir-se escenaris que perjudiquessin la viabilitat del projecte?*

Econòmicament parlant, sembla difícil que un petit pressupost de manteniment del software pugui posar en perill la seva viabilitat. Altres situacions hipotètiques com les presentades anteriorment podrien suposar més problemes i s'haurien d'estudiar el pressupost amb deteniment.

### 7.3. Dimensió ambiental

- *Respecte el Projecte posat en producció (PPP): ¿Has estimat l'impacte ambiental que tindrà la realització del projecte?*

L'impacte ambiental serà mínim. L'únic element a tenir en consideració seria el consum d'electricitat necessari per utilitzar el hardware descrit, i és realment baix considerant els 4 mesos de desenvolupament del treball.

- *T'has plantejat minimitzar-ne el impacte, per exemple, reutilitzant recursos?*

Gran part dels recursos, principalment hardware i software són elements que ja tenia en possessió abans de començar el projecte. L'únic element que podria tenir en consideració és una reducció del consum elèctric (poc probable) o una impressió de la memòria amb paper reciclat o similar.

- *Respecte el Projecte posat en producció (PPP): Has quantificat l'impacte ambiental de la realització del projecte?*

L'únic impacte ambiental que ha tingut la realització del projecte seria el consum elèctric produït per l'ordinador portàtil durant les hores de treball. Aquest cost ja ha estat quantificat en kWh en la Gestió econòmica.

- *Quines mesures has pres per reduir l'impacte?*

Deixant de banda l'impacte inicialment petit que ja tenia el treball, reduir l'impacte seria una conseqüència de disminuir el nombre d'hores que s'està fent ús de l'ordinador. Com que una enorme part del projecte implica la seva utilització, és difícil prendre mesures per consumir menys electricitat.

- *Has quantificat aquesta reducció?*

Quantificar una hipotètica reducció en el consum és pràcticament impossible, ja que no es pot saber amb exactitud la quantitat que ha consumit un únic aparell elèctric durant aquest temps.

- *Respecte el Projecte posat en producció (PPP): Si fessis el projecte de nou, el podries realitzar amb menys recursos?*

Difícilment, ja que l'únic recurs material utilitzat ha estat el meu ordinador personal, i sense ell no hauria pogut programar el software ni escrit aquesta memòria.

- *Respecte la Vida Útil: Com es resol actualment el problema que vols abordar (estat de l'art)?, i. ¿En què millorarà ambientalment la teva solució respecte a les existents?*

Aquest no és un producte que afecti el medi ambient ja que es software únicament. En comparació a solucions existents no hi haurà una diferència rellevant.

- *Respecte la Vida Útil: Quins recursos estimes que s'utilitzaran durant la vida útil del projecte?*

Principalment ordinadors per poder fer-lo servir i possiblement algun servidor si s'adapta en un futur proper.

- *Quin serà l'impacte ambiental d'aquests recursos?*

El seu consum elèctric. Difícil de quantificar ja que estarà en relació a l'ús que se'n faci.

- *Respecte la Vida Útil: El projecte permetrà reduir l'ús d'altres recursos?*

No és el seu objectiu.

- *Globalment, l'ús del projecte millorarà o empitjorarà l'empremta ecològica?*

Cap de les dues, no és el seu objectiu.

- *Respecte els Riscos: Podrien produir-se escenaris que fessin augmentar l'empremta ecològica del projecte?*

No s'esperen.

---

## 7.4. Dimensió social

- *Respecte el Projecte posat en producció (PPP): Què creus que t'aportarà a nivell personal la realització d'aquest projecte?*

A nivell personal significa una cerca de la perfecció. És a dir, un projecte amb el que mai podré dir que estic satisfet al 100% perquè encara es pot millorar més. Evidentment, uns resultats realment positius o millors que la competència serien molt satisfactoris.

- *Respecte el Projecte posat en producció (PPP): La realització d'aquest projecte ha implicat reflexions significatives a nivell personal, professional o ètic de les persones que hi han intervingut?*

Les reflexions més significatives han estat en relació al procés de disseny del treball i la seva execució. A nivell personal o ètic no se'n poden destacar realment.

- *Respecte la Vida Útil: Com es resol actualment el problema que vols abordar (estat de l'art)? i. ¿En què millorarà socialment (qualitat de vida) la teva solució respecte les existents?*

Les solucions actuals ja poden resoldre el problema amb molt bones taxes d'encert, però en aquest projecte s'investigaran i aplicaran un ventall de tècniques i algoritmes per intentar superar els resultats actuals. La millora social serà gairabé inapreciable, però si es millora lleugerament la eficàcia del software, aquesta millora existirà.

- *Respecte la Vida Útil: Existeix una necessitat real del projecte?*

No existeix una necessitat real del projecte. Ja existeixen productes amb una bona taxa d'encerts.

- *Respecte la Vida Útil: Qui es beneficiarà de l'ús del projecte?*

Existeixen una sèrie de grups que es poden beneficiar amb aquest projecte. Primerament podem pensar en entitats que vulguin conèixer la reacció dels seus clients davant d'un dels seus productes. També es podria pensar en altres persones que estiguin desenvolupant un software similar i vulguin utilitzar-lo per implementar noves idees o millores addicionals. Aquest projecte també pot tenir com a finalitat un ús més casual, en el que una persona el volgués utilitzar sense cap implicació en el seu desenvolupament o ús comercial.

- *Hi ha algun col·lectiu que es pugui veure perjudicat pel teu projecte? En quina mesura?*

El col·lectiu més vulnerable a un mal ús del projecte serien els usuaris que en fan un ús més comú de l'aplicació. Cal recordar que encara que aquest projecte no tracta amb la identitat de les persones que l'utilitzen, s'estan enregistrant les imatges i existeix la probabilitat que algú en pogués fer un mal ús.

- *Respecte la Vida Útil: En quina mesura soluciona el projecte el problema plantejat inicialment?*

El soluciona de forma molt satisfactòria. Més detalls es podran trobar més endavant.

- *Respecte els Riscos: Podrien produir-se escenaris que fessin que el projecte fos perjudicial per algún segment particular de la població?*

Com s'ha mencionat anteriorment, existeix el risc que es pugui fer un mal ús de l'aplicació pel fet que s'enregistren les cares de les persones que l'utilitzen. Sí es garanteix però, la protecció d'identitat ja que en cap moment es demana informació personal.

- *Respecte els Riscos: Podria crear el projecte algun tipus de dependència que deixes als usuaris en posició de debilitat?*

Com ja s'ha dit abans, la informació personal dels usuaris no es requereix, l'únic element enregistrat és la seva cara.

---

## 8. Conclusions

Les primeres conclusions que es poden treure d'aquest treball és que els seus resultats han estat molt positius. Obtenir taxes d'encert superiors al 90% ha estat molt satisfactori després de tota la investigació i el disseny de diferents mètodes per resoldre el problema inicial. Han aparegut alguns problemes durant la consecució dels diversos objectius, principalment amb el segon, on ha costat més de l'esperat obtenir un software amb un encert mínimament acceptable. El fet de trobar-se amb diverses característiques que no eren suficient per diferenciar emocions va ser un petit inconvenient. Però la problemàtica principal ha estat el haver d'adaptar el software per poder tractar vídeos. El fet de no obtenir bons resultats a la detecció de cares era un problema que s'havia de solucionar el més aviat possible. Per sort, tot s'ha pogut solucionar en uns terminis acceptables que no han afectat greument al desenvolupament del projecte.

Deixant de banda el desenvolupament del software, hi ha altres factors que cal destacar. Més concretament a l'anàlisi de resultats s'ha comparat el nostre software amb altres que hi ha al mercat que tenen funcionalitats similars. Aquestes comparacions són una mica enganyoses pel fet que la base de dades que s'ha tractat en el projecte estava limitada per diverses restriccions. Aquestes restriccions, ja descrites en capítols anteriors, fan al·lusió a la posició de la persona en la càmera, la il·luminació, etc. Els softwares del mercat estan dissenyats per poder donar respostes en condicions no tan òptimes com les treballades aquí. És a dir, són molt més flexibles amb situacions més casuals de la vida quotidiana. És natural pensar que els seus resultats estan condicionats per aquestes dificultats a les que el nostre software no s'ha enfrontat.

No obstant, els resultats obtinguts segueixen sent meritoris ja que l'objectiu del treball ha estat sempre la detecció d'emocions. El poder captar la cara d'una persona i les seves característiques es podria aconseguir amb un preprocessament més elaborat de les imatges que no formava part del projecte en cap moment.

El que podem aprendre amb el desenvolupament d'aquest projecte i amb el que s'ha discutit fins ara és la capacitat d'afirmar que aquest és un problema molt difícil de resoldre. No difícil a l'hora de crear un simple software mínimament funcional que realitzi una classificació decent, si no a l'hora de fer un programa infalible. En moltes altres disciplines, aconseguir sistemes que garanteixin resultats impecables és factible, però no és el cas en el reconeixement d'emocions. Observant productes com l'oferit per Emotient, el qual té un treball enorme al darrera i enormes bases de dades, no és capaç d'arribar a un encert del 100%, ens adonem de la dificultat del problema. Aquesta és una de les grans desavantatges que té qualsevol problema que requereix lògica difusa com a tècnica per resoldre'l. Per tant, encara es pot dir que el reconeixement d'emocions és encara un camp en expansió, i que encara hi ha marge de millora. Aquest projecte ha tractat de proposar idees de com dissenyar solucions, mesclant tècniques ja existent amb altres més creatives. Aquest és el camí a seguir per seguir millorant i perfilant el coneixement que pot arribar a tenir una màquina.

## 8.1. Treball futur

El treball realitzar durant aquest projecte ha estat molt destacable, però el seu potencial encara no acaba aquí. Principalment per culpa de la manca de temps, hi ha elements o idees que no s'han pogut arribar a implementar o fins i tot a tenir en consideració. Això no significa, que en un futur no es puguin realitzar per millorar la qualitat d'aquests quatre mesos de treball.

Probablement, el primer que caldria fer sobre la estructura general del software és unificar-lo. Fins ara tenim dues parts ben diferenciades, la detecció de cares i el reconeixement d'emocions. Cadascuna d'elles està programada amb un llenguatge de programació ben diferent, de forma justificada com s'ha explicat abans, però al cap i a la fi, difícilment compatibles. Una solució seria reprogramar una de les parts del codi, preferentment la detecció de cares, en Matlab. Tenint en compte que Matlab és un llenguatge que dona moltes facilitats a l'hora de tractar amb imatges, no seria una mala idea. Recordar que la detecció de cares va ser programada amb Javascript per fer ús de la llibreria *clmtrackr*. Calia realitzar aquesta part amb un curt espai de temps i havia de proporcionar una taxa d'encert molt elevada per garantir un bon funcionament de la resta del projecte. Llavors, si ens oblidem de la limitació de temps, fer aquesta unificació és una tasca factible per fer en un futur proper.

Seguidament, si la idea d'unificar el codi del software s'arriba a aplicar, el següent pas seria adaptar-lo per poder fer-ne un ús més casual. Essencialment, el que caldria millorar seria la interacció amb l'usuari, ja que fins ara, la part del reconeixement d'emocions és molt poc adient. Afortunadament, existeixen maneres de transportar un programa escrit en Matlab a una interfície web. Situant el resultat en un servidor, facilitaria molt l'accés als usuaris i amb un disseny intuïtiu, faria molt més senzill el seu ús. El resultat podria ser fàcilment similar al vist en la part del software de detecció de cares, que ja està programat directament en HTML.

Deixant de banda la part més dedicada a l'usuari, cal tornar a mirar a la part més tècnica del software. Com ja s'ha explicat diverses vegades durant la memòria, el dataset utilitzat per aquest projecte era ideal, però no totes les situacions ho són. Les condicions d'il·luminació, la posició de la persona (o persones) a la imatge, entre altres, no sempre seran les que ens hem trobat a RAVDESS. Per això, caldria dissenyar i implementar una nova part del software, que seria previa a tot el que s'ha fet fins ara, un preprocessament d'imatges. Mitjançant un ventall de tècniques diferents, es pot manipular una imatge per tal que el software de detecció de cares sigui capaç de veure on hi ha una persona. Com a conseqüència de la "simplicitat" amb la que s'ha dissenyat el nostre software a la planificació inicial, aquest no és capaç de detectar cares amb una bona taxa d'encert en condicions no òptimes. La millor part de realitzar aquesta millora és el fet que no altera la resta del codi, ja que seria un mòdul a part que s'executaria abans que la resta. Això també podria suposar una millora opcional, com s'ha intuït abans, que seria el fet de realitzar el reconeixement d'emocions a totes les persones detectades en la imatge, no només a una.

Finalment, la darrera tasca a tenir en consideració en el futur és la de seguir buscant noves estratègies per assolir millors taxes d'encert. Un altre cop per culpa del límit de temps, el temps concedit a la investigació i implementació de noves tècniques per optimitzar el software era molt limitat. Evidentment, els resultats obtinguts en el projecte ja poden ser considerats com a molt satisfactoris, però després de l'aplicació d'un preprocessament d'imatges i de la utilització d'altres bases de dades, potser ja no ho són tant. Per això, investigar noves formes de calcular característiques o considerar altres formes de classificar-les és una tasca pel futur.

## 9. Annex

### 9.1. Documentació del software

La carpeta del software està composta de 5 subcarpetes (build, models, src, emotion recognition, face detection). Les tres primeres fan referència a la llibreria de Javascript utilitzada en el treball, *clmtrackr*. Les altres dos contenen els elements pròpiament desenvolupats pel treball i s'explicaran amb més deteniment a continuació.

#### Detecció de cares

La part dedicada a la detecció de cares està continguda a la carpeta “face detection”. En ella podem trobar dos fitxers HTML.

##### *face\_detect\_image.html*

Aquest fitxer pot realitzar la detecció de cares en imatges. Per tal de fer-lo servir cal obrir-lo en un servidor amb un navegador qualsevol, com ara Google Chrome. Si l'obrim podem veure dues parts ven diferents. A l'esquerra hi ha un “canvas” on apareix una imatge de mostra i on es mostraran les imatges que s'analitzin. A la dreta hi ha un petit menú amb un sèrie d'elements que permeten manipular les opcions del detector. Aquestes són:

- Nom imatge(s): el nom de la imatges o grup d'imatges a analitzar. Aquestes s'han de trobar a la carpeta “imatges” dins la carpeta “media”, inclosa en el mateix nivell on està aquest mateix arxiu. Si no la trova, l'anàlisi no començarà.
- Número d'imatges: quantitat d'imatges a analitzar. Per fer que el programa reconeixi les imatges han de tenir el nom i format següent: “nom\_imatge (número).jpeg”. Els números comencen en 1 i acaben en el número entrat en aquest camp.
- Descarregar característiques: si s'activa, es descarregarà un fitxer .txt amb les coordenades de tots els punts del model per cada imatge.
- Nom fitxer(s): si s'ha activat la descàrrega de característiques, aquest serà el nom del fitxer descarregat. Si n'hi ha més de un, els seus noms tindran el format següent: “nom\_fitxer (número).txt”.
- Iniciar: comença la detecció de cares amb la informació entrada. Si hi ha camps obligatoris buits, sortirà un error per pantalla.

##### *face\_detect\_video.html*

Aquest fitxer pot realitzar la detecció de cares en vídeos. Estructura molt similar al fitxer anterior i mateixos requeriments per fer-lo funcionar. Les opcions del menú són:

- Nom del vídeo: el nom del vídeo en format .mp4, on l'anirà a buscar a la carpeta “videos”, també dins de la carpeta “media”.
- Carregar vídeo: carrega el vídeo en el canvas. Necessari per realitzar el seu anàlisi.

- Descarregar característiques i Nom fitxer(s): igual que en *face\_detect\_imatge.html*. L'únic element a tenir en compte és el fet que descarrega fitxers a una freqüència de 5 per segon.
- Iniciar: comença la detecció de cares amb la informació entrada. Si hi ha camps obligatoris buits, sortirà un error per pantalla.

## Reconeixement d'emocions

La part dedicada al reconeixement d'emocions està continguda a la carpeta “emotion recognition”. En ella podrem trobar fins a set fitxers de Matlab.

### *entrada\_learn.m*

Fitxer encarregat de carregar la entrada pel procés de *learning* del software. En ell es pot manipular la base de dades que volem utilitzar (imatges o vídeos), el nombre de fitxers que es carregaran per emoció, el path fins a la carpeta “media” i el nom de les carpetes/emocions.

El resultat seran dues matrius de 4 dimensions amb les coordenades de cada imatge de cada emoció. Una d'elles contindrà els valors tal qual i l'altre normalitzats. També s'obtindrà la informació del rectangle on està continguda la imatge, que serà d'utilitat més endavant.

### *learning.m*

Part del software encarregada de generar el classificador i les característiques. En aquest fitxer podem trobar 8 seccions diferents, que corresponen als 8 mètodes de càlcul de característiques explicats al treball. El classificador utilitzat per cadascun dels mètodes és el TreeBagger, però en el mètode “Angles combinats sense contorn + HOG” es pot escollir qualsevol de les altres alternatives presentades.

### *load\_predictors.m*

Un cop realitzada la fase de learning, no cal generar el classificador diverses vegades si ja l'hem guardat anteriorment (Matlab disposa de la funcionalitat de guardar variables del workspace). En aquest fitxer es podran carregar 3 classificadors, corresponent als que s'han analitzat en el treball amb les característiques “Angles combinats sense contorn + HOG”. Desgraciadament, per limitacions d'espai, l'únic classificador disponible és el TreeBagger (el que té una millor taxa d'incert). Aquest es pot trobar a la carpeta “predictors”.

### *entrada\_test.m*

Fitxer practicament idèntic a *entrada\_learn.m*, però adaptat per poder llegir la part de la base de dades de *testing* i guardar la seva informació en unes matrius diferents a les de l'aprenentatge.

*testing.m*

Fitxer molt similar a *learning.m* pel fet que apareixen els mateixos mètodes de càlcul de característiques. En canvi, ara s'utilitzaran per cridar al predictor i no per generar el classificador. D'aquesta manera, obtindrem els resultats en funció del mètode que utilitzem.

Nota: cal utilitzar el mateix mètode que s'ha fet servir per l'aprenentatge, sino el programa donarà error o no s'obtidran els resultats desitjats.

*filters.m*

Petit fitxer encarregat de filtrar les dades obtingudes a la fase de *testing*. En ell es pot escollir entre els sis tipus de filtres presentats en el treball.

*output.m*

Fitxer encarregat de mostrar els resultats obtinguts. Aquests es presentaran de tres formes diferents. Per recordar els detalls, veure la secció "5.2. Format dels resultats".

Nota: aquest fixter dona l'opció de presentar els resultats amb filtre o sense. Si no s'han filtrat les dades i es demana una representació filtrada dels resultats, donarà error.

---

## 10. Bibliografia

### 10.1. Referències

[1] FaceFirst, Jesse Davis West. August 01, 2017. *A brief history of face recognition*. (Consulta: Setembre 2018)

URL: <https://www.facefirst.com/blog/brief-history-of-face-recognition-software/>

[2] Apple Developer, Core Programming Guide. *Detecting Faces in an Image*. (Consulta: Setembre 2018)

URL: [https://developer.apple.com/library/archive/documentation/GraphicsImaging/Conceptual/CoreImaging/ci\\_detect\\_faces/ci\\_detect\\_faces.html](https://developer.apple.com/library/archive/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html)

[3] Hansoninc, Josh Jacob. June 01, 2018. *Mobile Vision, Part 1: Face Detection*. (Consulta: Setembre 2018)

URL: <https://www.hansoninc.com/mobile-vision-part-1-face-detection/>

[4] MathWorks, Documentation. *What is Fuzzy Logic?* (Consulta: Setembre 2018)

URL: <https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html>

[5] SAVRAN, Arman et al. *Emotion Detection in the Loop from Brain Signals and Facial Images*. In: *Proceedings of the eNTERFACE 2006 Workshop*. Dubrovnik. (Consulta: Setembre 2018)

URL: <https://archive-ouverte.unige.ch/unige:47926>

[6] Zeng, Z., Fu, Y., Roisman, G.I., Wen, Z., Hu, Y., & Huang, T.S. (2006). *Spontaneous Emotional Facial Expression Detection*. *Journal of Multimedia*, 1, 1-8. (Consulta: Setembre 2018)

URL: <https://pdfs.semanticscholar.org/a603/5473e288015e8c67062fd6a27ae6c7f1e611.pdf>

[7] P. Pal, A. N. Iyer and R. E. Yantorno, "Emotion Detection From Infant Facial Expressions And Cries," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Toulouse, 2006, pp. II-II. doi: 10.1109/ICASSP.2006.1660444. (Consulta: Setembre 2018)

URL: <https://ieeexplore.ieee.org/abstract/document/1660444/>

[8] Microsoft Azure, *Emotion*. (Consulta: Setembre 2018)

URL: <https://azure.microsoft.com/es-es/services/cognitive-services/emotion/>

[9] auduno, GitHub repository, (2018), *clmtrackr*. (Consulta: 09/12/2018)

URL: <https://github.com/auduno/clmtrackr/>

[10] 7color94, Github, (2017-02-18), *Constrained Local Model*. (Consulta: Octubre 2018)

URL: <https://7color94.github.io/2017/02/constrained-local-model/>

- 
- [11] Saragih, J.M., Lucey, S. & Cohn, J.F. Int J Comput Vis (2011) 91: 200. *Deformable Model Fitting by Regularized Landmark Mean-Shift*. (Consulta: Octubre 2018)  
URL: [https://ci2cv.net/media/papers/2011\\_IJCV\\_Saragih.pdf](https://ci2cv.net/media/papers/2011_IJCV_Saragih.pdf)
- [12] Towards Data Science, Rohith Gandhi. Jun 13, 2018. *K Nearest Neighbours—Introduction to Machine Learning Algorithms*. (Consulta: Desembre 2018)  
URL: <https://towardsdatascience.com/k-nearest-neighbours-introduction-to-machine-learning-algorithms-18e7ce3d802a>
- [13] uc3m (Departamento de Estadística), Tema 6: *Análisis discriminante*. (Consulta: Desembre 2018)  
URL: <http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/AMult/tema6am.pdf>
- [14] Mathworks, Documentation. *TreeBagger*. (Consulta: Desembre 2018)  
URL: <https://es.mathworks.com/help/stats/TreeBagger.html?lang=en>
- [15] Livingstone SR, Russo FA (2018) *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English*. PLoS ONE 13(5): e0196391. (Consulta: 09/12/2018)  
URL: <https://zenodo.org/record/1188976#.XAvgkFVKjIW>
- [16] Discuss, jinhyuncheong. Mar 13, 2018. *Face analysis software comparison: Affective (AFFDEX) vs OpenFace vs Emotient (FACET)*. (Consulta: Gener 2019)  
URL: <http://jinhyuncheong.com/jekyll/update/2018/03/13/Face-analysis-software-comparison-revisited.html>
- [17] Stöckli, Sabrina & Schulte-Mecklenbeck, Michael & Borer, Stefan & Samson, Andrea. (2017). Facial expression analysis with AFFDEX and FACET: A validation study. Behavior Research Methods. 50. 10.3758/s13428-017-0996-1. (Consulta: Gener 2019)  
URL: <http://www.schulte-mecklenbeck.com/wp-content/uploads/2017/12/Stockli2017.pdf>

## 10.2. Figures

Figura 1:

[https://developer.apple.com/library/archive/documentation/GraphicsImaging/Conceptual/CoreImaging/art/face\\_detection\\_2x.png](https://developer.apple.com/library/archive/documentation/GraphicsImaging/Conceptual/CoreImaging/art/face_detection_2x.png)

Figura 2:

<https://www.cds.caltech.edu/~murray/wiki/images/d/d5/Fuzzylogic.jpg>

Figura 3:

<https://c.slashgear.com/wp-content/uploads/2016/01/emotioenaag.jpg>

Figura 4: Diagrama realitzat amb Excel

Figura 5:

[https://auduno.github.io/clmtrackr/examples/media/facemodel\\_numbering\\_new.png](https://auduno.github.io/clmtrackr/examples/media/facemodel_numbering_new.png)

Figura 6:

<https://7color94.github.io/2017/02/constrained-local-model/>

Figures 7 i 8: Imatges modificades de la base de dades RAVDESS

Figures 9 i 10:

<https://www.slideshare.net/xavigiro/image-classification-dlcv-d112>

Figura 11:

[https://www.mathworks.com/help/examples/stats/win64/TrainEnsembleOfBaggedClassificationTreesExample\\_01.png](https://www.mathworks.com/help/examples/stats/win64/TrainEnsembleOfBaggedClassificationTreesExample_01.png)

Figures 12~25: Imatges generades pel software del projecte

Figura 26:

[https://es.mathworks.com/help/examples/vision/win64/ExtractAndDisplayImageWithHOGFeaturesUsingCellSizeExample\\_01.png](https://es.mathworks.com/help/examples/vision/win64/ExtractAndDisplayImageWithHOGFeaturesUsingCellSizeExample_01.png)

Figura 27: Imatges modificades de la base de dades RAVDESS

Figures 28~45: Imatges generades pel software del projecte

Figures 46 i 47: Imatges extretes de la base de dades RAVDESS

Figura 48:

<http://jinhyuncheong.com/jekyll/update/2018/03/13/Face-analysis-software-comparison-revisited.html>

Figura 49:

<http://www.schulte-mecklenbeck.com/wp-content/uploads/2017/12/Stockli2017.pdf>

Figura 50: Diagrama realitzat amb excel