



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Desenvolupament d'un sistema multiagent de detecció d'emocions en un robot bípede**

**TREBALL FI DE GRAU**

Grau en Enginyeria Informàtica

*Autor:* Christian Verdú Mateu

*Tutor:* Vicente Julián Inglada  
Carlos Carrascosa Casamayor

Curs 2015-2016



# Resum

L'habilitat d'identificar les expressions facials de les persones es trivial per als éssers humans. Per aquest motiu, un sistema automàtic amb aquesta capacitat seria un gran pas en el camí cap a la interacció persona-màquina. En aquest treball es descriu una aproximació per a intentar resoldre aquest problema. A més a més, el sistema deu poder-se utilitzar per un robot així com poder interactuar amb un sistema multiagent (MAS). El sistema utilitza un algorisme que es basa, primerament, en l'extracció de punts característics de les imatges facials. Mitjançant aquests punts es calculen certes distàncies preestablertes. Aquestes distàncies seran usades per a entrenar models estadístics, tals com: k-veïns més propers, xarxes neuronals artificials, màquines de vectors de suport i arbres aleatoris. A més a més, es formalitzen, avaluen i comparen tres aproximacions, cadascuna millorant l'anterior, que permeten el reconeixement automàtic d'emocions a partir d'imatges facials.

**Paraules clau:** aprenentatge, automàtic, reconeixement, expressió, facial, processament, imatges, svm, ann, mlp, mas, multiagent

---

# Resumen

La habilidad de identificar las expresiones faciales de las personas es trivial para los seres humanos. Por este motivo, un sistema automático con esta capacidad sería un gran paso en el camino hacia la interacción persona-máquina. En este trabajo se describe una aproximación para intentar resolver este problema. Además, el sistema debe poderse utilizar para un robot así como poder interactuar con una sistema multiagente (MAS). El sistema usa un algoritmo que se basa, primeramente, en la extracción de puntos característicos de las imágenes faciales. Mediante estos puntos se calculan ciertas distancias preestablecidas. Estas distancias serán usadas para entrenar modelos estadísticos, tales como: k-vecinos más cercanos, redes neuronales artificiales, máquinas de vectores de soporte y árboles aleatorios. Además, se formalizan, evalúan y comparan tres aproximaciones, cada una mejorando la anterior, que permiten el reconocimiento automático de emociones a partir de imágenes faciales.

**Palabras clave:** aprendizaje, automático, reconocimiento, expresión, facial, procesamiento, imágenes, svm, ann, mlp, mas, multiagente

---

# Abstract

The ability to identify facial expressions of people are trivial for humans. Therefore, an automated system with this capability would be a great step in the path to human-computer interaction. This paper describes an approach in order to solve this problem. In addition, the system must be able to be used by a robot as well as to interact with a multi-agent system (MAS). The system use an algorithm which is based primarily on extracting feature points of facial images. These points will be used in order to calculate certain preset distances. These distances will be used to train statistical models, such as k-nearest neighbors, neural networks, support vector machines and random trees. In addition, it will be described, it will be evaluated and it will be compared three approaches, each one improving the predecessor, in order to allow the automatic recognition of emotions from facial images.

**Key words:** machine, learning, emotion, face, expression, recognition, image, processing, svm, ann, mlp, mas, multi-agent

---



# Agraiments

---

*«Se'm considera un dels treballadors més dedicats i si el pensament és un equivalent de la labor potser ho sóc, doncs a ell he consagrat quasi totes les meues hores de vigília. Però si el treball s'interpreta com un rendiment determinat durant un temps específic d'acord amb una regla rígida, llavors pot ser que jo haja sigut el pitjor dels ganduls» - Nikola Tesla.*

En primer lloc voldria agrair als meus directors del projecte Vicente i Carlos pel suport, interès i el treball que han realitzat per a poder dur a terme aquest projecte.

Als meus companys de carrera que, durant aquests últims quatre anys m'han alegrat el dia a dia amb el seu humor i la seua forma de ser.

A la meua parella, per donar-me suport i aguantar-me en els moments bons i no tan bons.

Als meus iaies Maria i Vicent, que sé que els fa moltíssima il·lusió tindre un nét amb una carrera universitària, per tots els esforços que han realitzat per a que la vida dels seus fills i néts siga més agradable i satisfactòria.

Als meus pares Ana i Paco, perquè ho sou tot per a mi i perquè gràcies a vosaltres, estic on estic.

Però sobretot, aquest treball va dedicat a mon pare Paco i al meu iaio Vicent, que de segur, allà on estiguen, es sentiran orgullosos de mi. Va per a vosaltres, per ser els meus àngels de la guarda, els meus herois.

A tots vosaltres, Gràcies.



# Índex

---

Agraïments	v
Índex	vii
Índex de figures	ix
Índex de taules	ix

---

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	Motivació . . . . .	1
1.2	Objectius . . . . .	2
1.3	Estructura de la memòria . . . . .	2
<b>2</b>	<b>Estat de l'art</b>	<b>5</b>
2.1	Tècniques d'aprenentatge automàtic . . . . .	5
2.1.1	Màquines de vectors de suport . . . . .	5
2.1.2	Perceptró multicapa . . . . .	6
2.1.3	K-veïns més propers . . . . .	7
2.1.4	Tècniques de <i>boosting</i> . . . . .	7
2.1.5	Arbres aleatoris . . . . .	9
2.1.6	Eliminació recursiva de característiques . . . . .	9
2.2	Treballs relacionats . . . . .	10
2.3	Robòtica . . . . .	11
2.3.1	NAO . . . . .	11
2.4	Sistemes Multiagent . . . . .	12
2.4.1	FIPA . . . . .	13
2.4.2	SPADE . . . . .	14
2.5	Avaluació dels mètodes estadístics . . . . .	15
2.5.1	Validació creuada <i>Monte Carlo</i> . . . . .	15
2.5.2	Mètriques d'avaluació de models estadístics . . . . .	15
<b>3</b>	<b>Detecció d'expressions facials com a aplicació nativa</b>	<b>17</b>
3.1	Descripció del problema . . . . .	17
3.2	Base de dades d'expressions facials . . . . .	17
3.3	Recerca de característiques . . . . .	19
3.3.1	Punts de referència . . . . .	19
3.3.2	Distàncies entre punts . . . . .	20
3.4	Entrenament i experimentació . . . . .	21
3.5	Conclusions . . . . .	23
<b>4</b>	<b>Sistema basat en el model client/servidor</b>	<b>25</b>
4.1	Sistema multiagent . . . . .	25
4.1.1	Agent NAO . . . . .	25
4.1.2	Agent classificador . . . . .	25
4.1.3	Estudi del temps de predicció . . . . .	26
4.2	Estudi i integració de nous algorismes d'aprenentatge automàtic . . . . .	26
4.3	Entrenament i experimentació . . . . .	27
4.3.1	Arbres aleatoris . . . . .	27

---

4.3.2	K-veïns més propers	27
4.3.3	Boosting	28
4.3.4	Perceptró multicapa	28
4.4	Conclusions	28
<b>5</b>	<b>Sistema basat en agents classificadors</b>	<b>31</b>
5.1	Anàlisi de la sensibilitat de cada expressió facial respecte al model utilitzat	31
5.1.1	Anàlisi de la matriu de confusió	32
5.2	Sistema de votacions	33
5.3	Sistema multiagent amb agents classificadors	33
5.3.1	Agent NAO	33
5.3.2	Agent coordinador	34
5.3.3	Agents classificadors	34
5.4	Experimentació	35
5.5	Conclusions	38
<b>6</b>	<b>Conclusions i treball futur</b>	<b>39</b>
6.1	Conclusions	39
6.2	Treball futur	40
	<b>Bibliografia</b>	<b>41</b>



# Índex de figures

---

2.1	Exemple d'SVM . . . . .	6
2.2	Exemple de xarxa neuronal artificial . . . . .	7
2.3	Representació gràfica del l'algorisme <i>KNN</i> . . . . .	8
2.4	Diagrama de blocs de l'algorisme <i>AdaBoost</i> . . . . .	8
2.5	Exemple d'arbres aleatoris . . . . .	9
2.6	Distàncies utilitzades en (Jarkiewicz, Kocielnik i Marasek 2009) . . . . .	10
2.7	Sensors, actuadors i components del robot NAO . . . . .	12
2.8	Representació gràfica d'un agent intel·ligent . . . . .	13
3.1	Interacció entre NAO i una persona . . . . .	18
3.2	Subconjunt d'imatges que formen part de <i>KDEF</i> . . . . .	19
3.3	Punts obtinguts amb els mètodes <i>detector</i> i <i>predictor</i> a partir de la imatge <i>AF33HAS</i> pertanyent al conjunt de dades <i>KDEF</i> . . . . .	20
3.4	Distàncies obtingudes amb <i>RFE</i> de la imatge <i>AF33HAS</i> de la col·lecció <i>KDEF</i> . . . . .	21
3.5	Distàncies obtingudes amb <i>FE</i> de la imatge <i>AM20AFS</i> de la col·lecció <i>KDEF</i> . . . . .	21
3.6	Gràfica de la importància relativa de les característiques utilitzant <i>FE</i> . . . . .	22
4.1	Representació gràfica de l'arquitectura client/servidor proposada . . . . .	26
4.2	Configuració de la xarxa neuronal artificial usada en l'experimentació del model <i>MLP</i> . . . . .	29
5.1	Representació gràfica del sistema multiagent proposat. . . . .	34
5.2	Representació gràfica del format de les particions utilitzat per a l'experimentació. . . . .	35
5.3	Gràfica on es comparen els valors de <i>PPV</i> de cada emoció i per a cada model. . . . .	38

# Índex de taules

---

3.1	Precisió obtinguda per a cada model <i>SVM</i> amb el conjunt de dades <i>KDEF</i> . . . . .	22
3.2	Precisió obtinguda usant <i>SVM RBF</i> per a cada conjunt de dades amb 2278 característiques . . . . .	23
3.3	Precisió obtinguda usant <i>SVM RBF</i> per a cada conjunt de dades amb 364 característiques . . . . .	23
4.1	Precisió obtinguda dels diferents models d'arbres aleatoris depenent de la profunditat màxima establerta . . . . .	27
4.2	Precisió obtinguda dels diferents models basats en <i>KNN</i> depenent dels <i>k</i> -veïns emprats . . . . .	27

4.3	Precisió obtinguda del diferents models basats en tècniques de <i>boosting</i> depenent de la profunditat màxima establerta . . . . .	28
4.4	Precisió obtinguda del diferents models d' <i>MLP</i> , entrenats amb diferents configuracions de neurones per capa . . . . .	28
5.1	Taula de la precisió obtinguda per model i emoció . . . . .	31
5.2	Mètriques obtingudes per als models <i>SVM</i> i <i>MLP</i> de les diferents emocions	32
5.3	Mètriques obtingudes amb el primer conjunt de dades d'entrenament . .	36
5.4	Mètriques obtingudes amb el segon conjunt de dades d'entrenament . . .	37

---

---

# CAPÍTOL 1

## Introducció

---

Actualment, la forma d'interactuar amb les màquines està canviant de forma dràstica. Per exemple, a les videoconsoles, s'està abandonant el comandament físic per càmeres que reconeixen els gestos i moviments per permetre altra forma d'interactuar amb el joc. També existeixen moltes aplicacions, capaces de reconèixer objectes i cares, que permeten una interacció més humana amb el maquinari. Si ens parem a pensar, la comunicació humana no és sols la comunicació verbal i l'escrita; la comunicació gestual és la que més ens descriu, la que més importància aporta a la comunicació. Es per això que, aquest projecte tracta del reconeixement d'emocions a través d'imatges captades per un robot humanoide, en concret, pel robot *NAO*.

Si parlem de la robòtica, segons Issac Asimov, la primera llei de la robòtica és: cap robot causarà dany a un ésser humà o permetrà, amb la seua inacció, que un ésser humà resulte danyat. Respecte a l'afirmació anterior, podria un robot ferir els nostres sentiments? La violència psicològica no és sinó, un tipus de violència més? Doncs, amb un detector d'expressions emocionals, es podria tractar de millorar aquest problema de comunicació persona-màquina.

Per altra banda, hi ha persones que no actuen igual enfront a estímuls humans o provinents del medi extern, és per això que, un psicòleg, podria utilitzar el robot per introduir-li estímuls i poder avaluar com es comporta enfront a ells, sense la intervenció humana. Ens donaria l'oportunitat d'obindre una informació qualitativa de les persones, per tal de prendre decisions, depenent de l'estat anímic de la persona a la que s'està monitoritzant.

### 1.1 Motivació

---

Amb aquesta contribució, s'intenta aportar una nova forma d'interactuar amb el maquinari, que sumada a altres formes d'interacció, ens aporte un altre punt de vista respecte a la comunicació persona-màquina. Una perspectiva de comunicació més humana que, aplicada a la vida real, pot nodrir futurs projectes on, les emocions de les persones, són un dels factors més importants a tindre en compte. Un dels dominis d'aplicació d'un sistema de reconeixement d'emocions és la monitorització de persones amb discapacitat, persones menudes i persones majors per tal d'obindre dades d'interès enfront a nous estímuls externs. També podríem parlar de la integració d'aquest sistema amb el reproductor de música per tal de reproduir cançons més adients al nostre estat anímic. És, també, una gran motivació el contribuir en una línia d'investigació del Grup de Tecnologia Informàtica - Intel·ligència Artificial (*GTI-IA*) de la Universitat Politècnica de València, degut a que, aquest treball, forma part d'un projecte d'investigació finançat amb el programa Prometeo de la Generalitat Valenciana (PROMETEOII/2013/009).

---

## 1.2 Objectius

---

L'objectiu principal d'aquest projecte és el **reconeixement d'emocions** a través d'imatges, en un temps raonable, utilitzant el robot humanoide *NAO*. A més a més, per aconseguir aquest objectiu principal seran necessaris subobjectius, els quals li donaran uns valors i una qualitat afegida al present treball.

- **Estudi de les característiques.** Estudiar quines característiques i com de significatives són per tal d'ajudar a la millora en la classificació d'expressions facials.
- **Anàlisi de les diferents tècniques d'aprenentatge automàtic.** Analitzar les tècniques d'aprenentatge automàtic i quines s'adapten millor al nostre corpus de dades.
- **Predir l'emoció en un temps raonable.** Una interacció més fluida entre el robot i la persona, predisposarà a una percepció més agradable i més humana per part de la persona. Per aquest motiu, el temps de resposta no deurà ser superior als 10s.
- **Experimentació sobre la plataforma.** Es plantejaran diferents experiments al llarg del projecte per tal de validar la correcta implementació i funcionament del sistema.
- **Obtindre una precisió superior al 90% .** En total, la precisió del reconeixedor serà superior al 90% per a totes les emocions.

---

## 1.3 Estructura de la memòria

---

La resta d'aquest escrit s'organitza principalment en cinc capítols de la forma següent:

- El **Capítol 2: Estat de l'art** revisa breument la situació de les tècniques i algorismes d'aprenentatge automàtic que s'utilitzen actualment. A més a més, proporciona alguns coneixements bàsics referents als sistemes multiagent i a la situació actual de la robòtica. Aquest capítol també revisa treballs relacionats amb la problemàtica del reconeixement d'emocions. Per altra banda, els mètodes de validació i les mètriques emprades en aquest projecte són explicades al final del capítol.
- El **Capítol 3: Detecció d'expressions facials com a aplicació nativa** revisa breument la descripció del problema així com les bases de dades utilitzades en aquest treball. També s'explica el procés de recerca de característiques, incloent el procés d'extracció de punts característics i càlcul de distàncies. Finalment es proposa una experimentació per tal de validar el sistema i obtenir unes conclusions.
- El **Capítol 4: Sistema basat en el model client/servidor** s'explica com es millora l'aproximació anterior, respecte al temps emprat pel reconeixement, mitjançant la implementació d'un sistema multiagent. A més a més, s'integren noves tècniques d'aprenentatge automàtic per tal de comparar distints models de classificació estadístics. Finalment es plantejen una sèrie d'experimentacions per tal d'avaluar els distints models i poder obtenir unes conclusions dels resultats obtinguts.
- El **Capítol 5: Sistema basat en agents classificadors** té com objectiu descriure l'anàlisi del comportament dels models depenent de la emoció utilitzada. L'anàlisi és realitzarà amb les mètriques corresponents i utilitzant les tècniques de validació exposades. Finalment es proposa un model basat en agents classificadors que, mitjançant el vot, elegiran la predicció pertinent. Finalment, es realitzarà una experimentació del sistema, amb la seua corresponent validació, per obtenir les conclusions dels resultats obtinguts.

- El **Capítol 6: Conclusions i treball futur** resumeix de quina manera i com es compleixen els objectius plantejats en el present treball, A més a més, estableix les conclusions i esbossa el treball futur.



---

---

## CAPÍTOL 2

# Estat de l'art

---

En aquest capítol analitzarem la situació actual del problema a resoldre, així com, els conceptes bàsics previs a la resolució del problema. A més a més, la nostra aproximació, emprerà un classificador estadístic, per això, en la següent secció, introduïrem les distintes tècniques i algorismes d'aprenentatge automàtic que més s'usen en l'actualitat.

### 2.1 Tècniques d'aprenentatge automàtic

---

L'aprenentatge automàtic és una branca de la Intel·ligència Artificial (IA) amb l'objectiu de desenvolupar tècniques que permetin a les computadores aprendre. De manera més concreta, es tracta de crear programes capaços de generalitzar comportaments a partir d'una informació no estructurada subministrada en forma d'exemples. És, per tant, un procés d'inducció del coneixement. En moltes ocasions el camp d'actuació de l'aprenentatge automàtic es solapa amb el de l'estadística, ja que les dues disciplines es basen en l'anàlisi de dades. No obstant això, l'aprenentatge automàtic es centra més en l'estudi de la complexitat computacional dels problemes. Molts problemes són de classe *NP-durs*, pel que gran part de la investigació realitzada en aprenentatge automàtic està enfocada al disseny de solucions factibles a aquests problemes. L'aprenentatge automàtic pot ser vist com un intent d'automatitzar algunes parts del mètode científic mitjançant mètodes matemàtics.

L'aprenentatge automàtic té una àmplia gamma d'aplicacions, incloent motors de cerca, diagnòstics mèdics, detecció de frau en l'ús de targetes de crèdit, anàlisi del mercat de valors, classificació de seqüències d'ADN, reconeixement de la parla i del llenguatge escrit, jocs i robòtica.

Tal i com s'expliquen en les diverses publicacions, com en (Bishop 2006), (Alpaydin 2004) i (Murphy 2012), hi existeixen un gran nombre de tècniques d'aprenentatge automàtic. Entre aquestes podem trobar màquines de vectors de suport (*SVM*), perceptró multicapa o xarxes neuronals artificials (*MLP*), tècniques de *boosting*, arbres aleatoris i *k*-veïns més propers.

#### 2.1.1. Màquines de vectors de suport

Les màquines de vectors de suport és una de les tècniques més usades en l'actualitat. Es basen en l'ús d'algorismes d'aprenentatge supervisat i s'utilitzen en problemes de classificació i regressió. Els *SVM* (*Support Vector Machines*) construeixen un classificador estadístic a partir de les dades d'entrenament que se li proporciona. A més a més, busquen un hiperplà que separe de forma òptima els punts d'una classe de l'altra, que eventu-

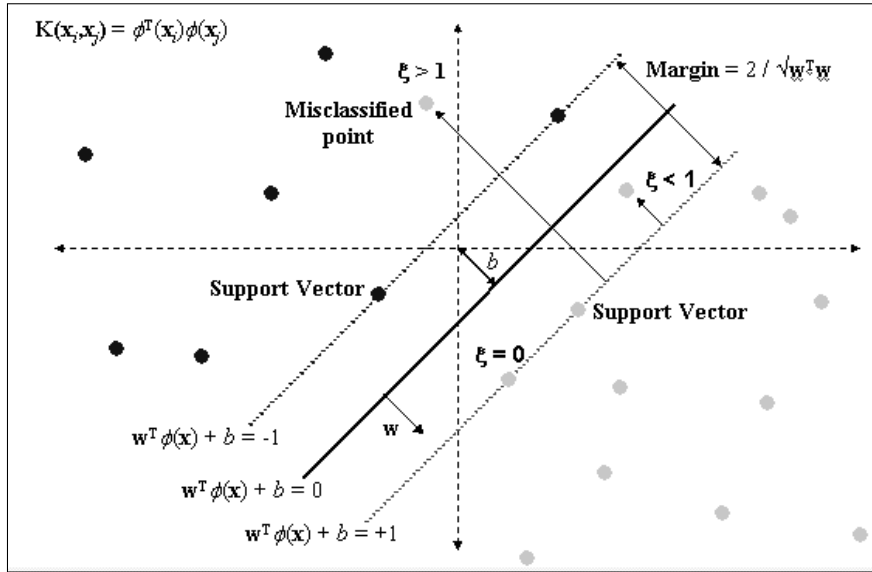


Figura 2.1: Exemple d'SVM

alment han pogut ser prèviament projectats a un espai de dimensionalitat superior. Un exemple de la correcta classificació de dos classes en un espai de dos dimensions es pot veure en la Figura 2.1. Aquest algorisme està implementat baix la llibreria *OpenCV*<sup>1</sup> i la seua API es pot consultar al portal web de la pròpia ferramenta.<sup>2</sup>

### 2.1.2. Perceptró multicapa

Les xarxes neuronals artificials és una de les tècniques que més utilitzades en els últims anys. Concretament, el perceptró multicapa (*MLP*) és una xarxa neuronal artificial formada per múltiples capes, açò li permet resoldre problemes que no són linealment separables, la qual cosa és la principal limitació del perceptró (també anomenat perceptró simple). El perceptró multicapa pot ser totalment o localment connectat. En el primer cas, cada eixida d'una neurona de la capa "i" és entrada de totes les neurones de la capa "i+1", mentre que en el segon cada neurona de la capa "i" és entrada d'una sèrie de neurones (regió) de la capa "i+1"

Es basen en el paradigma d'aprenentatge i processament de la informació automàtic i s'inspiren en el funcionament de les xarxes neuronals del sistema nerviós biològic. La configuració de l'estructura està formada per una capa d'entrada i una d'eixida, a més d'una o més capes ocultes. Cada una de les capes està formada per una o més neurones que posseïxen la seua corresponent funció d'activació. El resultat de cada neurona es enviat a la neurona següent per al seu corresponent càlcul. En la capa d'eixida es produeix l'eixida final de la *MLP*. Es pot veure un exemple d'*MLP* en la Figura 2.2 amb quatre neurones en la capa de entrada, dos en la d'eixida i una capa oculta amb quatre neurones. Aquest algorisme d'aprenentatge automàtic està implementat baix la llibreria *OpenCV* i la seua API es pot consultar al portal web de la pròpia ferramenta.<sup>3</sup>

<sup>1</sup>Consulta <http://opencv.org/> per a més informació.

<sup>2</sup>Consulta [http://docs.opencv.org/2.4/modules/ml/doc/support\\_vector\\_machines.html](http://docs.opencv.org/2.4/modules/ml/doc/support_vector_machines.html) per a més informació sobre l'API.

<sup>3</sup>Consulta [http://docs.opencv.org/2.4/modules/ml/doc/neural\\_networks.html](http://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html) per a més informació sobre l'API.



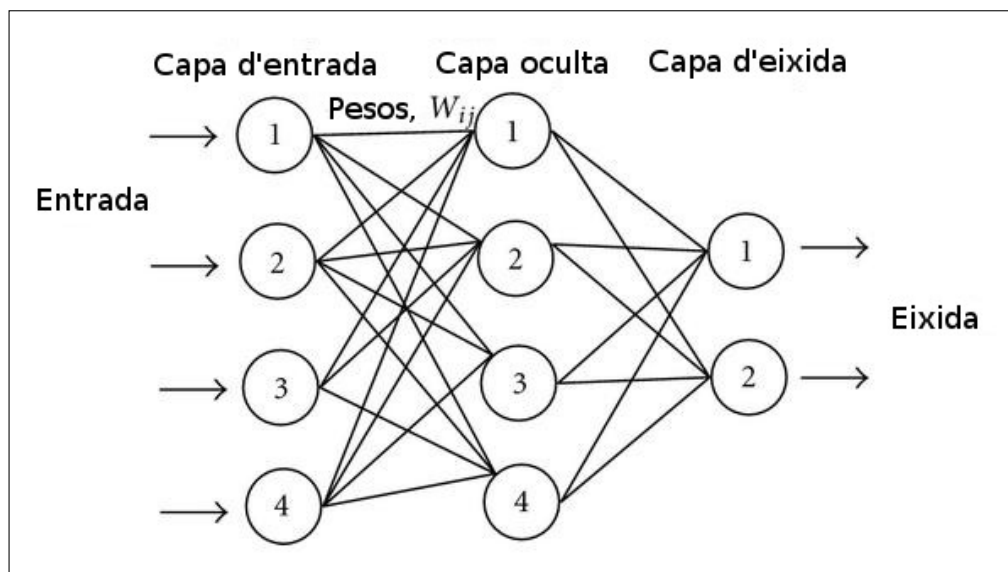


Figura 2.2: Exemple de xarxa neuronal artificial

### 2.1.3. K-veïns més propers

En el reconeixement de patrons, l'algorisme k-veïns més propers (*KNN*) és un mètode no paramètric utilitzat per a la classificació i la regressió. (Altman 1992) En els dos casos, l'entrada es compon dels k exemples d'entrenament més propers a l'espai de característiques. La sortida depèn de si *KNN* s'utilitza per a la classificació o regressió. En la classificació *KNN*, la sortida és una pertinença a una classe. Un objecte es classifica pel vot de la majoria dels seus veïns, per tant, se li assigna la classe més comú entre els seus k veïns més propers (k és un enter positiu, normalment petit). Si  $k = 1$ , llavors l'objecte simplement s'assigna a la classe que pertany aqueix veí més proper.

*KNN* és un tipus d'aprenentatge basat en exemples, o d'aprenentatge difús, on la funció només és aproximada a nivell local i tots els càlculs s'ajornen fins a la classificació. L'algorisme *KNN* està entre el més simples de tots els algorismes d'aprenentatge automàtic. Com podem veure a la Figura 2.3, la mostra senyalada serà classificada com a blanca, degut a que la votació dels 3 veïns més propers dona com a resultat la classe a la que pertanyen les blanques.

Els veïns s'obtenen a partir d'un conjunt de mostres per als quals es coneix la classe. Això és el conjunt d'entrenament per a l'algorisme. Un defecte de l'algorisme *KNN* és que és sensible a l'estructura local de les dades.

*KNN* està implementat baix la llibreria *OpenCV* i la seua *API* es pot consultar al portal web de la pròpia ferramenta. <sup>4</sup>

### 2.1.4. Tècniques de *boosting*

Les tècniques de *boosting* es un mètode d'aprenentatge automàtic, basat en el conjunt de models febles, que, primerament, redueix el soroll, i també la variància utilitzant l'aprenentatge supervisat, i forma part de la família d'algorismes d'aprenentatge automàtic que converteixen models febles en models més forts. Els models febles es defineixen per

<sup>4</sup>Consulta [http://docs.opencv.org/2.4/modules/ml/doc/k\\_nearest\\_neighbors.html](http://docs.opencv.org/2.4/modules/ml/doc/k_nearest_neighbors.html) per a més informació sobre l'*API*.

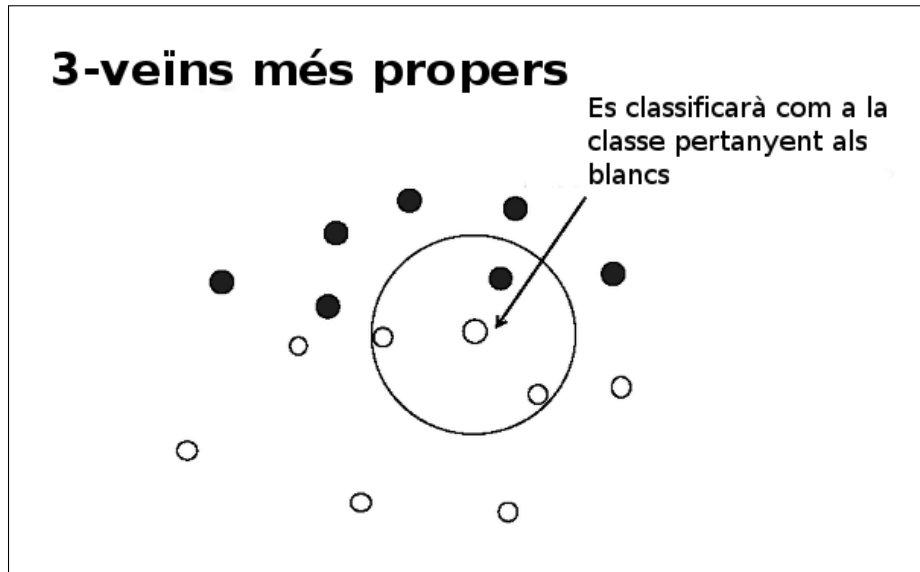


Figura 2.3: Representació gràfica del l'algorisme *KNN*

ser classificadors que estan lleugerament correlacionats amb la veritable classificació. Per contra, un model fort és un classificador que està arbitràriament ben correlacionat amb la veritable classificació. En la Figura 2.4 podem observar l'arquitectura que segueixen els classificadors basats en tècniques de *boosting*.

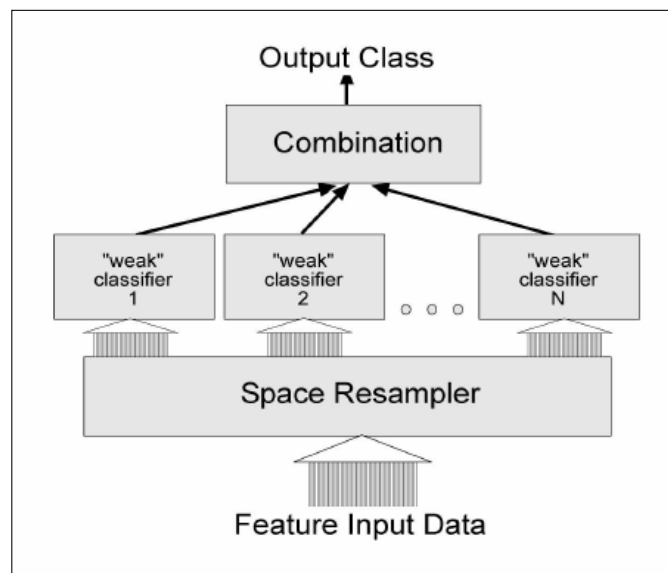


Figura 2.4: Diagrama de blocs de l'algorisme *AdaBoost*

Una de les tècniques de *boosting* més utilitzades és *AdaBoost* que es pot utilitzar en combinació amb molts altres tipus d'algoritmes d'aprenentatge per millorar el seu rendiment. La sortida dels altres algoritmes d'aprenentatge dels models febles es combina en una suma ponderada que representa la sortida final del classificador impulsat (model fort). Per a la formulació d'aquests models es sol utilitzar arbres de decisió.

*AdaBoost* està implementat baix la llibreria *OpenCV* i la seua *API* es pot consultar al portal web de la pròpia ferramenta.<sup>5</sup>

### 2.1.5. Arbres aleatoris

Els arbres aleatoris són un mètode d'aprenentatge, basats en la conjunció de classificadors febles, per a la classificació, regressió i altres tasques, que operen mitjançant la construcció d'una multitud d'arbres de decisió en el moment de la formació, on la sortida de la classe, serà la pertinència (o no) a les classes (classificació). Els arbres aleatoris corregeixen el sobreajustament al seu conjunt d'entrenament d'arbres de decisió. En la Figura 2.5 podem observar com les sortides dels diferents arbres aleatoris es combinen per tal de generar un únic resultat, la classe a la que pertany la mostra.

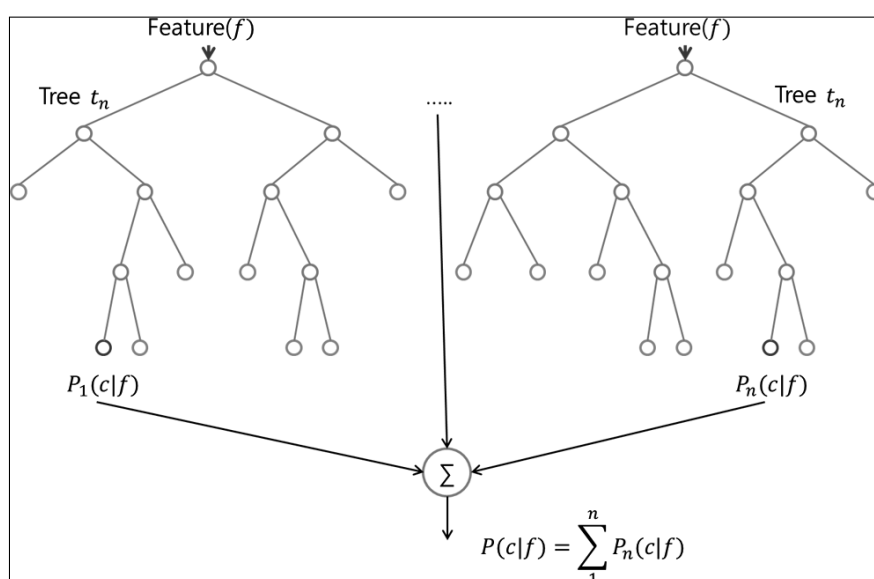


Figura 2.5: Exemple d'arbres aleatoris

Aquesta tècnica està implementada baix la llibreria *OpenCV* i la seua *API* es pot consultar al portal web de la pròpia ferramenta.<sup>6</sup>

### 2.1.6. Eliminació recursiva de característiques

Tenint en compte un estimador extern que assigna pesos a les característiques (per exemple, els coeficients d'un model lineal), l'objectiu de la funció recursiva d'eliminació (*RFE*) de característiques és seleccionar les característiques més significatives, considerant de forma recursiva conjunts més petits de característiques. En primer lloc, l'estimador és entrenat en el conjunt inicial de les característiques i els pesos són assignats a cada un d'ells. Llavors, les característiques, on el seu pes absolut és el més menut són eliminades del conjunt de característiques actuals. Aquest procediment es repeteix recursivament en el conjunt podat fins que finalment, s'arriba al nombre desitjat de característiques per seleccionar.

<sup>5</sup>Consulta <http://docs.opencv.org/2.4/modules/ml/doc/boosting.html> per a més informació sobre l'*API*.

<sup>6</sup>Consulta [http://docs.opencv.org/2.4/modules/ml/doc/random\\_trees.html](http://docs.opencv.org/2.4/modules/ml/doc/random_trees.html) per a més informació sobre l'*API*.

Aquest algorisme està implementat baix la llibreria *Sklearn*<sup>7</sup> i la seua *API* es pot consultar al portal web de la pròpia ferramenta.<sup>8</sup>

## 2.2 Treballs relacionats

Com a part fonamental del treball, s'han analitzat diversos treballs semblants, existents en la literatura actual, que tracten el problema d'identificar les expressions facials a partir d'imatges facials. En cadascun d'ells, s'utilitzen distintes tècniques i algorismes, per tal d'obtenir una aproximació a la solució òptima del problema. Com es pot veure en (Sohail i Bhattacharya 2008), l'autor utilitza un model antropomètric basat en nou punts i cinc distàncies per extraure zones d'interès d'una imatge facial. Per altra banda, en (Zhang, Zhao i Lei 2011) s'intenta fer una aproximació del reconeixedor utilitzant tècniques d'Anàlisi Discriminant Lineal (*LDA*) i l'algorisme d'aprenentatge automàtic K-veïns més propers, obtenint com a millor resultat un 87,70 % de precisió.

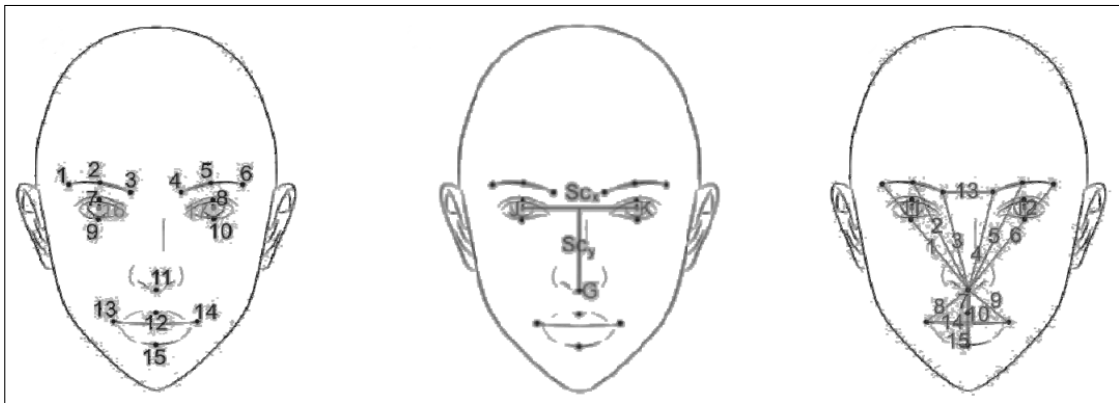


Figura 2.6: Distàncies utilitzades en (Jarkiewicz, Kocielnik i Marasek 2009)

Altra de les aproximacions que s'estudien (Bavkar, Rangole i Deshmukh 2015), on a partir de certs punts característics, obté unes distàncies entre aquests, utilitzant la distància euclidiana. Després s'entrena un classificador estadístic basat en màquines de vectors de suport, que s'explicarà en la secció 2.1. Amb aquest model, s'obté una precisió mitja del 90%. Per últim trobem una publicació (Jarkiewicz, Kocielnik i Marasek 2009) on podem observar com, a partir de les distàncies de la Figura 2.6, les usa com a característiques per a entrenar dos classificadors; un classificador basat en k-veïns més propers i l'altre en xarxes neuronals artificials. Amb aquesta aproximació, obté una precisió mitja del 86,36 % per al millor classificador, el basat en xarxes neuronals. Cal afegir que totes les publicacions classifiquen les emocions en felicitat, neutralitat, sorpresa, ira, disgust, por i tristesa, que conformen un total de set expressions facials a reconèixer.

Aquestes publicacions, ens proporcionen les nocions bàsiques per poder estudiar i proposar una aproximació que millori la precisió mitja respecte a totes les publicacions nombrades. Per altra banda, abans de procedir a explicar la solució del problema, haurem de posar l'accent en l'estat de l'art de la robòtica i dels dominis en els quals es desenvolupa.

<sup>7</sup>Consulta <http://scikit-learn.org> per a més informació.

<sup>8</sup>Consulta [http://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.RFE.html](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html) per a més informació sobre l'*API*.

## 2.3 Robòtica

---

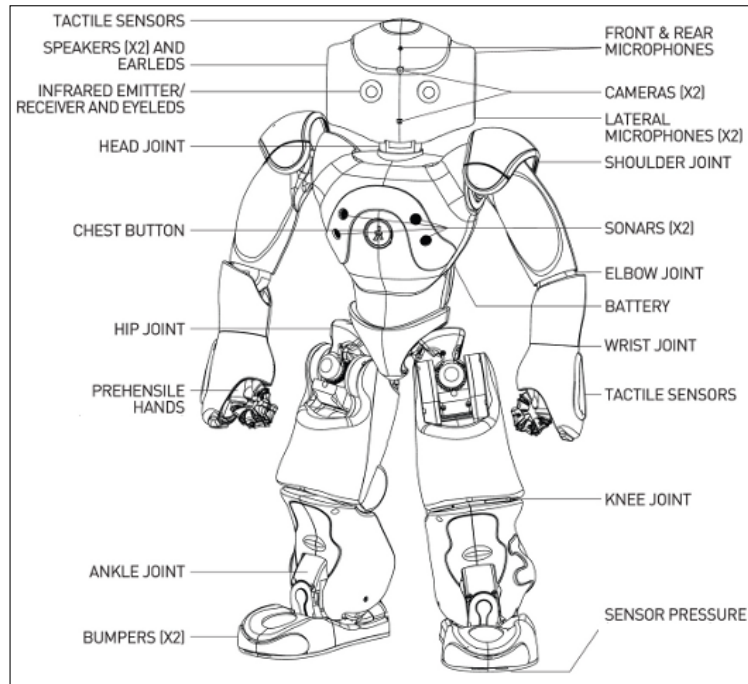
La robòtica sempre ha intentat simular d'alguna manera els comportaments humans per tal d'interactuar d'una forma més amable amb les persones. Existeixen moltes aplicacions i dominis on els robots són la clau per a resoldre els problemes, i més, si pensem en robots humanoides. Tal i com podem trobar en moltes revistes científiques, alguns projectes, com per exemple (Low et al. 2013), plantegen l'estratègia a seguir per tal de que un grup de robots juguen a futbol. Aquest problema és resolt mitjançant la col·laboració entre tots els robots, per tal de complir un objectiu comú. Aquests robots, usen per a comunicar-se el protocol *FIPA*, que s'explicarà més endavant. Cal afegir que, cada robot, actua com un agent intel·ligent, el qual compleix unes característiques determinades. El robot usat en l'article mencionat, és també el que s'usa en aquest treball, el robot *NAO*.

### 2.3.1. *NAO*

*NAO* és un robot humanoide programable i autònom de 58cm d'altura, desenvolupat per *Aldebaran Robotics* amb el qual pretenen aconseguir un company encantador i interactiu per a cada casa. El desenvolupament del robot començà en 2004. En 2007, *NAO* va substituir al gos robot *Aibo* de *Sony* com a plataforma estàndard per a la *Robocup (Robot Soccer World Cup)*, un concurs internacional de robòtica. L'edició acadèmica de *NAO*, fou desenvolupada per a les universitats i laboratoris amb finalitats d'investigació i d'educació. Així doncs, fou llançada en 2008 i es posà a disposició del públic general abans del 2011. El robot ha entrat ja en ús en nombroses institucions acadèmiques de tot el món, incloent la Universitat de Tokio, el IIT Kanpur de la Índia i la Universitat del Rei Fahd de Petrol i Minerals d'Arabia Saudita. En 2014 sortí al mercat una versió millorada del robot, que ofereix una major precisió en la síntesi de veu multilingüe.

*NAO*, obté les imatges usant dos càmeres de vídeo de 1,22 Mpx, que poden capturar 30 fotogrames per segon. La càmera superior, situada en la part davantera del cap del *NAO*, obté instantànies del camp de visió frontal mentre que la situada a l'altura de la boca, les obté del camp inferior de visió. La llibreria *Naoqi*, escrita en *Python*, que acompanya el robot, permet recuperar imatges i vídeos de ambdues càmeres anteriorment descrites. A més a més, *NAO* proporciona una sèrie d'algorismes per a detectar i reconèixer cares i formes. També pot reconèixer qui està parlant-li o encontrar una pilota o objectes més complexos. Aquests algorismes estan especialment desenvolupats pensant constantment en usar el menor nombre de recursos de processador. A més, al comptar amb *Python* com a llenguatge natiu, permet desenvolupar els nostres propis mòduls per a interactuar amb *OpenCV*, la llibreria de codi lliure de visió per computador, originalment desenvolupada per *Intel* i que compta amb un gran nombre de funcions i característiques orientades al processament d'imatges.

El robot usa quatre micròfons direccionals per a detectar sons. Recordem que uns dels dominis aplicables podria ser la interacció amb persones menudes. La localització del so, permet al robot identificar la direcció dels sons. Així, quan una font propera emetés un so, cadascun dels quatre micròfons del *NAO* rebrà l'onda de so en temps lleugerament diferents. Utilitzant aquestes característiques, un ampli rang de d'experiments i investigacions poden tindre lloc en els camps de la comunicació i la interacció persona-robot. A més de càmeres i micròfons, *NAO* està equipat amb sensors capacitius posicionats en tres seccions dalt del seu cap i en les mans. Se li pot proporcionar, per tant, informació al *NAO* mitjançant el tacte: pressionant una única vegada per a dir-li que s'apague, per exemple, o usant els sensors com una sèrie de botons per a activar una acció associada.



**Figura 2.7:** Sensors, actuadors i components del robot NAO

*NAO* està equipat amb dos canals de sonar: dos transmissors i dos receptors. Estos permeten a *NAO* estimar la distància dels obstacles del seu entorn. El rang de detecció és d'1 cm a 3 m. A menys de 15 cm, no torna informació de distància, sòls sap que hi ha un obstacle present. L'humanoide suporta connexió Wifi i Ethernet, així que podem usar la xarxa com a mitjà de comunicació. Esta combinació de tecnologies dota al *NAO* de l'habilitat d'obtenir característiques quantitatives del seu entorn. Però, ¿Pot interpretar el que detecta? Ací és on la llibreria *Naoqi* entra en acció. *Aldebaran* ha integrat aquesta llibreria baix un sistema operatiu basat en la distribució de *Linux gentoo*. Aquest sistema operatiu s'executa sobre la CPU principal del robot, un *Intel ATOM 1,6 GHz* situat en el cap, junt amb una CPU secundària localitzada en el tors.

Finalment, *Alderbaran* també distribueix una aplicació, per a ordinadors de sobretaula, que inclou una ferramenta gràfica de programació, *Choregraphe suite*, una aplicació de simulació creada exclusivament per als robots de l'empresa *Alderbaran*. Amb aquesta eina, es poden programar accions sobre el robot d'una forma simple: clicar i arrastrar.

A banda del maquinari, també necessitarem d'un sistema de comunicació que permeti al robot comunicar-se amb un altre per tal de prendre decisions conjuntament o per a formar part d'un col·lectiu de robots que actuen de manera racional.

## 2.4 Sistemes Multiagent

Un sistema multiagent (*SMA*), tal i com podem veure en (Wooldridge 2009) i (Russell i Norvig 2010) és un sistema compost per distints agents intel·ligents que interactuen entre si dintre d'un determinat entorn. Aquestos, són capaços d'assolir col·lectivament metes o objectius que serien difícils de realitzar per un agent individual o sistema monolític.

El bloc fonamental de la construcció d'un sistema multiagent és l'agent, tal i com podem veure en la Figura 2.8. Un agent intel·ligent és una entitat capaç de percebre el seu

entorn, processar dites percepcions i respondre o actuar de manera racional. També són capaços de comunicar-se a través d'un mecanisme de comunicació interprocés, utilitzant protocols de comunicació.

Depenent de la complexitat del agents, podem distingir tres tipus bàsics:

- Agents passius. Són agents molt simples que no disposen de cap objectiu.
- Agents actius. Són agents intel·ligents que disposen d'un objectiu simple, per exemple, esperar un missatge i reenviar-lo.
- Agents complexos. Són agents actius on el objectiu a assolir no és tan simple. Són capaços de realitzar càlculs complexos, així com de dur a terme un gran ventall de funcionalitats.

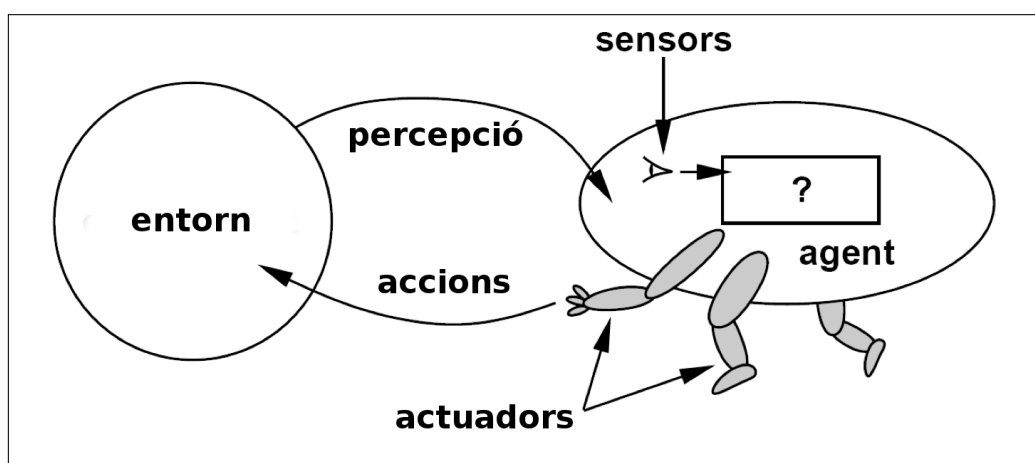


Figura 2.8: Representació gràfica d'un agent intel·ligent

Els agents intel·ligents tenen tres característiques importants:

- Autonomia. Els agents són, almenys, parcialment autònoms.
- Visió local. Cap agent té una visió global del sistema perquè es massa complexe per a un agent fer un ús pràctic d'eixos coneixements.
- Descentralització. No hi ha un agent de control designat, el sistema està distribuït.

Cal destacar que, aquests agents intel·ligents, podrien ser "el cervell" de qualsevol robot, per tal de donar-li al robot una forma d'actuar més racional. Per a la interacció entre agents, existeix un estàndard anomenat *FIPA-ACL* que seguidament explicarem.

#### 2.4.1. FIPA

La *Foundation for Intelligent Physical Agents (FIPA)* és un organisme per al desenvolupament i establiment d'estàndards de *software* per a agents heterogènis que interaccuen entre si i sistemes basats en agents.

*FIPA* fou fundada com una organització sense ànim de lucre l'any 1996 amb l'objectiu de definir un conjunt complet de normes, per a la implementació de sistemes en els que



es puguem executar agents, i definir una especificació de com els propis agents es deuen comunicar i interactuar. La organització es va dissoldre ne 2005 i es va crear un comitè d'estàndards *IEEE* en el seu lloc.

Per altra banda, una plataforma multiagent es un *framework* per a la creació i gestió del cicle de vida d'un sistema multiagent. Per a ser compatibles amb *FIPA*, les plataformes multiagent requereixen quatre requisits mínims:

- Un canal de comunicació dels agents o *ACC* que permeta als agents i a la plataforma comunicar-se entre ells.
- Un sistema de gestió d'agents o *AMS*. Un mecanisme que permet als agents registrar-se en la plataforma i ser assolibles per al contacte. Constitueix una espècie de pàgines blanques de la plataforma.
- Un facilitador de directoris o *DF*. És un tipus de servei públic en el qual els agents publiquen els serveis que hi ofereixen. El *DF* seria docs, com les pàgines grogues de la plataforma.
- Suport per al llenguatge de comunicació d'agents (*ACL*) de *FIPA*, que és un llenguatge per a tots els agents, on la seva sintaxis es basa en *XML*.

Un missatge *FIPA-ACL*, tal i com es pot veure en (Louis i Martinez 2006), conté un conjunt de un o més paràmetres. Precisar quins paràmetres són necessaris per a una comunicació entre agents efectiva, varia d'acord amb la situació. L'únic paràmetre que és obligatori en tots els missatges *ACL* és la performativa, a més s'espera que la majoria de missatges continguin els camps d'emissor, receptor i contingut o inclús la ontologia. Si un agent no reconeix o és incapaç de processar un o més paràmetres o valors d'estos, pot respondre amb el missatge apropiat de no-comprensió.

#### 2.4.2. *SPADE*

Havent repassat els conceptes anteriors, comentat en que es basen les plataformes multiagents i com es comuniquen els agents que hi formen part, en aquest projecte usarem la *Smart Python Agent Development Environment (SPADE)*<sup>9</sup>, desenvolupada pel grup *GTI-IA* de la *UPV*. *SPADE* és una plataforma multiagent que habilita de forma simple la interacció entre agents intel·ligents, ja que està basada en la tecnologia de missatgeria instantània *XMPP*, que és un estàndard en missatgeria. Aquest últim concepte és fonamental ja que els missatges que intercanvien els agents s'agrupen en converses, cosa que permet separar i identificar el contingut de la informació transmesa. A més d'açò, *SPADE* ofereix un parell d'utilitats per a crear un sistema multiagent. Existeix a la disposició del desenvolupador una llibreria per a *Python* que ajuda en la creació d'agents i en la implementació dels mètodes necessaris per a rebre i enviar missatges en el sistema.

A més a més, disposa d'una interfície web per a gestionar la plataforma i els agents que estiguen connectats a ella, a més de mantenir un històric de les converses. Parlant amb més detall de la llibreria, permet usar models d'agents que compten amb mecanismes per a connectar a la plataforma, un emissor de missatges i un conjunt de comportaments bàsics per a l'agent. Com es pot observar, aquestes característiques afavoreixen bastant la tasca de la implementació ja que donen feta una part bastant gran del procés de crear un sistema multiagent.

<sup>9</sup>Consulta <https://github.com/javipalanca/spade> per a més detalls de com obtenir *SPADE*



## 2.5 Avaluació dels mètodes estadístics

Per avaluar els nostres models, haurem de fer ús de les distintes tècniques de validació que existeixen en la literatura actual. És per açò què, en aquest treball, farem ús d'una tècnica de validació creuada, la *Monte Carlo*, a més a més d'obtindre unes mètriques a partir del resultat de la taula de confusió de cada model.

### 2.5.1. Validació creuada *Monte Carlo*

La validació creuada aleatòria, també anomenada *Monte Carlo* (Simon 2007), consisteix en dividir aleatòriament el conjunt de dades d'entrenament i el conjunt de dades de prova. Per a cada divisió la funció d'aproximació s'ajusta a partir de les dades d'entrenament i calcula els valors de sortida per al conjunt de dades de prova. El resultat final es correspon a la mitjana aritmètica dels valors obtinguts per a les diferents divisions. L'avantatge d'aquest mètode és que la divisió de dades entrenament-prova no depèn del nombre d'iteracions. Però, en canvi, amb aquest mètode hi ha algunes mostres que queden sense avaluar i altres que s'avaluen més d'una vegada, és a dir, els subconjunts de prova i entrenament es poden solapar.

### 2.5.2. Mètriques d'avaluació de models estadístics

Per a la posterior avaluació dels nostres models, farem ús de les distintes mètriques que s'obtenen a partir de la taula de confusió (Fawcett 2006). Totes les mètriques s'obtenen a partir de quatre valors calculats a partir dels resultats de la taula de confusió. Aquest valors són els següents:

- **TP**. Sumatori dels casos vertaders positius.
- **FN**. Sumatori dels casos falsos negatius.
- **FP**. Sumatori dels casos falsos positius.
- **TN**. Sumatori dels casos vertaders negatius.

A partir dels anteriors valors podem obtindre distintes mètriques, com ho són les següents:

- **TPR**. La taxa de vertaders positius o sensibilitat mesura la proporció de positius que s'han identificat correctament com a tal.

$$TPR = \frac{TP}{TP + FN}$$

- **SPC**. L'especificitat o taxa de vertaders negatius mesura la proporció de negatius que s'han identificat correctament com a tal

$$SPC = \frac{TN}{FP + TN}$$

- **PPV**. El valor de predicció positiu o precisió és la proporció de resultats positius que són veritables positius.

$$PPV = \frac{TP}{TP + FP}$$

- **NPV**. El valor de predicció negatiu és la proporció de resultats negatius que són veritables negatius.

$$NPV = \frac{TN}{TN + FN}$$

- **FPR**. La taxa de falsos positius és la proporció de tots els negatius que produeixen resultats positius. Esta mètrica és complementària a la especificitat.

$$FPR = \frac{FP}{FP + TN} = 1 - SPC$$

- **FDR**. La taxa de fals descobriment és la proporció de falsos positius respecte al total de casos positius. Esta mètrica és complementària al valor de predicció positiu.

$$FDR = \frac{FP}{TP + FP} = 1 - PPV$$

- **FNR**. La taxa de falsos negatius és la proporció de tots els negatius que produeixen resultats positius.

$$FNR = \frac{FN}{FN + TP}$$

- **ACC**. La precisió és la proporció del total de prediccions correctes.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

# Detecció d'expressions facials com a aplicació nativa

---

En aquest capítol descriurem el problema al qual s'enfrontem. Seguidament, es detallaran els subproblemes que s'han anat encontrant, com ho són el conjunt de mostres a utilitzar i la selecció de característiques. A més a més, entrenarem un model d'aprenentatge automàtic (secció 2.1) i l'avaluarem. Finalment, exposarem unes conclusions extretes d'aquest capítol.

## 3.1 Descripció del problema

---

El reconeixement d'expressions facials, concretament, les expressions de les persones quan sentim emocions és l'objectiu principal d'aquest projecte i, per tant, com obtindre aquestes emocions, mitjançant tècniques d'imatge digital, es converteix en el problema principal. Una vegada decidit com es representaran les emocions, dintre de l'aplicació, necessitarem un conjunt de fotografies, correctament etiquetades, on hi apareguen cares de persones expressant les distintes emocions.

A més a més, aquesta aplicació predictora d'emocions, haurà d'executar-se baix *Naoqi OS*, el sistema operatiu natiu del *NAO* (secció 2.3.1). Per tant, haurà de ser desenvolupada amb el llenguatge de programació *Python*, degut a que és un dels llenguatges nadius capaços d'executar-se baix *Naoqi OS*. Per tant, als problemes descrits anteriorment, haurèm de sumar el obtindre una precisió de qualitat, respecte a la predicció de les emocions, així com d'obtindre un cost temporal baix de processament d'aquestes.

Per a finalitzar, el problema presenta, també, la intencionalitat de poder-se usar com un agent que forme part d'una plataforma multiagent, capaç de prendre decisions a partir de l'estat anímic de les persones. En la Figura 3.1 podem apreciar com el robot *NAO* interactua amb una persona, perfectament podria estar predient l'estat anímic d'aquesta.

## 3.2 Base de dades d'expressions facials

---

Explicat el problema, se'ns fa necessària l'obtenció d'un conjunt de dades d'imatges de cares amb les diferents expressions facials, correctament etiquetades, per tal d'estudiar la naturalesa de les característiques que podem obtindre d'aquestes. Es per això que rere una minuciosa recerca, s'obtenen les següents bases de dades:

- **JAFFE**. La *Japanese Female Facial Expressions* (1998) és una base de dades que conté 217 fotos de 10 dones japoneses posant per expressar felicitat, tristesa, por, ira, sor-



Figura 3.1: Interacció entre NAO i una persona

presa, disgust i neutralitat.<sup>1</sup> Aquesta està formada per 30 imatges expressant ira, 29 de disgust, 32 de por, 31 de felicitat, 30 de neutralitat, 31 de tristesa i 30 de sorpresa. Usada com a conjunt de dades per a la realització de l'article (Dailey et al. 2010)

- **CAFE.** La *California Facial Expressions* és un conjunt de dades creat a la Universitat de Califòrnia, San Diego.<sup>2</sup> CAFE està composta per 135 fotografies de models provinents del sud de Califòrnia que representen la felicitat, la tristesa, la por, la ira, sorpresa i disgust, així com expressions neutrals. Aquesta col·lecció consta de 13 imatges de ira, 21 de disgust, 13 de por, 40 de felicitat, 13 de tristesa, 17 de sorpresa i 18 d'expressions neutrals. Usada també, al mateix article que l'anterior base de dades.
- **KDEF.** La *Karolinska Directed Emotional Faces* és una base de dades creada al Institut Karolinska, Estocolm, Suècia.<sup>3</sup> Conté les sis emocions anteriorment descrites amés d'imatges expressant neutralitat, en total la col·lecció consta de 980 fotografies. Aquesta ha sigut empleada en diferents articles, com per exemple (Calvo i Lundqvist 2008). La KDEF està composta de 140 fotografies per a cada expressió facial. A la Figura 3.2 podem trobar exemples de les imatges que formen part d'aquest conjunt de dades.
- **CFE.** Creada a partir d'imatges pròpies, obtingudes amb la càmera del robot NAO i mitjançant una plataforma web<sup>4</sup> creada per a recaptar imatges de tothom el que volguera participar amb el projecte. Aquesta pàgina web fou desenvolupada mitjançant *Nodejs*, que permet l'execució de programes escrits amb *Python*. Així, la integració del reconeixedor d'emocions, es podria implantar d'una forma molt senzilla, encara que aquest, no forma part dels objectius d'aquest projecte. La CFE

<sup>1</sup>Consulta <http://www.kasrl.org/jaffe.html> per a més detalls de com obtindre JAFFE

<sup>2</sup>Consulta <http://www.cse.ucsd.edu/~gary> per a més detalls de com obtindre CAFE

<sup>3</sup>Consulta <http://www.emotionlab.se/resources/kdef> per a més detalls de com obtindre KDEF

<sup>4</sup>Consulta <http://emotion.animaldesekia.info> per a visualitzar la ferramenta web

conté 168 fotografies de l'expressió de disgust, 168 de por, 197 de sorpresa, 164 de tristesa, 151 de por, 169 de felicitat i 162 de neutralitat.

En la següent secció, en serà necessari seleccionar una col·lecció de les anteriors, per tal d'avaluar i identificar les característiques més significatives que defineixen cada expressió facial. Així, utilitzarem la col·lecció *KDEF* degut a que les fotografies estan capturades baix les mateixes condicions lumíniques, posicionals i amb les expressions ben definides.



Figura 3.2: Subconjunt d'imatges que formen part de *KDEF*

### 3.3 Recerca de característiques

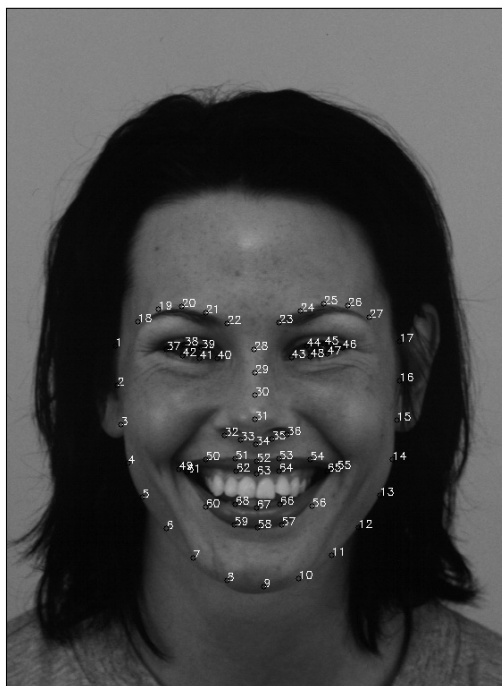
Com ja sabem, l'elecció de característiques, en qualsevol problema de reconeixement de patrons, és una de les tasques més importants dintre del preprocés, anterior al entrenament del model. Una bona elecció de les característiques que defineixen les distintes emocions, ens permetrà obtenir un model més precís i eficient de cara a la posterior classificació de noves mostres alienes al conjunt de dades d'entrenament.

#### 3.3.1. Punts de referència

Per a la classificació de les expressions facials, ens basarem en les distàncies entre diferents punts característics, obtinguts del rostre facial. Per obtenir aquests punts, utilitzarem dos mètodes de la llibreria *Dlib*<sup>5</sup>, denominats detector i predictor. Detector, s'encarregarà de trobar cares de persones en una imatge i predictor d'estimar la seua postura.

<sup>5</sup>Consulta <http://dlib.net/> per a més informació

Aquesta última pren la forma de 68 punts de referència. Estos són punts característics sobre la cara; com les cantonades de la boca, al llarg de les celles, en els ulls, i així successivament com podem observar en la Figura 3.3



**Figura 3.3:** Punts obtinguts amb els mètodes *detector* i *predictor* a partir de la imatge AF33HAS pertanyent al conjunt de dades KDEF

Aquest detector de cares està desenvolupat usant el clàssic Histograma de Gradients Orientats (HGO) com a característiques combinades amb un classificador lineal, una piràmide d'imatge, i l'esquema de detecció de finestra lliscant. L'estimador de postura fou creat a partir de la implementació de l'article de (Kazemi i Sullivan 2014) i fou entrenada amb el conjunt de dades de punts de referència de cares denominat *iBUG 300-W*.<sup>6</sup>

### 3.3.2. Distàncies entre punts

Utilitzant la idea proposada en (Sohail i Bhattacharya 2008) i (Sohail i Bhattacharya 2008) de calcular distàncies usant només uns pocs punts, es decideix estudiar, a partir de totes les possibles distàncies entre els distints punts, quines són les més significatives, per a proporcionar a l'entrenador, un conjunt de mostres amb menys soroll. A més a més, les dades o característiques que obtindrem, les normalitzarem utilitzant la funció que es mostra a continuació:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

per a  $x = (x_1, \dots, x_n)$  i  $z_i$  serà la  $i$ -èsima mostra normalitzada.

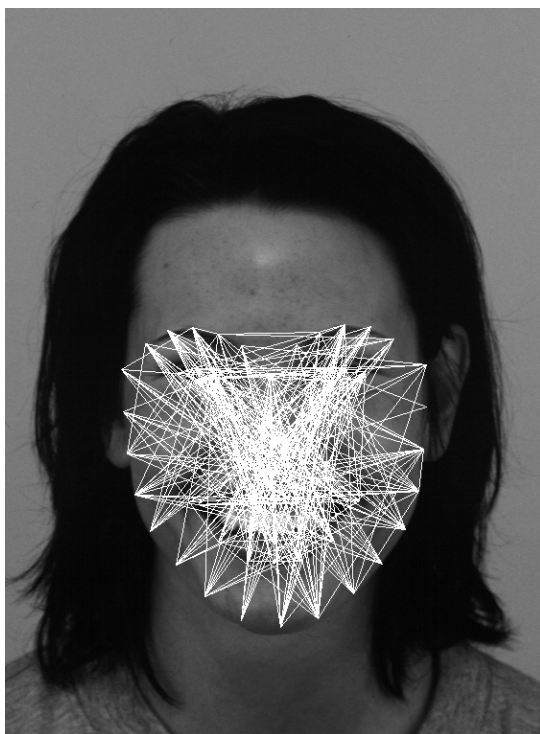
Amb açò, reduïrem el sobreajustament, millorarem la precisió i per últim, disminuïrem el temps d'entrenament del nostre classificador. La distància entre dos punts, on cada punt és una coordenada  $(x,y)$ , que s'utilitzarà per a quantificar la longitud entre els dos punts, és la distància euclídia i es mostra en l'exemple següent:

<sup>6</sup>Consulta <http://ibug.doc.ic.ac.uk/resources/300-W> per a més detalls de com obtenir 300-W

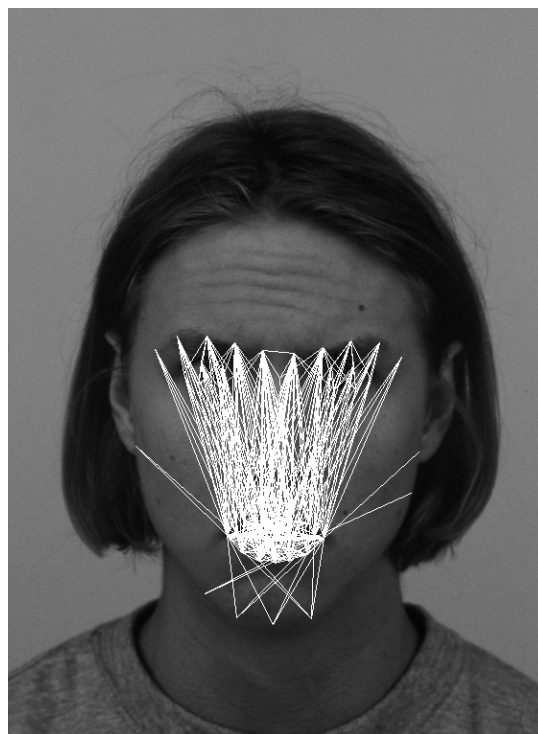


$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.2)$$

Per a determinar quines són les distàncies o característiques més significatives per a discriminar entre les diferents expressions facials, usarem el *Recursive Feature Elimination (RFE)* i *Feature Importance (FE)* (secció 2.1.6 i 2.1.5). Aquest últim, està basat amb arbres aleatoris. Per a fer aquest anàlisi, necessitarem construir un vector amb totes les possibles distàncies (en avant, característiques). En total, es calcularan 2278 característiques (68punts\*67punts/2) per a cada cara. Una vegada calculades per a tot el conjunt de dades, li'l proporcionarem als dos algorismes nombrats anteriorment. Amb *RFE* obtenim una classificació per ordre d'importància de cada característica. Per altra banda, amb el *FE*, s'obté un valor relatiu d'importància per a cada característica. Analitzant la gràfica obtinguda amb *FE*, ens n'adonem que tan sols un 16 % de les característiques calculades (364 distàncies), tal i com es pot veure a la Figura 3.6, ens aporten la major informació a l'hora de distingir entre les diferents expressions facials. Es per açò que, es decideix, per als dos algorismes d'anàlisi, seleccionar el percentatge de característiques anteriorment descrit. A més a més, es proporcionen dues imatges (figures 3.4 i 3.5), on es pot visualitzar quines són les característiques o distàncies seleccionades per ambdós algorismes.



**Figura 3.4:** Distàncies obtingudes amb *RFE* de la imatge *AF33HAS* de la col·lecció *KDEF*



**Figura 3.5:** Distàncies obtingudes amb *FE* de la imatge *AM20AFS* de la col·lecció *KDEF*

### 3.4 Entrenament i experimentació

Per a l'entrenament del nostre predictor, implementarem el model *SVM* proporcionat per *OpenCV* (secció 2.1.1). Aquest model, ens permet configurar els diferents paràmetres, ja siga el paràmetre *C* o *gamma* per a cert tipus de model. També el tipus de nucli a utilitzar. Gràcies a un mètode implementat al model *SVM*, denominat *train\_auto*, només cal proporcionar al model *SVM* quin tipus de nucli es vol utilitzar, així com uns paràmetres

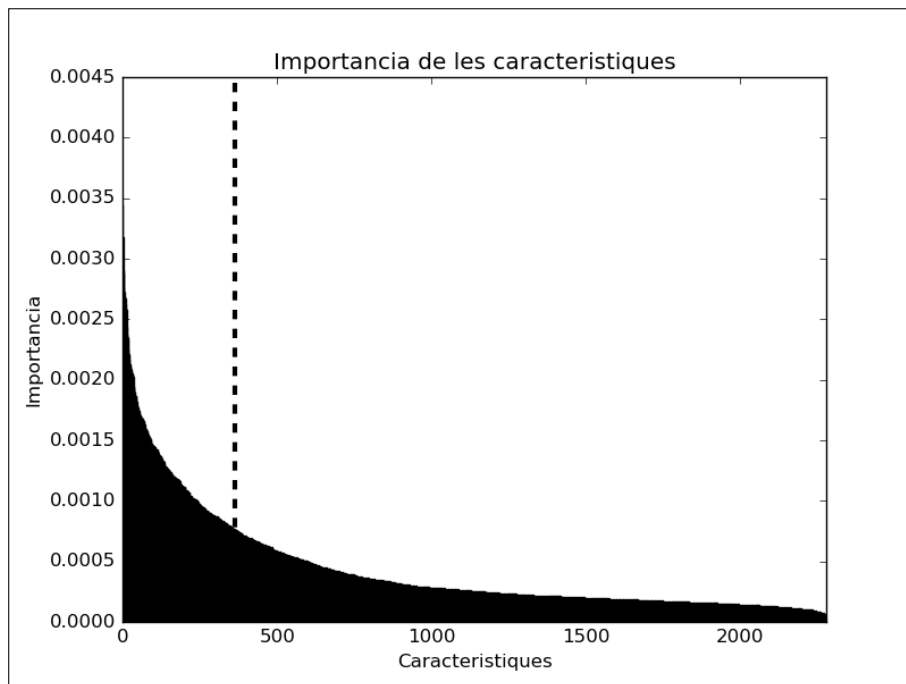


Figura 3.6: Gràfica de la importància relativa de les característiques utilitzant FE

predefinits. El mètode s'encarrega d'obtenir el millor model, caracteritzat pels paràmetres que el construeixen. Així es proposa una experimentació per a obtenir el millor model entre els diferents tipus de nucli i diferents tipus de formulació.

Per tant, tenim dos tipus de formulació *C-Support Vector Classification* (*C-SVC*) i  $\mu$ -*Support Vector Classification* ( $\mu$ -*SVC*), a més a més de quatre tipus de nucli: lineal, sigmoid, polinomial i de funció de base radial (*RBF*). S'estableix una bateria d'experimentació, composta per cent repeticions per a cada combinació possible de tipus de nucli i tipus de formulació, per a obtenir una precisió mitja. Per a validar el nostre model, utilitzarem el mètode de validació creuada *Monte Carlo* (secció 2.5.1), on, per a cada experimentació, seleccionarem aleatòriament un 80% de les mostres com a conjunt d'entrenament i el restant per a test. D'aquesta experimentació, s'obtenen els resultats que es mostren en la Taula 3.1.

Taula 3.1: Precisió obtinguda per a cada model SVM amb el conjunt de dades KDEF

	<i>linial</i>	<i>polinomial</i>	<i>RBF</i>	<i>sigmoid</i>
<i>C - SVC</i>	79,62	78,07	80,22	10,55
$\mu$ - <i>SVC</i>	80,04	79,66	<b>82,04</b>	51,53

A més a més, i una vegada obtinguda la millor configuració possible dels paràmetres del nostre model SVM, entrenarem el nostre model amb distint nombre de característiques per corroborar que reduint les característiques, augmentem la precisió i disminuïm temps de processament. Aquesta comparació la podem trobar a les taules 3.2 i 3.3.



**Taula 3.2:** Precisió obtinguda usant *SVM RBF* per a cada conjunt de dades amb 2278 característiques

	<i>Millor</i>	<i>Pitjor</i>	<i>Mitja</i>
<i>KDEF</i>	89,79	73,72	81,79
<i>CAFE</i>	90,90	36,36	67,27
<i>JAFFE</i>	100	68,18	87,90
<i>CFE</i>	97,91	85,41	<b>92,93</b>

**Taula 3.3:** Precisió obtinguda usant *SVM RBF* per a cada conjunt de dades amb 364 característiques

	<i>Millor</i>	<i>Pitjor</i>	<i>Mitja</i>
<i>KDEF</i>	92,85	75,51	84,06
<i>CAFE</i>	90,90	54,54	75,45
<i>JAFFE</i>	100	68,18	89,72
<i>CFE</i>	100	93,04	<b>96,85</b>

### 3.5 Conclusions

Després d'executar l'entrenament del model, s'adonem que la funció **detector** (secció 3.3.1) tarda una mitja de 13 s aproximadament en executar-se dintre del propi robot *NAO*. D'altra banda, en un ordinador personal amb 16 GB de memòria principal i un processador *Intel Core i5-6600* de 3,30 GHz, el mètode anterior s'executa en 0,2 s de mitja. Per les evidències anteriors, arribem a la conclusió de que és inviable l'execució de la llibreria *Dlib* com a aplicació nativa dintre del robot, donada la poca capacitat de memòria i processament del robot. Afegir que, el mètode **detector** s'executa, tan en l'entrenament com en la classificació, per a recollir els punts característics. És per açò que, es decideix el redisseny del sistema proposat, mitjançant el desenvolupament d'un sistema distribuït, per tal de llevar-li la màxima càrrega de processament possible al *NAO*.

Respecte als resultats obtinguts rere l'experimentació anteriorment explicada (veure Taula 3.1. ), es pot comprovar que, el millor resultat, respecte a precisió obtinguda, pertany al model  $\mu$ -SVC amb nucli *RBF* amb  $\gamma$  valent  $10^{-5}$  i  $\mu$  agafant el valor de  $10^{-3}$ , per tant, quan parlem del model *SVM*, n'estarem parlant del model amb *SVM* amb aquestos atributs.

Per altra banda, observant les taules comparatives (Taula 3.2 i Taula 3.3), podem afirmar que, mitjançant la reducció de les característiques, s'obtenen uns nivells de precisió superiors a quan aquesta reducció no s'aplica, per tant, el model a emprar, estarà basat amb aquestes distàncies seleccionades, en concret 364 distàncies.

A més a més, podem afirmar que el nostre classificador té una precisió del 96,85 % respecte a la col·lecció *CFE*. Cal afegir que, realitzades les proves pertinents agafant les imatges directes amb la càmera del *NAO*, els models entrenats amb les cares del conjunt

*CFE* prediuen amb més precisió l'expressió facial que es representa. L'anterior afirmació és deguda a que les imatges de *CFE* capturen diferents punts de vista i és per això, que, en un entorn dinàmic, la càmera no sempre estarà a la mateixa distància ni posició de l'individu.

Finalment, al següent capítol, tractarem d'abordar la problemàtica principal, el temps de processament de la imatge. Un problema lleu, que solucionarem utilitzant una arquitectura de sistema distinta a la explicada que es basarà en la introducció de nous agents, com s'explicarà en el capítol següent.

# Sistema basat en el model client/servidor

---

En aquest capítol, entrarem en detall en com millorar el sistema anteriorment proposat, per tal de reduir el temps de processament de les imatges, que és el procés que fa de coll de botella en la nostra solució. Aprofitant el redisseny de la proposta, també incorporarem noves tècniques d'aprenentatge automàtic per tal de comparar-les i seleccionar aquella que ens proporcione una millor precisió.

## 4.1 Sistema multiagent

---

Donada la necessitat de computar les imatges en un servidor extern, ja que, el robot *NAO* no té els recursos suficients com per a processar les imatges en temps real, es decideix emprar la plataforma multiagent *SPADE* (secció 2.4) com a plataforma de comunicació entre els diferents agents. A continuació, es descriurà l'arquitectura del sistema multiagent a implementar i les funcionalitats i comportaments de cadascun dels agents. Comencem per explicar els dos agents que interactuaran entre si, per tal d'enviar i rebre les imatges del robot al servidor i viceversa; l'agent *NAO* i l'agent classificador.

### 4.1.1. Agent *NAO*

L'agent *NAO*, allotjat al robot *NAO*, s'encarregarà d'obtenir la imatge de la càmera per tal d'enviar cada segon, una nova captura al servidor. Aquesta petició serà etiquetada amb la ontologia *predict-image*, per tal de que l'agent classificador ho reba en la bústia corresponent. Al mateix temps, també s'encarregarà de rebre resposta des de l'agent classificador. Aquesta vindrà etiquetada amb la ontologia *response-predict* i com a contingut, l'emoció que haja predit el classificador. Per altra banda, les imatges capturades, tindran una grandària de 680px d'amplada per 480px d'altura i l'espai de color en escala de grisos. La imatge capturada, deurà ser codificada en *base64* per tal de ser enviada mitjançant la missatgeria integrada d'*SPADE*.

### 4.1.2. Agent classificador

Un altre membre del sistema multiagent serà l'agent classificador, que s'encarregarà d'esperar peticions de l'agent *NAO*, o qualsevol altre, que demane una petició amb la ontologia del missatge correctament establerta, concretament anomenada *predict-image* i com a contingut del missatge, la imatge codificada en *base64*. Donat aquest punt, tan sols queda

descodificar el contingut del missatge, reconstruir la imatge a partir d'aquest, obtindre els punts i distàncies, reduir les distàncies fins a 364 i classificar-les amb el model prèviament entrenat. Per últim, tornarem el resultat de la predicció, com a contingut del missatge, al agent classificador, amb l'ontologia definida com *response-predict*, tal i com podem veure en la Figura 4.1

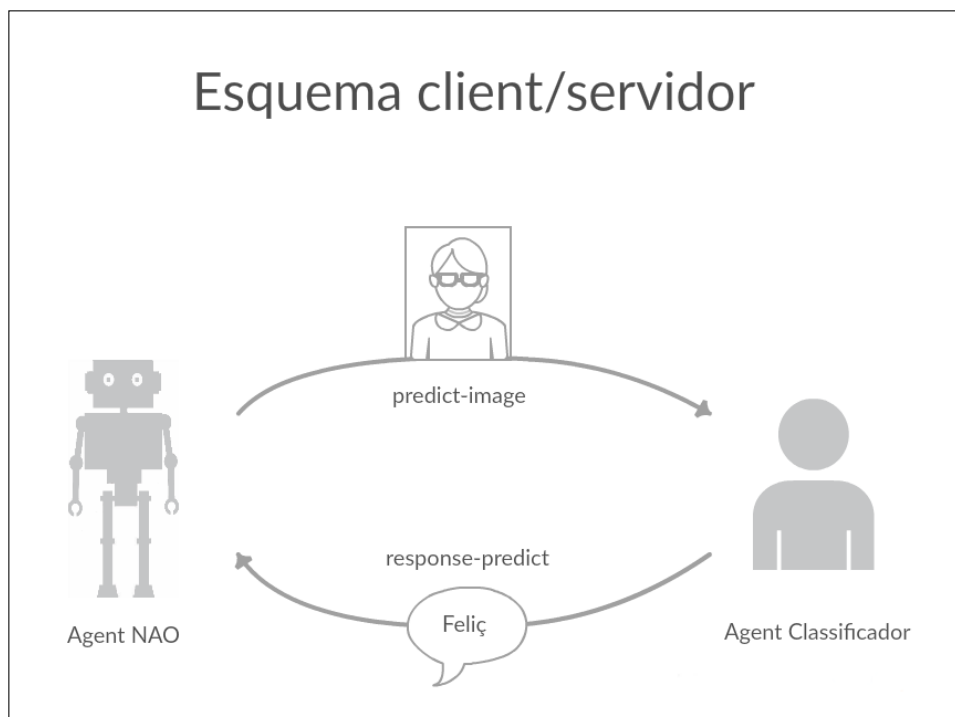


Figura 4.1: Representació gràfica de l'arquitectura client/servidor proposada

### 4.1.3. Estudi del temps de predicció

Una vegada plantejada l'arquitectura del nostre sistema multiagent, cal estudiar com es comporta el temps de predicció respecte a una imatge donada. En l'anterior proposta, el procés emprava una mitja de 13 s. Per a mesurar el temps de predicció, plantejarem una experimentació que es basarà en la predicció de cent imatges, de les quals s'obindrà el temps emprat pel mètode *detector*, de la llibreria *Dlib*, i el temps total emprat des de que s'envia una foto, des del robot, fins a que li arriba la resposta. El resultat serà la mitjana aritmètica dels temps emprats en els diferents processos. Així doncs, per al temps usat pel mètode *detector* serà de **0,3 s** de mitja. El temps mitjà total que transcorre des de que s'envia una foto fins que li arriba la resposta al robot és de **2,1 s**.

## 4.2 Estudi i integració de nous algorismes d'aprenentatge automàtic

Degut a que en el capítol anterior tan sols implementarem els *SVM*, en aquest capítol ens proposem estudiar el comportament de les altres tècniques d'aprenentatge automàtic. Com ja sabem, la llibreria *OpenCV* (secció 2.1) conté moltíssims recursos aprofitables i per tant, objecte d'estudi. Cap la possibilitat de que alguns dels algorismes implementats en la llibreria siguin d'utilitat i milloren la precisió. Per aquests motius, es decideix incorporar nous models al classificador tals com: arbres aleatoris, *k*-veïns, *boosting* i perceptró

multicapa. La solució de la integració passa per crear un model base, que integre els mètodes de cada model, que són bàsicament els mateixos, exceptuant que per a alguns models, és necessària l'adaptació de les dades d'entrada al entrenador i al predictor.

### 4.3 Entrenament i experimentació

En aquesta secció explicarem les diferents experimentacions realitzades per a cada model, així com la configuració dels paràmetres corresponents per tal d'augmentar la precisió. Aquestes experimentacions usaran el conjunt de dades *CFE* ja que és el conjunt de dades que millor s'adapta als nostres requeriments degut a que ofereix distints punts de vista de les expressions facials i aconseguix que els models s'adaptin millor per a treballar en un entorn dinàmic. A més a més, totes les experimentacions seran validades amb el mètode de validació creuada *Monte Carlo* (secció 2.5.1), així doncs, per a cada iteració de les següents experimentacions, seleccionarem aleatòriament un 80% de les mostres com a conjunt d'entrenament i el restant per a test.

#### 4.3.1. Arbres aleatoris

El model d'**arbres aleatoris** (secció 2.1.5), ens permet configurar un paràmetre, la profunditat màxima de recerca. Per tant, per a aconseguir el paràmetre que maximitza la precisió, es proposa una experimentació variant el paràmetre descrit, des d'un nivell de profunditat d'una unitat fins a cinquanta. A la Taula 4.1 podem trobar els cinc resultats amb més bona precisió de l'experimentació exposada, repetint aquest procés cinquanta iteracions i obtenint la mitja d'estos.

**Taula 4.1:** Precisió obtinguda del diferents models d'arbres aleatoris depenent de la profunditat màxima establerta

	86	91	217	218	230
<i>CFE</i>	89,95	<b>90,14</b>	89,76	89,63	88,72

#### 4.3.2. K-veïns més propers

El model basat en **k-veïns**, del anglès *k-nearest neighbors* (*KNN*) (secció 2.1.3), ens dona la possibilitat de configurar quants veïns (*k*) seran necessaris per a classificar una mostra. Per tant, es proposa una experimentació variant el paràmetre descrit, des d'un veí fins a cinc, amb repeticions de cent iteracions. A la Taula 4.2 podem trobar les precisions per a cada model entrenat amb l'algorisme *KNN*.

**Taula 4.2:** Precisió obtinguda dels diferents models basats en *KNN* depenent dels k-veïns emprats

	1	2	3	4	5
<i>CFE</i>	<b>84,69</b>	83,53	83,75	82,29	81,44

### 4.3.3. Boosting

El model basat en tècniques de *boosting* (secció 2.1.4), al igual que el model de arbres aleatoris (secció 4.3.1), ens permet la possibilitat de configurar el màxim nivell de profunditat al que pot arribar durant la creació dels arbres que utilitza per a la predicció del les distintes expressions facials. Per tant, estudiarem el comportament del model respecte a aquest paràmetre executant cent iteracions per a cada entrenament i predicció amb el nivell de profunditat màxima corresponent que minvarà des d'un nivell fins a cinquanta. A la Taula 4.3 podem trobar les cinc millors precisions obtingudes depenent del nivell de profunditat utilitzat.

**Taula 4.3:** Precisió obtinguda del diferents models basats en tècniques de *boosting* depenent de la profunditat màxima establerta

	22	23	24	26	28
CFE	<b>78,27</b>	78,18	77,88	78,13	77,97

### 4.3.4. Perceptró multicapa

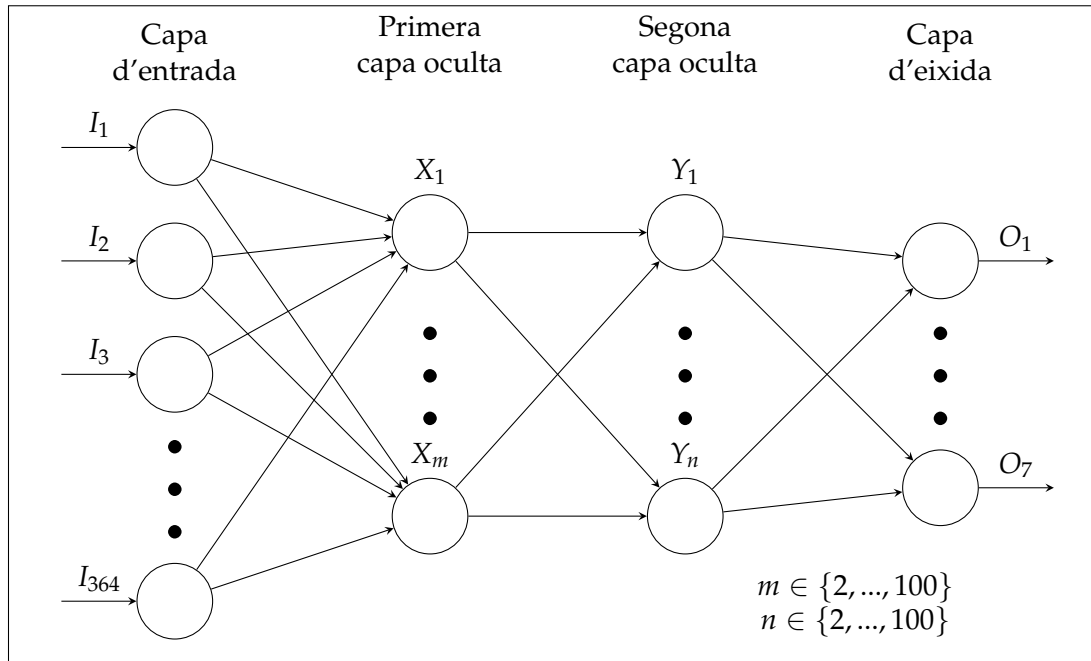
La implementació del **perceptró multicapa** (*MLP*) o xarxes neuronals artificials (secció 2.1.2) de la llibreria *OpenCV*, ens proporciona la facilitat d'incorporar una xarxa neuronal artificial de tal forma que podem configurar l'estructura d'esta. Ens centrarem en estudiar el comportament de la nostra xarxa configurant l'arquitectura amb quatre capes variant el nombre de neurones de les dues capes ocultes. El nombre de neurones en les dues capes ocultes ( $m$  i  $n$ , tal i com es pot comprovar en la Figura 4.2) variarà des de 2 fins a 100 neurones per capa, amb repeticions de deu iteracions. Una vegada realitzada aquesta experimentació, es procedirà a replantejar l'experimentació per als trenta millors resultats obtinguts amb l'anterior experimentació i amb repeticions de cinquanta iteracions per obtenir un resultat més estable. La causa de programar dues experimentacions recau en que el temps de cada entrenament del model és elevat, del ordre de 20s de mitja.

**Taula 4.4:** Precisió obtinguda del diferents models d'*MLP*, entrenats amb diferents configuracions de neurones per capa

	m={ Nombre de neurones en la primera capa oculta }				
	n={ Nombre de neurones en la segona capa oculta }				
	$m = 17$	$m = 16$	$m = 14$	$m = 16$	$m = 15$
	$n = 75$	$n = 52$	$n = 86$	$n = 88$	$n = 76$
CFE	96,73	96,82	97,00	97,08	<b>97,34</b>

## 4.4 Conclusions

Una vegada executades les diferents experimentacions anteriorment descrites, concloem que els millors models a utilitzar són el *SVM* i *MLP* degut a la alta precisió que propor-



**Figura 4.2:** Configuració de la xarxa neuronal artificial usada en l'experimentació del model MLP

cionen. Així, i per ordre de pitjor a millor precisió, podem afirmar que, amb el model de *boosting*, la millor precisió obtinguda és de **78,27 %**, per a *KNN* és de **84,69 %**, per al model de *arbres aleatoris* és de **90,14 %** i per a *MLP* és de **97,34 %**. En comparació amb el capítol anterior, on per al model *SVM* obteníem una precisió del **96,85 %**, podem afirmar que el model d'*MLP* és quasi **un punt més precís** que *SVM*.

També caldria afegir que amb aquesta millora, el procés que abans tardava 13 s, el detector de cares, ara tan sols tarda 0,3 s degut a que l'extracció de cares i de punts característics, a més a més de la predicció de l'emoció, es realitza en un servidor extern i no en el robot, com es comentava en el capítol anterior. Però cal afegir, que, donat que la comunicació entre els diferents agents es realitza mitjançant Internet, el temps que s'emptra en la comunicació és de 1,85 s de mitja. Per tant, el temps total en realitzar una predicció des de que s'obté la foto fins que el sistema torna una resposta és de 2,1 s de mitja.

No obstant, amb aquesta aproximació, la predicció d'algunes expressions facials podrien ser millorables respecte a la seua precisió. Aquesta recerca, que es basarà en extreure i calcular la precisió per a cada emoció i proposar una nova arquitectura de sistema, es veurà en el capítol següent.





---

---

## CAPÍTOL 5

# Sistema basat en agents classificadors

---

En aquest capítol es planteja la hipòtesi de la possible millora, respecte a la precisió, del nostre classificador mitjançant l'adició d'un conjunt de classificadors amb característiques diferenciades. Per a aconseguir aquests predictors, primerament, analitzarem com es comporten les diferents expressions facials depenent del model utilitzat. Finalment, construirem una nova arquitectura de sistema on tinga cabuda aquest nou conjunt de predictors.

### 5.1 Anàlisi de la sensibilitat de cada expressió facial respecte al model utilitzat

---

En aquesta secció, analitzarem com es comporten els models estadístics depenent de l'emoció a predir. Per a realitzar l'anàlisi, plantejarem un conjunt d'experimentació on avaluarem la taxa de verdaters positius o sensibilitat de cada expressió facial. Per a l'experimentació, en cada iteració, usarem el 80% de les mostres per a entrenar un model i el 20% restant les emprarem per a classificar-les i obtindre la matriu de confusió. Aquest procés serà repetit cent iteracions per a cada model donat. Amb les matrius de confusió obtingudes per a cada model, calculem la sensibilitat dels nostres classificadors, respecte a cada emoció. Els resultats obtinguts rere aquesta experimentació es mostren en la Taula 5.1.

**Taula 5.1:** Taula de la precisió obtinguda per model i emoció

	<i>felicitat</i>	<i>neutralitat</i>	<i>disgust</i>	<i>por</i>	<i>sorpresa</i>	<i>tristesa</i>	<i>ira</i>
<i>SVM</i>	98,32	95,04	99,88	93,10	94,54	94,52	100
<i>Knn</i>	86,41	77,86	95,35	76,23	86,68	74,23	99,30
<i>Rtrees</i>	91,07	87,35	93,54	83,83	90,09	78,08	99,35
<i>Boost</i>	88,88	72,35	85,90	67,51	73,88	62,22	87,60
<i>MLP</i>	99,10	93,85	100	88,68	97,38	95,83	99,37

Taula 5.2: Mètriques obtingudes per als models SVM i MLP de les diferents emocions

		felicitat	neutralitat	disgust	por	sorpresa	tristesa	ira
TP	svm	1668	1619	1640	1368	1597	1550	1654
	mlp	1636	1582	1694	1322	1706	1554	1646
FN	svm	25	68	2	104	83	122	0
	mlp	29	43	6	141	45	91	5
FP	svm	54	148	6	40	72	82	2
	mlp	38	94	11	46	94	63	14
TN	svm	9753	9665	9852	9988	9748	9746	9844
	mlp	9797	9781	9789	9991	9655	9792	9835
TPR	svm	98,52	95,96	99,87	92,93	95,05	92,70	100
	mlp	98,25	97,35	99,64	90,36	97,43	94,46	99,69
SPC	svm	99,44	98,49	99,93	99,60	99,26	99,16	99,97
	mlp	99,61	99,04	99,88	99,54	99,03	99,36	99,85
PPV	svm	96,86	91,62	99,63	97,15	95,68	91,97	98,87
	mlp	97,72	94,39	99,35	96,63	94,77	96,10	99,15
NPV	svm	99,74	99,30	99,97	98,96	99,15	98,76	100
	mlp	99,70	99,56	99,93	98,60	99,53	99,07	99,94
FPR	svm	0,55	1,50	0,06	0,39	0,73	0,83	0,02
	mlp	0,38	0,95	0,11	0,45	0,96	0,63	0,14
FDR	svm	3,13	8,37	0,36	2,84	4,31	5,02	0,12
	mlp	2,27	5,60	0,64	3,36	5,22	3,89	0,84
FNR	svm	1,47	4,03	0,12	7,06	4,94	7,29	0
	mlp	1,74	2,64	0,35	9,63	2,56	5,53	0,30
ACC	svm	99,31	98,12	99,93	98,74	98,65	98,22	99,98
	mlp	99,41	98,80	99,85	98,37	98,79	98,66	99,83

Donat que els millors resultats s'obtenen per als models MLP i SVM, farem un anàlisi més profund sobre aquests models, obtenint totes les mètriques explicades en la secció 2.5.2.

### 5.1.1. Anàlisi de la matriu de confusió

Per a realitzar un anàlisi amb més profunditat del comportament de cada emoció, sols emprarem el models MLP i SVM, dels quals obtindrem un conjunt de mètriques que ens permetran avaluar-los amb més exactitud. Per tant, l'experimentació constarà de cent iteracions amb les mateixes condicions anteriors, és a dir, el 80% del conjunt de mostres s'utilitzarà per al entrenament mentre que el restant, s'utilitzarà per a la classificació. Els resultats d'aquesta experimentació es poden consultar en la Taula 5.2

Si comparem els resultats, podem observar com el **valor predictiu negatiu (NPV)** és superior al 99% per a totes les emocions excepte per a *disgust*, que es del 98,96% per a SVM i del 98,60% per a MLP. A més a més, per a la *tristesa* i usant el model SVM s'obté un 98,76%.

El **valor predictiu positiu (PPV)** o la proporció de casos positius que són correctament identificats és variable per a cada expressió facial. Per ordre descendent, el PPV més alt correspon al *disgust* amb més del 99,58% usant SVM. A continuació la *ira* amb el 99,17% per a MLP. Seguidament la *felicitat* amb 97,95% usant MLP. Per a la *por* i SVM un 95,54%. Per a la sorpresa obtenim un 94,47% emprant el model SVM. Finalment per a la tristesa

i la neutralitat obtindríem un 92,53 % i 90,20 % respectivament, usant ambdós el model *MLP*.

Per altra banda, si ens centrem en la **especificitat** (SPC), podem observar com tots els casos superen el 99% exceptuant *disgust* amb el model *SVM* del qual s'obté un 98,49 %,

Per últim, la **sensibilitat** (TPR), coneguda en la literatura anglesa com a *Recall*, que n'obtenim per a la *ira* assoleix el 100%, mentre que per a *disgust* obtenim un 99,87 % i en descens fins a arribar a la *por* usant *SVM* que és del 92,93 %.

En resum, els nostres classificadors són molt bons predient la no pertinença a una classe, però en canvi, no ho són tant per a classificar la pertinença a certa classe. Per aquest motiu, proposarem i estudiarem un sistema de votacions, capaç d'aglutinar els resultats dels distints models, per tal d'augmentar la predicció de la classe *por*.

## 5.2 Sistema de votacions

---

Tal com dèiem en la secció anterior, la predicció de la classe *por* és bastant baixa respecte a les altres emocions. Seguint amb la hipòtesi de poder millorar la predicció del reconeixedor, a partir d'un conjunt de classificadors, entrenarem quatre models distints i, amb la predicció que ens proporcionen, realitzarem una votació per a seleccionar la classe resultant.

El sistema de votacions, recollirà la predicció dels distints models i realitzarà una votació entre les respostes obtingudes. A més a més, el procés de votació consistirà en comptabilitzar els vots de tots els models entrenats per tal d'elegir quina és la predicció final. La classe que més vots tinga serà la classe escollida, excepte en cas d'empat, que la decisió es prendrà aleatòriament entre les dos primeres classes amb més vots. Aquest sistema de votacions serà emprat en el sistema multiagent que s'explica en el següent apartat.

## 5.3 Sistema multiagent amb agents classificadors

---

Tal i com s'explica en la secció anterior, per a la integració d'aquests nous predictors, afegirem quatre agents que implementaran cadascun dels quatre models anteriorment descrits. A més a més, mitjançant l'adició d'un agent coordinador, podrem aglutinar i processar la resposta dels quatre agents. El processament d'aquesta resposta es basarà en el sistema de votacions anteriorment descrit (secció 5.2). A continuació, formalitzarem els agents que formaran part del sistema, així com les funcionalitats que implementen. En la Figura 5.1 podem observar com quedaria configurat el sistema.

### 5.3.1. Agent NAO

L'agent *NAO* s'encarregarà, com en l'anterior proposta, de capturar imatges i enviar-les a un altre agent. Aquest agent, en aquests cas, es tracta del agent coordinador. L'ontologia utilitzada per a l'enviament d'aquest missatge serà la mateixa que l'utilitzada anteriorment, *predict-image*. Finalment, aquest agent també implementarà la funcionalitat de rebre missatges del agent coordinador, etiquetats amb la ontologia *response-predict*, on el contingut serà la classe a la qual es classificarà la imatge enviada anteriorment.

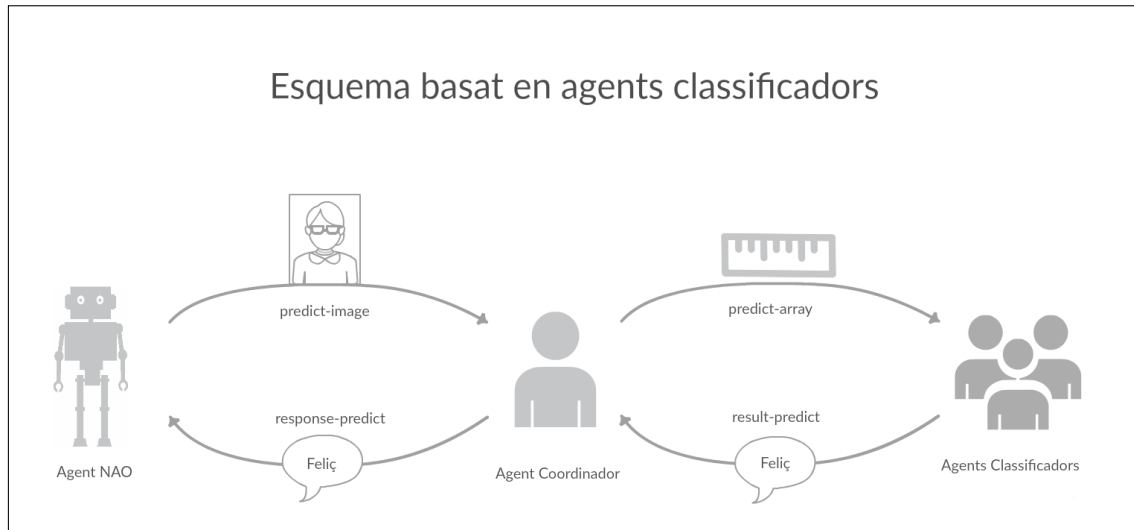


Figura 5.1: Representació gràfica del sistema multiagent proposat.

### 5.3.2. Agent coordinador

Continuant amb l'especificació dels agents, en aquesta secció es presenta l'agent coordinador. Aquest agent s'encarregarà, primerament, de rebre un missatge, correctament etiquetat amb l'ontologia *predict-image*, on el contingut serà una imatge. Una vegada rebuda aquesta imatge, serà processada amb el mètode *detector* i, si escau, es calcularan les distàncies entre els punts característics obtinguts. En cas contrari, s'enviarà al agent *NAO* que no s'ha trobat cap cara. Si s'ha trobat una cara i, per tant, s'han calculat unes distàncies, aquestes seran enviades a tots els agents classificadors registrats als sistema. Les distàncies s'enviaran encapsulades dintre d'un missatge que serà etiquetat amb la ontologia *predict-array*.

A més a més, per a registrar els agents classificadors, l'agent coordinador implementarà la funcionalitat de rebre peticions de registre, on l'ontologia usada serà *login-classificador*. Finalment, i com a última funcionalitat, serà capaç de rebre totes les respostes dels diferents agents classificadors. Aquestes respostes seran tractades com a vots a una determinada classe o emoció. Aquests vots es rebran amb l'ontologia *result-predict* i es comptabilitzaran com s'ha explicat en la secció 5.2.

### 5.3.3. Agents classificadors

L'últim grup d'agents que formaran part del nostre sistema multiagent és l'agent classificador. En la nostra proposta s'especifiquen quatre agents classificadors, què, tal i com diu la paraula, s'encarregaran de classificar a quina emoció pertanyen les distàncies rebudes. L'única diferència entre ells serà el model que usaran per predir l'emoció. Per altra banda, les altres funcionalitats seran exactament les mateixes. Aquestes funcionalitats es basaran en la implementació de la petició de *login-classificador*, tan prompte com l'agent comence a executar-se. Per altra banda, una vegada rebudes les distàncies provinents de l'agent coordinador, amb l'ontologia *predict-array*, l'agent s'encarregarà de classificar-les, mitjançant el model entrenat, i respondre un missatge on el contingut serà la classe o emoció predita. Aquest missatge s'etiquetarà amb l'ontologia *result-predict* i serà enviat a l'agent coordinador.

## 5.4 Experimentació

Per a poder avaluar aquest nou model, format pel conjunt dels quatre models, plantejarem una sèrie d'experimentacions que es basaran en realitzar cent iteracions i entrenar quatre models distints, que seran usats pels agents classificadors. Abans de realitzar l'entrenament en cada iteració, realitzarem diverses particions en les mostres d'entrenament. Primerament, el conjunt de mostres totals serà dividit en dues particions, concretament una del 80% i la restant del 20%. Reservarem aquesta última com a conjunt de test. La partició més gran, la dividirem en dues particions. Aquestes particions estaran parcialment interseccionades, és a dir, les particions compartiran un 40% de les mostres d'entrenament. L'elecció d'aquest format de les particions, s'ha pres mitjançant una experimentació de cent iteracions. En cada iteració, s'entrenaran els dos models (*SVM* i *MLP*) i s'obtidran resultats de predicció per a determinar la grandària òptima en la qual maximitzen els valors de *PPV* i *TPR*.

Per a realitzar aquestes particions, seleccionarem el 70% (56% del total de mostres) de les primeres mostres del conjunt més gran per al primer conjunt d'entrenament. Per a la segona partició utilitzarem el 70%, de les últimes mostres provinents del conjunt gran. En la Figura 5.2 es mostra gràficament com es divideixen les distintes particions. Per tant, obtindrem tres particions; la partició del 20%, reservada per a la classificació, i dos conjunts que ens serviran per a l'entrenament.

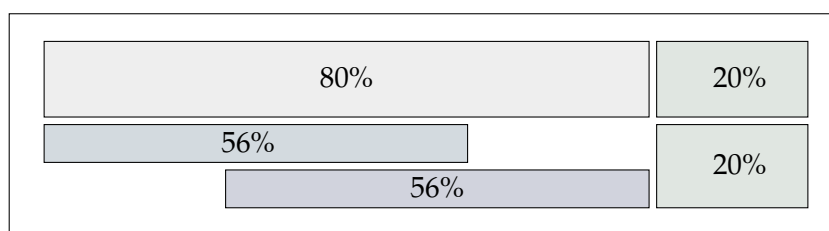


Figura 5.2: Representació gràfica del format de les particions utilitzat per a l'experimentació.

Amb el primer dels dos conjunts d'entrenament anteriorment formalitzats, entrenarem un parell de models. En concret, un dels dos conjunts serà usat en l'entrenament del model amb l'algorisme *SVM* i l'altre amb *MLP*. Amb el segon conjunt restant realitzarem el mateix procediment de tal forma que obtindrem dos classificadors més que faran un total de quatre.

Els resultats dels quatre models es mostren en les taules comparatives (Taula 5.3 i Taula 5.4). Cada taula correspon als resultats d'un model, *SVM* o *MLP*, comparats amb el model format pel conjunt de classificadors (etiqueta *MIX*).

Finalment, tan sols ens queda comparar els resultats obtinguts pel model que acabem de construir amb la proposta anterior. Per això, analitzarem els diferents valors de *TPR*, *SPC*, *PPV* i *NPV*.

Si ens centrem en la **taxa de veritables positius** (*TPR*), ens n'adonem que, a comparació dels resultats obtinguts pels submodels entrenats, la millora és bastant significativa. El model *MIX*, el que hem plantejat en aquest capítol, millora tots els resultats dels seus submodels. En canvi, si comparem el model *MIX* amb els models *SVM* i *MLP* (models monolítics), que hem obtingut en la secció 5.1.1, comprovem com els resultats no varien significativament per a les emocions *felicitat*, *disgust* i *ira*. En canvi, per a les restants emocions, aconseguim apropar-nos al valor més alt proporcionats pels dos models monolítics.

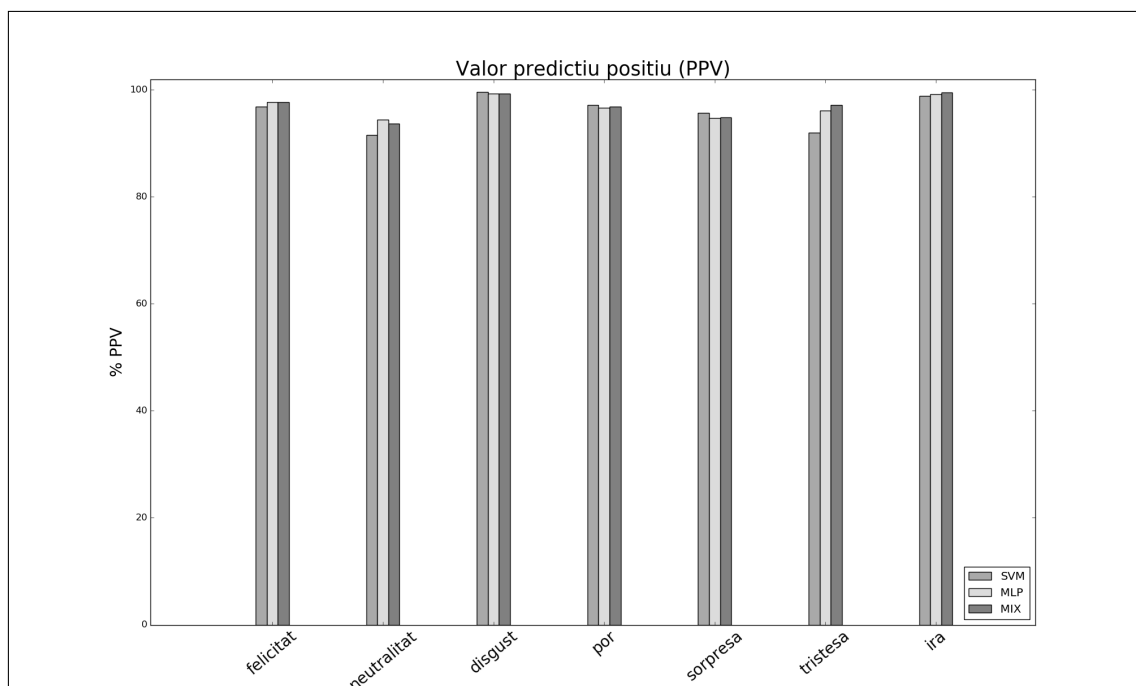
Taula 5.3: Mètriques obtingudes amb el primer conjunt de dades d'entrenament

		felicitat	neutralitat	disgust	por	sorpresa	tristesa	ira
TP	svm	3328	3041	3272	2644	3177	3082	3373
	mlp	3377	3056	3270	2627	3224	3013	3361
	mix	3389	3156	3291	2681	3258	3164	3374
FN	svm	117	211	27	292	180	251	5
	mlp	68	196	29	309	133	320	17
	mix	56	96	8	255	99	169	4
FP	svm	114	351	11	150	193	229	35
	mlp	84	353	34	139	222	193	47
	mix	78	213	25	87	177	92	15
TN	svm	19441	19397	19690	19914	19450	19438	19587
	mlp	19471	19395	19667	19925	19421	19474	19575
	mix	19477	19535	19676	19977	19466	19575	19607
TPR	svm	96,60	93,51	99,18	90,05	94,64	92,47	99,85
	mlp	98,03	93,97	99,12	89,48	96,04	90,40	99,50
	mix	98,37	97,05	99,76	91,31	97,05	94,93	99,88
SPC	svm	99,42	98,22	99,94	99,25	99,02	98,84	99,82
	mlp	99,57	98,21	99,83	99,31	98,87	99,02	99,76
	mix	99,60	99,01	99,87	99,57	99,10	99,53	99,92
PPV	svm	96,69	89,65	99,66	94,63	94,27	93,08	98,97
	mlp	97,57	89,65	98,97	94,97	93,56	93,98	98,62
	mix	97,75	93,68	99,25	96,86	94,85	97,17	99,56
NPV	svm	99,40	98,92	99,86	98,55	99,08	98,73	99,97
	mlp	99,65	99,00	99,85	98,47	99,32	98,38	99,91
	mix	99,71	99,51	99,96	98,74	99,49	99,14	99,98
FPR	svm	0,58	1,78	0,06	0,75	0,98	1,16	0,18
	mlp	0,43	1,79	0,17	0,69	1,13	0,98	0,24
	mix	0,40	1,08	0,13	0,43	0,90	0,47	0,08
FDR	svm	3,31	10,35	0,34	5,37	5,73	6,92	1,03
	mlp	2,43	10,35	1,03	5,03	6,44	6,02	1,38
	mix	2,25	6,32	0,75	3,14	5,15	2,83	0,44
FNR	svm	3,40	6,49	0,82	9,95	5,36	7,53	0,15
	mlp	1,97	6,03	0,88	10,52	3,96	9,60	0,50
	mix	1,63	2,95	0,24	8,69	2,95	5,07	0,12
ACC	svm	99,00	97,56	99,83	98,08	98,38	97,91	99,83
	mlp	99,34	97,61	99,73	98,05	98,46	97,77	99,72
	mix	99,42	98,66	99,86	98,51	98,80	98,87	99,92

Taula 5.4: Mètriques obtingudes amb el segon conjunt de dades d'entrenament

		felicitat	neutralitat	disgust	por	sorpresa	tristesa	ira
TP	svm	3300	3038	3277	2641	3181	3074	3376
	mlp	3441	3197	3289	2831	3313	3269	3374
	mix	3389	3156	3291	2681	3258	3164	3374
FN	svm	145	214	22	295	176	259	2
	mlp	3449	3307	3309	3041	3401	3397	3382
	mix	56	96	8	255	99	169	4
FP	svm	123	346	14	133	204	260	33
	mlp	2652	4475	2093	4069	3087	4456	2454
	mix	78	213	25	87	177	92	15
TN	svm	19432	19402	19687	19931	19439	19407	19589
	mlp	36458	35021	37309	36059	36199	34878	36790
	mix	19477	19535	19676	19977	19466	19575	19607
TPR	svm	95,79	93,42	99,33	89,95	94,76	92,23	99,94
	mlp	49,94	49,15	49,85	48,21	49,34	49,04	49,94
	mix	98,37	97,05	99,76	91,31	97,05	94,93	99,88
SPC	svm	99,37	98,25	99,93	99,34	98,96	98,68	99,83
	mlp	93,22	88,67	94,69	89,86	92,14	88,67	93,75
	mix	99,60	99,01	99,87	99,57	99,10	99,53	99,92
PPV	svm	96,41	89,78	99,57	95,21	93,97	92,20	99,03
	mlp	56,47	41,67	61,11	41,03	51,77	42,32	57,89
	mix	97,75	93,68	99,25	96,86	94,85	97,17	99,56
NPV	svm	99,26	98,91	99,89	98,54	99,10	98,68	99,99
	mlp	91,36	91,37	91,85	92,22	91,41	91,12	91,58
	mix	99,71	99,51	99,96	98,74	99,49	99,14	99,98
FPR	svm	0,63	1,75	0,07	0,66	1,04	1,32	0,17
	mlp	6,78	11,33	5,31	10,14	7,86	11,33	6,25
	mix	0,40	1,08	0,13	0,43	0,90	0,47	0,08
FDR	svm	3,59	10,22	0,43	4,79	6,03	7,80	0,97
	mlp	43,53	58,33	38,89	58,97	48,23	57,68	42,11
	mix	2,25	6,32	0,75	3,14	5,15	2,83	0,44
FNR	svm	4,21	6,58	0,67	10,05	5,24	7,77	0,06
	mlp	50,06	50,85	50,15	51,79	50,66	50,96	50,06
	mix	1,63	2,95	0,24	8,69	2,95	5,07	0,12
ACC	svm	98,83	97,57	99,84	98,14	98,35	97,74	99,85
	mlp	86,74	83,08	88,26	84,54	85,90	82,93	87,31
	mix	99,42	98,66	99,86	98,51	98,80	98,87	99,92

Per altra banda, els resultats d'**especificitat** (SPC) i del **valor predictiu negatiu** (NPV) del model *MIX* no varien significativament dels resultats dels seus submodels. Tampoc ho fan en comparació amb els models monolítics.



**Figura 5.3:** Gràfica on es comparen els valors de *PPV* de cada emoció i per a cada model.

Finalment, si comparem el **valor predictiu positiu** (*PPV*) dels resultats obtinguts pel model *MIX* amb els dels seus submodels, la diferència no és significativa. No obstant, si la comparació la realitzem entre el model *MIX* i els models monolítics (Figura 5.3), la millora sí és significativa degut a que, per a tots els casos, el *PPV* del model *MIX* s'acosta al valor més alt obtingut pels models monolítics.

## 5.5 Conclusions

Com a resum d'aquest capítol, hem construït quatre models diferents que, independentment, emetran una predicció. Aquesta predicció serà recollida pel sistema de votacions i processada, per tal d'arribar a un consens on es trie la classe a la qual prediem que pertany.

Amb la nostra proposta, hem aconseguit millorar el *TPR* i *PPV* mitjançant el model *MIX* en comparació al conjunt dels models monolítics, concretament per a les emocions *neutralitat*, *por*, *sorpresa* i *tristesa*. No obstant, es basava en la hipòtesi de la possible millora en la classificació de l'expressió facial *por*. Per tant, amb aquesta aproximació no hem aconseguit el nostre propòsit degut a que hem igualat el resultat al millor obtingut pel model monolític.

Per a acabar amb aquesta conclusió, farem un especial èmfasi en la millora respecte a l'escalabilitat i disponibilitat del sistema, ja que, amb aquesta implementació hem aconseguit distribuir els càlculs entre els distints agents i replicar els distints agents per proporcionar-li una major disponibilitat a més de proporcionar una implementació capaç de ser escalable en un moment donat.



# Conclusions i treball futur

---

En el present capítol exposarem les conclusions obtingudes rere l'estudi, implementació i avaluació d'un predictor d'expressions facials a través del robot *NAO* i baix una plataforma multiagent. A més a més, i per finalitzar, relatarem el treball futur.

### 6.1 Conclusions

---

En este projecte final de grau s'ha formalitzat, avaluat i implementat un sistema multiagent per a la detecció d'emocions en un robot bípede. Per a complir l'objectiu principal ha sigut necessari estudiar la naturalesa de les característiques, en aquest cas les distàncies. Un total de tres-centes seixanta-quatre característiques han estat seleccionades per a intervindre en el càlcul del nostre model estadístic.

A més a més, s'han avaluat i implementat les tècniques d'aprenentatge automàtic més usades en l'actualitat, obtenint com a millors models classificadors els *SVMs* i *MLPs*. A més a més, i gràcies al sistema multiagent, hem aconseguit que el temps mitjà de resposta per part del robot siga de 2,1 s, un temps raonable per a que la percepció de la interacció per part de la persona siga bastant agradable.

Per tal de validar les nostres aproximacions, s'han obtingut unes mètriques mitjançant una tècnica de validació creuada aleatòria. Els resultats obtinguts certifiquen el correcte funcionament de la nostra aproximació així com les altes mètriques obtingudes, obtenint una taxa de precisió per a cada emoció superior al 98%.

A nivell de desenvolupament, la implementació realitzada en aquest projecte ha suposat tot un repte per la gran quantitat de tecnologies i algorismes que s'han hagut d'integrar. Es tracta de tecnologies que han sigut desenvolupades recentment (*OpenCV*, *Dlib*, *SPADE*) i que estan en plena expansió degut a la gran acceptació que estant tenint per part de la comunitat de programadors. Açò que a priori suposa un avantatge, afegeix doble dificultat al desenvolupament de l'aplicació:

- El desconeixement de cadascuna de les ferramentes ha suposat un desafiament.
- La poca informació que es troba disponible a més està subjecta a canvis de versió, ja que les llibreries estan en desenvolupament.

Encara que s'ha tractat d'un projecte molt ambiciós que, pels seus requisits, precisaria d'un esforç considerable per part d'un equip de treball, estem molt satisfets del bon resultat obtingut, respecte a les altes mètriques obtingudes per a les diferents expressions facials analitzades. A més a més pot constituir una base per a futures línies de desenvolupament.

lupament i investigació. Finalment, en el següent enllaç web <sup>1</sup> es troba el repositori on podem consultar i descarregar tot el codi font realitzat en el present treball.

## 6.2 Treball futur

---

L'actual projecte s'ha realitzat pensant en oferir a l'usuari una infraestructura i uns mètodes, per tal de reconèixer l'emoció de les persones. Moltes de les millores i ampliacions han quedat fora de l'abast d'aquest projecte a causa fonamentalment a l'enorme envergadura del mateix. Per tant, no és d'estranyar que l'eina, conforme ha estat implementada i desenvolupada, presente una infinitat de millores i de possibles ampliacions. A continuació, enumerem algunes de les funcionalitats que poden millorar o completar de l'aplicació desenvolupada:

- **Modularitzar l'aplicació.** Per tal d'oferir al públic general aquesta aproximació, serà necessària la modularització en forma de paquet, per tal de ser usable fàcilment com a llibreria.
- **Especificació d'una API.** Degut al punt anterior, també serà necessària l'especificació d'una API, per tal de proporcionar als usuaris les funcions i mètodes implementats.
- **Autoaprenentatge.** Implementar l'autoaprenentatge dels distints models, per tal de millorar la taxa de precisió amb el temps.
- **Integració en una aplicació real.** Implementar aquesta proposta en una aplicació del món real per tal d'avaluar i millorar el seu rendiment.
- **Integració amb la Microsoft Emotion API.** Implementar un agent que interactue en la plataforma web de Microsoft per tal de reforçar el sistema de predicció.
- **Integració de diferents bases de dades.** La incorporació de noves dades d'entrenament, junt a la implementació dels agents que puguen usar-les, seria una millora considerable, ja que augmentaria la precisió del sistema de reconeixement.

---

<sup>1</sup>[https://bitbucket.org/chverma\\_/lendmarkemotiondetection](https://bitbucket.org/chverma_/lendmarkemotiondetection)

# Bibliografia

---

- Alpaydin, E. (2004). *Introduction to Machine Learning*. Adaptive computation and machine learning. MIT Press. ISBN: 9780262012119 (v. la pàg. 5).
- Altman, N. S. (1992). "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression". en. A: *The American Statistician* 46.3, pàg. 175 - 185. ISSN: 0003-1305, 1537-2731. DOI: [10.1080/00031305.1992.10475879](https://doi.org/10.1080/00031305.1992.10475879) (v. la pàg. 7).
- Bavkar, S. S., J. S. Rangole i V. U. Deshmukh (2015). "Article: Geometric Approach for Human Emotion Recognition using Facial Expression". A: *International Journal of Computer Applications* 118.14. Full text available, pàg. 17 - 22 (v. la pàg. 10).
- Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer. ISBN: 9780387310732 (v. la pàg. 5).
- Calvo, Manuel G. i Daniel Lundqvist (2008). "Facial expressions of emotion (KDEF): Identification under different display-duration conditions". A: *Behavior Research Methods* 40.1, pàg. 109 - 115. ISSN: 1554-351X, 1554-3528. DOI: [10.3758/BRM.40.1.109](https://doi.org/10.3758/BRM.40.1.109) (v. la pàg. 18).
- Dailey, Matthew N. et al. (2010). "Evidence and a computational explanation of cultural differences in facial expression recognition." A: *Emotion* 10.6, pàg. 874 (v. la pàg. 18).
- Fawcett, Tom (2006). "An introduction to {ROC} analysis". A: *Pattern Recognition Letters* 27.8. {ROC} Analysis in Pattern Recognition, pàg. 861 - 874. ISSN: 0167-8655. DOI: [10.1016/j.patrec.2005.10.010](https://doi.org/10.1016/j.patrec.2005.10.010) (v. la pàg. 15).
- Jarkiewicz, Julia, Rafał Kocielnik i Krzysztof Marasek (2009). "Anthropometric Facial Emotion Recognition". A: *Human-Computer Interaction. Novel Interaction Methods and Techniques*. Ed. de David Hutchison et al. Vol. 5611. Berlin, Heidelberg: Springer Berlin Heidelberg, pàg. 188 - 197. ISBN: 978-3-642-02576-1 978-3-642-02577-8 (v. la pàg. 10).
- Kazemi, V. i J. Sullivan (2014). "One millisecond face alignment with an ensemble of regression trees". A: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pàg. 1867 - 1874. DOI: [10.1109/CVPR.2014.241](https://doi.org/10.1109/CVPR.2014.241) (v. la pàg. 20).
- Louis, Vincent i Thierry Martinez (2006). "An Operational Model for the FIPA-ACL Semantics". A: *Agent Communication II*. Ed. de Frank P. M. Dignum, Rogier M. van Eijk i Roberto Flores. Vol. 3859. Berlin, Heidelberg: Springer Berlin Heidelberg, pàg. 1 - 14. ISBN: 978-3-540-68142-7 978-3-540-68143-4 (v. la pàg. 14).

- Low, Cheng Yee et al. (2013). "Strategy planning for collaborative humanoid soccer robots based on principle solution". en. A: *Production Engineering* 7.1, pàg. 23 - 34. ISSN: 0944-6524, 1863-7353. DOI: [10.1007/s11740-012-0416-4](https://doi.org/10.1007/s11740-012-0416-4) (v. la pàg. 11).
- Murphy, K.P. (2012). *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning series. MIT Press. ISBN: 9780262018029 (v. la pàg. 5).
- Russell, S.J. i P. Norvig (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall. ISBN: 9780136042594 (v. la pàg. 12).
- Simon, Richard (2007). "Resampling Strategies for Model Assessment and Selection". A: *Fundamentals of Data Mining in Genomics and Proteomics*. Ed. de Werner Dubitzky, Martin Granzow i Daniel Berrar. Boston, MA: Springer US, pàg. 178. ISBN: 978-0-387-47509-7. DOI: [10.1007/978-0-387-47509-7\\_8](https://doi.org/10.1007/978-0-387-47509-7_8) (v. la pàg. 15).
- Sohail, Abu Sayeed Md i Prabir Bhattacharya (2008). "Detection of Facial Feature Points Using Anthropometric Face Model". en. A: *Signal Processing for Image Enhancement and Multimedia Processing*. Ed. d'Ernesto Damiani et al. Multimedia Systems and Applications Series 31. Springer US, pàg. 189 - 200. ISBN: 978-0-387-72499-7 978-0-387-72500-0. DOI: [10.1007/978-0-387-72500-0\\_17](https://doi.org/10.1007/978-0-387-72500-0_17) (v. les pàg. 10, 20).
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. Wiley. ISBN: 9780470519462 (v. la pàg. 12).
- Zhang, Shiqing, Xiaoming Zhao i Bicheng Lei (2011). "Facial Expression Recognition Using Local Fisher Discriminant Analysis". A: *Advances in Computer Science, Environment, Ecoinformatics, and Education*. Ed. de Song Lin i Xiong Huang. Vol. 214. Berlin, Heidelberg: Springer Berlin Heidelberg, pàg. 443 - 448. ISBN: 978-3-642-23320-3 978-3-642-23321-0 (v. la pàg. 10).