

Universitat Oberta de Catalunya

# Estudi i implementació de tècniques de reconeixement facial

Jordi Soler Velasco

## **Agraïments**

Aquest treball no hagués estat possible sense el suport i ajuda de la família, amics, companys de feina i coneguts, fent especial menció a Irene Gonzalez.

Aprofito per donar les gràcies a David Massip.

## RESUM

### INTRODUCCIÓ I OBJECTIUS

La tecnologia de reconeixement d'imatges es basa en l'anàlisi de certes característiques facials per a reconèixer a un individu. Gràcies a aquest anàlisi es poden detectar els canvis en les cares dels individus.

Actualment molts serveis estan utilitzant tècniques de reconeixement, com poden ser les xarxes socials i les càmeres fotogràfiques digitals.

En aquest projecte es pretén estudiar i desenvolupar els mètodes de reconeixement més comuns basats en aparença (eigenfaces i fisherfaces) i basats en models (active shape models). Al finalitzar aquesta fase s'integren els anteriors mètodes dins d'una aplicació prototipus per mostrar el seu funcionament conjunt formant un petit sistema de reconeixement facial.

Per dur a terme la implementació esmentada s'ha optat per utilitzar un entorn sobre Windows que consta de Matlab, Visual Studio 2008 i OpenCV. Matlab és una plataforma molt utilitzada per la recerca en visió per computador ja que permet una fàcil manipulació de les imatges i matrius. A més, disposa de moltes funcions matemàtiques.

### DESENVOLUPAMENT DEL PROJECTE

Un sistema de reconeixement facial no només consta d'un component de reconeixement si no que està format per altres components necessaris per assolir l'objectiu final, que no és cap altre que reconèixer individus dins d'imatges. Aquests components són, el component de detecció i el component de normalització.

Dins d'aquest projecte s'utilitzen tots els components mencionats anteriorment per mostrar el funcionament dels procediments de reconeixement facial dins d'un sistema complet. Per dur a terme la detecció de rostres s'utilitza la funcionalitat de Viola&Jones que ja incorpora OpenCV i per dur a terme l'alineament s'utilitza un sistema d'alineament i normalització bàsic que escala i rota les cares prenent com a referència l'ull esquerre (centre de rotació) i la distància entre ulls, per dur a terme la detecció dels ulls també s'utilitza Viola&Jones.

Pel que fa els mètodes de reconeixement estudiats, aquests són dos mètodes basats en aparença (eigenfaces i fisherfaces) i un mètode basat en models (active shape model).

Els sistemes basats en aparença s'utilitzen directament sobre les imatges sense fer ús de models 3D. Aquests tipus de sistemes representen un objecte en funció de diferents vistes del mateix. En aquests sistemes cada imatge es representa com un punt en un subespai vectorial, de forma que la comparació entre la imatge de test i les imatges de referència es realitza en un subespai vectorial. L'objectiu d'aquests algorismes és classificar les diferents cares en el nou subespai, per això serà necessari entrenar prèviament el sistema amb imatges de diferents cares amb diferents vistes.

- **Eigenfaces**

La base d'aquest procediment és l'anàlisi de components principals. Dins d'eigenfaces s'utilitza l'anàlisi de components principals per tractar el problema del reconeixement facial com si fos un problema de reconeixement en dues dimensions, sense la necessitat de recuperar la

geometria en tres dimensions de qualsevol objecte. D'aquesta forma s'aprofita l'avantatge de que, en principi, les cares es troben en posició vertical en una imatge i, així, es poden arribar a descriure com un petit conjunt de característiques en dues dimensions.

- **Fisherfaces**

És una variació d'eigenfaces, de fet, va ser proposat com una millora. Tenint en compte que el conjunt d'entrenament està etiquetat, és possible utilitzar aquesta informació per reduir la dimensionalitat de l'espai de característiques.

Aquest mètode té com a objectiu portar les cares a un espai on es pugui distingir de millor forma les diferents classes entre si, maximitzant la distància entre imatges de diferents classes i a la vegada minimitzant la distància entre imatges d'una mateixa classe

Els sistemes basats en models, intenten construir un model el més descriptiu possible de la cara humana. Són capaços de detectar amb precisió la variacions facials. Aquests sistemes han d'obtenir les característiques biomètriques de les imatges per a realitzar el reconeixement (distància entre ulls, gruix del nas, etc...). Habitualment aquestes tècniques requereixen d'imatges de gran resolució. Quan s'utilitzen aquests sistemes, l'algorisme sap amb antelació l'objecte que ha de representar i el que intenta fer és que es correspongui la cara real amb el model.

- **Active Shape Model**

La idea principal resideix en el fet que tot i saber quin és l'objecte (en aquest cas la cara) que es busca dins de la imatge i la forma que té, aquest pot aparèixer amb alguna deformació. Així doncs, es defineixen uns punts, que donen la forma a l'objecte, els quals accepten una petita variació segons cap on evoluciona la figura.

## RESULTATS I CONCLUSIONS

Degut a problemes per falta de temps, no s'ha pogut implementar l'algorisme basat en Active Shape Models, només se n'ha pogut fer un estudi i queda pendent com a treball futur. Pel que fa als altres dos algorismes estudiats, els resultats obtinguts han estat els esperats. Eigenfaces i Fisherfaces ofereixen resultats molt similars, ja que Fisherfaces també utilitza PCA per reduir la dimensionalitat, però en entorns amb il·luminació variant Fisherfaces mostra un millor comportament, ja que l'anàlisi de discriminant lineal (LDA) que s'utilitza en Fisherfaces permet retenir l'estructura de la cara.

Cal notar que els dos mètodes implementats no es poden considerar mètodes de reconeixement fiables, segurament ASM (Active Shape Models) obté més bons resultats ja que és capaç d'adaptar-se millor als canvis d'expressió i posició degut a que s'utilitzen models genèrics deformables que s'adapten a cada rostre en funció de la ubicació de certs punts característics. Els mètodes basats en aparença estudiats estan molt lligats a factors com la il·luminació, posició, orientació, etc... pel correcte funcionament del sistema de reconeixement és vital que l'objecte de reconeixement, en aquest cas una cara, estigui ben il·luminada, ben centrada i ben escalada i això en entorns del món real a vegades resulta complicat aconseguir-ho.



**ÍNDIX**

<b>1. Introducció</b> .....	<b>13</b>
1.1 Aplicacions.....	13
1.2 Tècniques i algorismes existents.....	14
1.3 Aspectes tècnics.....	14
1.4 Motivació.....	15
1.6 Abast.....	15
1.7 Entorn de treball.....	16
1.8 Activitats i tasques.....	17
1.9 Estructura de la documentació.....	19
<b>2. Estat de l'art</b> .....	<b>20</b>
2.1 Funcionament d'un sistema de reconeixement.....	20
2.2 Detecció de cares.....	21
2.3 Reconeixement facial.....	21
2.3.1 Eigenfaces, PCA (Principal Component Analysis).....	23
2.3.2 Fisherfaces, LDA (Linear Discriminant Analysis).....	23
2.3.3 ASM (Active Shape Model).....	24
<b>3. Anàlisi dels mètodes emprats i desenvolupament</b> .....	<b>25</b>
3.1 Mètode Viola & Jones (detecció).....	25
3.1.1 Característiques Haar - Like.....	25
3.1.2 Imatge integral.....	26
3.1.3 Entrenament dels classificadors.....	27
3.1.4 Cascada de classificadors.....	29
3.1.5 Detecció.....	31
3.1.6 Problemes.....	31
3.2 Mètodes de reconeixement.....	32
3.2.1 Distància entre imatges.....	32
3.2.2 Eigenfaces.....	33
3.2.2.1 PCA (Anàlisi de components principals).....	33
3.2.2.2 Procediment.....	34
3.2.3 Fisherfaces.....	36

3.2.3.1 LDA (Anàlisi discriminant lineal) .....	37
3.2.3.2 Procediment .....	37
3.2.4 ASM .....	40
3.2.4.1 Procediment .....	40
3.3 Aspectes tècnics .....	42
3.3.1 Detecció de cares i ulls .....	43
3.3.2 Alineament .....	44
3.3.3 Reconeixement.....	46
3.3.3.1 Base de dades.....	46
3.3.3.2 Reconeixement.....	48
3.3.4 Anàlisi i disseny.....	49
3.3.4.1 Metodologia .....	49
3.3.4.2 Eines de desenvolupament .....	50
3.3.4.3 Requeriments i especificacions del sistema .....	51
3.3.4.4 Disseny.....	52
3.3.5 Experiments i resultats.....	56
3.3.5.1 Bases de dades de cares .....	56
3.3.5.2 Experiments amb els mètodes de reconeixement .....	57
3.3.5.3 Resultats visuals de l'aplicació.....	60
<b>4. Conclusions</b> .....	63
4.1 Valoració final .....	63
4.1.1 Valoració econòmica .....	65
4.2 Conclusions generals.....	66
4.3 Treball futur .....	66
<b>Annexos</b> .....	68

**LLISTAT DE FIGURES**

<b>1. Introducció</b> .....	<b>13</b>
Fig 1.1 Taula d'aplicacions del reconeixement facial .....	14
Fig 1.2 Taula de tasques .....	17
Fig 1.3 Calendaris de planificació .....	19
<b>2. Estat de l'art</b> .....	<b>20</b>
Fig 2.1 Diagrama general per a qualsevol sistema de reconeixement facial .....	20
Fig 2.2 Alguns mètodes de reconeixement .....	22
Fig 2.3 Eigenfaces d'un conjunt d'imatges .....	23
Fig 2.4 Fisherfaces d'un conjunt d'imatges .....	24
<b>3. Anàlisi dels mètodes emprats i desenvolupament</b> .....	<b>25</b>
Fig 3.1 Característiques Haar.....	25
Fig 3.2 Característiques Haar rectangulars.....	26
Fig 3.3 Imatge integral.....	26
Fig 3.4 Càlculs en la imatge integral.....	27
Fig 3.5 Procediment de l'algorisme de detecció .....	31
Fig 3.6 Distància Euclidiana entre dos punts 2D .....	32
Fig 3.7 Punts característics.....	41
Fig 3.8 Iteracions amb el mean shape.....	41
Fig 3.9 Cerca de punts .....	42
Fig 3.10 Diagrama de l'arquitectura del sistema.....	43
Fig 3.11 Etapa de detecció .....	43
Fig 3.12 Detecció de cara i ulls .....	44
Fig 3.13 Angle de rotació perquè els ulls quedin alineats.....	44
Fig 3.14 Offset entre ulls .....	45
Fig 3.15 Exemple alineament .....	46
Fig 3.16 Etapa alineament.....	46
Fig 3.17 Exemples de cares .....	47
Fig 3.18 Pantalla per inserir nous elements a la base de dades.....	47
Fig 3.19 Estructura de directoris base de dades .....	48
Fig 3.20 Alta d'una imatge a la base de dades .....	48
Fig 3.21 Etapa de reconeixement.....	48



Fig 3.22 Metodologia emprada .....	49
Fig 3.23 Diagrama de context general de l'aplicació.....	52
Fig 3.24 Diagrama d'activitat del reconeixement de cares .....	54
Fig 3.25 Diagrama d'activitat per afegir cara a bdd .....	54
Fig 3.26 Diagrama d'activitat entrenament eigenfaces .....	55
Fig 3.27 Diagrama d'activitat entrenament fisherfaces.....	55
Fig 3.28 Diagrama d'activitat sistema de reconeixement.....	56
Fig 3.29 Extracte base de dades de cares AT&T.....	57
Fig 3.30 Taxa reconeixement descartant fins a quatre high eigenfaces.....	58
Fig 3.31 Taxa reconeixement descartant low eigenfaces .....	58
Fig 3.32 Taxa reconeixement en funció d'eigenvectors descartats .....	59
Fig 3.33 Taxa reconeixement en funció d'eigenvectors utilitzats.....	59
Fig 3.34 Taxa reconeixement en funció d'eigenvectors utilitzats (il·luminació variant).....	60
Fig 3.35 Reconeixement correcte amb eigenfaces .....	60
Fig 3.36 Reconeixement incorrete amb eigenfaces (cara descentrada).....	61
Fig 3.37 Reconeixement correcte amb fisherfaces (canvis il·luminació) .....	61
Fig 3.38 Reconeixement incorrecte amb eigenfaces (canvis il·luminació).....	61
Fig 3.39 Reconeixement eigenfaces a temps real, imatges capturada des de webcam.....	62
Fig 3.40 Reconeixement fisherfaces a temps real, imatges capturada des de webcam.....	62
<b>4. Conclusions .....</b>	<b>63</b>
Fig 4.1 Temporalització prevista activitat implementació i integració .....	63
Fig 4.2 Temporalització prevista per l'activitat implementació d'una interfície gràfica.....	63
Fig 4.3 Temporalització final activitat implementació i integració .....	64
Fig 4.4 Temporalització final per l'activitat implementació d'una interfície gràfica .....	64

## 1. INTRODUCCIÓ

Aquest projecte està emmarcat dins de l'àmbit de la visió per computador, concretament en la vessant que estudia la detecció i especialment el reconeixement d'objectes dins imatges o vídeos.

El reconeixement facial s'ha convertit en els últims anys en una àrea d'investigació activa que engloba diverses disciplines, com processat d'imatges, reconeixement de patrons i xarxes neurals. Involucra tan a investigadors de l'àrea d'informàtica com a neurocientífics. Es podria considerar també dins el camp del reconeixement d'objectes, on la cara és un objecte tridimensional subjecte a variacions d'il·luminació, postura, etc... I ha de ser identificada basada en la seva projecció 2D (excepte quan s'utilitzen tècniques 3D).

L'objectiu d'un sistema de reconeixement facial és, generalment, el següent: donada una imatge d'una cara desconeguda, cal trobar una imatge de la mateixa cara dins d'un conjunt d'imatges conegudes, o imatges d'entrenament. La gran dificultat afegida és la d'aconseguir que aquest procés es pugui realitzar en temps real. El sistema ha d'identificar automàticament les cares presents dins d'imatges o vídeos.

Un sistema de reconeixement facial pot operar de dues maneres:

- **Verificació o autenticació de cares**

Es compara una imatge de la cara amb una altre imatge amb la cara de la que es vol conèixer la identitat. El sistema ha de confirmar o refusar la identitat de la cara.

- **Identificació o reconeixement de cares**

Es compara la imatge d'una cara desconeguda amb totes les imatges de cares conegudes que es troben a la base de dades per tal de determinar la seva identitat.

### 1.1 APLICACIONS

Actualment el reconeixement facial s'utilitza molt en sistemes de seguretat per al reconeixement d'usuaris. En aquest tipus de sistemes un lector defineix les característiques del rostre, i quan l'usuari sol·licita l'accés, la identitat d'aquest es verifica comparant les dades obtingudes amb les de la base de dades. Aquests sistemes, però, presenten un inconvenient molt important, no resulten útils a mesura que passen els anys, ja que les característiques facials varien i quan es sol·licita l'accés, aquestes característiques ja no coincideixen amb les dades de la base de dades. Per solucionar aquest problema es pot optar per anar renovant la base de dades.

El reconeixement facial també s'utilitza per aplicacions d'interacció persona-ordinador, en gestió multimèdia i en programari com Picasa, facebook, Apple iPhoto, Asus Smart Logon, Google Picasa, Sony's Picture Motion Browser (PMB), etc...

En un futur no molt llunyà no és descabellat pensar que es puguin establir les tècniques de reconeixement facial a nivell d'usuari, per exemple utilitzar els sistemes de reconeixement facial en caixers automàtics on per a operar sigui necessari un reconeixement facial en comptes de l'actual PIN.

A continuació es mostra una taula on es mostren les àrees i les aplicacions específiques del reconeixement facial.

Àrees	Aplicacions específiques
<b>Biometria</b>	Llicència de conduir, programes de dret, immigració, DNI, passaports, registre de votants, frau
<b>Seguretat de la informació</b>	Inici de sessió, seguretat en aplicacions, seguretat en bases de dades, xifratge de la informació, seguretat a Internet, accés a Internet, registres mèdics, terminals de comerç segur, caixers automàtics
<b>Compliment de la llei i vigilància</b>	Videovigilància avançada, control CCTV, seguiment de sospitosos, investigació
<b>Targetes intel·ligents</b>	Autenticació d'usuaris
<b>Control d'accés</b>	Accés a instal·lacions, accés a vehicles

**Fig 1.1.** Taula d'aplicacions del reconeixement facial.

## 1.2 TÈCNiques I ALGORISMES EXISTENTS

El reconeixement de cares és un problema que pot ser solucionat (o no) de moltes maneres. Els estudis que s'han realitzat al respecte han proposat tot tipus de mètodes, algorismes i estratègies. Aquests mètodes i estratègies es poden separar en els següents grups:

- **Basats en aparença**

Els sistemes basats en aparença s'utilitzen directament sobre les imatges sense fer ús de models 3D. En aquests sistemes cada imatge es representa com un punt en un subespai vectorial, de forma que la comparació entre la imatge de test i les imatges de referència es realitza en un subespai vectorial. L'objectiu d'aquests algorismes és classificar les diferents cares en el nou subespai.

- **Basats en models**

Són mètodes que intenten construir un model el més descriptiu possible de la cara humana. Obtenen les característiques biomètriques de les imatges per a realitzar el reconeixement (distància entre ulls, gruix del nas, etc...).

## 1.3 ASPECTES TÈCNICS

Un sistema de reconeixement facial no és perfecte i a vegades difícil d'implementar en certes condicions. Una de les principals debilitats d'aquests sistemes és deguda a l'angle en el que es troba el rostre que es vol conèixer. Estudis han confirmat que el reconeixement actua correctament fins als 20 graus, una vegada superat aquest angle comencen a sorgir problemes. És per aquest motiu que s'està investigant el reconeixement en 3D amb el qual aquest inconvenient desapareixeria.

Un altre inconvenient és el mal funcionament en situacions de poca llum, a més, tenir el cabell llarg, ulleres de sol o altres objectes que cobreixin part del rostre dificulta molt la tasca de l'algorisme. L'algorisme no sempre és capaç de distingir els rostres si l'expressió d'aquest és diferent a la que està emmagatzemada a la base de dades.

## 1.4 MOTIVACIÓ

La motivació principal d'aquest projecte de final de carrera és la de crear una aplicació capaç de detectar persones (cares) dins d'una seqüència de vídeo o dins d'una imatge estàtica i que a la vegada, mitjançant els mètodes de reconeixement estudiats sigui capaç també reconèixer qui és la persona en qüestió (segons si la té entrada a la base de dades o no). Per fer això cal disposar d'un software que sigui robust, però a la vegada prou ràpid per treballar en temps real.

## 1.5 OBJECTIUS

L'objecte d'estudi d'aquest projecte són tècniques de reconeixement facial basades en aparença (Fisherfaces i Eigenfaces) i en models (Active Shape Models). Per demostrar el funcionament d'aquestes es pretén implementar un sistema complet de reconeixement fent ús de l'algorisme Viola & Jones per dur a terme la detecció de cares i un procediment d'alineament que s'explicarà més detalladament en apartats posteriors.

A partir d'aquí cal demostrar el correcte funcionament dels mètodes detectant cares noves i reconeixent-les.

Per tant, els objectius que es plantegen en el següent projecte són els següents:

- **Estudiar l'algorisme Viola & Jones i la seva implementació en OpenCV**, en la seva aplicació en la detecció de cares.
- **Desenvolupar un mètode d'alineament/normalització**, en la seva aplicació en l'alineament del rostre detectat abans de realitzar el reconeixement.
- **Estudiar, dissenyar i implementar els algorismes de reconeixement facial**, en l'aplicació per a reconèixer cares.
- **Adaptar i modificar els mètodes** per adaptar-los al màxim al reconeixement de cares en imatges estàtiques i en seqüències de vídeo capturades a temps real.
- **Fer un estudi sobre el comportament** de cada una de les aplicacions dels mètodes de reconeixement.
- **Realitzar una aplicació prototipus** que permeti realitzar tant la detecció com el reconeixement (utilitzant els mètodes de reconeixement implementats) de forma automàtica.

## 1.6 ABAST

Una eina de detecció i reconeixement d'objectes en imatges pot abastar una gran quantitat de possibilitats. En aquest projecte el sistema de reconeixement es limita a detectar, alinear i reconèixer cares de persones, centrant especial atenció en els mètodes de reconeixement. D'aquesta manera l'aplicació prototipus serà capaç de detectar cares i de donar la identitat de la cara en qüestió sempre hi quan les cares de mostra de l'individu estiguin guardades prèviament dins una base de dades d'entrenament.

Un cop finalitzat el projecte, es disposarà d'una aplicació de Matlab, que permetrà fer tots els diferents passos i proves fins a arribar a obtenir els resultats esperats. Dins d'aquesta aplicació es disposarà de la possibilitat d'obtenir dades per a la base de dades d'entrenament, fer els entrenaments de les dades, fer petites execucions de cada mòdul per veure com funciona, i finalment comprovar el funcionament conjunt dels mètodes de reconeixement, detecció i alineament sobre una seqüència de vídeo o d'una imatge estàtica amb la finalitat de reconèixer la identitat dels rostres que hi apareixen. A més de l'aplicació visual es disposarà d'un seguit de procediments de test per a poder comparar els resultats dels mètodes de reconeixement implementats.

S'han definit un seguit de restriccions/especificacions per l'aplicació principal:

- Per la detecció d'objectes (rostres i ulls) s'utilitza l'algorisme Viola & Jones (s'explica detalladament en capítols posteriors) implementat en les OpenCV.
- S'opta per no entrenar Viola & Jones, ja que la versió per defecte de OpenCV ja ve entrenada amb més de 100.000 imatges.
- No es crea una base de dades d'entrenament de zero. S'utilitzen bases de dades ja existents, prenent una mostra d'un 40 imatges (10 per individu) per dur a terme l'entrenament. Les bases de dades utilitzades són: <http://cswww.essex.ac.uk/mv/allfaces/> , <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html> i <http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html>.
- El nucli i la part més important del projecte és la implementació dels algorismes de reconeixement facial.
- Les imatges de les cares a reconèixer han de ser frontals.
- Els mètodes de reconeixement han de ser sensibles a canvis d'aspecte. L'aplicació haurà de ser capaç de classificar la cara invariant a canvis com: gorres, ulleres, nas de pallasso, etc... Això es du a terme per posar a prova el sistema.

## 1.7 ENTORN DE TREBALL

Per a la implementació d'aquest sistema s'ha escollit l'entorn de programació Matlab R2010, Visual C++ 2008 Express edition (per fer ús de les llibreries especialitzades en visió per computador OpenCV).

Les raons per les quals s'ha escollit aquest entorn de programació són les següents:

- Matlab ofereix un gran ventall d'eines matemàtiques.
- Permet una interacció senzilla amb imatges.
- És de fàcil integració amb el llenguatge de programació C++. Per aquest motiu s'utilitza el Visual C++ i les llibreries OpenCV.
- Permet la creació d'interfícies gràfiques senzilles.
- Permet la reutilització de molt de codi, ja que està organitzat en funcions.

- Ja es disposava de certa experiència en aquest entorn degut a treballs anteriors, més concretament per la realització del projecte de final de carrera d'ETIS (Enginyeria tècnica en informàtica de sistemes).

## 1.8 ACTIVITATS I TASQUES

A continuació es mostren les tasques amb la seva estimació de temps i es descriuen breument cadascuna d'aquestes:

Fase	Tasca
1	Recerca de documentació, elecció de l'entorn de desenvolupament i definició del pla de treball.
2	Repàs del llenguatge de programació que s'utilitza.
3	Estudi dels mètodes Viola & Jones i de reconeixement.
4	Implementació i integració dels algorismes de reconeixement.
5	Implementació d'una interfície gràfica.
6	Període de proves.
7	Extracció de conclusions un cop analitzats els resultats obtinguts.
8	Redactat de la memòria.

Fig 1.2. Taula de tasques.

A continuació es descriuen les activitats que cal dur a terme per cada una de les tasques esmentades anteriorment:

**1. Recerca de documentació, elecció de l'entorn de desenvolupament i definició del pla de treball.**

Conversacions amb el tutor del projecte per determinar en que consistirà aquest. Un cop decidit l'àmbit del projecte, cercar articles i documentació relacionada amb els mètodes de detecció i reconeixement facial que s'ha decidit estudiar en aquest projecte.

Redacció de la introducció, el pla de treball del projecte i configuració de l'entorn de treball.

*Duració: 10 dies*

**2. Repàs del llenguatge de programació que s'utilitza.**

Primera aproximació a les llibreries i al llenguatge de programació que s'utilitzaran durant la implementació del projecte. S'aprofita per refrescar una mica els coneixements sobre el llenguatge.

*Duració: 8 dies*

**3. Estudi dels mètodes Viola&Jones, reconeixement basat en aparença i reconeixement basat en models.**

S'estudien els diferents passos a seguir per tal de poder aplicar els mètodes en qüestió. Com ja s'ha comentat anteriorment la col·lecció de llibreries OpenCV ja duu incorporada la implementació del algorisme Viola & Jones, per tant s'opta per utilitzar aquesta funcionalitat ja implementada i entrenada amb més de 100.000 imatges. Això permetrà centrar el projecte en els algorismes de reconeixement facial (Eigenfaces, Fisherfaces i ASM).

*Duració: 14 dies*

**4. Implementació i integració dels algorismes de reconeixement.**

Aquesta tasca es divideix en diverses subtasques, que són les següents:

- Anàlisi i implementació dels algorismes de reconeixement.
- Implementació d'un petit component d'alineament i normalització d'imatges de cares.
- Implementació de la crida a Viola & Jones de OpenCV.
- Integració dels dos algorismes amb el mòdul de detecció "Viola&Jones" i el component d'alineament.

Pel desenvolupament dels mètodes de reconeixement és necessari partir de bases de dades d'imatges per tal de poder entrenar els algorismes amb l'objectiu de que detectin les cares. Es descarta la creació d'una base de dades perquè es parteix de bases de dades ja existents a la xarxa Internet. *Duració: 28 dies*

**5. Implementació d'una interfície gràfica.**

Creació d'una interfície gràfica per facilitar la comunicació entre el sistema i l'usuari, i per mostrar el funcionament de les implementacions realitzades. L'usuari podrà:

- Comprovar el funcionament de Viola&Jones en imatges estàtiques i en seqüències de vídeo.
- Afegir noves cares a la base de dades d'entrenament.
- Comprovar el funcionament dels mètodes de reconeixement, introduint la imatge d'una cara ja alineada i normalitzada per dur a terme el seu reconeixement.
- Comprovar el funcionament del sistema complet (integració de la detecció de cares, alineament i reconeixement)

*Duració: 7 dies*

**6. Període de proves.**

Obtenció i comparació de resultats després d'aplicar els algorismes sobre imatges.

*Duració: 3 dies*

**7. Extracció de conclusions un cop analitzats els resultats obtinguts.**

Anàlisi detallat dels resultats per tal de detectar possibles errors o millores per corregir i/o en futurs treballs/projectes.

*Duració: 2 dies*

**8. Redactat de la memòria.**

Redacció de la memòria final partint de la documentació generada en cada fase del projecte.

*Duració: 12 dies*

A continuació es mostren a sobre dels calendaris els títols de les tasques amb la seva estimació de temps. A la figura **Fig 1.3** hi ha la planificació realitzada sobre calendaris.

Fase	Tasca
1	Recerca de documentació, elecció de l'entorn de desenvolupament i definició del pla de treball.
2	Repàs del llenguatge de programació que s'utilitza.
3	Estudi dels mètodes Viola & Jones i reconeixement.
4	Implementació i integració dels algorismes de reconeixement.
5	Implementació d'una interfície gràfica.
6	Període de proves.

<b>7</b>	Extracció de conclusions un cop analitzats els resultats obtinguts.
<b>8</b>	Redactat de la memòria.

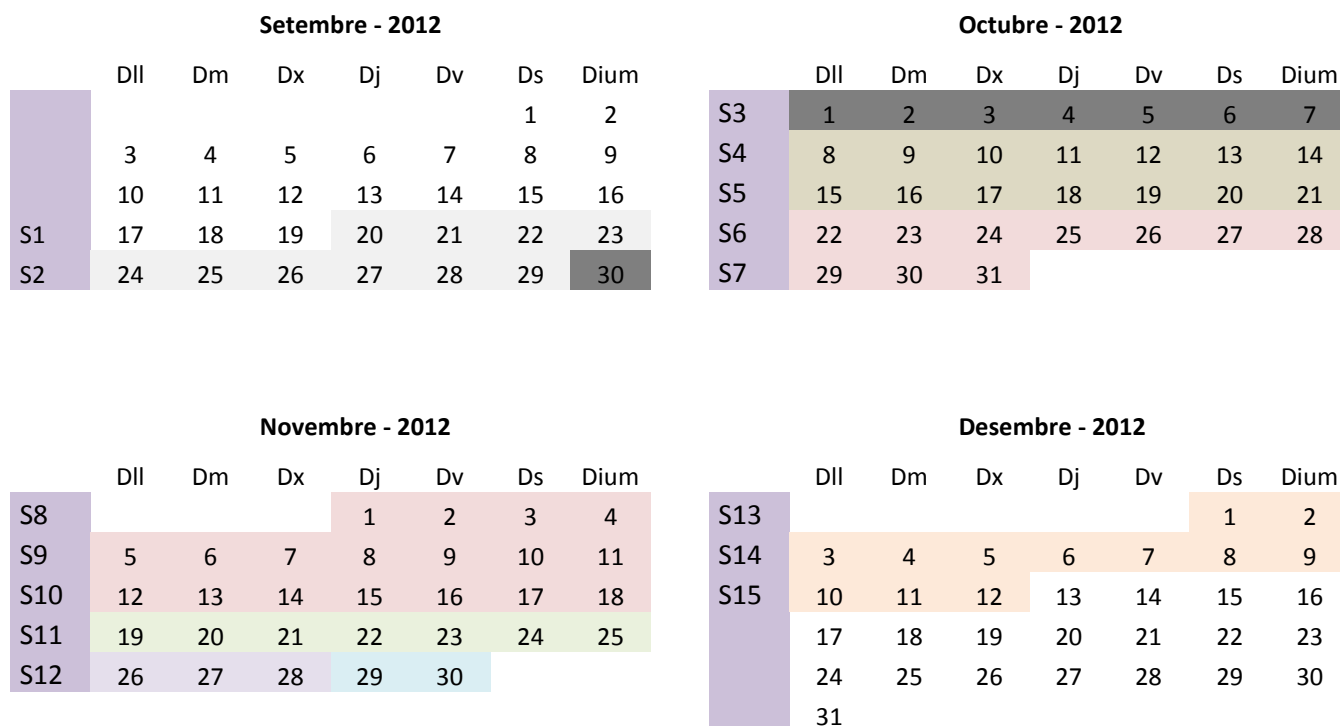


Fig 1.3. Calendaris de planificació.

## 1.9 ESTRUCTURA DE LA DOCUMENTACIÓ

La memòria que es presenta explica les tasques realitzades durant el procés del projecte. Aquests són els capítols que componen la memòria d'aquest projecte:

### 1. Introducció

Es presenta la temàtica del projecte juntament amb la definició dels objectius, motivacions, abast i planificació de les activitats i tasques que es duen a terme al llarg del projecte.

### 2. Estat de l'art

Introducció i estudi previ de les tècniques de detecció i reconeixement basades en aparença existents. Fent menció especial a les que s'utilitzen en aquest projecte.

### 3. Anàlisi dels mètodes emprats i desenvolupament

És el capítol principal del projecte on s'exposa el contingut teòric de cada un dels mètodes i les tècniques utilitzades en les aplicacions de reconeixement.

Es proporcionen detalls de les implementacions, es mostren i es comenten els experiments duts a terme així com també es descriu l'anàlisi i disseny de l'aplicació visual.

### 4. Conclusions

### 5. Valoració econòmica.



## 2. ESTAT DE L'ART

En un sistema de reconeixement facial hi intervé un mòdul de detecció i un mòdul de reconeixement. Actualment existeixen varies tècniques de detecció i reconeixement, en aquest projecte s'utilitza Viola & Jones per fer la detecció i s'estudien les tècniques de reconeixement, més concretament: PCA, LDA i ASM. En les posteriors seccions d'aquest capítol s'expliquen les diverses tècniques més importants dins de l'àmbit de la detecció i reconeixement facial.

### 2.1 FUNCIONAMENT D'UN SISTEMA DE RECONeixEMENT

El procés de reconeixement consta de cinc mòduls principals:

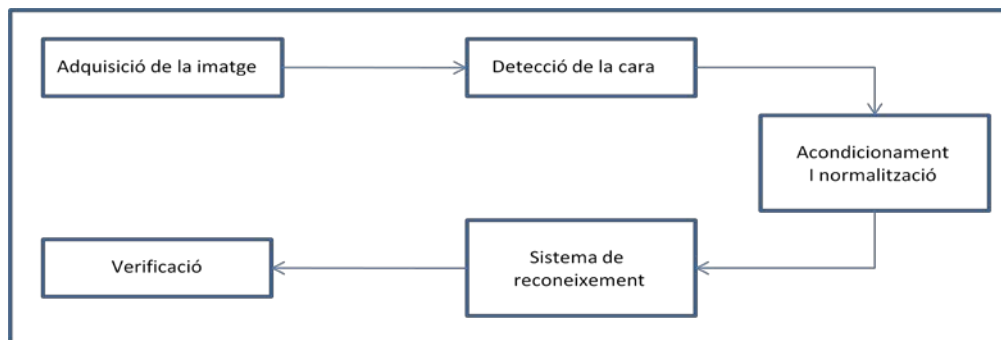


Fig 2.1. Diagrama de blocs general per a qualsevol sistema de reconeixement facial

#### 1. Detecció de la cara

Detecta que hi ha una cara a la imatge sense identificar-la. Si es tracta d'un vídeo també es pot fer un seguiment de la cara. Proporciona la localització i l'escala on es troba la cara.

#### 2. Alineació de la cara

Localitza els components de la cara, i mitjançant transformacions geomètriques i fotomètriques, es normalitzen les imatges. Per normalitzar les imatges de les cares es poden seguir diferents regles, com al distància entre les pupil·les o la posició del nas. També es pot definir la mida de les imatges i la gamma de colors. Normalment, per disminuir la càrrega computacional del sistema, s'acostuma a utilitzar imatges petites amb escala de grisos com a espai de color. A vegades també s'utilitza l'equalització del histograma.

#### 3. Extracció de característiques

Proporciona informació per distingir entre les cares de diferents persones segons variacions geomètriques i/o fotomètriques.

#### 4. Reconeixement

El vector de característiques extret de la imatge on es vol practicar el reconeixement es compara amb els vectors de característiques extrets de les cares de la base de dades. Si se'n troba un amb un percentatge elevat de similitud, el sistema retorna la identitat de la cara; si no, indica que es tracta d'una cara desconeguda.

Els resultats obtinguts depenen de les característiques extretes per representar el patró de la cara i dels mètodes de classificació utilitzats per distingir rostres, però per extreure aquestes característiques apropiadament és necessari localitzar i normalitzar la cara adientment.

## 2.2 DETECCIÓ DE CARES

La detecció de cares és un treball de Visió per Computador que porta prop de 50 anys en investigació. Un sistema de detecció facial és una aplicació dirigida per ordinador per detectar automàticament a una persona en una imatge digital mitjançant la comparació de determinades característiques facials i una base de dades.

Durant tots aquests anys d'investigació s'ha anat resolent el problema de la detecció de cares de diferents maneres:

- **Tècniques basades en els trets**  
Es basa en un anàlisi a priori basat en el coneixement. S'exploten propietats com ara el color de la pell i la geometria facial.
- **Tècniques basades en la imatge**  
Aquestes tècniques estan basades en l'estudi de la pròpia imatge. El coneixement previ s'incorpora implícitament en esquemes d'entrenament. Es treballa directament amb una representació de la imatge a la que se li apliquen algorismes d'entrenament i anàlisi.

Alguns dels mètodes més coneguts són els següents:

- **Viola & Jones [1]**  
Està basat en la utilització de classificadors dèbils. Això vol dir que amb un sol classificador no es tindrien resultats fiables. A més de detectar cares es pot entrenar per a la detecció d'altres objectes.
- **Schneiderman & Kanade [2]**  
Es representen les estadístiques de l'aparició d'una cara i d'una "no cara" utilitzant un histograma. Cada histograma representa un subconjunt de coeficients i les posicions de les cares.
- **Rowley, Baluja & Kanade [3]**  
Es basa en l'ús xarxes neuronals per a la detecció de cares. S'entrenen les xarxes neuronals amb imatges de cares i altres que no ho són.

Com s'ha comentat el mètode escollit en aquest projecte, per ser estudiat i estudiar la detecció facial, és el mètode Viola & Jones. El fet d'haver escollit aquest mètode és perquè es tracta d'un procés senzill i a la vegada robust que permet obtenir molt bons resultats. En capítols posteriors s'explica de forma més detallada.

---

## 2.3 RECONeixEMENT FACIAL

Existeixen dos famílies de tècniques de reconeixement facial: tècniques basades en aparença i tècniques basades en models. En cada una d'aquestes famílies es troben varis mètodes per caracteritzar la imatge.

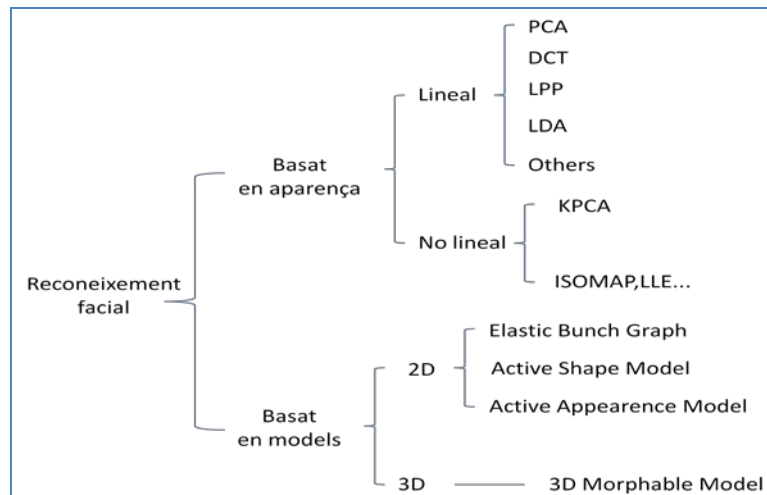


Fig 2.2. Alguns mètodes de reconeixement

Els sistemes basats en aparença s'utilitzen directament sobre les imatges sense fer ús de models 3D. Aquests tipus de sistemes representen un objecte en funció de diferents vistes del mateix. En aquests sistemes cada imatge es representa com un punt en un subespai vectorial, de forma que la comparació entre la imatge de test i les imatges de referència es realitza en un subespai vectorial. L'objectiu d'aquests algorismes és classificar les diferents cares en el nou subespai, per això serà necessari entrenar prèviament el sistema amb imatges de diferents cares amb diferents vistes.

Els sistemes basats en models intenten construir un model el més descriptiu possible de la cara humana capaç de detectar amb precisió variacions facials. Aquests sistemes tracten d'obtenir característiques biomètriques de les imatges per realitzar el reconeixement. Habitualment aquestes tècniques requereixen imatges de gran resolució. Quan s'utilitzen aquests sistemes, l'algorisme sap amb antelació l'objecte que ha de representar i busca correspondències entre la cara real i el model. El procés que es sol seguir quan s'utilitzen aquestes tècniques consta de 2 passos:

- Construcció del model.
- Ajustar el model a la imatge de test.
- Utilitzar els paràmetres del model ajustat per calcular la similitud entre la imatge de test i les imatges de referència per a realitzar el reconeixement.

Alguns dels avantatges i inconvenients dels sistemes basats en l'aparença en front als basats en models són els següents:

- Avantatges:
  - Més ràpids.
  - Requereixen d'una menor mida de les imatges.
  - Menor complexitat.
  - No requereixen d'un coneixement previ de les imatges.
- Inconvenients:
  - Més afectats per canvis a la orientació o expressió de la cara.
  - Més dificultats en front a canvis a la il·luminació.

A continuació es fa una breu descripció, de la literatura que existent, sobre alguns dels mètodes més rellevants exposats en la figura anterior.

### 2.3.1 EIGENFACES, PCA (PRINCIPAL COMPONENT ANALYSIS)[4],[5]

Es considera una de les tècniques que proporciona un millor rendiment. La idea principal del PCA és obtenir un conjunt de vectors ortogonals (*eigenvectors*) que, d'una forma òptima, representen la distribució de la informació amb la desviació del error quadràtic mig (m.s.e).

La idea bàsica d'aquest algorisme és aconseguir reconèixer una imatge facial comparant-la amb les característiques obtingudes a partir d'algunes cares que es coneixen prèviament. Es tracta el problema de reconeixement facial com si fos un problema de reconeixement en dues dimensions, sense la necessitat de recuperar la geometria en tres dimensions de qualsevol objecte. D'aquesta forma s'aprofita l'avantatge de que, en principi, les cares es troben en posició vertical en una imatge i, així, es poden arribar a descriure com un petit conjunt de característiques en dues dimensions. El sistema funciona projectant les imatges facials sobre un espai de faccions que engloba les variacions significatives entre les imatges facials conegudes. Les faccions significatives s'anomenen *eigenfaces*, ja que són els *eigenvectors* (components principals) del conjunt de cares. La projecció caracteritza la imatge facial d'un individu com la suma dels diferents pesos de totes les faccions (*eigenfaces*) i, de la mateixa manera, per reconèixer una imatge facial en particular només es necessita comparar aquests pesos amb aquells individus coneguts prèviament.

En la figura següent es mostren les 32 primeres *eigenfaces* d'un conjunt d'imatges, on la primera imatge és la imatge mitjana de totes elles. Les *eigenfaces* estan ordenades segons el seu corresponent *eigenvalue* (de més gran a més petit).

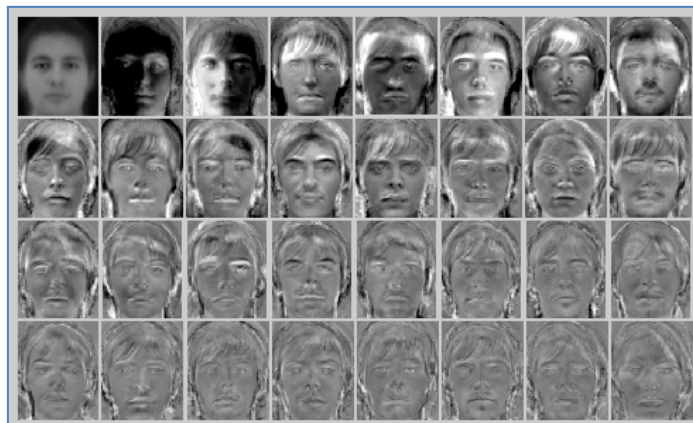


Fig 2.3. *Eigenfaces* d'un conjunt d'imatges.

Les *eigenfaces* també es poden utilitzar per representar les imatges de test per ser identificades. Això s'aconsegueix al projectar les imatges de test sobre les *eigenfaces*. La primera *eigenface* és la que conté la major part de la informació, en el que a l'error quadràtic mig es refereix.

### 2.3.2 FISHERFACES, LDA (LINEAR DISCRIMINANT ANALYSIS)

L'esquema LDA permet utilitzar la informació entre membres de la mateixa classe per desenvolupar un conjunt de vectors de característiques on les variacions de les diferents cares s'emfatitzen mentre que els canvis deguts a expressió facial i orientació de la cara no ho fan. Explicat d'una altra manera: el PCA no té en compte la informació de quines imatges pertanyen a un mateix individu, mentre que el LDA sí que ho fa. Per aquest motiu el inconvenient del PCA sobre el LDA és que maximitza la variància de totes les mostres sense tenir en compte la classe a la que pertanyen. Per la seva part, el LDA intenta

maximitzar la variància de les mostres entre classes; i al mateix temps minimitzar la variància entre mostres de la mateixa classe.



Fig 2.4. Fisherfaces d'un conjunt d'imatges.

Els dos mètodes anteriors es s'exposen més detalladament en capítols posteriors.

### 2.3.3 ASM (ACTIVE SHAPE MODELS)[7],[8]

Un dels algorismes més utilitzats dins dels mètodes basats en models és ASM (Active Shape Models).

La idea principal resideix en el fet que tot i saber quin és l'objecte (en aquest cas la cara) que es busca dins de la imatge i la forma que té, aquest pot aparèixer amb alguna deformació. Així doncs, es defineixen uns punts, que donen la forma a l'objecte, els quals accepten una petita variació segons cap on evoluciona la figura.

Aquest mètode es descompon en dues etapes: entrenament i cerca. Dins la fase d'entrenament es calcula un model estadístic que es fa variar en la fase de cerca fins a obtenir un resultat final que s'adapti a la nova imatge.

### 3. ANÀLISI DELS MÈTODES EMPRATS I DESENVOLUPAMENT

Abans d'iniciar aquest projecte s'han llegit diferents articles relacionats amb Viola & Jones i les tècniques de reconeixement facial que s'implementen en aquest projecte.

En les següents seccions s'exposa detalladament el contingut teòric de cada un dels mètodes i tècniques utilitzades en les aplicacions de reconeixement. Primer es comença pel mètode de detecció (Viola & Jones) i es segueix amb els mètodes de reconeixement Eigenfaces, Fisherfaces i ASM.

#### 3.1 MÈTODE VIOLA & JONES (DETECCIÓ)

El mètode Viola & Jones és un algorisme per a la detecció d'objectes. Va ser publicat al 2001 per **Paul Viola i Michael Jones [1]** amb la intenció de construir un entorn per a la detecció de cares dins d'imatges. A partir de la seva publicació es va produir un canvi molt important en la detecció de patrons en imatges, ja que l'algorisme és capaç de processar imatges molt més ràpidament que els algorismes pensats fins llavors, a la vegada que obté uns ràtios d'èxit elevats.

L'algorisme aporta tres grans contribucions. La primera és la introducció d'una nova representació de la imatge, anomenada "imatge integral", que permet computar ràpidament les característiques de la imatge usades pel detector. La segona contribució és la utilització d'uns classificadors simples i eficients, construïts mitjançant l'algorisme d'aprenentatge AdaBoost (**Freud i Schapire 1995**). Aquests serveixen per seleccionar un petit nombre de característiques crítiques d'un conjunt enorme de característiques potencials. Finalment, la tercera contribució és el mètode de combinar classificadors simples en "cascada". Es distribueixen de manera que es descarten ràpidament les regions de fons de la imatge per centrar més l'anàlisi en aquelles zones més susceptibles de ser l'objecte que es vol detectar.

##### 3.1.1 CARACTERÍSTIQUES HAAR-LIKE

El procediment que segueix el detector d'objectes consisteix en classificar les imatges a partir del valor de característiques simples. Hi ha diferents motius per usar característiques enlloc d'usar el valor dels píxels directament. El motiu més comú és que es necessiten menys imatges d'exemple per obtenir bons resultats quan s'utilitzen les característiques respecte a quan s'utilitzen operacions píxel a píxel. A part d'això la principal motivació és que els sistemes basats en les característiques operen molt més ràpid que els basats en píxel.

Una característica Harr de la imatge consisteix en una sèrie d'àrees, on cada àrea té assignat valor 0 o 1. El resultat d'una característica Harr consistirà en avaluar la suma dels valors dels píxels de la imatge que pertanyen a les àrees amb valor 1 i restar la suma de píxels de les àrees on el valor sigui 0. En la següent figura es poden observar alguns dels exemples de característiques Haar:

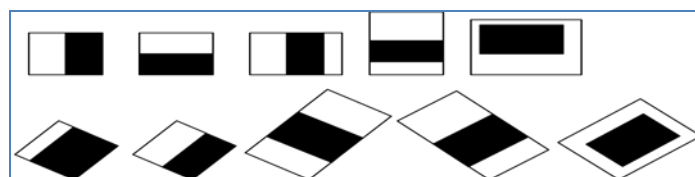
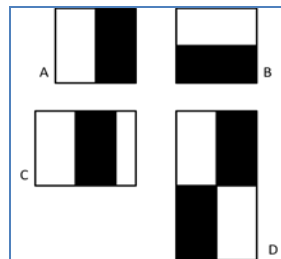


Fig 3.1. Característiques Haar.

Concretament, l'algorisme Viola&Jones només utilitza tres tipus de característiques de Haar rectangulars per poder aprofitar-se dels beneficis de la imatge integral. En primer lloc, la característica 2-rectangles, avaluada com la diferència entre la suma de píxels de les dues regions rectangulars. Les regions tenen la mateixa mida i són adjacents horitzontalment o verticalment. En segon lloc, la característica 3-rectangles, que es calcula com la suma dels dos rectangles exteriors restant la suma del rectangle central. I finalment, la característica 4-rectangles, computada com la diferència entre les dues parelles de rectangles agrupades en diagonal.

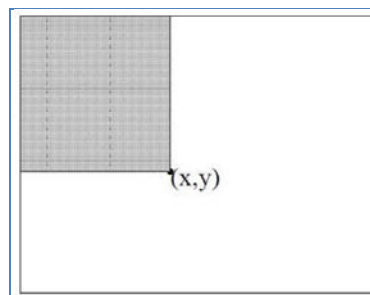


**Fig 3.2.** Característiques Haar rectangulars.

Quan s'analitza la imatge amb les característiques, es fa per a diferents mides de la característica Haar. Per a cada mida cal provar totes les posicions possibles dins de la imatge. Això és degut a que no es sap la mida que té el patró que s'està buscant a la imatge. Per exemple, si la finestra de cerca (o d'entrenament) del detector és de 24 x 24, el conjunt exhaustiu de característiques rectangulars possibles és 160.000, és a dir, un nombre força elevat de característiques a calcular.

### 3.1.2 IMATGE INTEGRAL

Les característiques rectangulars poden ser computades ràpidament gràcies a una representació intermèdia de la imatge que s'anomena "imatge integral".



**Fig 3.3.** Imatge integral.

La posició  $(x,y)$  de la imatge conté la suma dels píxels superiors a  $y$  i a l'esquerra de  $x$ , ambdós inclosos:

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y')$$

on  $ii(x,y)$  és la imatge integral i la imatge original és  $i(x,y)$ .

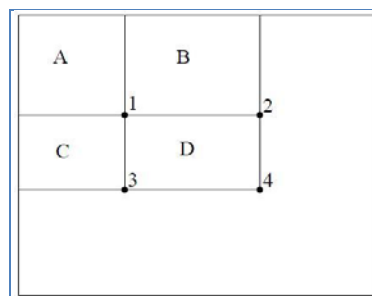
Usant un parell de recurrències, la imatge integral es pot obtenir recorrent la imatge una sola vegada:

$$s(x, y) = x(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

on  $s(x, y)$  és la suma acumulada de la fila  $x$  fins a  $y$ ,  $ii(-1, y) = 0$  i  $s(x, -1) = 0$

Sabent com s'obté, cal veure que usant la imatge integral qualsevol suma rectangular es pot calcular en només 4 accessos a una taula. A més, la diferència entre les sumes de dos rectangles es pot computar en 8 accessos. En la figura 3.4 el càlcul de  $D$  es pot obtenir sumant els punts de la següent manera:  **$D = 4 - 3 - 2 + 1$** .



**Fig 3.4.** Càlculs en la Imatge integral.

Al utilitzar les característiques de Haar rectangulars es pot fer servir aquest mètode per avaluar cada característica. Fins i tot, es poden reutilitzar punts que ja s'han consultat a memòria per obtenir la suma de rectangles anteriors, en cas de que siguin adjacents.

Resumint, es té que els 2-rectangles es computen en 6 accessos, 8 accessos pels 3-rectangles i 9 pels 4-rectangles. Aquesta és una de les aportacions més importants de l'algorisme de Viola&Jones i un dels motius de la seva gran rapidesa.

### 3.1.3 ENTRENAMENT DELS CLASSIFICADORS

Donat un conjunt de característiques, un conjunt d'exemples positius (és a dir, imatges que contenen l'objecte que es vol saber identificar) i un conjunt d'exemples negatius (imatges que no el contenen), es pot usar qualsevol algorisme per tal d'aprendre les funcions de classificació. El conjunt de funcions de classificació apreses s'anomena classificador.

Una funció de classificació relaciona l'espai de característiques (en aquest cas, la subfinestra de cerca) amb una classe o etiqueta (l'etiqueta determinarà si el patró apareix a l'espai de cerca o no).

Cal recordar que, en un cas típic, hi ha 160.000 rectangles de característiques associats a cada subfinestra de cerca. S'ha vist que cada característica pot ser computada eficientment, però computar el conjunt complet de característiques per cada subfinestra és quelcom prohibitiu.

Les hipòtesis dels autors és que un petit conjunt d'aquestes característiques pot ser combinat per construir un classificador eficaç. Així doncs, el repte és trobar quines són aquestes característiques.



El Viola&Jones utilitza una variant de l'algorisme AdaBoost per fer dues coses, en primer lloc seleccionar les característiques significatives i en segon lloc per entrenar el classificador. En la seva forma habitual, l'algorisme d'aprenentatge AdaBoost és utilitzat per millorar el rendiment d'algorismes d'aprenentatge simples. Això ho fa combinant una col·lecció de classificadors febles en un classificador fort.

Una manera pràctica de veure l'analogia entre classificadors febles de l'AdaBoost i les característiques del Viola&Jones és restringir el classificador feble a un conjunt de funcions de classificació, cadascuna de les quals depèn d'una sola característica de Haar.

Concretament pel Viola&Jones, l'algorisme d'aprenentatge del classificador febles és dissenyat per seleccionar la característica rectangular que millor separi els exemples positius dels negatius. Per cada característica, l'entrenament determina el llindar òptim de la funció de classificació tal que s'obtingui el mínim nombre d'exemples mal classificats (tan positius, com negatius).

Els classificadors febles resultants de l'aprenentatge que usa Viola&Jones, es poden veure com un únic node d'un arbre de decisió. Quan s'analitzi una nova subfinestra per realitzar la detecció el classificador feble determinarà si aquesta compleix el patró o no, seguint la següent inequació:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{si } pf(x) < p\theta \\ 0 & \text{altrament} \end{cases}$$

on  $f(x)$  és l'avaluació de la característica sobre la subfinestra  $x$ ,  $\theta$  el llindar i  $p$  la polaritat del classificador.

A continuació es descriu el procediment que segueix l'algorisme AdaBoost per construir el classificador fort combinant els classificadors febles:

**Es construeixen T hipòtesis usant les característiques simples. La hipòtesis final és una combinació lineal ponderada de les T hipòtesis on el pes és inversament proporcional als errors de l'entrenament.**

Donades les imatges d'exemple  $(x_1, y_1), \dots, (x_n, y_n)$  on  $y_i=0,1$  pels negatius i els positius respectivament.

Inicialitzar els pesos  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  per  $y_i=0,1$  respectivament, on  $m$  i  $l$  són el nombre d'exemples negatius i positius respectivament.

Per  $t=1 \dots T$ :

Normalitzar els pesos,  $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$

Seleccionar el millor classificador feble en funció de l'error ponderat

$$\varepsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

Definir  $h_t(x) = h(x, f_t, p_t, \theta_t)$  on  $f_t, p_t$  i  $\theta_t$  minimitzen l'error  $\varepsilon_t$ .

Actualitzar els pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

On  $e_i = 0$  si l'exemple  $x_i$  és classificada correctament,  $e_i = 1$  altrament.  $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$

El classificador final és:

$$f(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{altrament} \end{cases}$$

On  $\alpha_t = \log \frac{1}{\beta_t}$

### 3.1.4 CASCADA DE CLASSIFICADORS

L'última gran aportació de l'algorisme és la disposició en cascada dels classificadors simples, ja que aquesta permet obtenir una major eficàcia a la vegada que es redueix de manera important el temps de computació.

El fet clau és construir classificadors petits i eficients capaços de descartar moltes de les subfinestres negatives a la vegada que detecten, com a mínim, totes les positives. Els classificadors més simples descarten la majoria de subfinestres abans que els classificadors més complexes es cridin per aconseguir uns ràtios de falsos positius menors.

El procés que construeix la cascada esta guiat per uns objectius de detecció i rendiment inicials. Donada una cascada de classificadors ja entrenada, el ràtio de falsos positius és:

$$F = \prod_{i=1}^K f_i$$

On **F** és el ràtio de falsos positius de la cascada, **K** és el nombre de classificadors i  $f_i$  és el ràtio de falsos positius de l'*i*-èssim classificador sobre els exemples que arriben a ell. El ràtio de detecció és:

$$D = \prod_{i=1}^K d_i$$

On **D** és el ràtio d'encert de la cascada, **K** és el nombre de classificadors i  $d_i$  és el ràtio d'encert de l'*i*-èssim classificador sobre els exemples que arriben a ell.

Els ràtios de cada etapa del classificador en cascada venen determinats pels objectius de ràtios per falsos positius i de deteccions globals. Per exemple, un ràtio de detecció del 0.9 es pot aconseguir mitjançant 10 etapes que tinguin un ràtio de 0.99 (ja que  $0.9 \approx 0.99^{10}$ ).

Tot i que aconseguir aquesta taxa d'encert pot semblar una tasca molt costosa, només cal afrontar-ho amb un ràtio de falsos positius del 30% per aconseguir una taxa global de l'ordre de  $10^{-6}$  (ja que  $0.3^{10} \approx 6 \times 10^{-6}$ ), que es considera més que acceptable.

Per estimar quin és el nombre de característiques que són avaluades quan s'analitza una imatge es necessita un procés probabilístic. Qualsevol subfinestra pot progressar per la cascada de classificadors

fins que es decideix que és una finestra negativa, o bé pot continuar fins al final de la cascada i ser marcada com a positiva (fet que succeeix molt rarament per una imatge).

Una mesura de cada classificador és el “ràtio de positius”, que és la proporció de finestres que es marquen com a potencialment contenedores de l’objecte. Així, el nombre de característiques avaluades es pot estimar com:

$$N = n_0 + \sum_{i=1}^K \left( n_i \prod_{j<i} p_j \right)$$

On  $N$  és el nombre esperat de característiques avaluades,  $K$  és el nombre de classificadors,  $p_j$  és el ràtio de positius de l’ $i$ -èsim classificador i  $n_i$  és el nombre de característiques en el classificador  $i$ -èsim. El fet interessant és que, donat que són molt poques les vegades que una finestra positiva es correspon realment a un objecte, el “ràtio de positius” equival al ràtio de falsos positius de la imatge.

El procés complet d’entrenament implica uns avantatges i desavantatges. En la majoria de casos els classificadors amb més característiques assoleixen ràtios d’encert més elevats i ràtios de falsos positius més baixos. Al mateix temps, els classificadors amb més característiques requereixen més temps de computació.

Un entrenament òptim consistiria en modificar:

- El nombre d’etapes del classificador.
- El nombre de característiques,  $n_i$ , de cada etapa.
- El llindar de cada etapa.

amb l’objectiu de minimitzar el nombre de característiques avaluades  $N$ , per uns objectius  $F$  i  $D$ . Desafortunadament, trobar el valor òptim d’aquests paràmetres és un problema molt complex.

A continuació s’explica l’algorisme per construir la cascada de classificadors:

- L’usuari selecciona valor per  $f$  el ràtio màxim acceptable de falsos positius per etapa i  $d$  el ràtio mínim acceptable de detecció per etapa.
- L’usuari selecciona el ràtio de falsos positius global desitjat  $F_{target}$ .

$P$ =conjunt d’exemples positius

$N$ =conjunt d’exemples negatius

$F_0=1.0; D_0=1.0$

$i=0$

mentre  $F_i > F_{target}$

$i = i + 1$

$n_i = 0; F_i = F_{i-1}$

mentre  $F_i > f \times F_{i-1}$

$n_i = n_i + 1$

- Usar  $P$  i  $N$  per entrenar el classificador amb  $n_i$  característiques usant AdaBoost.
- Avaluar l'actual classificador en cascada per determinar  $F_i$  i  $D_i$ .
- Disminuir el llindar per l' $i$ -èsim classificador fins que l'actual classificador tingui, com a mínim, el ràtio  $d \times D_i$  (també afectarà a  $F_i$ ).
- $N=0$ .
- Si  $F_i > F_{target}$  avaluar l'actual detector en cascada sobre el conjunt de negatives i posar al conjunt  $N$  les que tinguin falsos positius.

### 3.1.5 DETECCIÓ

El resultat de l'entrenament és un classificador en cascada que funciona com un arbre de decisió. Per cada imatge que es vol analitzar per crear l'objecte, es té un conjunt de finestres candidates a processar. Cada finestra ha de recórrer l'arbre de decisió, avaluant a cada etapa el valor pel classificador. Un resultat positiu del primer classificador llença l'avaluació del segon classificador, que ha estat construït per tenir millors ràtios de detecció. Un resultat positiu del segon llença l'avaluació del tercer i així successivament. Un resultat negatiu a qualsevol punt del procés descarta immediatament la subfinestra que s'està analitzant.

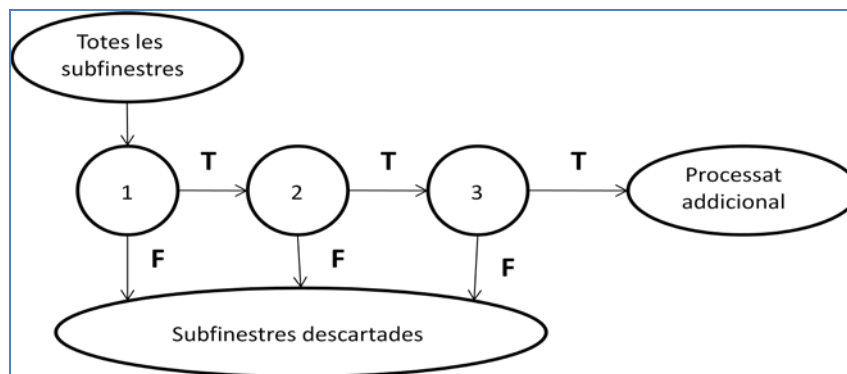


Fig 3.5. Procediment de l'algorisme de detecció.

Aquesta estructura en cascada basa el seu èxit i bon rendiment en que la majoria de subfinestres de la imatge són negatives. Per això la cascada intentarà descartar el més aviat possible les negatives mentre que una finestra positiva llençarà l'avaluació de tots els classificadors que formen la cascada, sent aquest fet molt poc habitual.

### 3.1.6 PROBLEMES

Es tracta d'un bon mètode per detectar qualsevol tipus d'objecte pel qual ha sigut entrenat. En el cas d'aquest projecte cares. El principal problema en la detecció de cares és que aquest mètode és capaç de detectar cares frontals, en canvi li costa detectar cares de perfil.

Es tracta doncs d'un mètode robust a l'hora de detectar cares frontals, però no tant quan s'ha de detectar cares de perfil.

## 3.2 MÈTODES DE RECONeixEMENT

El reconeixement facial és el procés de donar nom a una cara. Això és un cop detectada la cara, saber dir de quina cara es tracta.

### 3.2.1 DISTÀNCIA ENTRE IMATGES

Abans de començar amb els mètodes de reconeixement, s'explica el concepte de distància entre imatges ja que és molt important sobretot pels mètodes Eigenfaces i Fisherfaces.

La distància en el paper original és mesurada punt a punt (distància Euclidiana). En dos dimensions i tres dimensions, respectivament, la distància Euclidiana entre dos punts  $P_1$  i  $P_2$  és:

$$d_{12} = \sqrt{(dx^2 + dy^2)} \text{ (en 2D)}, \quad d_{12} = \sqrt{(dx^2 + dy^2 + dz^2)} \text{ (en 3D)}$$

on  $dx=x_2-x_1$ ,  $dy=y_2-y_1$  i  $dz=z_2-z_1$

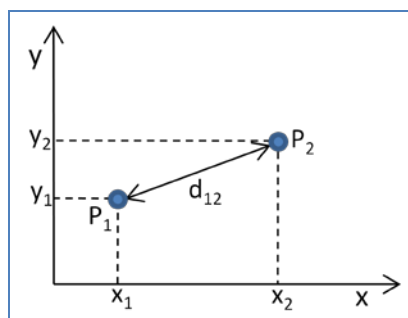


Fig 3.6. Distància Euclidiana entre dos punts en 2D.

En un espai de dues dimensions com el de la figura 3.6, les dimensions són els eixos X i Y. En un espai de tres dimensions les dimensions són els eixos X, Y i Z. Ara bé, quines són les dimensions d'una imatge facial? La resposta és que els mètodes basats en aparença consideren cada localització d'un píxel com una dimensió a part, però abans cal aplicar una reducció dimensional. Abans d'entrar en més detalls s'explica perquè es necessita una reducció dimensional.

Qualsevol imatge facial per petita que sigui està formada per una quantitat considerable de píxels. Considerant la mida més comuna d'imatges en el reconeixement facial, 50 x 50 píxels, es disposa d'una imatge amb una mida de 2500 píxels. Per computar la distància entre dues imatges es calcula la distància Euclidiana entre dues imatges, utilitzant els píxels com a dimensions.

Del càlcul anterior només s'obté un únic valor, per tant, partint de la base que en tota imatge hi ha soroll cal tenir en compte com afecta aquest soroll en el valor de la distància. S'entén per soroll qualsevol cosa que afecta a la "brillantor" del píxel; dues imatges no són exactament iguals degut al soroll, que canvia la "brillantor" dels píxels. Per tant, si cada un dels 2500 píxels duu incorporat una petita fracció de soroll, per molt petita que sigui, com que es té en compte cada píxel pel càlcul de la distància entre imatges, el nivell de soroll final acumulat dels 2500 píxels serà molt elevat, aquest fet pot dificultar molt la fase de reconeixement.

En mig de totes aquestes contribucions de soroll, segur que hi ha la informació suficient per identificar individualment les cares, però amb 2500 píxels afegint soroll resulta complicat. Una solució és utilitzar menys píxels i que aquests siguin els que aportin la informació suficient per poder fer la identificació.

Per aconseguir el que s'ha enunciat en el paràgraf anterior cal utilitzar una reducció dimensional.

---

## 3.2.2 EIGENFACES

Eigenfaces és un senzill algorisme de reconeixement facial. És el primer algorisme de reconeixement facial que tot estudiant de visió per computador ha de conèixer. **Turk and Pentland [4]** a l'any 1991 van publicar el paper que descriu el mètode de Eigenfaces.

Els passos utilitzats en el mètode Eigenfaces són utilitzats per molts altres mètodes. De fet, els experts en visió per computador recomanen estudiar aquest mètode pels qui estan interessats en aprendre els fonaments de la visió per computador, encara que no es dediquin al reconeixement facial. Una de les raons per les quals aquest mètode és tan important, és perquè molts dels principis bàsics que hi ha darrera d'aquest mètode (PCA i distance-based matching) apareixen en nombroses aplicacions de visió per computador i en aplicacions d'intel·ligència artificial, com les *machine learning applications*.

A continuació s'expliquen els passos a seguir per dur a terme el reconeixement facial: Donat un conjunt d'imatges d'exemple de cares de diferents persones, més una imatge d'una cara desconeguda (la cara a reconèixer),

1. Computar la distància entre la nova imatge i cada una de les cares d'exemple.
2. Seleccionar la imatge d'exemple més propera a la imatge de la cara a identificar.
3. Si la distància entre la imatge d'exemple i la imatge de la cara desconeguda és inferior a un llindar determinat, es senyala la cara com a reconeguda, altrament es classifica com a persona desconeguda.

### 3.2.2.1 PCA (ANÀLISI DE COMPONENTS PRINCIPALS)

---

Genèricament l'anàlisi de components principals, en estadística, és una tècnica utilitzada per reduir la dimensió d'un conjunt de dades per a poder-les representar gràficament en gràfics de dues o tres dimensions agrupant diverses variables de les dades en factors, o components, compostos per l'agrupació de diverses variables.

Dins l'àmbit d'Eigenfaces s'utilitza PCA per tractar el problema del reconeixement facial com si fos un problema de reconeixement en dues dimensions, sense la necessitat de recuperar la geometria en tres dimensions de qualsevol objecte. D'aquesta forma s'aprofita l'avantatge de que, en principi, les cares es troben en posició vertical en una imatge i, així, es poden arribar a descriure com un petit conjunt de característiques en dues dimensions. Aquest sistema funciona projectant les imatges facials sobre un espai de faccions que engloba les variacions significatives entre les imatges facials conegudes. Les faccions significatives s'anomenen eigenfaces, ja que són els eigenvectors (components principals) del conjunt de cares. D'altra banda no és necessari que aquestes eigenfaces siguin faccions com els ulls, les orelles o el nas. La projecció caracteritza la imatge facial d'un individu com la suma dels diferents pesos de totes les faccions (eigenfaces) i, de la mateixa manera, per reconèixer una imatge facial en particular només es necessita comparar aquests pesos amb aquells dels individus coneguts prèviament.

### 3.2.2.2 PROCEDIMENT

Per tal de reconèixer les cares s'ha de passar per 2 etapes clarament diferenciades: l'entrenament, amb les dades disponibles, i el reconeixement.

#### ENTRENAMENT

Per poder realitzar l'entrenament dels eigenfaces s'han de seguir els 7 passos següents:

1. Obtenir les imatges de la base de dades d'entrenament. És molt important que les imatges estiguin centrades i tinguin la mateixa mida.
2. Cal representar cada imatge  $I_i$  com un vector  $\Gamma_i$ . Això s'aconsegueix concatenant les columnes de  $I_i$ . A continuació es mostra un exemple de com representar una imatge com un vector.

$$I_1 = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1j} \\ P_{21} & P_{22} & \cdots & P_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ P_{i1} & P_{i2} & \cdots & P_{ij} \end{pmatrix}_{i \times j} ; \Gamma_1 = \begin{pmatrix} P_{11} \\ P_{21} \\ \vdots \\ P_{i1} \\ P_{12} \\ P_{22} \\ \vdots \\ P_{i2} \\ P_{13} \\ \vdots \\ P_{ij} \end{pmatrix}_{n \times 1}$$

3. Considerant que es disposa de  $M$  imatges, cal calcular la mitjana dels vectors. Aquest vector mitjana s'anomena  $\Psi$ , i es calcula segons la següent equació:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

4. Restar la mitjana  $\Psi$  a tots els vectors que representen les imatges  $\Gamma_i$  i formar la matriu  $A$  amb els vectors resultants de restar  $\Psi$  amb cada vector  $\Gamma_i$ :

$$\Phi_i = \Gamma_i - \Psi$$

$$A = (\Phi_1 \quad \Phi_2 \quad \cdots \quad \Phi_M)_{n \times M}$$

5. Calcular la matriu de covariància  $C$ :

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = \frac{1}{M} A A^T = A A^T, (\text{matriu } n \times n)$$

6. Per trobar els components principals, partint de la matriu de covariància  $C$  s'obtenen els eigenvectors i els seus corresponents eigenvalues. Trobant els eigenvalues més grans i els seus eigenvectors associats, permetrà obtenir un espai dimensional que descriurà la variació entre el conjunt d'imatges d'entrenament. El problema en aquest càlcul recau en que computar els eigenvectors de  $C$  és molt costós, per tant s'opta per calcular els eigenvectors de la matriu  $A^T A$ , que conté  $M$  eigenvectors que també són eigenvectors de la matriu  $C$ .

Considerant el vector  $\mathbf{u}$ , si és un eigenvector de  $C$ , també ho és de  $AA^T$ :

- 1)  $\exists \lambda \in \mathbb{R} \ni Cu = \lambda u \Rightarrow \exists \lambda_M \in \mathbb{R} \ni AA^T u = \lambda_M u$
- 2)  $\exists \lambda \in \mathbb{R} \ni AA^T v = \lambda v \Rightarrow \exists \lambda_C \in \mathbb{R} \ni Cv = \lambda_C v$

Suposant que  $\mathbf{u}$  és un eigenvector de  $C \Rightarrow Cu = \lambda u, \lambda \in \mathbb{R}$ . Tenint en compte que  $C = \frac{1}{M} AA^T \Rightarrow \frac{1}{M} AA^T u = \lambda u \Rightarrow AA^T u = M\lambda u$  ( $\lambda_M = M\lambda$ ) es conclou que  $\mathbf{u}$  també és eigenvector de  $AA^T$ .

Si  $\mathbf{v}$  és eigenvector de  $AA^T \Rightarrow AA^T v = \lambda v, \lambda \in \mathbb{R}$ . Tenint en compte que  $C = \frac{1}{M} AA^T \Rightarrow MCv = \lambda v \Rightarrow Cv = \frac{\lambda}{M} v = \lambda_C v$  ( $\lambda_C = \frac{1}{M} \lambda$ ) es conclou que  $\mathbf{v}$  també és eigenvector de  $C$ .

En aquest procediment es demostra que per 2 eigenvectors qualsevols  $\mathbf{u}, \mathbf{v}$  de les matrius  $C$  i  $AA^T$  respectivament,  $\mathbf{u}$  és també eigenvector de  $AA^T$  i  $\mathbf{v}$  és eigenvector de  $C$ . El que es pretén demostrar és que la matriu de  $M \times M$   $A^T A$  té  $M$  eigenvectors que també són eigenvectors de  $C$ .

Suposant que  $\mathbf{k}$  és un eigenvector de  $M \times 1$  de  $A^T A$ , es demostra que qualsevol eigenvector  $\mathbf{k}_{M \times 1}$  de  $A^T A$  és un eigenvector de  $AA^T$  i també un eigenvector de  $C$ :

$$A^T A k = \lambda k \quad (\lambda \in \mathbb{R}) \Rightarrow AA^T A k = A \lambda k \Rightarrow (AA^T) A k = \lambda A k \Rightarrow C A k = \lambda_C A k \quad (\text{on } \lambda_C = \frac{1}{M} \lambda)$$

Dels eigenvectors, cal quedar-se amb els millors.

7. Es guarden únicament els  $K$  eigenvectors millors, que són els que corresponen als  $K$  eigenvalues més grans.

Cada cara  $\Phi_i$  de l'entrenament menys la mitjana pot ser representada com una combinació lineal dels seus  $K$  millors eigenvectors:

$$\Phi_i - \Psi = \sum_{j=1}^K w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

Cada cara  $\Phi_i$  és representada pel següent vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \dots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \dots, M$$

## RECONeixEMENT

Donada una imatge d'una cara, l'objectiu és saber reconèixer-la, és a dir, saber identificar-la. Per fer això s'utilitzen els següents 5 passos:

1. Cal normalitzar la cara, és a dir, de la mateixa manera com s'ha fet durant la fase d'entrenament de dades (passos 2,3 i 4 entrenament). Cal representar imatge en un vector  $\Gamma$ . Un cop fet això, es busca la mitjana  $\Psi$  (obtinguda durant l'entrenament).



2. Un cop es té la cara normalitzada es projecta sobre l'eigenespai:

$$\hat{\Phi} = \sum_{j=1}^K w_j u_j, (w_j = u_j^T \Phi)$$

3. Es representa  $\phi$  com:

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_K \end{bmatrix}$$

4. Es busca la diferència mínima entre la representació de la imatge tractada i la representació de totes les cares d'entrenament utilitzades. Per calcular aquest error podem utilitzar els algorismes de la distància Euclidiana o bé Mahalanobis:

$$\|\Omega - \Omega^k\| = \sum_{i=1}^K \frac{1}{\lambda_i} (w_i - w_i^k)^2$$

S'indicarà que s'ha mostrat una cara quan s'ha trobat la representació de la base de dades amb la distància mínima amb la cara introduïda. El fet, és que és possible que la cara que s'intenta classificar no estigui a la base de dades, per tant, s'ha de determinar un llindar.

## PROBLEMES

Hi ha varis factors a tenir en compte a l'hora de realitzar el reconeixement de les cares amb el mètode eigenfaces:

- Tant les imatges a entrenar com les imatges de les cares per reconèixer s'han d'introduir centrades i sense gaire fons.
- Les condicions de llum poden crear degradats en la imatge, aquest fet pot alterar el reconeixement de les cares.
- Cal escalar totes les imatges per tal que tinguin la mateix mida.
- L'angle de rotació pot influir en el procés.

### 3.2.3 FISHERFACES

Fisherfaces és una variació d'Eigenfaces, de fet, va ser proposat com una millora del Eigenfaces. Tenint en compte que el conjunt d'entrenament està etiquetat, és possible utilitzar aquesta informació per reduir la dimensió de l'espai de característiques.

A diferència de Eigenfaces aquest mètode no busca mantenir la topologia de l'espai original, és a dir, l'objectiu ara és portar les cares a un espai on es pugui distingir de millor forma les diferents classes entre si, maximitzant la distància entre imatges de diferents classes i a la vegada minimitzant la distància entre imatges d'una mateixa classe. En aquesta nova representació, quan es projecti una cara desconeguda s'espera que aquesta quedi a prop d'un grup de cares i lluny de la resta de cares, d'aquesta forma el reconeixement serà confiable.

Resumint, aquest mètode produeix una millor discriminació entre classes, preservant al mateix temps la propietat de baixa discriminació de la representació d'Eigenfaces dins de cada classe. També es considera que Fisherfaces és l'aplicació de l'anàlisi discriminant lineal (LDA) sobre l'espai generat per PCA.

### 3.2.3.1 LDA (ANÀLISI DE DISCRIMINANT LINEAL)

L'esquema LDA permet utilitzar la informació entre membres de la mateixa classe (imatges del mateix individu) per desenvolupar un conjunt de vectors de característiques on les variacions de les diferents cares s'emfatitzen mentre que els canvis d'expressió facial i orientació de la cara no ho fan.

Explicat d'una altra manera, el PCA no té en compte la informació de quines imatges pertanyen a un individu, mentre que el LDA sí que ho fa. Per aquest motiu el inconvenient del PCA sobre el LDA és que maximitza la variància de totes les mostres sense tenir en compte la classe a la que pertanyen. Per la seva part, el LDA intenta minimitzar la variància entre mostres de la mateixa classe. Formalment, això s'aconsegueix construint dos matrius de dispersió a partir de les imatges d'entrenament; en una de les quals es representa la dispersió entre les diferents classes (diferents individus), i a l'altra es representa la dispersió entre membres de la mateixa classe.

Una de les limitacions que té LDA, és que es requereix de matrius de dispersió no-singulars. Això en aplicacions de reconeixement facial és complicat degut a que en moltes ocasions l'alta dimensionalitat de les imatges és molt major que el nombre d'imatges, fet que comporta un problema de matrius singulars. Per tal de solucionar aquest problema s'han proposat diferents alternatives. L'exemple més clar és el que proposa [6] amb el que es coneix amb el nom de *Fisherfaces*, en honor de Robert Fisher, qui va desenvolupar la tècnica de discriminació lineal per al reconeixement de patrons. El nom de Fisher Linear Discriminant és equivalent al LDA.

### 3.2.3.2 PROCEDIMENT

Per tal de reconèixer les cares s'ha de passar per 2 etapes clarament diferenciades: l'entrenament amb les dades disponibles i el reconeixement.

#### ENTRENAMENT

Amb l'objectiu de maximitzar la variància de les mostres entre classes i minimitzar la variància entre mostres de la mateixa classe es construeixen dos matrius de dispersió a partir de les imatges d'entrenament; en una de les quals es representa la dispersió entre membres de la mateixa classe (imatges del mateix individu). Les matrius es defineixen com:

$$S_B = \sum_{i=1}^c P_i (\mu_i - \mu) \cdot (\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^c P_i \sum_{j=1}^{N_i} (x_j^{(i)} - \mu_i) \cdot (x_j^{(i)} - \mu_i)^T$$

On s'assumeix que les  $c$  classes es donen amb una probabilitat a priori  $P_i$ . El nombre d'imatges per classe és  $N_i$ . Les imatges en el conjunt d'entrenament es representen mitjançant vectors de  $n$  dimensions on  $x_j^{(i)}$  denota la  $j$ -èsima imatge que pertany a la classe  $i$ . El vector mitjana d'una mateixa classe es calcula a partir de:

$$\mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_j^{(i)}$$

essent  $\mu$  la mitjana total de tots els vectors d'entrenament:

$$\mu = \frac{1}{\sum_{i=1}^c N_i} \cdot \sum_{i=1}^c \sum_{j=1}^{N_i} x_j^{(i)}$$

La matriu de dispersió entre classes  $S_B$  recull la dispersió dels vectors mitjana de cada classe respecte el vector mitjana total. El factor  $P_i$  en aquesta matriu representa la probabilitat de cada classe, i generalment s'estima com el quocient entre  $N_i$  i el nombre total d'imatges en el conjunt d'entrenament. D'altra banda, la matriu de dispersió dels elements d'una mateixa classe  $S_W$  representa la dispersió dels elements d'una mateixa classe respecte el vector mitjana d'aquella mateixa classe. Aquesta matriu és la suma de les diferents probabilitats ponderades de la matriu de covariància entre membres de la mateixa classe. Per tant,  $S_W$  calcula, d'alguna manera, el soroll de cada imatge d'entrenament respecte el vector mitjana de la seva classe.

El mètode LDA proposa projectar les imatges sobre un conjunt de  $k$  vectors ortogonals que maximitza la següent funció:

$$V_{opt} = \arg \max_V \left\{ \frac{|V^T S_B V|}{|V^T S_W V|} \right\}$$

on  $V_{opt}$  representa una matriu amb  $k$  vectors-columna ortogonals amb  $n$  components cadascun.

Un cop es determina la matriu  $V_{opt}$ , la informació es projecta sobre un subespai reduït de dimensió  $k$  a partir de:

$$y = V_{opt}^T x$$

on  $x$  és el vector de dimensió  $n$  que representa la informació (píxels de la imatge en forma de vector).

Clarament, el criteri d'optimització anterior implica que  $V_{opt}$  maximitza la distància projectada entre els vectors que pertanyen a les diferents classes però també intenta recopilar les imatges projectades que pertanyen a la mateixa classe de manera conjunta. Aquest mètode utilitza la informació de la matriu de distorsió d'imatges de la mateixa classe per optimitzar l'agrupació de la informació després de la projecció. D'altra banda, quan es construeix la matriu de covariància en el mètode PCA, no s'introdueix cap tipus d'informació sobre l'existència de diferents classes.

La diferència entre els mètodes PCA i LDA és que el mètode PCA torba el vector de projecció òptim en la direcció de màxima energia de la informació mentre que el sistema LDA optimitza el vector de projecció per obtenir la discriminació màxima.

Sempre i quan la matriu de distorsió d'imatges de la mateixa classe sigui no singular es pot reduir el problema anterior segons la següent equació generalitzada:

$$S_B \cdot v_i = \lambda_i S_W v_i, i = 1, 2, \dots, k$$

on  $v_i$  són els eigenvectors (columnes de  $V_{opt}$ ) associades als  $k$  eigenvalues més grans. No obstant, per al problema de reconeixement facial, l'equació anterior no és viable, ja que la matriu  $S_W$  sempre és no singular.

## RECONeixEMENT

Per al problema de reconeixement facial, l'equació anterior no és viable, ja que la matriu  $S_W$  sempre és no singular. Això és degut a la seva construcció, el rang és com a molt  $N-c$  i, en general, el nombre d'imatges en el conjunt d'entrenament ( $N$ ) és bastant més petit que el nombre de píxels de les imatges ( $n$ ).

Per tal de solucionar aquest problema s'han proposat diferents alternatives. L'exemple més clar és el que proposa Robert Fisher (Fisherfaces), qui va desenvolupar la tècnica de discriminació lineal per al reconeixement de patrons. El nom de Fisher Linear Discriminant (FLD) s'usa també per al procediment descrit anteriorment i és equivalent al LDA.

La tècnica de les Fisherfaces es basa en reduir la dimensionalitat dels vectors que contenen la informació mitjançant l'algorisme de PCA abans d'aplicar el mètode FLD. Després de projectar el conjunt d'imatges sobre un subespai de dimensions més petites la matriu resultant  $S_W$  és no singular i se li pot aplicar el mètode de classificació fisher.

Això s'aconsegueix utilitzant l'algorisme PCA per reduir la dimensió de l'espai de característiques a  $N-c$  i després aplicant l'estàndard LDA per reduir la dimensió a  $c-1$ . Formalment, la matriu de projecció es descompon en dues parts:

$$V_{opt}^T = V_{fld}^T \cdot V_{pca}^T$$

on

$$V_{pca} = \arg \max_V \{|V^T S_T V|\}$$

$$V_{fld} = \arg \max_V \left\{ \frac{|V^T V_{pca}^T S_B V_{pca} V|}{|V^T V_{pca}^T S_W V_{pca} V|} \right\}$$

En el mètode original la optimització per  $V_{pca}$  es calcula a partir de matrius amb columnes ortonormals de  $n \times (N-c)$ , mentre que la optimització per  $V_{fld}$  s'aconsegueix, també, mitjançant matrius amb columnes ortonormals de  $(N-c) \times m$ . En resum, a l'hora de calcular  $V_{pca}$ , només es descarten les  $c-1$  components principals més petites.

El reconeixement es resol obtenint la distància mínima entre els vectors de característiques i els de test (imatge facial a reconèixer).

## PROBLEMES

Fisherfaces obté millors resultats que Eigenfaces quan les condicions d'il·luminació són variables (hi ha diverses condicions d'il·luminació dins de les imatges d'entrenament i de test). L'anàlisi FLD permet retenir l'estructura de la cara sota aquestes condicions variables, mentre que no succeeix amb el PCA. Això s'explica perquè les diferents il·luminacions es poden modelar utilitzant les superfícies de Lambert.

Els problemes sorgeixen quan hi ha variacions en les expressions facials de les imatges. Això és degut a que quan s'entrena un sistema amb un conjunt d'imatges d'individus amb diferents expressions facials, les fisherfaces que s'utilitzen donen menys importància als píxels situats a prop de la boca, que és la zona més variable als canvis d'expressió. El sistema dona més pes als ulls, al nas i a les galtes ja que són zones més invariables en les diferents imatges d'una persona.

També és molt important que les imatges tinguin mateixa mida i que estiguin ben alineades.

---

### 3.2.4 ASM

La idea principal d'aquest mètode resideix en el fet que tot i saber quin és l'objecte (en aquest cas la cara) que es busca dins de la imatge i la forma que té, aquest pot aparèixer amb alguna deformació. Així doncs, es defineixen uns punts, que donen la forma a l'objecte, els quals accepten una petita variació segons cap on evoluciona la figura.

#### 3.2.4.1 PROCEDIMENT

---

### ENTRENAMENT

La forma de l'objecte es pot definir segons punts que delimiten el contorn, aquests punts han d'acceptar variabilitat, és a dir, han de moure's per tal d'adaptar-se a la forma de la figura. Aquesta forma base que posteriorment s'utilitzarà per fer les variacions rep el nom de forma mitjana o mean shape. Les variacions de la forma mitjana venen donades per la variació dels punts de la imatges d'entrenament (expressades per la matriu de covariàncies).

Dit d'una altra manera, en l'etapa d'entrenament es calcula un model estadístic sobre com varia la forma a través d'unes imatges de mostra. Tot seguit s'analitza com es duu a terme el càlcul d'aquest model.

La forma que es troba de l'objecte (cara) a les imatges d'entrenament es delimita a través de **n landmarks** distribuïts al llarg del contorn de l'objecte. Les variacions en les coordenades d'aquests punts característics seran les que marcaran les variacions de forma de l'objecte. Tenint aquests punts es pot definir la forma de l'objecte com un vector:

$$x = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)$$

El següent pas, l'avaluació i organització dels punts obtinguts a l'entrenament es fa aplicant un anàlisi de components principals (PCA) que calcula els eixos principals en els quals estan les dades distribuïdes contant que el sistema no té dimensions.

Dins el context del reconeixement facial, en la següent imatge es mostra com es marquen els punt característics:



Fig 3.7. Punts característics.

També es mostra un exemple on es mostra la mitjana d'una cara en el punt inicial per començar l'adaptació a la forma, les iteracions són un paràmetre que es pot configurar (no sempre més iteracions corresponen a més convergència). És molt important el punt inicial on es col·loca el mean shape.



Fig 3.8. Iteracions amb el mean shape.

## CERCA

La segona part del mètode és la de fer variar el model obtingut a l'entrenament fins obtenir un resultat final que s'adapti a la forma de la nova imatge. Tenint en compte els paràmetres es crea un model inicial  $X$  on es defineix la posició, orientació i escala i es van aproximant aquests valors amb el següent procés iteratiu:

1. Examinar la regió de la imatge al voltant de cada punt del model  $X$  per trobar el millor  $X'_i$ .
2. S'actualitzen els paràmetres amb el punt nou que és millor que l'anterior.
3. Repetir 1 i 2 fins a la convergència final.

Cada punt del model és examinat segons la imatge. Tot i aquesta transformació tan ideal no sempre els punts característics estan col·locats en el lloc correcte, a la imatge poden haver-hi més regions o formes i que, per qüestions de l'entrenament, un punt s'hagi fixat dins d'algun d'aquests altres perfils existents.

Es presenten dos aproximacions per tal de localitzar punts correctes del contorn. Un dels mètodes consisteix en construir un model estadístic de l'estructura de la imatge de manera que quan es cerquin els punts s'acceptaran els que més s'ajustin al model.

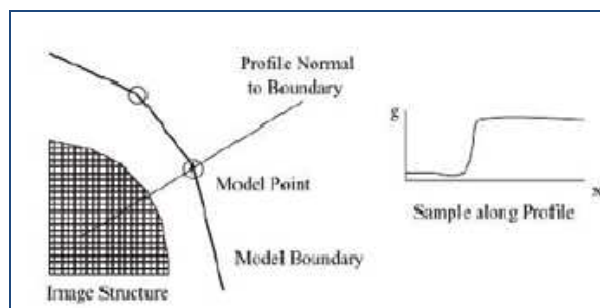


Fig 3.9. Cerca de punts.

El primer mètode per tal d'analitzar els punts de l'estructura rep el nom de *Modeling local structure*. El procediment que es segueix és agafar un punt del contorn i els píxels veïns creant un model estadístic del nivell de gris del contorn de la imatge. Cada vegada que es tingui un punt a analitzar serà comparat amb aquest model i si hi ha correspondència s'acceptarà creant així un nou punt a analitzar, d'aquesta manera la forma mitja anirà evolucionant fins a trobar la forma o fins que el procés iteratiu arribi al final.

El segon mètode contempla la possibilitat de tractar la tasca d'evolució com una classificació. Es recullen exemples que es sap que són certs, a la vegada que es recullen característiques de la imatge que estiguin situades a prop del contorn i que es sàpiga que són incorrectes, és a dir, que de cap manera pertanyen al contorn. Una vegada es tenen les mostres es creen unes classes que permetran diferenciar punts correctes de punts incorrectes i a l'hora de buscar punts o analitzar-los s'escolliran els que corresponguin als models certs i els que més difereixin i continguin menys característiques de les classificades com a negatives.

## CONSIDERACIONS

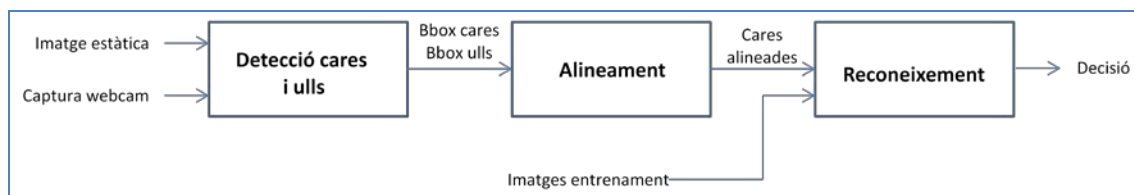
Amb aquest mètode es modela l'estructura de forma efectiva i bàsicament el que farà serà recorre els punts de la normal del perfil en els contorns de l'entrenament per finalment construir el model estadístic en nivell de gris.

### 3.3 ASPECTES TÈCNICS

En aquesta secció es donen a conèixer els detalls de la implementació dels components de detecció, alineament i reconeixement.

Un dels objectius és integrar els components anteriors amb la intenció de disposar d'un sistema amb dues entrades, una imatge estàtica o bé la captura realitzada per una webcam. El mòdul de detecció mitjançant Viola & Jones extreu les cares i els ulls de cada cara. El mòdul d'alineament rep les coordenades de les "bounding box" d'ulls i cares i, prenent com a referència l'ull esquerre de cada cara escala i centra la cara per tal de facilitar el reconeixement.

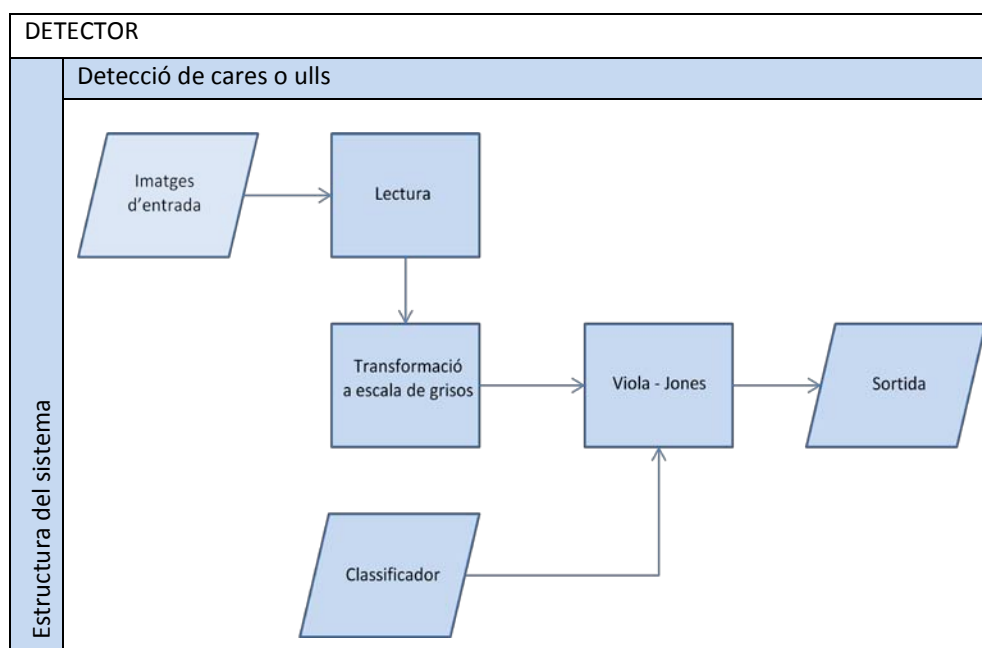
Les cares correctament centrades i alineades passen al mòdul de reconeixement on mitjançant el mètode Eigenfaces o Fisherfaces es reconeix el personatge de la cara.



**Fig 3.10.** Diagrama de l'arquitectura del sistema.

### 3.3.1 DETECCIÓ DE CARES I ULLS

Per realitzar aquesta tasca més ràpida i eficient i no haver d'implementar de zero un algorisme de detecció, s'opta per utilitzar el mètode Viola & Jones que incorpora la llibreria OpenCV. Com que OpenCV són llibreries per C++ i en aquest projecte s'utilitza com a entorn de programació Matlab, la crida al procediment de detecció es fa mitjançant l'ús d'un fitxer Mex.



**Fig 3.11.** Etapa de detecció.

L'esquema d'aquest detector s'utilitza per extreure les cares i els ulls de les persones que hi ha a la imatge d'entrada. Per trobar l'objecte que es vol buscar es transforma la imatge a escala de grisos i s'hi aplica el mètode Viola & Jones per dur a terme la detecció.

En funció de l'objecte que es vol detectar s'utilitza un classificador diferent, en aquest cas per detectar cares s'utilitza un classificador entrenat per detectar cares i per detectar els ulls s'utilitza un classificador entrenat per detectar ulls.

La sortida del sistema de detecció són les bounding box dels objectes detectats, en aquest cas, cares o ulls.



Per la detecció es parteix d'una factor escala mínim de 1.1 i el procediment de detecció duu a terme varies iteracions sobre la imatge augmentant el factor escala en cada iteració fins a trobar objectes.

Es disposa de més informació sobre com configurar Matlab per generar fitxers Mex en els annexos.

### 3.3.2 ALINEAMENT

Quan s'extreuen les cares de les persones a reconèixer aquestes poden presentar diferents dimensions, poden estar rotades i descentrades. Per facilitar el reconeixement i obtenir bons resultats és molt important normalitzar les imatges de les cares que reben els algorismes de reconeixement, és a dir, aquestes imatges cal que estiguin ben escalades (han de tenir la mateixa mida que les imatges d'entrenament dels algorismes de detecció) i centrades.

Actualment hi ha moltes maneres de normalitzar les imatges però és una tasca complicada que requereix molt de temps i dedicació, i per limitacions de temps no es contempla dins d'aquest projecte. Però si que s'ha implementat un senzill mètode d'alineament suficientment vàlid per ser integrat entre el mòdul de detecció i reconeixement.

El mètode en qüestió rep les cares i la posició dels ulls de cada cara procedents del component de detecció (en cada cara detectada s'hi aplica també Viola & Jones per detectar els ulls) i llavors s'inicia el procediment següent, tenint en compte que es coneix la posició dels ulls, "l'offset" que hi ha d'haver entre ulls i marge esquerre i dret i les dimensions que ha de tenir la imatge normalitzada:

- **Es parteix de la cara i la posició dels ulls.**

Amb Viola & Jones es detecta la cara i els ulls.

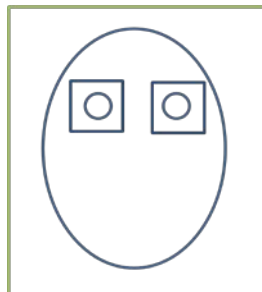


Fig 3.12. Detecció de cara i ulls.

- **L'angle entre ull esquerre i ull dret ha de ser de zero graus.**

Es calcula l'angle entre l'ull esquerre i l'ull dret, i es rota la imatge en funció de l'angle. De tal forma que l'angle que formen els dos ulls sigui de zero graus:

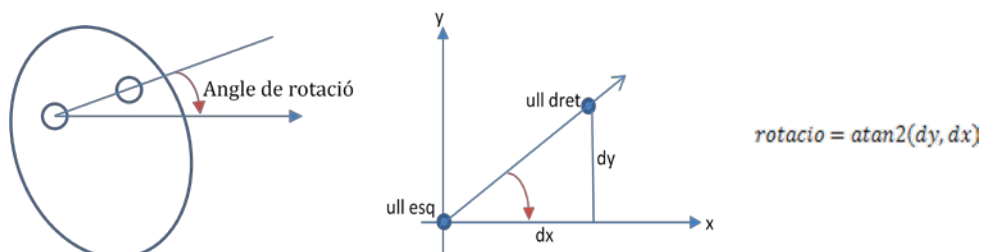


Fig 3.13. Angle de rotació perquè els ulls quedin alineats.

- **Offset horitzontal.**

Es defineix un offset entre ull esquerre i ull dret. L'offset entre els ulls i els marges es fixa en un 30%.

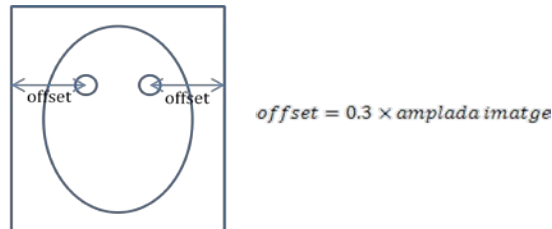


Fig 3.14. Offsets entre ulls.

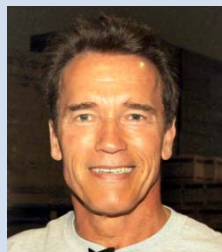
- **Redimensió de la imatge i crop.**

S'escala la imatge a unes dimensions determinades mantenint la proporcionalitat, és a dir, que la cara no quedi deformada, posteriorment es duu a terme un procés de crop (retallar la regió exacta de la cara). La mida estàndard utilitzada per les imatges és 92 píxels d'amplada i 112 d'alçada.

Aplicant aquesta normalització es redueix la càrrega computacional dels procediments posteriors. És molt important que les cares siguin frontals, ja que el sistema de reconeixement només reconeix cares frontals. A continuació es mostra un exemple per una fotografia real:

**Offset i dimensions (alçada, amplada)**

**Imatge original**



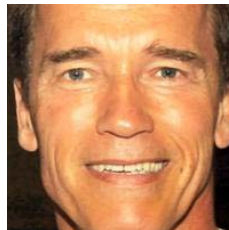
0.1 (10%), (200,200)



0.2 (20%), (200,200)



0.3 (10%), (200,200)



0.2 (10%), (70,70)



Fig 3.15. Exemple alineament.

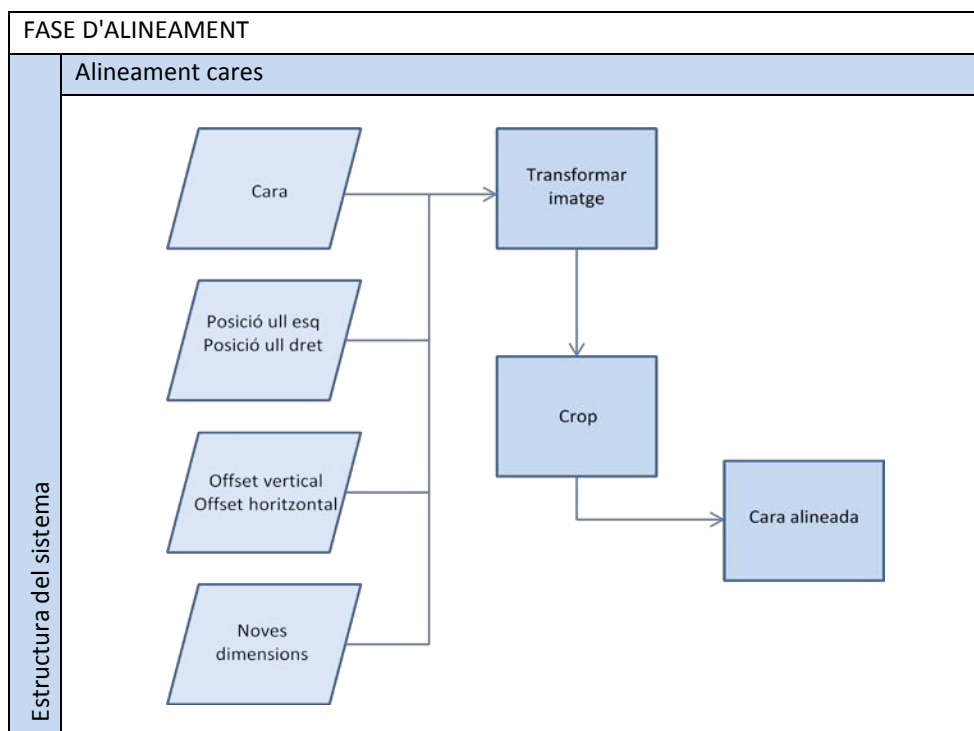


Fig 3.16. Etapa d'alineament.

### 3.3.3 RECONeixEMENT

Per limitacions temporals i problemes sorgits en les implementacions no s'han pogut implementar tots els mètodes de reconeixement (dins de les conclusions es justifiquen més detalladament els problemes les desviacions temporals). Els mètodes implementats han estat els mètodes basats en aparença, Eigenfaces i Fisherfaces.

Per dur a terme la implementació dels mètodes en qüestió s'han considerat els següents factors, que s'exposen en els següents apartats.

### 3.3.3.1 BASE DE DADES

Es disposa d'una base de dades de cares de persones. Per cada una d'aquestes persones és necessari tenir un nombre d'imatges, en el cas d'aquest projecte s'utilitzen 10 imatges de la cara de l'individu en diferents posicions.

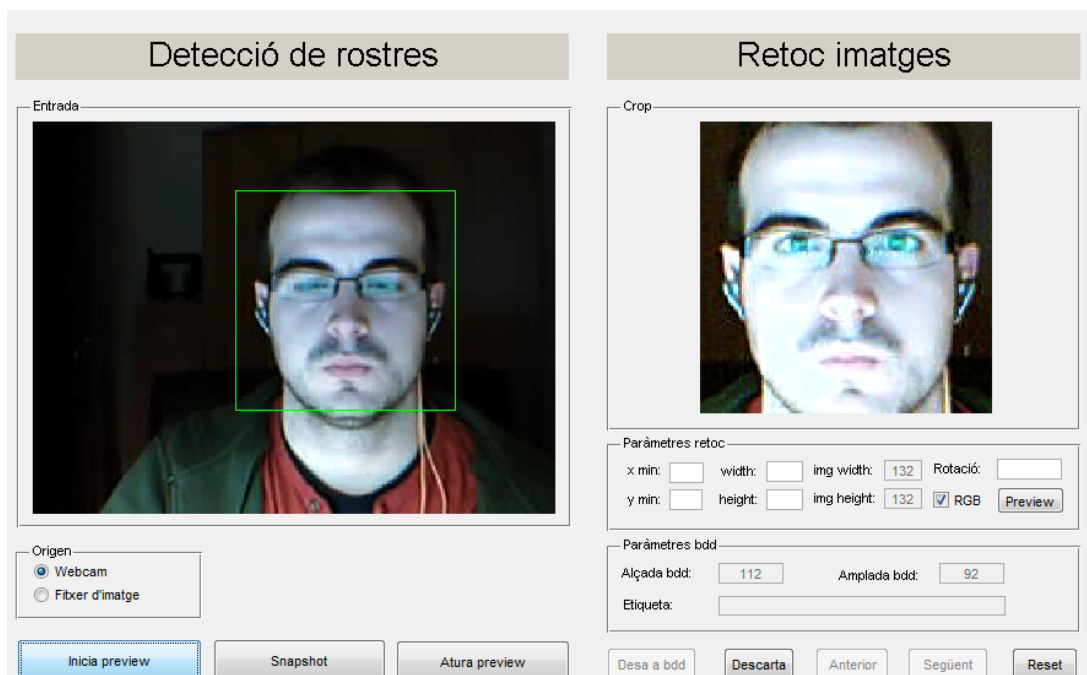


**Fig 3.17.** Exemple de les cares que es guarden per una persona a la base de dades pels mètodes eigenfaces i fisherfaces.

Una mostra de 10 imatges ja es considera suficientment representativa ja que amb aquesta quantitat els algorismes de reconeixement ja obtenen bons resultats i el cost de computació de l'entrenament no és molt elevat.

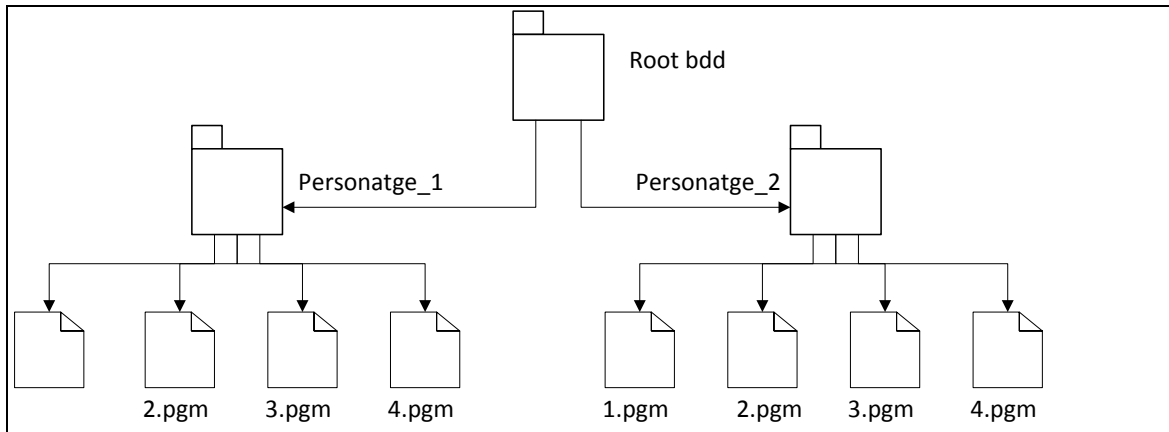
Les cares de la base de dades d'entrenament poden estar guardades en escala de grisos o bé en l'espai de color RGB (en el moment de carregar la base de dades pel seu posterior ús en el reconeixement les imatges ja es converteixen a escala de grisos).

Per tal d'incorporar noves cares, s'ha desenvolupat una petita aplicació per tal que l'usuari pugui inserir la seva imatge a la base de dades de forma fàcil i intuïtiva. A continuació es mostra una captura de pantalla de l'aplicació per inserir nous elements a la base de dades.

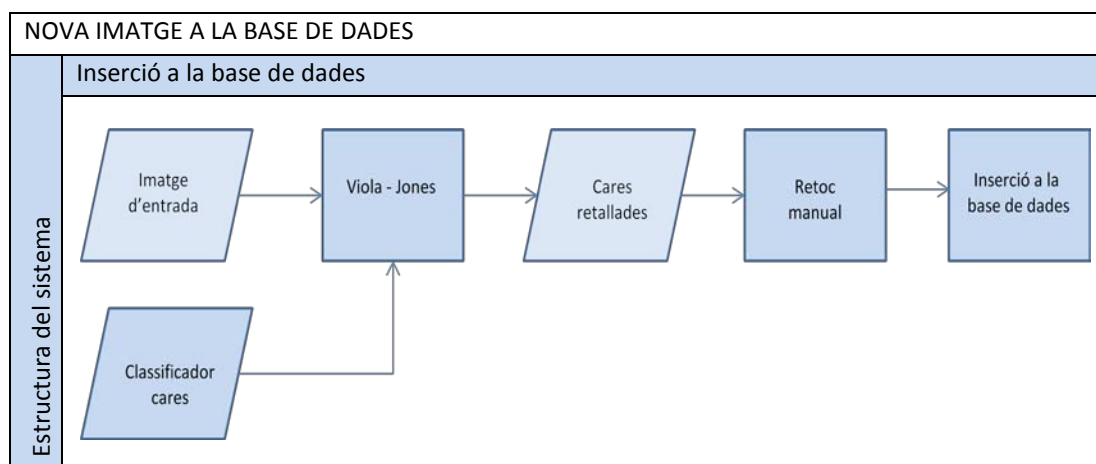


**Fig 3.18.** Pantalla per inserir nous elements a la base de dades.

La imatge es pot retocar manualment (redimensionar, canviar l'espai de color i rotar) per tal de guardar-la normalitzada, i es pot guardar en RGB o escala de grisos. La lògica de l'estructura de directoris que segueix la base de dades utilitzada és la següent:



**Fig 3.19.** Estructura de directoris de la base de dades d'imatge d'entrenament.

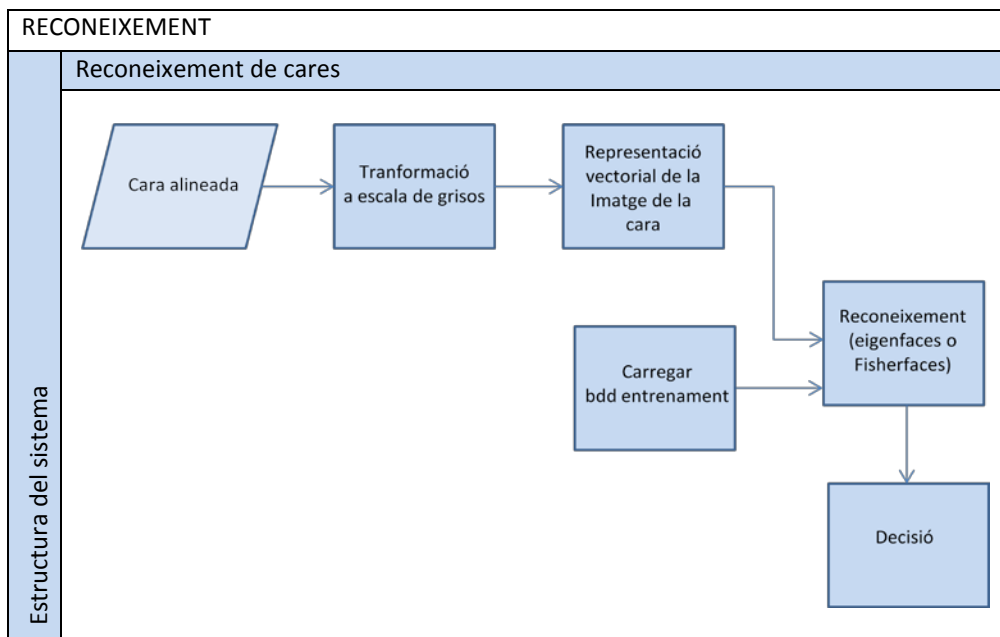


**Fig 3.20.** Alta d'una imatge a la base de dades.

### 3.3.3.2 RECONeixEMENT

En el mòdul de reconeixement es reben les cares normalitzades procedents del mòdul d'alineament, es càrrega la base de dades d'entrenament, s'aplica el mètode de reconeixement (Eigenfaces o Fisherfaces) i es mostra el resultat del reconeixement a l'usuari indicant el nom de la persona reconeguda a la imatge.

Per tal de no estar llegint del sistema de fitxers les imatges d'entrenament en la fase de reconeixement, aquestes ja es guarden vectoritzades dins d'un fitxer ".mat" de Matlab.



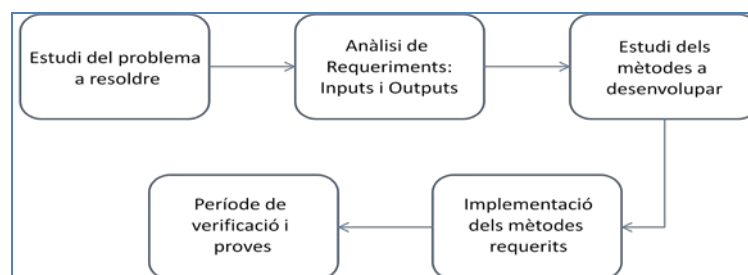
**Fig 3.21.** Etapa de reconeixement.

### 3.3.4 ANÀLISI I DISSENY

En aquesta secció es mostra l’anàlisi i disseny de l’aplicació prototipus, és a dir, els requeriments i especificacions.

#### 3.3.4.1 METODOLOGIA

Per tal d’elaborar l’aplicació s’ha seguit la metodologia descrita en la següent figura. Es tracta d’un model en “cascada”, on es tenen en compte els passos imprescindibles per tal de realitzar una bona aplicació.



**Fig 3.22.** Metodologia emprada.

En primer lloc s’ha estudiat el problema a resoldre, fins a quin nivell es vol arribar, i quins aspectes ha d’abastar aquesta aplicació.

Tot seguit s’han analitzat quines són les accions que ha de poder realitzar l’usuari, i quins són els elements que necessitarà l’aplicació per funcionar i de quina manera retornarà els resultats obtinguts. Un cop presa la decisió de com ha de ser l’aplicació, s’ha iniciat la integració amb els mètodes desenvolupats.

Posteriorment s'ha passat a implementar tots els mètodes de l'aplicació i també les interfícies de comunicació amb l'usuari.

Un cop desenvolupat el prototipus de l'aplicació, s'han realitzat les proves i tests pertinents per validar el correcte funcionament de l'aplicació envers a les diferents situacions possibles.

## ESTUDI DEL PROBLEMA A RESOLDRE

El primer pas és marcar quin és el problema a resoldre. Un cop fet això comença un estudi dedicat a resoldre aquest problema. En aquest cas, el problema és realitzar l'aplicació prototipus capaç de detectar i reconèixer les cares procedents d'una imatge estàtica o bé des d'una webcam mitjançant dos algorismes de reconeixement facial: Fisherfaces i Eigenfaces, per poder, posteriorment analitzar el comportament de cadascun.

Per fer possible això cal la correcta integració dels mètodes de detecció facial, alineament i reconeixement facial descrits en seccions anteriors.

## ANÀLISI DELS REQUERIMENTS

El programa ha de ser capaç de rebre una seqüència de vídeo a través d'una webcam instal·lada a l'equip i analitzar frame a frame per tal de detectar les cares i reconèixer-les (tot aquest procés ja ha estat explicat amb anterioritat). El sistema també ha de poder treballar amb imatges estàtiques (fotografies).

A més l'aplicació també ha de ser capaç de demostrar el correcte funcionament dels mètodes principals per separat (detecció i reconeixement), afegir nous elements a la base de dades per tal de tenir-los en compte per a properes execucions, etc...

L'aplicació retorna la informació a l'usuari de forma visual a través de la seva interfície d'usuari. Quan detecta una cara marca un requadre al seu voltant, mentre que quan reconeix una cara coneguda surt la informació que te sobre ella a la part superior del requadre.

## IMPLEMENTACIÓ DELS MÈTODES

En aquesta fase de la metodologia s'han implementat els mètodes de detecció, alineament i reconeixement prèviament estudiats.

## PERÍODE DE VERIFICACIÓ I PROVES

Durant aquest període s'han realitzat les proves pertinents per tal de provar el conjunt de l'aplicació en totes les situacions possibles. A més s'ha aprofitat per realitzar un estudi tenint en compte alguns factors com la quantitat de dades a la base de dades, el nombre de deteccions en una imatge etc... A més, si durant aquest període s'ha detectat algun error, o possible canvi s'ha aprofitat per refinar-lo.

### 3.3.4.2 EINES DE DESENVOLUPMANET

Com ha s'ha comentat anteriorment l'entorn de desenvolupament utilitzat ha estat Matlab, però a més s'ha fet ús d'altre programari. S'han utilitzat eines dins del mateix entorn Matlab. Una d'aquestes eines

ha estat el **GUIDE** per crear interfícies d'usuaris semblants a les típiques Windows Forms que te el Visual Studio.

Consta d'una plataforma on es col·loquen els elements que s'utilitzaran (desplegables, botons, quadres de text...). L'aplicació permet modificar les característiques dels elements (mida, color, nom).

Un cop s'ha dissenyat l'aparença de l'aplicació, falta donar-li a cada element, la funcionalitat que li pertoca. Per fer-ho el **GUIDE** crea un arxiu esquelet on se li dona aquesta funcionalitat. Tant es poden donar funcionalitats d'inici (és a dir que s'executin quan s'iniciï l'aplicació), com funcionalitats interactives (és a dir la funcionalitat que fa si es prem, per exemple, un botó).

Fora l'entorn de desenvolupament, també s'han utilitzat altres eines per tal de complementar el treball pràctica i/o ajudar a desenvolupar la documentació:

- **Microsoft Word 2007:** Per la realització de la documentació.
- **Paint .NET:** Utilitzat pel retoc d'algunes imatges, fons de pantalles i creació d'icones.
- **PdfCreator:** Per a transformar la documentació a un tipus d'arxiu PDF.
- **Acrobat Reader:** Utilitzat per poder llegir arxius de tipus PDF.
- **Microsoft Visio 2010:** Per fer alguns esquemes i figures.
- **OpenProj:** Per a la temporalització del temps del treball.

### 3.3.4.3 REQUERIMENTS I ESPECIFICACIONS DEL SISTEMA

---

Els requeriments del sistema són els que s'exposen a continuació. Per una part els requeriments funcionals, que descriuen les funcionalitats que ha d'oferir l'aplicació, i per l'altra, els requeriments no funcionals, dedicats a quina tecnologia ha estat la utilitzada, així com la manera de treballar amb l'aplicació.

#### REQUERIMENTS FUNCIONALS

Qualsevol usuari disposa de les següents funcionalitats:

1. **Afegir contingut a la base de dades.**  
En l'aplicació un usuari podrà entrar un nou element a la base de dades d'entrenament que utilitzen els mètodes Eigenfaces i Fisherfaces.
2. **Permet treballar amb els diferents mètodes i bases de dades.**  
Sempre i quan l'usuari disposi d'una base de dades d'imatges aquest podrà realitzar un reconeixement amb les dades de la base de dades en qüestió.
3. **Permet veure els resultats obtinguts d'una cerca.**  
El programa mostra els resultats de la cerca: imatge més propera, distància entre aquesta i la imatge de test i la etiqueta de qui apareix a la imatge de test en qüestió.
4. **Defineix el model d'interès.**  
Quan s'insereix dades a la base de dades d'entrenament es crea un model que serà amb el que es treballarà posteriorment en els mètodes Eigenfaces i Fisherfaces.



### 5. Disposa d'una interfície de fàcil ús.

Es disposa d'una interfície on l'usuari pot realitzar totes les accions, aquesta interfície és altament intuïtiva.

## REQUERIMENTS NO FUNCIONALS

### 1. El sistema disposa d'una sèrie de bases de dades.

En l'aplicació hi ha una funció que permet a l'usuari afegir elements a una d'aquestes bases de dades.

### 2. Entrada de dades.

- Els valors dels paràmetres no podran superar els límits establerts.
- Només es treballarà amb imatge o vídeo, i serà l'usuari que triï quina operació vol realitzar. Qualsevol de les accions triades requerirà la presència d'una o varies bases de dades específiques.
- Els formats de les imatges seran els preestablerts per l'explorador. Mentre que el vídeo serà el detectat per la webcam pertinentment connectada i configurada en la màquina.

### 3.3.4.4 DISSENY

Un cop detallada la proposta de l'aplicació, es procedeix a fer el disseny del model conceptual de dades. En l'entorn de programació s'han utilitzat classes però són estàtiques, per tant, no es pot desenvolupar el diagrama de classes, per aquesta raó s'ofereixen els següents esquemes per veure el funcionament de l'aplicació.

## DIAGRAMA DE CONTEXT

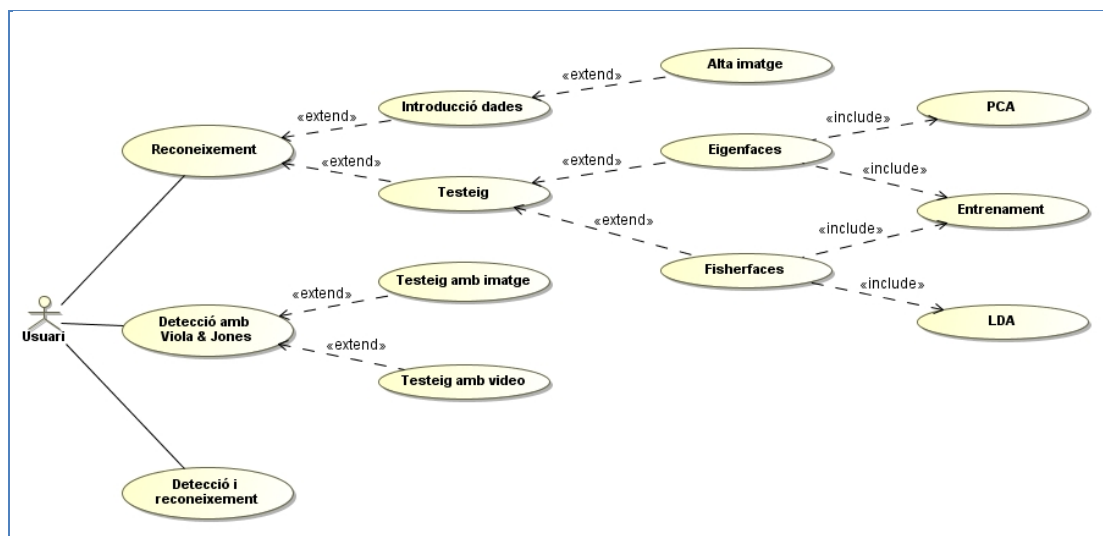


Fig 3.23. Diagrama de context general de l'aplicació.

## FITXES DE CAS D'ÚS

L'usuari pot iniciar l'aplicació a través de l'entorn Matlab. Un cop aquest hagi iniciat l'aplicació, se li presenten 3 grups amb diverses opcions:

- **Detecció de cares.**
- **Reconeixement facial (Fisherfaces i Eigenfaces).**
- **Detecció i reconeixement de cares (integració dels mètodes anteriors).**

Qualsevol tipus d'usuari pot realitzar les operacions de qualsevol dels anteriors grups. Com a precondició cal tenir en compte que qualsevol usuari que vulgui utilitzar aquesta aplicació necessita tenir instal·lat en el seu equip l'entorn Matlab, les llibreries OpenCV (versió 2.4.3), l'entorn Visual Studio 2008 express edition i una webcam. En cas contrari, no es podran realitzar cap d'aquestes 3 accions.

En el primer grup de detecció de cares l'usuari pot escollir entre les següents opcions:

- **Detectar cares en imatges.**
- **Detectar cares en seqüències de vídeo.**

En el segon grup de reconeixement facial hi ha diverses accions disponibles:

- **Afegir a la base de dades.**  
Permet afegir noves imatges de cares a la base de dades.
- **Executar reconeixement facial.**  
Es realitza l'entrenament de les dades de la base de dades i posteriorment es reconeix la cara introduïda per l'usuari. El reconeixement pot ser utilitzant Eigenfaces o bé Fisherfaces.

En el tercer grup, detecció i reconeixement de cares, només es disposa de dues opcions. Es tracta de la integració sobre seqüència de vídeo o imatges estàtica dels mètodes de detecció i reconeixement anteriors. El sistema executarà una sèrie d'operacions per tal de dur a terme les accions de detecció i reconeixement:

- **Detecció.**  
S'utilitza el mètode estudiat per a la detecció de les cares (Viola & Jones) en cadascun dels frames de la seqüència de vídeo o la imatge estàtica.
- **Alineament.**  
Es normalitza la imatge de la cara detectada per tal d'afavorir el reconeixement.
- **Reconeixement.**  
Quan l'usuari ho indica, es reconeixen les cares detectades pel mòdul de detecció. Hi ha dos mètodes per dur a terme el reconeixement: Eigenfaces i Fisherfaces.
- **Decisió.**  
Si es reconeix la persona es mostra el nom d'aquesta sobre la bounding box que enquadra la cara detectada.

**DIAGRAMES D'ACTIVITAT**

Mitjançant els diagrames d'activitat es mostren les diferents accions que pot dur a terme l'aplicació desenvolupada en aquest projecte.

- **Reconeixement de cares:**

Aquest diagrama és vàlid pels procediments de reconeixement Fisherfaces i Eigenfaces. Descriu pas a pas els passos a seguir per tal de realitzar el reconeixement de cares.

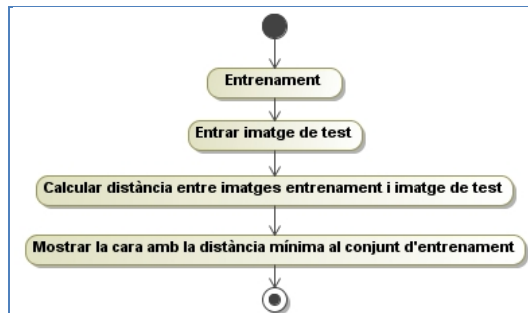


Fig 3.24. Diagrama d'activitat "del reconeixement de cares".

- **Afegir una cara a la base de dades:**

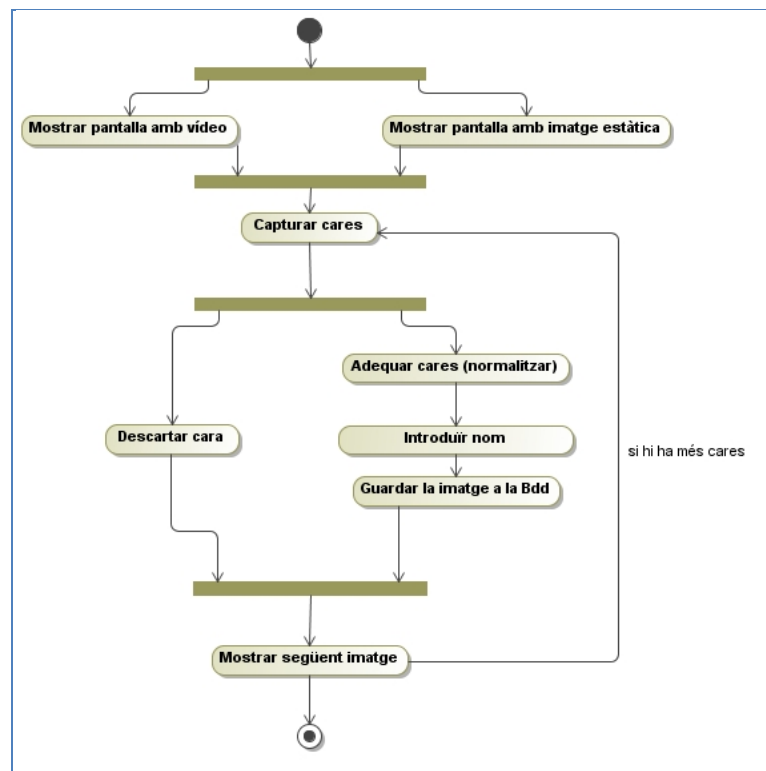


Fig 3.25. Diagrama d'activitat "afegir cara a base de dades".

El diagrama anterior mostra els passos que segueix l'aplicació per tal de guardar correctament la cara d'una persona a la base de dades.

- **Entrenament Eigenfaces:**

L'entrenament dels eigenfaces és una fase que cal explicar per separat, però el programa no té una opció específica per això, s'executa conjuntament amb el programa Eigenfaces.

En l'esquema següent es mostren els passos que es segueixen per tal de realitzar un entrenament amb les dades que hi ha a la base de dades.

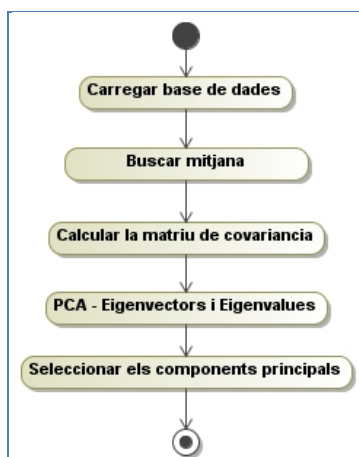


Fig 3.26. Diagrama d'activitat "entrenament eigenfaces".

- **Entrenament Fisherfaces:**

L'entrenament dels fisherfaces és una fase que cal explicar per separat, però el programa no té una opció específica per això, s'executa conjuntament amb el programa Fisherfaces.

En l'esquema següent es mostren els passos que es segueixen per tal de realitzar un entrenament amb les dades que hi ha a la base de dades.

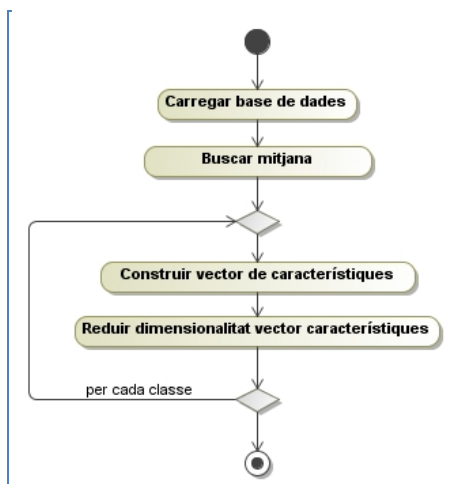
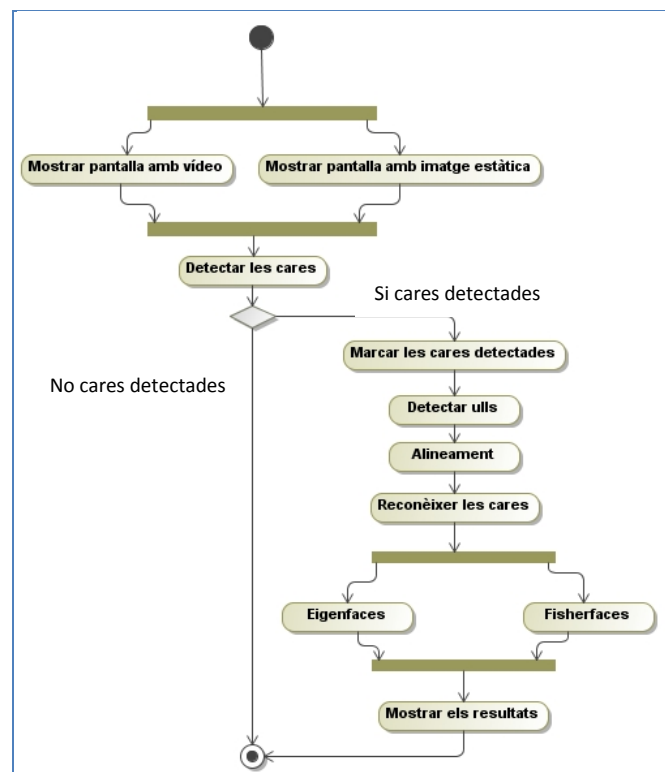


Fig 3.27. Diagrama d'activitat "entrenament fisherfaces".

Pel procediment de detecció de cares no hi ha diagrames ja que s'aprofita la implementació que ja ve programada en les llibreries OpenCV.

- **Aplicació del sistema de reconeixement complet:**



**Fig 3.28.** Diagrama d'activitat "sistema de reconeixement".

Integrant els mètodes (detecció i reconeixement) es realitza una aplicació que és capaç de detectar cares dins d'una seqüència de vídeo o imatge estàtica i reconèixer les cares detectades. En el diagrama anterior es poden observar els passos d'aquest procés.

### 3.3.5 EXPERIMENTS I RESULTATS

En aquesta secció es mostren els experiments realitzats amb els mètodes de reconeixement, els resultats obtinguts i un anàlisi d'aquests mateixos.

#### 3.3.5.1 BASES DE DADES DE CARES

Existeixen més d'una vintena de bases de dades amb imatges de cares per provar qualsevol algorisme. Com ja s'ha comentat a la introducció s'utilitzen les bases de dades Yale, AT&T i les del Dr. Libor Spacek. El lloc web <http://www.face-rec.org> recomana les més utilitzades. L'accés a cada una d'aquestes bases de dades és més o menys públic i es poden trobar referències en multitud d'articles i altres treballs a l'ús de bases de dades d'imatges de cares com la Yale o la de AR (entre les més fàcils d'aconseguir, les més grans).

Els resultats que es mostren en aquest capítol s'han obtingut fent ús de la base de dades de AT&T. S'ha escollit aquesta base de dades per tenir un nombre de cares suficient i per tenir les cares centrades a la foto. En altres bases de dades això no és així, per la qual cosa es requereix un preprocessament de la imatge per seleccionar la part on existeix la cara.

A la base de dades AT&T existeixen 400 imatges fetes en un laboratori entre 1992 i 1994. Són imatges de 40 individus diferents, als quals se'ls hi van fer 10 fotos en cada sessió amb variacions de llum, expressió facial i punt de vista i (en alguns casos) presència o no d'ulleres. A la figura següent es poden veure les fotografies fetes a 3 d'aquests 40 personatges:



**Fig 3.29.** Extracte de la base de dades d'imatges de cares AT & T.

Pels experiments sobre els mètodes de reconeixement s'han considerat aquest tipus d'imatges, ja suficientment enquadrades. La dimensió de les fotografies és de 92 d'ample per 112 d'alçada. S'ha utilitzat un protocol de "fold cross validation", utilitzant el 90% de les imatges per l'entrenament, el 10% de les imatges per testejar i s'ha anat rotant, és a dir, suposant que s'utilitzen 400 imatges, de la 1 a la 40 per test i de la 41 a la 400 per entrenament, després es repeteix el procés però agafant de la 41 a la 80 per testejar, i de la 1 a la 40 i 81 a 400 per entrenament, i així successivament repetint el procés 10 cops. Finalment s'agafa la mitja dels 10 resultats.

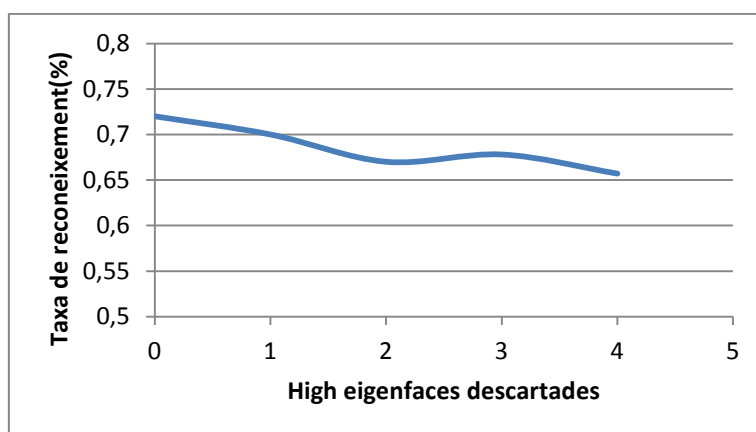
### 3.3.5.2 EXPERIMENTS AMB ELS MÈTODES DE RECONeixEMENT

En aquesta secció es mostra el percentatge d'encert de cada un dels mètodes de reconeixement. Es presenten per separat i es van comentat els gràfics. Del mètode Eigenfaces es vol monitoritzar la taxa d'encert en funció de les Eigenfaces descartades (high i low) i del mètode Fisherfaces es vol monitoritzar la taxa d'encert en funció del nombre d'Eigenvectors.

#### EIGENFACES

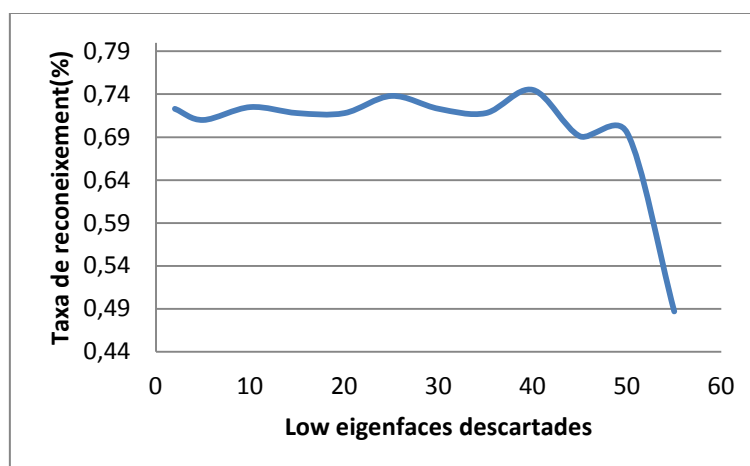
En aquesta prova es representa la taxa d'encert en funció del nombre d'eigenfaces descartades. En cada prova s'escull el nombre d'eigenfaces descartades i llavors es duen a terme els testos utilitzant el protocol enunciat anteriorment. La base de dades utilitzada és la de AT&T.

En el següent gràfic es pot veure com la taxa de reconeixement no presenta millora tenint en compte les high eigenfaces descartades. Si es descarten més eigenfaces, aquesta millora desapareix fins a convertir-se en una taxa de reconeixement més petita. Això es produeix perquè les eigenfaces amb eigenvalues més grans són les que tenen major pes a l'hora de desxifrar quina identitat és la correcta.



**Fig 3.30.** Taxa de reconeixement descartant fins a quatre high eigenfaces.

La següent prova s'ha basat en variar les low eigenfaces descartades. Segons la teoria del PCA no és necessari utilitzar totes les eigenfaces per tal d'aconseguir la taxa de reconeixement màxima per a una mateixa base de dades:



**Fig 3.31.** Taxa de reconeixement descartant les low eigenfaces.

Els resultats obtinguts demostren que no és cert utilitzar totes les eigenfaces disponibles per obtenir la taxa de reconeixement més òptima. Això és degut a que les eigenfaces amb eigenvalues més petits tenen un pes molt baix a l'hora de realitzar el reconeixement, en alguns casos però, es pot introduir un cert error enlloc d'afavorir el reconeixement en sí. Per ser més exactes, descartant al voltant de 40 eigenfaces s'obtenen els valors més alts de reconeixement.

En aquesta última prova es comprova la baixa taxa de reconeixement d'eigenfaces en entorns on hi ha variacions en la il·luminació. Per dur a terme aquesta prova s'ha utilitzat una base de dades on les imatges presenten diferents il·luminacions, la base de dades de Yale. La dimensió de les fotografies també és de 92 d'ample per 112 d'alçada. El protocol de test utilitzat continua essent fold cross validation utilitzant 400 imatges.

La taxa obtinguda no ve determinada per culpa d'una imatges d'entrenament que es sobreconeix, sinó que és degut al algorisme en sí mateix. Això ofereix un pobre rendiment al algorisme PCA sobre canvis d'il·luminació.

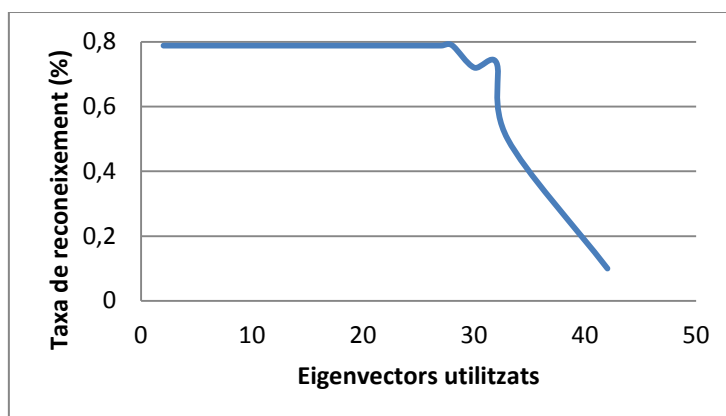


Fig 3.32. Taxa de reconeixement en funció del nombre d'eigenvectors utilitzats.

### FISHERFACES

Amb els següents experiments es mostra el potencial de Fisherfaces. Les proves es duen a terme prenent com a paràmetre variable els eigenvectors. La base de dades utilitzada és la de AT&T:

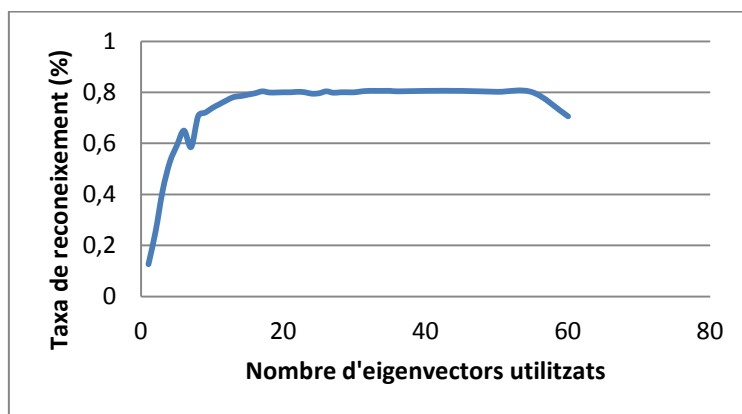


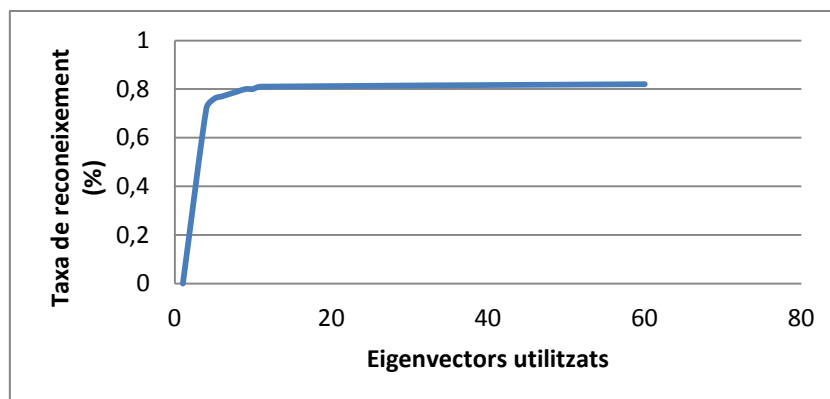
Fig 3.33. Taxa de reconeixement en funció dels eigenvectors utilitzats.

La taxa de detecció és molt similar a la que presenta el mètode Eigenfaces i cal notar que el comportament dels dos algorismes és molt semblant, no és necessari utilitzar la totalitat dels eigenvectors per obtenir resultats òptims, també és necessari un mínim d'eigenvectors per obtenir valors de reconeixement alts. Això és degut a que LDA utilitza PCA per reduir la dimensionalitat.

En la següent prova es comprova el millor comportament de Fisherfaces davant d'entorns on la il·luminació va variant. El gràfic següent mostra com la taxa de reconeixement és molt elevada utilitzant un nombre petit d'eigenvectors, a diferència del PCA on la taxa de reconeixement és molt baixa. El protocol de test és el mateix que s'ha utilitzat per testear el mètode Eigenfaces sota condicions d'il·luminació variants.



En aquesta prova també s'utilitza la base de dades de Yale.



**Fig 3.34.** Taxa de reconeixement en funció dels eigenvectors utilitzats (il·luminació variant).

### 3.3.5.3 RESULTATS VISUALS DE L'APLICACIÓ

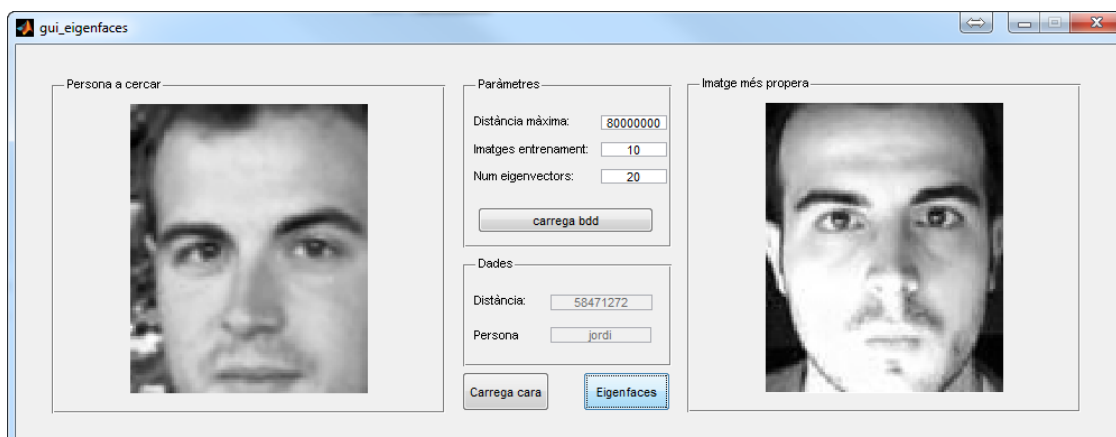
En aquest apartat es mostren els resultats de l'aplicació i cadascun dels seus subapartats. Cal notar que el temps d'execució varia segons la màquina utilitzada. En el moment d'executar les aplicacions desenvolupades s'ha disposat d'una màquina amb les següents prestacions:

- AMD Phenom X4 3200Mhz.
- 6Gb de memòria Ram DDR3.
- ATI Radeon HD 4670 512MB GDDR3.
- WebCam logitech.

No es comenta res sobre la implementació de Viola & Jones ja que aquest mètode no ha estat desenvolupat des de zero, com s'ha dit, s'ha aprofitat la implementació d'OpenCV.

### EIGENFACES I FISHERFACES

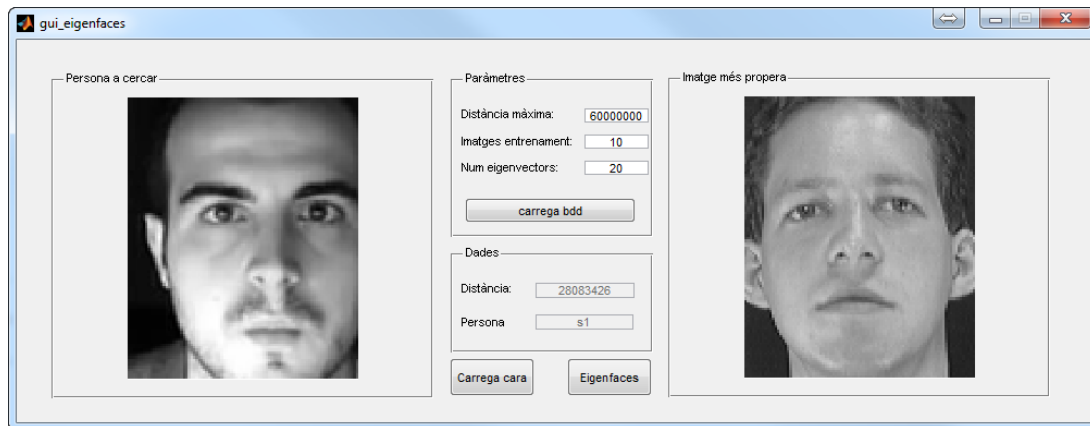
Les proves més "visuals" realitzades dels mètodes Eigenfaces i Fisherfaces, han estat proves fetes tant amb cares sense complements, com amb cares amb ulleres de sol, etc...



**Fig 3.35.** Eigenfaces, prova de reconeixement correcte.

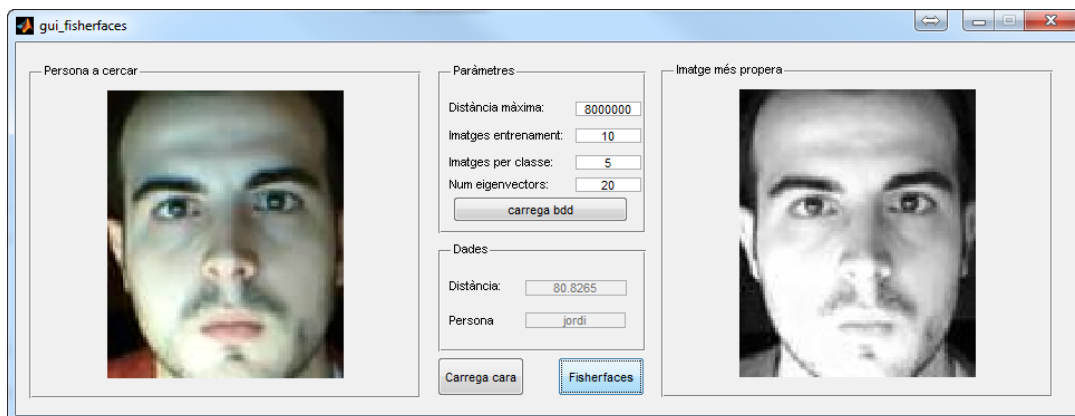
Altres proves que s'han realitzat han estat descentrant la imatge de la cara o bé amb diferents nivells de lluminositat:

Imatge descentrada:

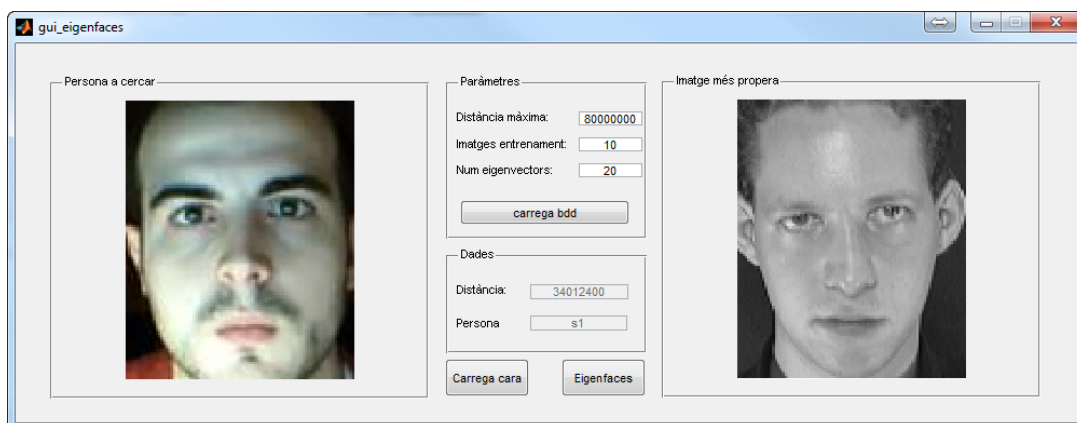


**Fig 3.36.** Eigenfaces, prova de reconeixement amb cara descentrada.

Amb il·luminació variant:



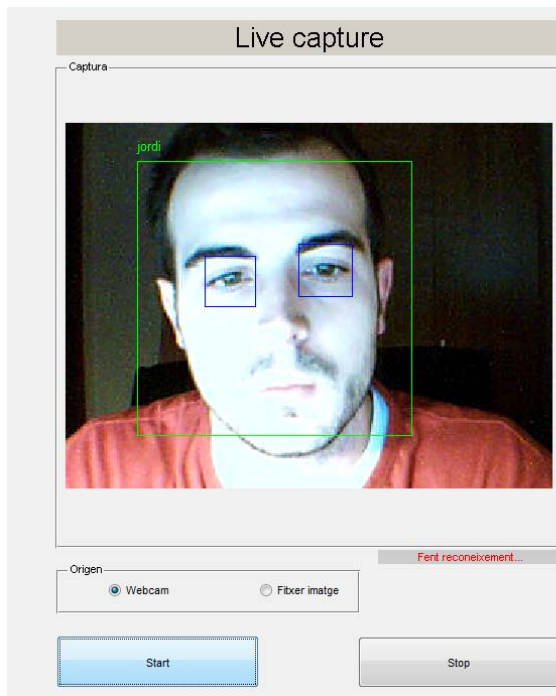
**Fig 3.37.** Fisherfaces, prova de reconeixement correcte amb canvis il·luminació.



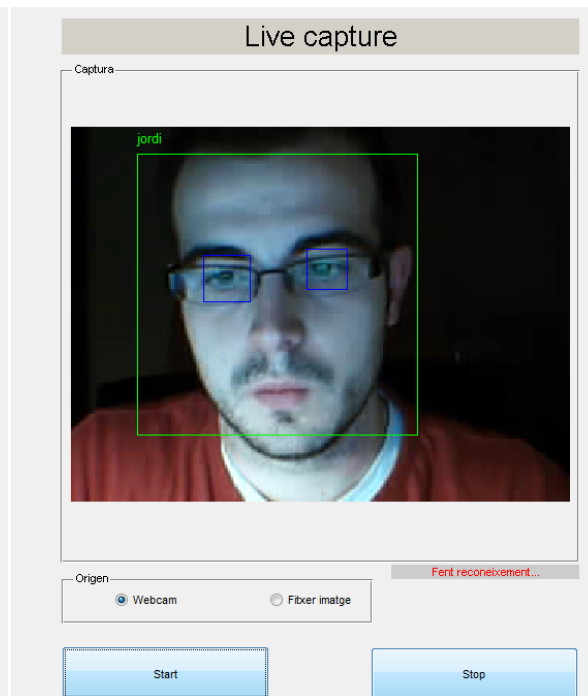
**Fig 3.38.** Eigenfaces, prova de reconeixement incorrecte amb canvis il·luminació.

### INTEGRACIÓ DELS COMPONENTS QUE FORMEN EL SISTEMA DE RECONeixEMENT

A continuació es mostren una sèrie d'exemples de resultats obtinguts durant la prova del sistema complet de reconeixement facial.



**Fig 3.39.** Eigenfaces prova a temps real, imatge capturada des de webcam.



**Fig 3.40.** Fisherfaces prova a temps real, imatge capturada des de webcam.

A partir dels resultats experimentals obtinguts en l'execució de les aplicacions anteriors s'arriba a la conclusió que si es vol treballar en temps real, com més informació hi ha a la base de dades més lent és el procés de reconeixement. A més, tenir una base de dades molt gran no implica augmentar el nivell d'encert. La informació que hi ha a les bases de dades ha de ser de qualitat.

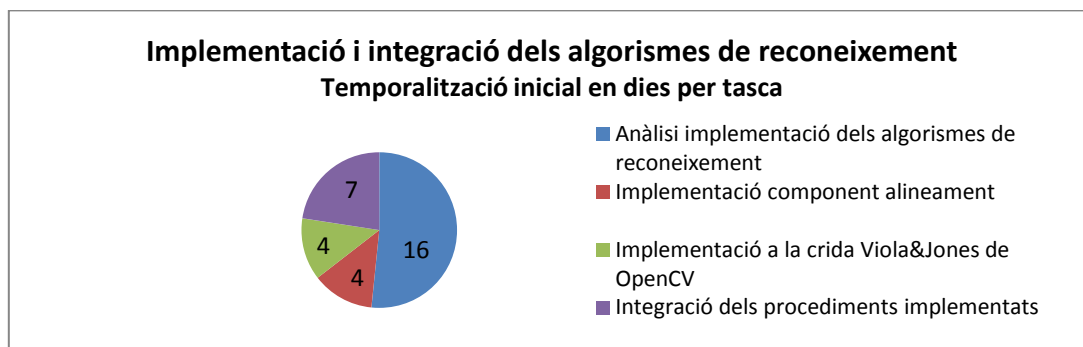
## 4. CONCLUSIONS

### 4.1 VALORACIÓ FINAL

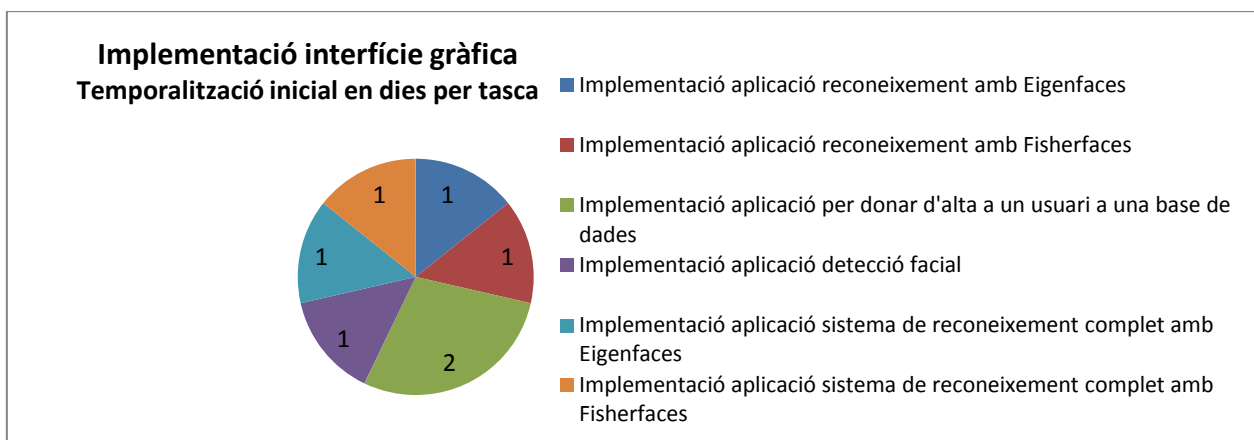
L'objectiu principal ha estat l'estudi i implementació de mètodes que s'apliquen per resoldre la problemàtica del reconeixement facial i la demostració del seu funcionament mitjançant una aplicació completa de reconeixement facial.

Per resoldre un problema de reconeixement cal que intervinguin mètodes de detecció, alineament i reconeixement. Aquest projecte s'ha centrat més en la part de reconeixement, per aquest motiu, s'ha aprofundit més en tres dels algorismes més utilitzats Eigenfaces, Fisherfaces i ASM. Pel que fa a la implementació de l'aplicació prototipus, aquesta ha estat duta a terme principalment sota el llenguatge Matlab.

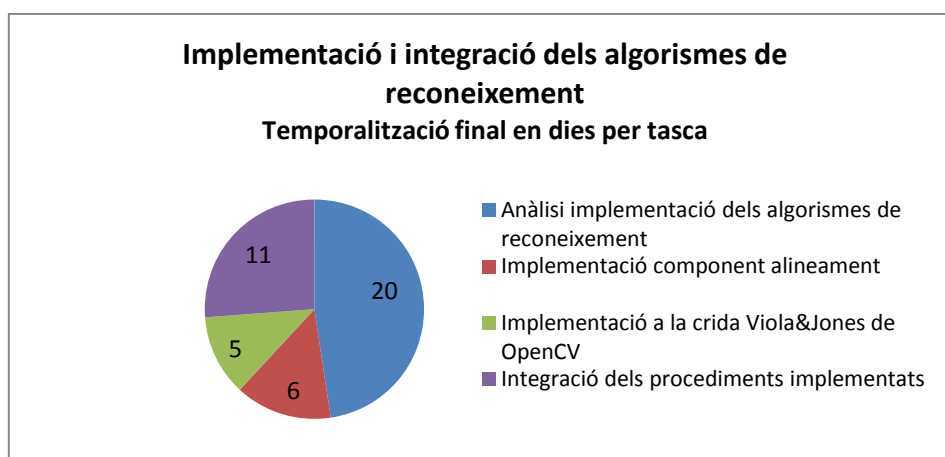
No es pot considerar que s'hagin assolit tots els objectius ja que hi ha hagut desviacions temporals durant l'execució de certes tasques, fet que ha provocat que no hagi estat possible la implementació del mètode ASM. En el següents gràfics es mostren temporalitzacions previstes i finals en les activitats d'implementació i integració dels algorismes de reconeixement i implementació de la interfície gràfica:



**Fig 4.1.** Gràfic de temporalització prevista per l'activitat implementació i integració dels algorismes de reconeixement.

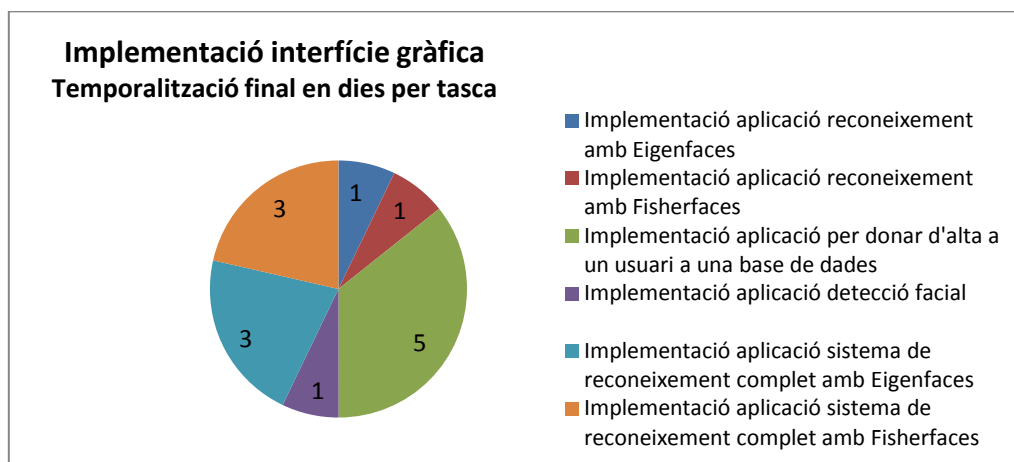


**Fig 4.2.** Gràfic de temporalització prevista per l'activitat implementació d'una interfície gràfica.



**Fig 4.3.** Gràfic de temporalització final per l'activitat implementació i integració dels algorismes de reconeixement.

On hi ha hagut les variacions més significatives ha estat en les tasques d'implementació dels algorismes de reconeixement i en la integració de tots els components (component de detecció, alineament i reconeixement). Les tasques de desenvolupament les mètodes d'Eigenfaces i Fisherfaces s'han allargat més del previst degut a la dificultat per comprendre bé la base matemàtica dels mètodes de reconeixement Eigenfaces, Fisherfaces i ASM; s'ha hagut de dedicar més temps del previst al repàs de geometria.



**Fig 4.4.** Gràfic de temporalització final per l'activitat implementació d'una interfície gràfica.

En aquesta activitat, els problemes han sorgit durant la implementació de l'aplicació visual per afegir cares a la base de dades d'entrenament. Les dificultats han sorgit a l'hora d'implementar els procediments de retoc manual de cada cara trobada, es podria haver optat per fer-ho automàtic però és molt important que les imatges s'insereixin correctament centrades, escalades, etc.. a la base de dades, aplicant un retoc manual dóna més bon resultats que aplicant un retoc automàtic. La resta de desviacions venen derivades en els retards en les implementacions dels algorismes de reconeixement.

El període de proves també s'ha allargat més del previst, inicialment es van estimar 3 dies, però s'han necessitat 9 dies. La causa principal ha estat la gran quantitat de temps de computació necessari per dur a terme els entrenaments, inicialment es partia de 100 imatges d'entrenament per individu, però al observar la lentitud de la tasca, es va optar per utilitzar 10 imatges per individu.

Pel que fa als resultats, tot i que no han estat dolents no es pot parlar que l'objecte d'estudi i implementació d'aquest projecte procediments reconeixement fiables, ja que els mètodes utilitzats estan molt lligats a factors com la il·luminació, posició, orientació, etc...

Pel correcte funcionament del sistema de reconeixement és vital que l'objecte de reconeixement, en aquest cas una cara, estigui ben il·luminada, ben centrada i ben escalada i això en entorns del món real a vegades resulta complicat aconseguir-ho.

Resumint, es pot dir que els objectius no s'han assolit del tot. S'ha vist el comportament i el funcionament d'un sistema de reconeixement que utilitza els mètodes Eigenfaces i Fisherfaces però no ha estat possible pel mètode basat en models Active Shape Model.

#### 4.1.1 VALORACIÓ ECONÒMICA

El cost estimat del projecte es divideix en recursos hardware, software i hores de treball humanes.

- Hardware:
  - PC desenvolupament i proves 500€.

No es pot imputar tot el cost total del hardware al projecte ja que aquest posteriorment tindrà altres usos. Així doncs, el cost ve donat per les hores d'utilització:

Període	Hores
Recerca documentació	24
Definició del pla de treball	3
Configuració entorn	3
Repàs llenguatge programació	24
Estudi previ	42
Implementacions	105
Proves	9
Extracció conclusions i resultats	6
Redactat memòria	36
<b>TOTAL</b>	<b>252</b>

Assumint que la vida útil d'un PC són 6 anys, si s'usa 21 dies al mes durant 3 hores al dia, el cost per hora és:

$$cost = \frac{500€}{6 \text{ anys} \times 12 \text{ mesos} \times 21 \text{ dies} \times 3 \text{ hores}} = 0.110€$$

Per tant, el cost del hardware imputable és:

	Hores	Cost
Pc desenvolupament i proves	252	27.72€

- Software:
 

Ja sigui perquè s'ha utilitzat software lliure (OpenCV) o bé gràcies a l'aliança entre la UOC i les empreses propietàries del software privatiu (Microsoft i MathWorks), el cost final del software utilitzat és exactament 0€.

- Recursos humans:**  
 L'estimació d'hores dedicades al projecte ve definida al capítol introductor, en 84 dies. De mitjana s'hi ha dedicat 5 hores diàries. Partint d'això el nombre total d'hores seria de 420 hores.  
 S'ha decidit que el cost d'aquestes hores de treball és 0€ ja que sóc estudiant i la realització del projecte és part de la meva formació.
- Cost total:**  
 Així doncs, el cost total estimat és de 27.22€.

Com que hi ha hagut desviacions temporals això ha implicat més hores de recursos humans i més hores d'ús de maquinari. Com ja s'ha dit les hores de recursos humans s'estableixen en 0€ i per tant aquesta part no afecta al cost del projecte. Les hores de maquinari s'han vist incrementades en la fase d'implementacions (algorismes i interfície gràfica), al còmput inicial cal afegir 54 hores aproximadament, de tal forma que el cost aproximat del projecte ascendeix a 33.66€:

	Hores	Cost
<b>Pc desenvolupament i proves</b>	252+54	33.66€

## 4.2 CONCLUSIONS GENERALS

Els mètodes de reconeixement i detecció utilitzats poden solucionar problemes de reconeixement facial sobre imatges frontals, encara que presentin lleugeres variacions en l'expressió de la cara, amb una taxa de reconeixement força alta.

Una de les limitacions amb les que s'ha de jugar amb aquests mètodes de reconeixement és en l'adquisició de la imatge a reconèixer. Molts cops les captures no presenten les condicions idònies per afavorir el reconeixement. Es poden tenir situacions, com ja s'ha explicat, de canvis d'il·luminació no previstos o imatges parcials en les que no està present tota la cara.

Un sistema de reconeixement facial ha de poder resoldre tots aquests problemes que es poden arribar a presentar en la vida real. En la següent taules es poden veure resultats aconseguits en cada una de les situacions esmentades:

	Eigenfaces	Fisherfaces
<b>Canvis expressió</b>	82%	80%
<b>Variacions il·luminació</b>	77%	82%
<b>Rotació imatges</b>	19%	14%

Es molt important doncs disposar també d'un bon sistema de captura i normalització de les imatges d'entrada dels mètodes de reconeixement. Tot això, sense perdre de vista el rendiment del sistema, ja que en la majoria de casos el requeriment més important és que un sistema de reconeixement facial funcioni en temps real.

## 4.3 TREBALL FUTUR

Respecte a possibles millores, cal dir que com a tot projecte es poden fer futures millores. Algunes de les millores que es plantegen són les següents:

- Millorar la base de dades. Seria recomanable comptar amb una base de dades que representi millors característiques, sobretot en el que fa referència al conjunt d'entrenament. Aquestes imatges han de ser preses en un entorn molt controlat, és a dir que totes segueixin un mateix patró. Això ajudaria a l'hora d'obtenir un increment de la taxa de reconeixement.
- Realitzar la implementació del mètode ASM, ja que ha quedat pendent de desenvolupament en aquest projecte.
- Resultaria molt interessant provar altres mètodes de reconeixement on no s'hagi de comptar amb tant de tractament previ de les imatges a reconèixer, i també poder disposar de més informació relacionada amb les imatges; com per exemple algun sistema que pugui utilitzar informació 3D (a més de les dues dimensions de les imatges poder tenir profunditat).
- Pel que fa a la implementació, un bon treballar futur és reimplementar l'aplicació sencera utilitzant el llenguatge de programació C++ i fer ús de les llibreries OpenCV. Això podria millorar molt els temps d'execució de l'aplicació.
- Un altre aspecte relacionat amb la implementació, és dur-ne a terme una sobre dispositius mòbils, per tasques de seguretat (desbloquejar el mòbil mitjançant el reconeixement de l'usuari).



## REFERÈNCIES

- [1] Paul Viola and Michael J. Jones. *Rapid object detection using a boosted cascade of simple features*. In *Computer Vision and Pattern Recognition*, pages 511-518, 2001.
- [2] Schneiderman & Kanade. *Object detection using the statistics of parts*. *International Journal of Computer Vision*, pages 151-177, 2004.
- [3] Rowley, Baluja & Kanade. *Neural network-base face detection*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 203-208, 1996.
- [4] Turk, M., Pentland, A., *Face recognition using eigenfaces*. *Journal of Cognitive Neuroscience*, pages 71-86, 2001.
- [5] L. Sirovich M. Kirby. *Low-dimensional procedure for the characterization of faces*. *Journal of the Optical Society of America A*, pages 519-524, 1991.
- [6] P.N Belhumer, J.P Hespanha, D.J Kriegman. *Eigenfaces vs Fisherfaces: Recognition Using Class Specific Linear Projection*. *IEEE transaction on Pattern Analysis and Machine Intelligence*, pages 711-720, 1997.
- [7] T. Cootes and C. Taylor. *Modelling object appearance using the grey-level surface*. In E. Hancock, editor, *British Machine Vision Conference*, pages 479-488, York, England, September 1994. BMVA Press.
- [8] T. Cootes, C. Taylor, D. Cooper and J. Graham. *Active Shape Models, their training and application*. Department of Medical Biophysics, University of Manchester. Engalnt, April 1994.

## ANNEXOS

## 1. CONFIGURACIÓ DE L'ENTORN DE TREBALL

A continuació s'expliquen tots els passos a seguir per preparar l'entorn de desenvolupament del projecte en una sola màquina. Primer s'explica com s'ha dut a terme la instal·lació del software involucrat i després els passos necessaris per configurar l'entorn.

### 1.1 INSTAL·LACIÓ DE OPENCV

OpenCV conté les llibreries que usen les utilitats que s'encarreguen de l'entrenament del classificador i implementa la funció que realitza l'algorisme Viola & Jones. S'ha utilitzat la versió 2.4.3. El software es pot aconseguir en la següent URL: <http://downloads.sourceforge.net/project/opencvlibrary/opencv-win/2.4.3/OpenCV-2.4.3.exe>.

Un cop descarregat l'executable, només cal seguir els passos de la instal·lació acceptant la llicència i deixant la resta de paràmetres per defecte.

La instal·lació afegirà la ruta dels binaris al PATH del sistema.

### 1.2 INSTAL·LACIÓ DE MATLAB

S'ha instal·lat la versió R2010a. És necessari disposar del codi d'activació i del fitxer de la llicència (license.dat).

1. Iniciar l'executable de la instal·lació.
2. Seleccionar treballar sense Internet.
3. Llegir i acceptar la llicència.
4. Introduir el codi d'activació.
5. Seleccionar la configuració típica, que instal·larà les Toolboxes necessàries:
  - a. Matab.
  - b. Matlab Compiler.
  - c. Image Acquisition toolbox.
  - d. Image Processing toolbox.
6. Especificar la ruta del directori on s'instal·larà Matlab.
7. Especificar la ruta del license.dat.
8. Confirmar els detalls de la instal·lació i esperar que acabi el procés.

### 1.3 INSTAL·LACIÓ DE MICROSOFT VISUAL STUDIO 2008 EXPRESS EDITION SP1

Aquest software servirà com a compilador de MEX per a Matlab.

1. Iniciar l'executable de la instal·lació.
2. Seleccionar l'opció instal·lar Microsoft Visual Studio 2008.
3. Acceptar la llicència d'acord i introduir el codi d'activació.
4. Seleccionar el tipus d'instal·lació per defecte i introduir la ruta d'instal·lació.
5. Fi de la instal·lació.

#### 1.4 CONFIGURAR VISUAL STUDIO PER TREBALLAR AMB OPENCV

Per poder treballar amb la biblioteca OpenCV des del Visual Studio abans cal editar la configuració global del Visual Studio per afegir les rutes als fitxers d'OpenCV. Es configura des de Visual Studio -> Tools -> Options -> VC++ Directories.

1. Seleccionant "Include files" per afegir les rutes següents:
  - a. C:\opencv\2.4.3\include
  - b. C:\opencv\2.4.3\build\include
2. Seleccionant "Library file" per afegir les rutes següents:
  - a. C:\opencv\2.4.3\build\x86\vc9\lib

#### 1.5 CONFIGURAR LA VARIABLE D'ENTORN DEL SISTEMA "PATH"

Afegir al path del sistema:

1. C:\opencv\2.4.3\build\x86\vc9\bin

#### 1.6 CONFIGURACIÓ DEL COMPILADOR DE MEX

En aquest projecte s'utilitza un mex-file per realitzar les crides des de Matlab a la classe ViolaJones.cpp implementada en C++. Per poder generar el mex-file cal definir el compilador de MEX.

1. Iniciar Matlab.
2. Introduir la següent instrucció a la finestra de comandes de Matlab:
  - a. `mex -setup`

Matlab preguntarà si vol que busqui els compiladors mex existents al sistema. Cal indicar que si i apareixerà el compilador "Microsoft Visual C++ 2008 SP1" entre d'altres.

3. Seleccionar el compilador de C++ de Visual Studio 2008 i confirmar la operació.
4. Modificar el fitxer %APPDATA%\MathWorks\MATLAB\R2010b\mexopts.bat afegint:
  - a. `set OPENCVDIR=C:\OpenCV2.4.3`
  - b. Afegir en el path %OPENCVDIR%\build\x86\vc9\bin
  - c. Afegir a la variable include:  
`%OPENCVDIR%\build\include;%OPENCVDIR%\build\include\opencv;%OPENCVDIR%\build\include\opencv2;`
  - d. Inicialitzar la variable linkflags amb els valors:  
`/dll /export:%ENTRYPOINT% /LIBPATH:"%LIBLOC%" opencv_calib3d243d.lib  
opencv_contrib243d.lib opencv_core243d.lib opencv_features2d243d.lib  
opencv_flann243d.lib opencv_gpu243d.lib opencv_highgui243d.lib  
opencv_imgproc243d.lib opencv_legacy243d.lib opencv_ml243d.lib  
opencv_objdetect243d.lib opencv_video243d.lib libmx.lib libmex.lib libmat.lib  
/MACHINE:X86 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib  
shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /manifest  
/incremental:NO /implib:"%LIB_NAME%.x"  
/MAP:"%OUTDIR%%MEX_NAME%%MEX_EXT%.map"`

## 1.7 COMPILAR FACEDETECT.CPP I EYEDTECT.CPP

Per compilar les classes FaceDetect.cpp i EyeDetect.cpp (funcions per dur a terme les detecció de cares i ulls respectivament) cal executar les instruccions:

```
mex FaceDetect.cpp
```

Aquesta instrucció genera els fitxer FaceDetect.mexw32 necessari per realitzar les crides des dels scripts de matlab.

```
mex EyeDetect.cpp
```

Aquesta instrucció genera els fitxer EyeDetect.mexw32 necessari per realitzar les crides des dels scripts de matlab.

## 2.CLASSIFICADOR XML

Per generar classificadors XML, s'utilitzen dos aplicacions: createsamples i haartraining. Aquestes dos aplicacions venen incloses a les llibreries OpenCV i la seva utilització és molt senzilla, ja que simplement cal executar-les des de la línia de comandes amb els paràmetres adequats.

Abans d'introduir la generació del detector de cares es fa un repàs al funcionament de l'algorisme de detecció facial que incorpora OpenCV i, posteriorment, s'explica la creació d'un classificador.

### 2.1FUNCIONAMENT DE L'ALGORISME DE DETECCIÓ DE CARES

Aquest algorisme s'encarrega de detectar si en una imatge Mat de OpenCV hi ha alguna cara humana. En cas de detectar-la retorna una llista de rectangles que delimiten cada cara humana.

Primerament l'algorisme de Viola&Jones s'encarrega de suavitzar la imatge i passar-la a escala de grisos, ja que els colors no són rellevants, perquè a l'hora de fer la cerca només es tenen en compte les formes geomètriques. Per aconseguir tenir la imatge a escala de grisos s'utilitza la següent funció:

```
cvtColor(orgImg, imgGray, Imgproc.COLOR_BGR2GRAY,4)
```

Quan s'obté la imatge amb les dimensions i colors adients es procedeix a la detecció de cares a la imatge. Per això s'utilitza la següent funció:

```
mCascade.detectMultiScale(mGray, faces,1,1,2,2,new Size(faceSize, faceSize));
```

Aquesta funció troba regions rectangulars dins la imatge d'entrada que tenen una gran probabilitat de contenir objectes per als que l'objecte *mCascade* (detector utilitzat) ha estat entrenat, retornant aquestes regions com una seqüència de rectangles. Aquest mètode escaneja la imatge varies vegades a diferent escala. Aplicant algorismes d'heurística per reduir el nombre de regions analitzades.

L'objecte *mCascade* s'ha d'haver inicialitzat anteriorment passant un fitxer XML (CvHidHaarClassifierCascade) entrenat per detectar cares com a paràmetre.

## 2.2 CREACIÓ DEL DETECTOR DE CARES

En aquest punt s'explica com es crea el detector de cares. La creació d'un detector es pot dur a terme en quatre etapes. Aquestes etapes s'executen seqüencialment :

- El primer pas que s'ha de seguir per crear el classificador és prendre les imatges que seran utilitzades com a imatge d'entrenament del classificador. Es necessiten dos tipus d'imatges:
  - Imatges positives: Són imatges en les que es troba l'objecte que es vol detectar de forma clara. En aquest cas imatges amb cares.  
Segons la documentació es recomana utilitzar unes 5000 imatges positives.
  - Imatges negatives: Són imatges que no contenen l'objecte que es vol trobar.
- Un cop es té el conjunt d'imatges definit, es passa a la següent etapa que és crear la mostra. Per dur-la a terme s'utilitza l'aplicació OpenCV create samples passant un fitxer amb les rutes de totes les imatges positives.

```
createsamples.exe -info positives/info.txt -vec infovec.vec -num 2147 -w 20 -h 20
```

On num és el nombre d'imatges positives, w i h la longitud que de les imatges de sortida, -vec el vector de sortida on es guardaran totes les imatges normalitzades i info.txt és l'arxiu de text amb les rutes de totes les imatges.

Després de realitzar aquest procés es disposa d'un vector que conté les imatges de mostra, aquestes imatges estaran normalitzades, en escala de grisos i amb la mida adient.

- Amb la mostra preparada el següent pas és dur a terme l'entrenament. S'utilitza l'aplicació haartraining amb els paràmetres adients a cada cas. Aquí es veu un exemple de crida:

```
haartraining.exe -data trainout -vec infovec.vec -bg negatives.txt -nstages 14 -nsplits 2 -minhitrate 0.999 -maxfalsealarm 0.5 -npos 2147 -nneg 2889 -w 20 -h 20 -mem 700 -mode ALL
```

El significat de cada un dels paràmetres és el següent:

- data: és la ruta de sortida del detector. Quan finalitza l'aplicació s'obté un fitxer XML amb el nom escollit, que serà el que s'ha d'utilitzar amb l'algorisme de Viola&Jones per dur a terme la detecció.
- vec: és el vector que conté la mostra d'imatges positives normalitzades que s'obtenen com a resultat de l'execució de createsamples.
- bg: en aquest paràmetre si passa un fitxer de text en el que cada una de les línies conté la ruta de cada una de les imatges negatives que es volen utilitzar. En aquest te 2889 línies, en cada una de les quals hi ha la ruta de la imatge negativa.
- nstages: nombre d'etapes que seran entrenades. Teòricament a major nombre d'etapes més potent és el classificador.

- nsplits: Indica les divisions internes de nodes que utilitzarà el classificador. OpenCV recomana deixar-ho en dos.
- minhitrate: és el mínim rang d'encert que es desitja aconseguir en cada una de les etapes d'entrenament. Si aquest valor és excessiu serà difícil d'aconseguir i si el seu valor és molt baix el detector serà poc robust.
- maxfalsealarm: és el màxim rang de falses alarmes que es permet en cada etapa del detector. Com més gran sigui el valor, pitjor serà el classificador.
- npos: és el número d'imatges positives existents en el vector que conté les imatges.
- nneg: és el nombre d'imatges negatives utilitzades. En principi ha de coincidir amb el nombre de línies del fitxer amb les rutes de les imatges negatives.
- w: és l'amplada de les imatges d'entrenament, és molt important que coincideixi amb la que s'utilitza a l'aplicació createsamples.
- mem: memòria RAM en Mb que s'assigna al procés.
- mode: permet escollir les característiques Haar, si es destitja el conjunt bàsic o el conjunt estès. La documentació aconsella utilitzar el mode ALL.

Quan l'entrenament hagi finalitzat es disposarà d'un detector en format XML que es pot utilitzar en la detecció de cares amb l'algorisme Viola&Jones.

---

### 3. MANUAL DE L'APLICACIÓ

L'aplicació que es presenta és una aplicació que permet treballar amb Viola & Jones i els mètodes de reconeixement Eigenfaces i Fisherfaces per tal de detectar i reconèixer cares de forma fàcil i senzilla.

Aquesta aplicació va dirigida a qualsevol persona que vulgui treballar amb els mètodes esmentats prèviament. Aquest persona tindrà la possibilitat de poder treballar, a partir d'una senzilla interfície gràfica, per veure el funcionament dels mètodes de detecció i reconeixement.

Per tal de poder executar l'aplicació tan sols cal una màquina capaç de poder executar l'entorn Matlab 7.10 R2010a. En ser una aplicació en la que es treballa amb imatges, és recomanable que la màquina disposi de una memòria RAM àmplia.

En qüestió de programari, s'ha de dir que l'aplicació únicament funciona sobre entorn Windows ja s'utilitzen fitxers Mex compilats amb el compilador Visual Studio C++ 2008 express utilitzant les llibreries OpenCV 2.4.3 per windows.

---

#### 3.1 INSTRUCCIONS

S'explica com instal·lar l'aplicació i com executar-la per proves els mètodes implementats.

### 3.1.1 INSTAL·LACIÓ

Abans de poder treballar amb l'aplicació, cal instal·lar-la. Per fer-ho tan sols cal copiar la carpeta mòdul a la màquina local a on es vulgui treballar.

Un cop copiada aquesta carpeta dins del disc dur, ja es pot procedir a executar l'aplicació.

### 3.1.2 EXECUTAR L'APLICACIÓ

S'inicia el Matlab i s'introdueix la direcció del directori des del que es vol treballar, a on s'ha copiat l'aplicació.

Abans de començar a treballar cal configurar un seguit de variables, per aquest motiu cal anar al menú de l'aplicació principal (executables startup.m), opció paràmetres i establir les rutes dels directoris i fitxers tenint en compte a on s'ha copiat l'aplicació.

The screenshot shows a window titled "Definició de paràmetres" (Parameter Definition) with the following sections:

- Definició de directoris:**
  - Directorí arrel: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi
  - Directorí fitxers "Mat": C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\mat
  - Directorí base de dades entrenament: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\data\train\Bdd1
  - Directorí base de dades test: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\data\test\Bdd1
- Definició de fitxers MAT:**
  - Mat file mètode PCA: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\mat\pca\trainDatabase.mat
  - Mat file mètode LDA: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\mat\lda\trainDatabase.mat
  - Mat file test: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\mat\test\Database.mat
  - Mat file cares: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\mat\candidates.mat
- Paràmetres imatges:**
  - Amplada imatges: 92
  - Alçada imatges: 112
  - Escala pel reconeixement d'ulls: 1.5
- Paràmetres Viola & Jones:**
  - Classificador cares: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\src\haar\cascade\_frontface\_at2.xi
  - Classificador ulls: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\src\haar\cascade\_eye\_tree\_eyeglas
  - Escala mínima detecció ulls: 1.1
  - Escala mínima detecció cares: 1.1
- Icones:**
  - Img no disponible: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\src\src\mat\_available.jpg
  - Img no reconegut: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\src\src\mat\_notfound.jpg
  - Img processant: C:\Users\Jordi\Dropbox\UOC\Projecte\Codi\src\src\processing.jpg
  - Filtres img: \*.jpg; \*.png; \*.bmp; \*.pnm
- Tipus operacions webcams:**
  - Preview code: prev
  - Codi PCA reconeixement video: reconvid
  - Codi PCA reconeixement estàtic: recongst
  - Codi video: vid
  - Codi foto: ph
  - Codi LDA reconeixement video: recfldvid
  - Codi LDA reconeixement estàtic: recfldstst
- Paràmetres PCA i LDA:**
  - Nombre imatges entrenament: 10
  - Distància màxima PCA: 80000000
  - Distància màxima LDA: 80000000
  - Nombre d'eigenvectors: 20
  - Nombre d'imatges per classe: 5

At the bottom of the window is a button labeled "Guardar paràmetres".

Si és el primer cop que, s'executa l'aplicació apareix una pantalla amb indicacions a tenir en compte perquè l'aplicació funcioni correctament:

The screenshot shows a window titled "Primer inici" (First Run) with the following text:

**Primer inici**

És el primer cop que s'inicia el programa, perquè funcioni correctament cal:

- Tenir instal·lat el Microsoft de Visual Studio C++ 2008 express SP1
- Configurar el Matlab amb el compilador de C++ per poder utilitzar les funcionalitats de detecció d'objectes de Viola&Jones d'OpenCV
- Tenir instal·lada la versió d'OpenCV 2.4.3 a c:\opencv2.4.3
- Modificar el fitxer mexopts.bat de %APPDATA%\Roaming\MathWorks\MATLAB\R2010a\per tal de que el compilador mex utilitzi les llibreries OpenCV. Dins de la carpeta rsc hi ha un fitxer mexopts.bat d'exemple.
- Si ja s'han seguit aquests passos, cal tancar la finestra i tornar a iniciar el programa

At the bottom of the window is a button labeled "Tanca".

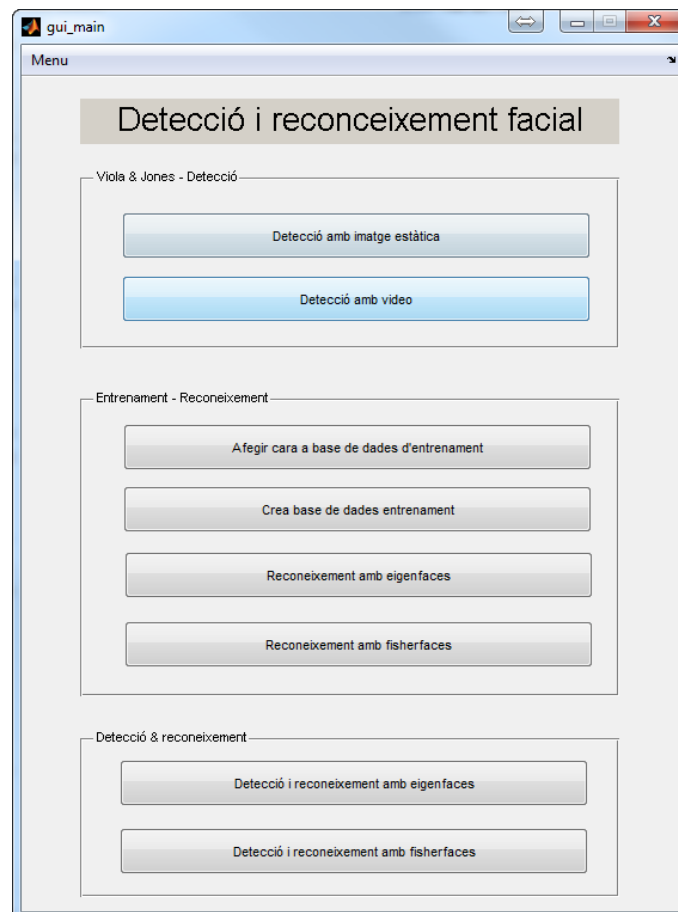
### 3.1.3 POSSIBILITATS DE L'APLICACIÓ

Com ja s'ha dit abans l'aplicació permet treballar tant amb Viola & Jones (per detectar cares) com Eigenfaces i FisherFaces, a més també es pot executar una subaplicació que realitza les dues tasques a l'hora.

#### 3.1.3.1 PANTALLA PRINCIPAL

A continuació es pot veure la pantalla principal, des de on es té accés a totes les opcions de l'aplicació. Dins d'aquesta pantalla es poden distingir els botons que indiquen l'acció a realitzar i un menú on es defineixen els paràmetres dels mètodes.

Dins d'aquesta pantalla principal, hi ha un menú on es defineixen tots els paràmetres de l'aplicació tals com paths de bases de dades, icones, classificadors pel mètode viola & Jones, etc... Cal notar que tot els paràmetres estan inicialitzats amb un valor per defecte, es recomana només tocar els path de directoris i fitxers en funció de la ubicació on s'hagi instal·lat l'aplicació.



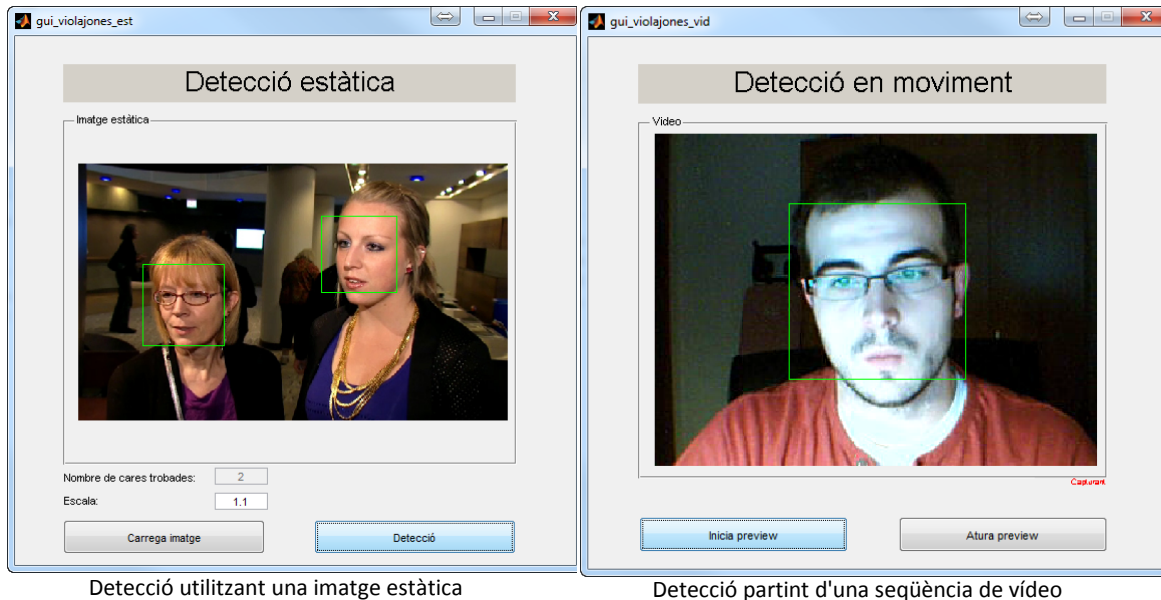
#### 3.1.3.1 VIOLA & JONES - DETECCIÓ

Aquest és el grup sobre la detecció facial. En aquesta secció es poden realitzar totes les accions implementades referents al mètode Viola & Jones: detecció de cares en imatges estàtiques (primera figura) i detecció de cares en seqüències de vídeo (segona figura).



Per fer la detecció partint d'una imatge estàtica cal utilitzar el botó "Detecció amb imatge estàtica", per fer la detecció partint d'una seqüència de vídeo, cal fer clic sobre el botó "Detecció amb vídeo".

En el cas de la detecció estàtica cal fer clic al botó "carrega imatge" per escollir la imatge i llavors fer clic al botó "detecció" per detectar les cares. En el cas de la detecció en moviment cal fer clic al botó "inicia preview" per iniciar la captura des de webcam, cada cop que una cara entri en escena i sigui detectada s'emmarcarà dins d'un requadre verd, per aturar la captura cal fer clic en el botó "atura preview".



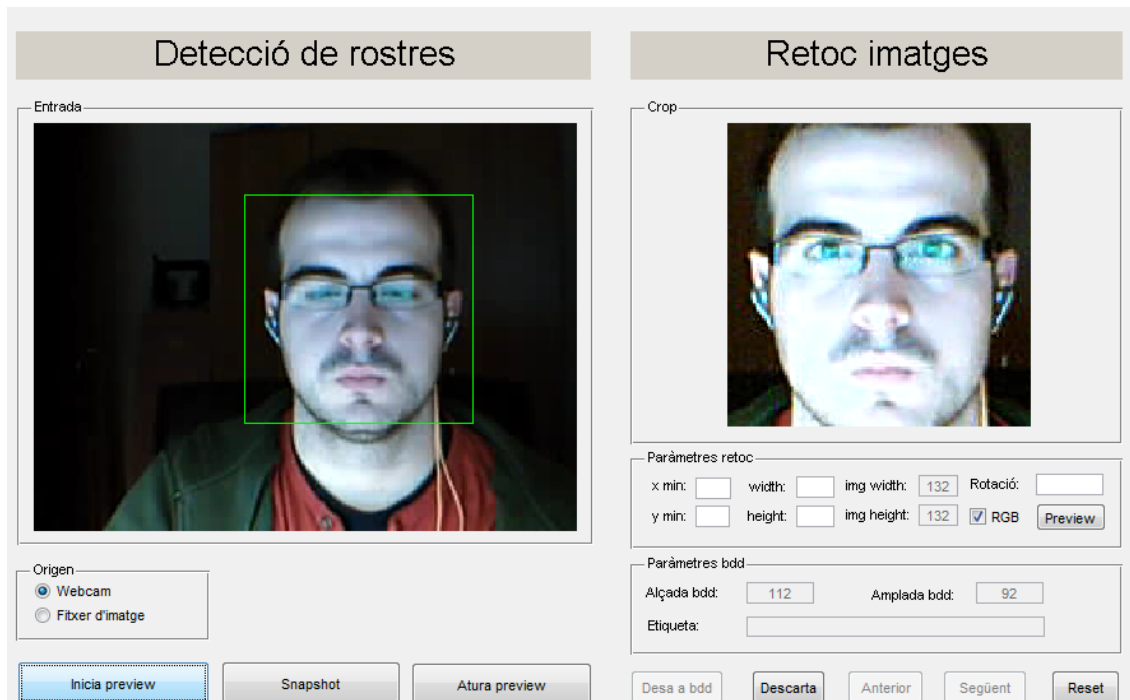
### 3.1.3.2 AFEGIR UNA CARA A LA BASE DE DADES

La pantalla està pensada per tal de que qualsevol persona pugui inserir la seva cara a la base de dades que s'utilitzarà en el procés d'entrenament dels algorismes de reconeixement facial. En el moment de detecció de la cara l'usuari té la possibilitat de capturar la cara i retocar-la (centrar, rotar i escalar la imatge), etiquetar-la i guardar-la a la base de dades.

L'obtenció de cares es pot fer partint d'imatges estàtiques o bé d'una seqüència de vídeo, l'usuari pot fer aquesta elecció mitjançant el panell inferior esquerre "Origen". Un cop es detecten les cares mitjançant el botó "Snapshot" es poden passar a retocar a la secció "Retoc imatges".

El retoc es pot dur a terme sobre totes les cares detectades. Dins del panell "paràmetres retoc" l'usuari estableix la mida de la imatge a través dels camps de text "width" i "height". La mida que ha de tenir la imatge es veu especificada en el panell "paràmetres bdd" a través dels valors "alçada bdd" i "amplada bdd".

Els camps de text s'utilitzen per desplaçar la imatge sobre l'eix x i y, el camp "rotació" per rotar la imatge un nombre de graus determinat i el botó "preview" per veure els canvis aplicats sobre la imatge. Si un cop fet el preview es decideix que la imatge és bona, es pot afegir a la base de dades indicant el nom del personatge en qüestió (camp "tag") i prement el botó "Desa a bdd".



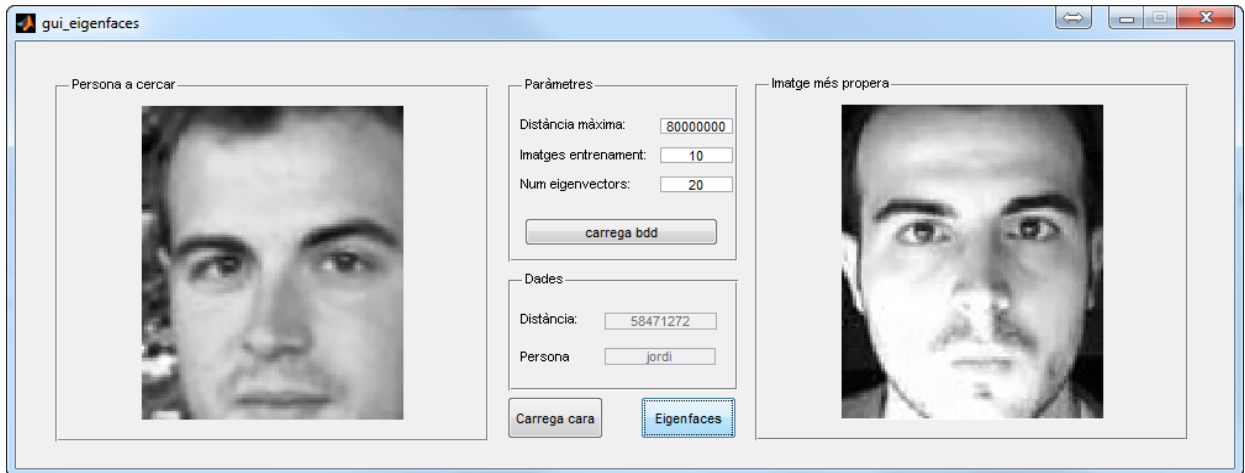
### 3.1.3.3 RECONeixEMENT FACIAL - EIGENFACES I FISHERFACES

Aquestes opcions permeten veure el funcionament dels mètodes eigenfaces i fisherfaces. Donada una cara es busca dins de la base de dades (prèviament carregada) la persona que més se li assembla.

Durant aquest procés, abans de reconèixer la cara cal dur a terme la fase d'entrenament de les dades que hi ha a la base de dades. A l'aplicació se li ha de proporcionar la distància màxima permesa entre imatges, el nombre d'imatges d'entrenament que s'utilitzaran, el nombre d'eigenvectors i el nombre d'imatges per classe (aquest en cas que s'utilitzi fisherfaces.)

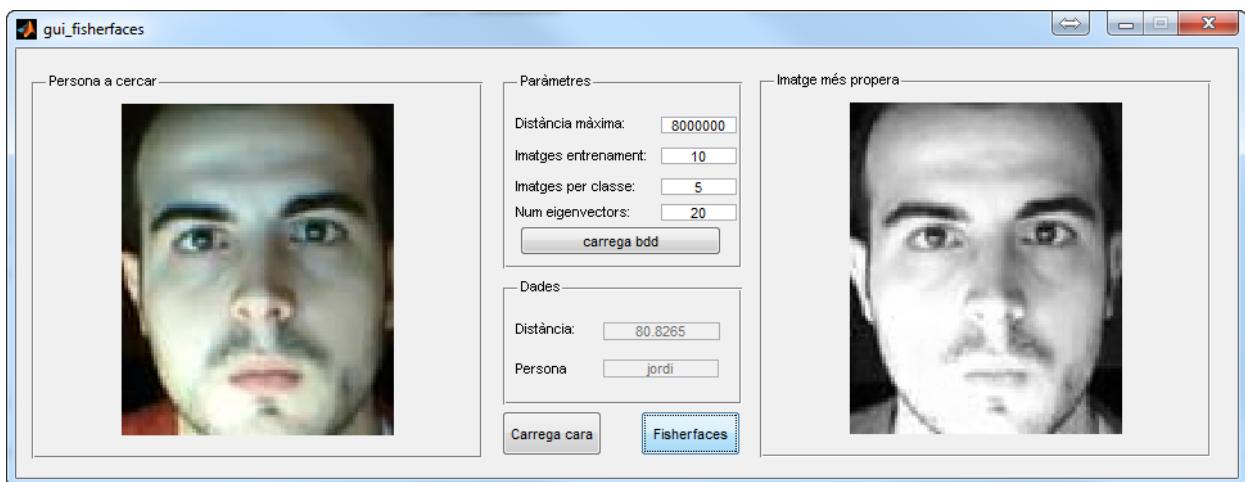
#### EIGENFACES:

- S'ha de proporcionar la distància màxima permesa entre imatges.
- El nombre d'imatges que es que s'utilitzaran per realitzar l'entrenament.
- El nombre d'eigenvectors utilitzats.
- Prémer el botó "carrega bdd" i indicar el path de la base de dades d'entrenament (directori amb les imatges).
- Prémer el botó "carregar cara" per carregar la fotografia d'un personatge. La fotografia ha de tenir la mateixa mida que les imatges utilitzades en l'entrenament.
- Prémer el botó "Eigenfaces" per fer la tasca de reconeixement.



### FISHERFACES:

- S'ha de proporcionar la distància màxima permesa entre imatges.
- El nombre d'imatges que es que s'utilitzaran per realitzar l'entrenament.
- El nombre d'imatges per classe.
- El nombre d'eigenvectors utilitzats.
- Prémer el botó "carrega bdd" i indicar el path de la base de dades d'entrenament (directori amb les imatges).
- Prémer el botó "carregar cara" per carrega la fotografia d'un personatge. La fotografia ha de tenir la mateixa mida que les imatges utilitzades en l'entrenament.
- Prémer el botó "Fisherfaces" per fer la tasca de reconeixement.



### 3.1.3.4 DETECCIÓ I RECONeixEMENT FACIAL

La pantalla que es mostra és l'aplicació completa. Permet veure en acció els mètodes de detecció, alineament i reconeixement funcionant com un bloc. En el moment que l'usuari prem el botó de "start" la webcam comença a captura i per cada cara detectada el programa fa el reconeixement i mostra el nom de la persona a la part superior esquerra de la bounding box de la cara detectada.

Si l'usuari ha escollit com a origen una imatge estàtica (fitxer d'imatge) la prémer el botó "start" se li demanarà que busqui una fotografia.

