



SISTEMA DE REONEIXEMENT DE CARES BASAT EN MÚLTIPLES VISTES

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica

realitzat per

Adriana Romero Soriano

i dirigit per

Jose Manuel Alvarez Lopez

Bellaterra, 11 de juny de 2010

El sotasignat, Jose Manuel Alvarez Lopez

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per Adriana Romero Soriano.

I per tal que consti firma la present.

Bellaterra, Juny de 2010.

Firmat: Jose Manuel Alvarez Lopez

A tots aquells que porto al cor.

Agraïments

M'agradaria agrair l'esforç i recolzament del meu director de projecte Jose Manuel Alvarez. Gràcies per atendre'm i resoldre els meus dubtes. Gràcies pels consells i ànims.

Gràcies David per la teva paciència i el teu suport, sense tu això no hagués estat possible.

Gràcies a tots els grans amics i amigues que m'han animat a seguir endavant, en especial al Toni, que tot i trobar-se en la mateixa situació que jo, sempre ha sabut trobar temps per donar-me un cop de mà.

Gràcies Josep Vallverdú per dedicar part del teu temps a donar-me consells d'última hora.

Gràcies a totes les persones que m'envolten i que amb petits detalls també han sabut ajudar-me.

Índex

1. Introducció	1
1.1. Estat de l'art i motivacions.....	1
1.2. Objectius	3
1.3. Metodologia	3
1.4. Estructura de la memòria.....	4
2. Anàlisi de requisits	5
2.1. Requisits funcionals	5
2.2. Requisits no funcionals	5
2.2.1. Interfícies Software	5
2.2.2. Interfícies Hardware	6
2.2.3. Memòria	6
2.3. Estudi de viabilitat.....	6
2.3.1. Viabilitat tècnica	6
2.3.2. Viabilitat econòmica	6
2.3.3. Viabilitat legal	7
2.4. Planificació temporal.....	7
2.4.1. Tasques.....	7
2.4.2. Terminis	9
3. Procediment.....	11
3.1. Camps aleatoris de Markov.....	11
3.1.1. Problema d'etiquetatge.....	11
3.1.2. Veïnatge i observacions	12
3.1.3. Definició.....	13
3.1.4. Inferència d'etiquetes	14
3.1.5. Observacions i evidències locals.....	17
3.2. Aprenentatge	20
3.2.1. Model de pell.....	20

3.2.2.	Model de no-pell	23
3.3.	Esquema general	24
4.	Disseny.....	25
4.1.	Disseny general	25
4.1.1.	Disseny general del bloc d'aprenentatge.....	25
4.1.2.	Disseny general del bloc de classificació.....	26
4.1.3.	Disseny general del bloc avaluador.....	26
4.2.	Etapas	27
4.2.1.	Etapas de l'aprenentatge.....	28
4.2.2.	Etapas de la classificació.....	31
5.	Resultats	35
5.1.	Proves	35
5.1.1.	Veïnatge	35
5.1.2.	Número d'iteracions del Belief Propagation	36
5.1.3.	Paràmetres de la propagació de missatges	38
5.1.4.	Pesos.....	39
5.2.	Resultats finals	40
5.3.	Planificació final	42
6.	Conclusions i treballs futurs.....	43
6.1.	Conclusions	43
6.2.	Treballs futurs	43
7.	Bibliografia	47

Índex de figures

<i>Fig. 1 Diagrama de Gantt de la planificació inicial.</i>	9
<i>Fig. 2 Assignació d'etiquetes als píxels d'una imatge.</i>	12
<i>Fig. 3 Veïnatge d'un emplaçament. (a) Veïnatge 4. (b) Veïnatge 8.</i>	12
<i>Fig. 4 Graf d'un píxel amb veïnatge 4.</i>	13
<i>Fig. 5 Graf d'un píxel amb veïnatge 8.</i>	14
<i>Fig. 6 CAM on els emplaçaments estan connectats als seus veïns i a les seves observacions.</i>	14
<i>Fig. 7 Aplicació del Belief Propagation sobre un CAM.</i>	15
<i>Fig. 8 Propagació de missatge del píxel 'i' cap al píxel 'j'.</i>	15
<i>Fig. 9 CAM on els píxels estan connectats a la seva mitjana μ i a la seva desviació típica σ.</i>	17
<i>Fig. 10 (a) Model de no-pell. (b) Model de pell.</i>	18
<i>Fig. 11 Probabilitat de pell d'un píxel.</i>	19
<i>Fig. 12 Probabilitat de no-pell d'un píxel.</i>	19
<i>Fig. 13 (a) Histograma de pell (b) Funció de densitat de pell.</i>	20
<i>Fig. 14 Modelatge de la pell.</i>	21
<i>Fig. 15 Procés per obtenir el model de pell.</i>	22
<i>Fig. 16 Transformació imatge i selecció de regions de no-pell.</i>	23
<i>Fig. 17 Modelatge de no-pell.</i>	24
<i>Fig. 18 Mapa conceptual de l'algorisme de detecció de pell proposat.</i>	24
<i>Fig. 19 Diagrama d'aprenentatge.</i>	26
<i>Fig. 20 Diagrama de classificació.</i>	26
<i>Fig. 21 Imatge a classificar i màscara de la imatge a classificar.</i>	27
<i>Fig. 22 Diagrama d'avaluació.</i>	27
<i>Fig. 23 Diagrama detallat d'aprenentatge.</i>	28
<i>Fig. 24 Diagrama de flux de la preparació de dades de l'aprenentatge del model de pell.</i>	29
<i>Fig. 25 Diagrama de flux de l'aprenentatge del model de pell.</i>	29
<i>Fig. 26 Diagrama de flux de la preparació de dades de l'aprenentatge del model de no-pell.</i>	30
<i>Fig. 27 Diagrama de flux de l'aprenentatge del model de no-pell.</i>	31
<i>Fig. 28 Diagrama detallat de classificació.</i>	32
<i>Fig. 29 Diagrama de flux de preparació de dades per crear el CAM.</i>	32
<i>Fig. 30 Diagrama de flux de la classificació de píxels.</i>	33
<i>Fig. 31 Augment del temps de classificació en funció del veïnatge.</i>	36
<i>Fig. 32 Augment del consum de memòria en funció del veïnatge.</i>	36
<i>Fig. 33 Augment de l'efectivitat en funció del nombre d'iteracions del Belief propagation.</i>	37

<i>Fig. 34 Augment del temps de classificació en funció del nombre d'iteracions del Belief propagation.</i>	37
<i>Fig. 35 Detecció de cares per diferents matrius de propagació.</i>	38
<i>Fig. 36 Detecció de cares amb diferents pesos.</i>	39
<i>Fig. 37 Algunes imatges de la base de dades Caltech utilitzades per testejar.</i>	40
<i>Fig. 38 (a) Classificació sense veïnatge. (b) Classificació amb veïnatge.</i>	41
<i>Fig. 39 Diagrama de Gantt de la planificació final.</i>	42

Índex de taules

<i>Taula 1 Viabilitat econòmica.</i>	7
<i>Taula 2 Terminis previstos.</i>	9
<i>Taula 3 Influència del veïnatge sobre els resultats de la classificació.</i>	35
<i>Taula 4 Influència de la matriu de propagació sobre els resultats de la classificació.</i>	38
<i>Taula 5 Influència dels pesos sobre els resultats de la classificació.</i>	39
<i>Taula 6 Resultats de detecció facial.</i>	40
<i>Taula 7 Rendiment de diversos mètodes de detecció facial extrets de [2] i del mètode emprat en el projecte.</i>	41
<i>Taula 8 Terminis finals.</i>	42

1. Introducció

Des del seu naixement, els éssers humans desenvolupen l'habilitat de reconèixer i distingir cares. Aquest procés humà, igual que molts altres, s'intenta entendre i imitar des d'un punt de vista científic. La ciència que intenta entendre i imitar la percepció visual humana és la visió per computador. Aquesta analitza i identifica components d'imatges i, per tant, pot ser utilitzada per intentar reproduir l'habilitat humana de veure i detectar cares. La imitació d'aquesta darrera habilitat mitjançant *software* no neix fins la dècada dels 60. Des d'aleshores, aquest àmbit ha estat en constant evolució donant lloc als sistemes de detecció facial.

Els sistemes de detecció facial són aplicacions que permeten identificar cares a partir de determinades característiques extrems d'imatges digitals. Donada una imatge, pretenen determinar si hi apareixen cares i, en cas afirmatiu, ubicar-les. Avui en dia existeixen diferents mètodes i algoritmes que permeten millorar les tècniques utilitzades en la detecció de cares.

1.1. Estat de l'art i motivacions

Els algoritmes de detecció facial actuals es poden dividir en dos grups. El primer grup fa referència a tots aquells algoritmes que fan una anàlisi *top-down*, en la que es busquen models de cara basant-se en un conjunt de regles humanes. L'algoritme podria extreure la posició relativa i la mida dels ulls, nassos i boques presents en una imatge per tal de decidir si determinades parts de la imatge corresponen a possibles cares. El segon grup fa referència a tots aquells algoritmes que fan una anàlisi *bottom-up*, en la que s'extreuen característiques de baix nivell (a nivell de píxel, com el color) per tal de trobar regions que puguin pertànyer a una cara.

En aquest projecte ens centrarem en l'anàlisi de característiques de baix nivell per trobar regions que potencialment puguin pertànyer a una cara. Ens basarem en la informació del color de la pell per determinar els píxels que són de pell i, que per tant, poden pertànyer a la cara. En conseqüència, efectuarem la detecció de cares mitjançant la detecció de píxels de pell.

Cal remarcar que detectar píxels de pell és un procés complicat donat que existeix una gran variabilitat en el seu color. A més, les condicions d'adquisició de la imatge dificulten la tasca ja que la detecció de pell és susceptible a la il·luminació. Degut a la il·luminació, poden aparèixer ombres i, per tant, un canvi en les condicions d'il·luminació produeix un canvi del color de la

pell a la imatge. En ambients més il·luminats la pell té un color més clar i, de la mateixa manera, en ambients més foscos la pell té un color més fosc.

No obstant, s'ha observat que els mètodes de detecció de pell que es basen en el color proporcionen informació eficient a l'hora de detectar regions pertanyents a la cara [1]. Els passos a seguir per dur a terme la detecció de pell utilitzant informació sobre el color són: representar els píxels de la imatge en un espai de color adequat, modelar els píxels de pell i de *no-pell* mitjançant distribucions adequades i, finalment, realitzar la classificació [1]. És important destacar que l'elecció de l'espai de color determina l'eficiència de la classificació i, en conseqüència, sempre és una tasca difícil.

L'espai de color RGB és l'espai de color per defecte i les transformacions cap a altres espais de color milloren la classificació de píxels de pell disminuint la superposició entre els píxels de pell i els de *no-pell* [1]. A fi de reduir la influència de la llum sobre el color dels píxels de pell, podem normalitzar els components RGB de manera que la seva suma sigui 1. D'altra banda, l'espai de color HSV és invariant a determinats tipus de llum (llum blanca, llum d'ambient...) i pot proporcionar bons resultats a l'hora de detectar píxels de pell [1]. No obstant, la majoria d'espais de color segueixen depenent de les condicions d'adquisició de la imatge. Per això, Álvarez, Gevers i López [2] proposen un mètode que combina diferents espais de color per aconseguir invariància respecte les condicions d'il·luminació. Aquest mètode es basa en l'anàlisi de la correlació entre diferents espais de color per aconseguir ser invariant a les variacions de llum. Donat que aquest últim mètode proporciona bons resultats respecte als altres i ofereix la possibilitat de treballar amb probabilitats i en un espai d'una única dimensió, l'utilitzem per aconseguir minimitzar els efectes de les condicions d'adquisició de la imatge a l'hora de detectar píxels de pell.

D'altra banda, cal destacar que els mètodes exposats fins ara prenen decisions a nivell de píxel. En conseqüència, no tenen en compte el veïnatge per determinar si un píxel és de pell o de *no-pell*. El veïnatge defineix una dependència contextual important per a interpretar la informació visual. En una imatge, si un píxel està envoltat de potencials píxels de pell, el més probable és que es tracti d'un píxel de pell. De la mateixa manera, si un píxel està envoltat de píxels que potencialment no són de pell, el més probable és que es tracti d'un píxel de *no-pell*. Per tenir en compte la influència dels veïns sobre un píxel, utilitzem un model gràfic en el que els nodes fan referència als píxels de la imatge i els arcs representen les dependències entre aquests, és a dir, el seu veïnatge. Establim aquest model mitjançant camps aleatoris de Markov que, amb una estructura de graf i un algoritme de propagació de missatges, ens permeten determinar les probabilitats de cada píxel de ser de pell i de *no-pell*.

1.2. Objectius

A partir dels fets exposats en aquesta introducció i donada la importància de la detecció d'àrees de pell, ens proposem, com a objectiu general, implementar un sistema capaç de detectar qualsevol regió de pell sota qualssevol condicions d'il·luminació.

Com hem vist al llarg de la introducció, un dels principals problemes dels mètodes de detecció de pell actuals és que prenen les decisions a nivell de píxel. Per superar la darrera mancança, volem proposar un algoritme que tingui en compte el veïnatge a l'hora de prendre les decisions de classificació. Superant aquesta mancança volem millorar el mètode proposat a l'article [2].

Per tal d'assolir els darrers objectius ens marquem les següent fites:

- Revisar l'estat de l'art dels algoritmes de detecció de pell.
- Revisar l'estat de l'art dels algoritmes que tenen en compte el veïnatge.
- Revisar l'estat de l'art dels camps aleatoris de Markov.
- Implementar l'algoritme proposat i validar-lo mitjançant la comparació dels resultats obtinguts amb els de l'estat de l'art.

1.3. Metodologia

Per assolir els objectius plantejats anteriorment, seguirem un model en cascada format per les següent etapes:

- Revisió de l'estat de l'art de com aconseguir invariància respecte les condicions de llum. L'article de Álvarez, Gevers i López [2] serà el nostre punt de partida donat que proposa un mètode que proporciona bons resultats respecte als altres i ofereix la possibilitat de treballar amb probabilitats i en un espai d'una única dimensió.
- Revisió de l'estat de l'art dels algoritmes que tenen en compte la influència del veïnatge. En aquesta etapa estudiarem l'estructura i el funcionament dels camps aleatoris de Markov.
- Realització d'una anàlisi de requisits per entendre les necessitats que tindrem a l'hora de dur a terme el projecte.
- Disseny d'una solució capaç de satisfer els requisits i objectius esmentats anteriorment.
- Programació de la solució dissenyada per parts:
 - Aprenentatge
 - Creació de la xarxa de Markov

- Classificació
- Testeig de la solució per comprovar que s'hagin assolit els objectius proposats.
- Valoració el treball realitzat i extracció conclusions.

No obstant, a l'etapa de programació, anirem testejant regularment que tot funcioni adequadament. D'aquesta manera evitarem trobar problemes de grans dimensions a l'etapa de proves.

1.4. Estructura de la memòria

La resta de la memòria està dividida en els següents apartats:

- **Anàlisi de requisits:** En aquest capítol analitzarem el projecte i identificarem tots els seus requisits. Tot seguit estudiarem la viabilitat del projecte i exposarem la planificació temporal que volem seguir.
- **Procediment:** En aquest capítol exposarem els conceptes necessaris per entendre el projecte: els camps aleatoris de Markov per classificar els píxels d'una imatge i l'entrenament previ necessari per dur a terme aquesta classificació.
- **Disseny:** En aquest capítol definirem l'esquema general del projecte a nivell de disseny i exposarem el seu desenvolupament. L'objectiu d'aquest capítol és obtenir un disseny que ens faciliti la implementació del projecte i que satisfaci tots els requisits descrits al capítol 2.
- **Resultats:** En aquest capítol presentarem les proves realitzades i els resultats obtinguts. Tot seguit, els compararem amb els resultats ja existents per tal de veure si el fet de tenir en compte el veïnatge millora la detecció facial. Finalment, exposarem i analitzarem la planificació final del projecte.
- **Conclusions i treballs futurs:** En aquest últim capítol de la memòria exposarem les conclusions a les que hem arribat a l'acabar el projecte i proposarem possibles treballs futurs que podrien continuar la feina ja presentada.

2. Anàlisi de requisits

En aquest capítol detallarem els requisits necessaris per al nostre projecte. El nostre projecte parteix de la proposta presentada per Álvarez, Gevers i López [2] per aconseguir invariància respecte les condicions d'adquisició d'una imatge a l'hora de detectar regions de pell i introdueix la influència del veïnatge per intentar millorar aquesta detecció.

2.1. Requisits funcionals

El requisit principal és proposar un mètode capaç de detectar els píxels de pell presents en una imatge de manera que la detecció sigui independent de les condicions d'adquisició de la imatge i tingui en compte el veïnatge. Aquest mètode ha de ser capaç de, donada una imatge d'entrada¹, retornar una imatge binària² en la qual les regions blanques corresponguin als píxels de pell i les regions negres als píxels de *no-pell*. Classificarem les imatges d'una en una de manera seqüencial.

2.2. Requisits no funcionals

En aquest apartat veurem els requisits no funcionals que hem de tenir en compte a l'hora de realitzar el nostre projecte.

2.2.1. Interfícies Software

Per implementar el projecte utilitzarem el llenguatge de programació Matlab, un requisit que se'ns demana al projecte i que ens permet realitzar càlculs de manera més ràpida que amb els llenguatges de programació tradicionals. Per tant, el primer requisit en quant a programari serà un sistema operatiu per al qual Matlab [3] estigui disponible. Això inclou Windows (32 i 64 bits), Linux (32 i 64 bits) i Mac OS X (32 i 64 bits).

¹ Formats d'entrada suportats: BMP, CUR, GIF, HDF4, ICO, JPEG, JPEG 2000, PBM, PCX, PGM, PNG, PPM, RAS, TIFF i XWD.

² En el nostre cas, la imatge binària de sortida serà un JPEG.

Òbviament també necessitarem el *software* de Matlab per poder realitzar el projecte i, com que treballem amb funcions de processament de imatges, finances i estadística, també requerirem els tres Toolbox de Matlab següents:

- Matlab Image Processing Toolbox.
- Matlab Financial Toolbox.
- Matlab Statistics Toolbox.

2.2.2. Interfícies Hardware

El programa haurà de ser capaç de córrer en una plataforma amb un processador Intel o AMD x86 que suporti el conjunt d'instruccions SSE2, necessari per fer servir el Matlab [3].

2.2.3. Memòria

El programa haurà de ser capaç de córrer en una plataforma amb com a mínim 1024 MB de RAM i entre 3 i 4 GB de disc lliures, necessaris pel funcionament i la instal·lació típica de Matlab [3].

2.3. Estudi de viabilitat

En aquest apartat realitzarem un estudi de viabilitat del projecte des d'un punt de vista tècnic, econòmic i legal.

2.3.1. Viabilitat tècnica

Per realitzar el projecte, primer haurem d'adquirir coneixements relacionats amb l'estat de l'art del problema per, finalment, poder proposar i implementar l'algoritme de detecció d'àrees de pell esmentat a la introducció. Donat que ja s'han dut a terme projectes d'una perspectiva similar a la nostra i que la universitat posa a la nostra disposició biblioteques i laboratoris amb la informació necessària per adquirir aquests coneixements, podem dir que el nostre projecte és viable tècnicament.

2.3.2. Viabilitat econòmica

Realitzarem el projecte en un entorn Windows 7. El *software* de desenvolupament serà, com hem esmentat anteriorment, Matlab i utilitzarem l'Office 2007 per escriure la memòria del projecte. Per tant, per dur a terme el projecte, necessitarem un pc que tingui Windows 7 com a sistema operatiu i Matlab i Office 2007 instal·lats. Suposem que el pc necessari té un preu de

800 euros. El sistema operatiu Windows 7 Professional té un cost de 285 euros [4] i una llicència d'Office 2007 té un cost de 140 euros [4]. El Matlab té un preu variable, no obstant, per una llicència d'estudiant individual que inclogui les *toolboxes* necessàries podem pagar aproximadament 820 euros [3]. Donat que no estem fent cap aplicació comercial, la llicència d'estudiant del Matlab serà suficient.

D'altra banda, el projecte té una durada d'unes 300 hores (corresponents a 15 crèdits). Si suposem un preu de 20 euros/hora i una única persona treballant en el projecte, el cost seria de 6000 euros. Per tant, el projecte tindria un preu total de 8045 euros. Donats els resultats que es volen obtenir, podem dir que es tracta d'un preu raonable. Per tant, el projecte també és viable econòmicament (*Taula 1*).

Producte	Cost
PC	800 €
Windows 7	285 €
Office 2007	140 €
Matlab	820 €
300 hores	6000 €
Total	8045 €

Taula 1 Viabilitat econòmica.

2.3.3. Viabilitat legal

Legalment, el projecte també és viable ja que no tracta dades de caràcter confidencial. D'altra banda, tampoc infringeix cap llei ja que disposem del Matlab amb llicència als ordinadors de la universitat i el codi que utilitzem en algunes parts del projecte és lliure.

2.4. Planificació temporal

A l'inici del projecte es va realitzar una planificació temporal. Per tal de fer una planificació temporal del treball més detallada, vam dividir el projecte en diferents tasques a les quals vam assignar una dedicació temporal.

2.4.1. Tasques

- Revisar l'estat de l'art del projecte (Durada: 5 setmanes).
 - Revisar l'estat de l'art dels algorismes de detecció de pell.

- Revisar l'estat de l'art dels algoritmes que treballen amb el veïnatge
- Realitzar l'anàlisi de requisits i el disseny (Durada: 2 setmanes)
 - Realitzar l'anàlisi de requisits
 - Dissenyar una solució general capaç de satisfer els objectius.
- Realitzar l'aprenentatge (Durada: 7 setmanes).
 - Dissenyar detalladament l'aprenentatge.
 - Aprendre a partir de mostres la distribució dels píxels de pell.
 - Aprendre a partir de mostres la distribució dels píxels de *no-pell*.
 - Dissenyar l'estructura de la xarxa de Markov i crear-la.
- Realitzar la classificació (Durada: 6 setmanes).
 - Dissenyar detalladament la classificació
 - Determinar la mitjana i la desviació de cada píxel.
 - Determinar les probabilitats de cada píxel de ser de pell i de *no-pell*.
 - Aplicar els camps aleatoris de Markov per tenir en compte la influència del veïnatge sobre cada píxel.
 - Classificar els píxels de la imatge en píxels de pell i de *no-pell*.
- Testejar (Durada: 4 setmanes).
 - Testejar que tot funcioni correctament.
 - Opcionalment, modificar alguns paràmetres a fi d'obtenir millors resultats.
- Escriure la memòria del projecte (Durada: 4 setmanes).

Per tal de seguir aquesta planificació es van preveure 3 hores de dedicació diàries. Cal destacar que la planificació es va fer de manera que quedés suficientment temps per resoldre possibles imprevistos.

2.4.2. Terminis

A continuació es presenta la taula i el diagrama amb els terminis que es van preveure (*Taula 2, Fig. 1*).

Name	Begin date	End date
Revisar l'estat de l'art (setmana 1)	10/13/09	10/20/09
Revisar l'estat de l'art (setmana 2)	10/20/09	10/30/09
Revisar l'estat de l'art (setmana 3)	12/16/09	12/22/09
Revisar l'estat de l'art (setmanes 4 i 5)	12/22/09	1/8/10
Analitzar els requisits i dissenyar solució general (setmana 1)	10/30/09	11/6/09
Analitzar els requisits i dissenyar solució general (setmana 2)	1/8/10	1/15/10
Realitzar l'aprenentatge (setmanes 1-3)	11/6/09	11/26/09
Realitzar l'aprenentatge (setmanes 5-9)	1/15/10	2/12/10
Realitzar la classificació (setmanes 1-1.5)	2/12/10	2/23/10
Realitzar la classificació (setmanes 1.5-2)	2/23/10	3/5/10
Realitzar la classificació (setmanes 3-6)	3/18/10	4/15/10
Testejar (setmanes 1)	3/10/10	3/18/10
Testejar (setmanes 2-4)	3/18/10	4/13/10
Escriure la memòria del projecte	4/27/10	5/25/10

Taula 2 Terminis previstos.

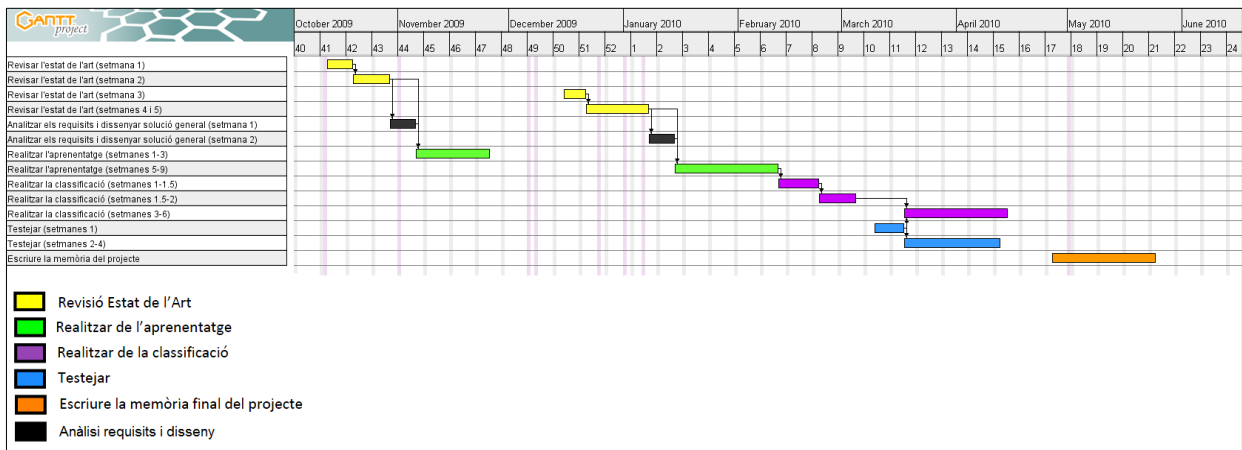


Fig. 1 Diagrama de Gantt de la planificació inicial.

3. Procediment

En aquest capítol exposarem els conceptes necessaris per entendre el projecte: la classificació de píxels d'una imatge tenint en compte el veïnatge mitjançant camps aleatoris de Markov i l'entrenament previ necessari per dur a terme aquesta classificació de manera invariant a les condicions d'adquisició de la imatge.

3.1. Camps aleatoris de Markov

Per permetre la detecció d'àrees de pell de manera més acurada és important tenir en compte les dependències contextuais. Els camps aleatoris de Markov (CAM) proporcionen una manera adequada i consistent de modelar les dependències contextuais d'un conjunt d'entitats, com els píxels d'una imatge. A més, tenen l'avantatge de no presentar relacions causa-efecte. Aquest avantatge és interessant donat que en el nostre cas volem que dos píxels veïns s'influïïn mútuament.

A continuació, introduïrem algunes definicions i conceptes bàsics per a la comprensió dels CAM i la seva funció dins del projecte.

3.1.1. Problema d'etiquetatge

A fi d'entendre la funció dels CAM dins del projecte, definim un problema d'etiquetatge en el que la solució és el conjunt d'etiquetes assignades als píxels de la imatge.

Un problema d'etiquetatge està format per un conjunt d'emplaçaments S i un conjunt d'etiquetes L .

Un emplaçament representa un punt o una regió d'una imatge. En el nostre cas, un emplaçament fa referència a un píxel de la imatge. Per tant, el conjunt d'emplaçaments S es correspon amb el conjunt de píxels de la imatge.

$$S = \{1, \dots, m\}$$

on $1, \dots, m$ són els índexs que corresponen als píxels de la imatge.

D'altra banda, una etiqueta és un dels estats que pot prendre un emplaçament. La detecció de pell és un problema de classificació que separa els píxels d'una imatge en dues classes: píxels de pell i píxels de *no-pell*. Com que estem classificant els píxels d'una imatge en píxels que pertanyen a la pell i píxels que no hi pertanyen, els píxels tindran únicament dos estats possibles: *pell* i *no_pell*. En conseqüència, el nostre conjunt d'etiquetes L està format pels dos estats anteriors.

$$L = \{pell, no_pell\}$$

Per tant, el problema d'etiquetatge consisteix en assignar una etiqueta f_i del conjunt L a cadascun dels emplaçaments del conjunt S . D'aquesta manera, definim el conjunt

$$f = \{f_1, \dots, f_m\}$$

que representa un etiquetatge possible dels emplaçaments del conjunt S [5]. En el nostre cas, el conjunt f representa un etiquetatge possible dels píxels d'una imatge (Fig. 2).

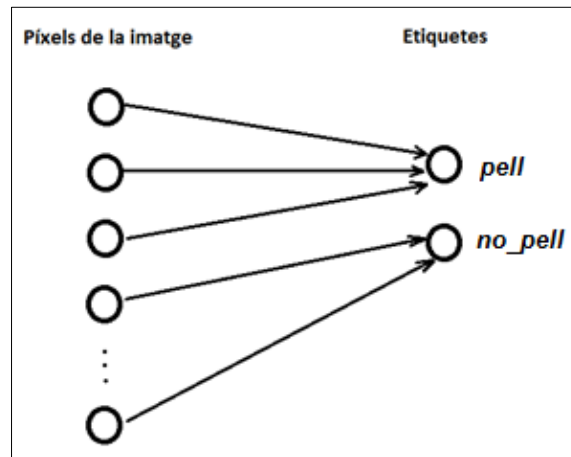


Fig. 2 Assignació d'etiquetes als píxels d'una imatge.

3.1.2. Veïnatge i observacions

Els CAM analitzen i modelen les dependències contextuais d'un conjunt d'emplaçaments S . Per tal de reflectir aquestes dependències, definim un sistema de veïnatge N

$$N = \{N_i \mid \forall i \in S\}$$

on N_i és el conjunt de veïns de i de manera que i no sigui veí de si mateix, és a dir, $i \notin N_i$ i que el veïnatge entre dos emplaçaments sigui mutu, és a dir, $i \in N_{i'} \Leftrightarrow i' \in N_i$ [5].

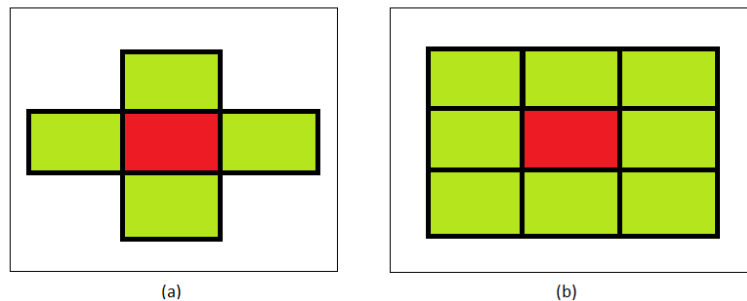


Fig. 3 Veïnatge d'un emplaçament. (a) Veïnatge 4. (b) Veïnatge 8.

No obstant, a part de tenir en compte el veïnatge, també hem de tenir present la informació pròpia de cada emplaçament. Anomenem observació a la informació pròpia d'un emplaçament.

Un emplaçament pot tenir una o vàries observacions. En el nostre cas, cada emplaçament (píxel) té dues observacions que ens permeten determinar la seva probabilitat de ser un píxel de pell i la de ser un píxel de *no-pell*.

3.1.3. Definició

Els CAM són un model gràfic que ens permet analitzar les dependències contextuais d'un conjunt de variables. Un model gràfic descriu les dependències entre un conjunt de variables aleatòries³ mitjançant un graf.

Els CAM estan formats per una família de variables aleatòries F definides sobre el conjunt d'emplaçaments S i que prenen un valor f_i dins del conjunt d'etiquetes L . F és un CAM sobre el conjunt S amb un veïnatge N si i només si

$$P(F = f) = P(F_1 = f_1, \dots, F_m = f_m) > 0 \quad \forall f$$

$$P(f_i | f_{S-\{i\}}) = P(f_i | f_{N_i})$$

on $f_{S-\{i\}}$ correspon a totes les etiquetes dels emplaçaments del conjunt $S - \{i\}$ i f_{N_i} correspon a totes les etiquetes del conjunt de veïns de l'emplaçament i [5]. De les dues propietats esmentades, la primera s'assumeix i la segona remarca que, en els CAM, la interacció existeix únicament entre etiquetes veïnes [5].

En el nostre cas, tenim un conjunt de píxels i volem modelar la influència que té el veïnatge sobre cadascun d'ells. Cal notar que les dependències entre píxels són mútues ja que si el píxel A és veí del píxel B, aleshores el píxel B és veí del píxel A. Per tant, donat que el veïnatge dels píxels sempre és mutu, podem representar les dependències contextuais mitjançant un graf no dirigit. En el nostre graf, els nodes fan referència als píxels de la imatge i els arcs representen les dependències entre aquests, és a dir, el seu veïnatge (Fig. 4, Fig. 5).

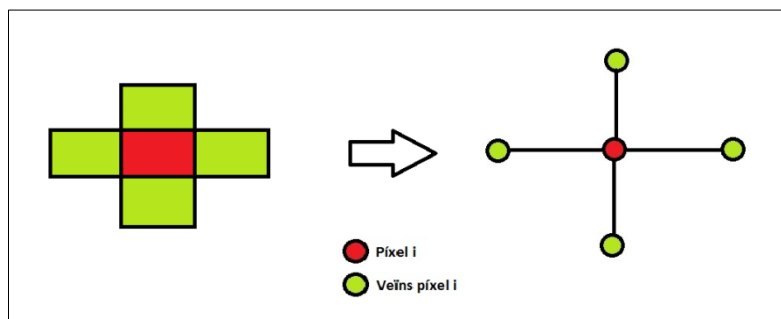


Fig. 4 Graf d'un píxel amb veïnatge 4.

³ S'anomena variable aleatòria una funció X que assigna a cada element ω de l'espai mostral Ω un nombre real $X(\omega) = x$ [8].

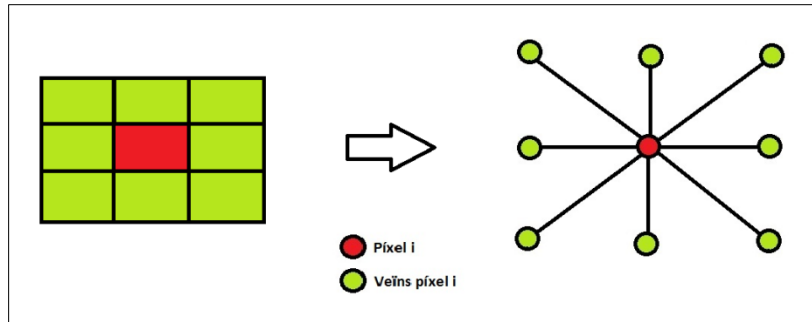


Fig. 5 Graf d'un píxel amb veïnatge 8.

A més, cal recordar que cada emplaçament té, en el nostre cas, dues observacions. Les observacions han d'estar connectades als emplaçaments mitjançant un arc. Per tant, cada node del graf està connectat als seus veïns i a les seves dues observacions (Fig. 6).

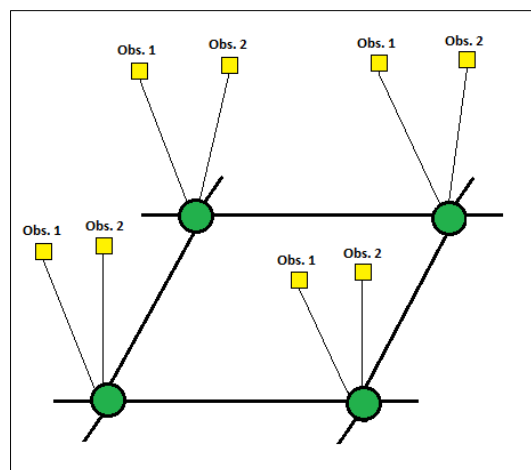


Fig. 6 CAM on els emplaçaments estan connectats als seus veïns i a les seves observacions.

Per tant, cada píxel es veu afectat per la seva pròpia probabilitat de ser de pell, la de ser *no-pell* i la influència dels seus veïns. A partir d'aquesta informació, ens preguntem com inferir l'etiquetatge dels píxels de la imatge.

3.1.4. Inferència d'etiquetes

Per tal de determinar l'etiquetatge dels píxels utilitzem el *Belief propagation*, un algoritme iteratiu de transmissió de missatges que fa inferència sobre models gràfics com els CAM. La inferència es calcula passant missatges entre nodes. Cada node envia i rep missatges que contenen informació estadística d'interès per als nodes veïns. Mitjançant aquest algoritme volem trobar l'assignació d'etiquetes més probable per al nostre model gràfic.

A l'aplicar l'algoritme anterior, els nodes del CAM estan connectats a través d'una matriu de propagació ψ_{ij} que defineix la relació entre els estats d'ambdós nodes. De la mateixa manera, els nodes estan connectats a les seves observacions a través de les evidències locals ϕ_i (Fig. 7).

Anomenem evidència local al vector que conté les probabilitats pròpies de cada emplaçament d'estar en els diferents estats, en el nostre cas, pell i *no-pell*. Les evidències locals es calculen a partir de les observacions de cada node.

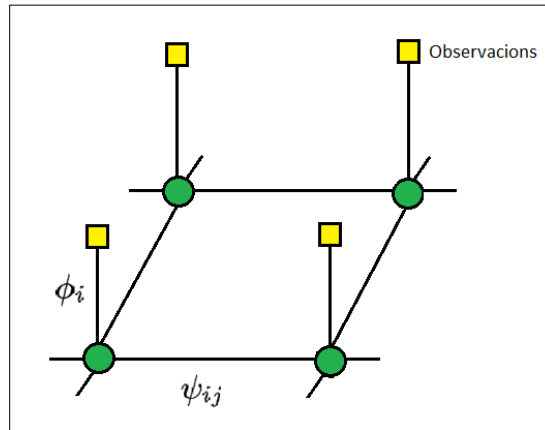


Fig. 7 Aplicació del Belief Propagation sobre un CAM.

L'algoritme *Belief propagation* infereix l'etiquetatge dels emplaçaments d'un CAM en funció del seu respectiu veïnatge. Per això, primer calcula la probabilitat pròpia de cada emplaçament d'estar en els diferents estats *i*, a partir d'aquesta informació, cada emplaçament determina quines són, segons ell, les probabilitats d'estar en els diferents estats dels seus veïns i les hi transmet a través d'un missatge. En el nostre cas particular, cada píxel calcula la seva probabilitat de ser píxel de pell i la de ser píxel de *no-pell* i determina les probabilitats dels seus veïns de ser píxels de pell i de *no-pell*. A continuació, cada píxel envia un missatge als seus veïns amb la informació estadística anterior.

El missatge que transmet un píxel (emissor) a un veí (receptor) es calcula a partir de les evidències locals del píxel emissor i dels seus missatges entrants, exceptuant el que prové del píxel receptor (Fig. 8).

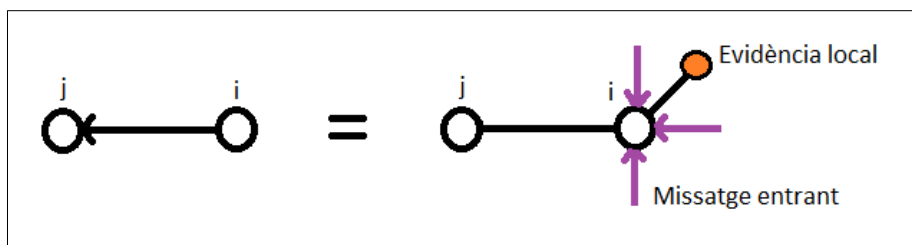


Fig. 8 Propagació de missatge del píxel 'i' cap al píxel 'j'.

Per a enviar un missatge, multipliquem tots els missatges entrants excepte el missatge que prové del node receptor i , tot seguit, multipliquem el resultat de la multiplicació anterior per la matriu de propagació i per l'evidència local.

$$M_{ij} = \phi_i * \psi_{ij} * \prod_{k \in N_i - j} M_{ki}$$

On M_{ij} fa referència al missatge generat pel node ' i ' per al node ' j ' i la multiplicació es fa terme a terme.

Donat que volem trobar l'assignació d'etiquetes més probable per al nostre model gràfic, fem servir la variant de l'algoritme anomenada *Max-product belief propagation* que ens permet buscar l'estat global més probable per al nostre problema. Per això, únicament transmetem el màxim de M_{ij} .

$$M_{ij} = \max(\phi_i * \psi_{ij} * \prod_{k \in N_i - j} M_{ki})$$

El missatge M_{ij} emès per un node conté dues probabilitats: les probabilitats de ser pell i *no-pell* del node receptor.

A partir dels missatges rebuts i de les evidències locals, cada node calcula les seves probabilitats de ser pell i *no-pell*. Anomenem *belief* al vector que conté les probabilitats anteriors. Remarquem que els *beliefs* es calculen a partir de les evidències locals i els missatges entrants de cada node.

$$b_i = \phi_i * \psi_i * \prod_{j \in N_i} M_{ij}$$

On b_i fa referència al *belief* del node ' i ' i la multiplicació es fa terme a terme.

A partir d'aquí, l'enviament de missatges es pot repetir fins que es satisfaci una condició de finalització determinada.

Al final de cada iteració de l'algoritme podem inferir l'etiquetatge de cada node a partir dels seus *beliefs*. Recordem que el *belief* d'un node està compost per la seva probabilitat de ser pell i la de ser *no-pell*. Per aquest motiu, realitzem la inferència comparant els *beliefs* de cada node. Si la probabilitat de ser pell és superior a la de ser *no-pell*, suposem que es tracta d'un píxel de pell. En cas contrari, suposem que es tracta d'un píxel de *no-pell*.

Finalment, podem resumir els passos de l'algoritme *Belief propagation* de la següent manera:

```

Per a cada node
    Calcula l'evidència local del node
Fi Per
Per a cada iteració
    Calcula els missatges enviats per a cada node cap als seus veïns

```

```
Envia els missatges
Calcula els beliefs de cada node
Infereix l'etiquetatge de cada node
Fi Per
```

3.1.5. Observacions i evidències locals

Fins ara hem vist què són els CAM, com ens permeten tenir en compte les dependències contextuais i la seva aplicació dins del projecte. Hem vist que els emplaçaments dels CAM tenen, en el nostre cas, dues observacions i que podem deduir les evidències locals dels emplaçaments a partir d'aquestes últimes. Però a què fan referència aquestes observacions? I com deduïm les evidències locals a partir d'aquestes últimes?

Les observacions dels nostres emplaçaments corresponen a la mitjana⁴ i la desviació típica⁵ de l'emplaçament. Per calcular les observacions d'un emplaçament, tenim en compte els valors dels emplaçaments que l'envolten. En el nostre cas particular, tenim en compte 25 píxels per tal de determinar la mitjana i la desviació típica de cada píxel. Els passos que seguim per dur a terme els càlculs anteriors són:

- Generar blocs de 5*5 píxels i associar un bloc de 5*5 píxels a cada píxel de la imatge.
- Calcular la mitjana del bloc de cada píxel. El resultat correspon a la mitjana del píxel associat al bloc.
- Calcular la desviació típica del bloc de cada píxel. El resultat correspon a la desviació típica del píxel associat al bloc.

D'aquesta manera generem les observacions de cada píxel del nostre CAM (Fig. 9).

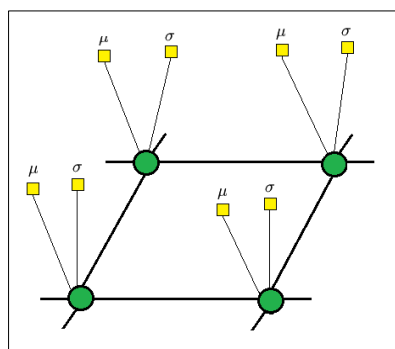


Fig. 9 CAM on els píxels estan connectats a la seva mitjana μ i a la seva desviació típica σ .

⁴ La mitjana simbolitza el valor central d'una distribució.

⁵ La desviació típica mesura la dispersió de la distribució, és a dir, la mitjana de llunyania dels punts de mostra respecte de la mitjana.

Les observacions ens han de permetre determinar les probabilitats de cada píxel de ser pell i *no-pell*. Per això, necessitem conèixer les distribucions de probabilitat de pell i *no-pell*. Recordem que una distribució de probabilitat assigna una probabilitat a un succés. A l'apartat 3.2, s'explica com trobar les distribucions anteriors i les decisions preses a l'hora de modelar-les. De moment, assumim que modelem la pell mitjançant una funció gaussiana i que coneixem la funció de densitat de probabilitat (f.d.p.)⁶ de *no-pell* (Fig. 10).

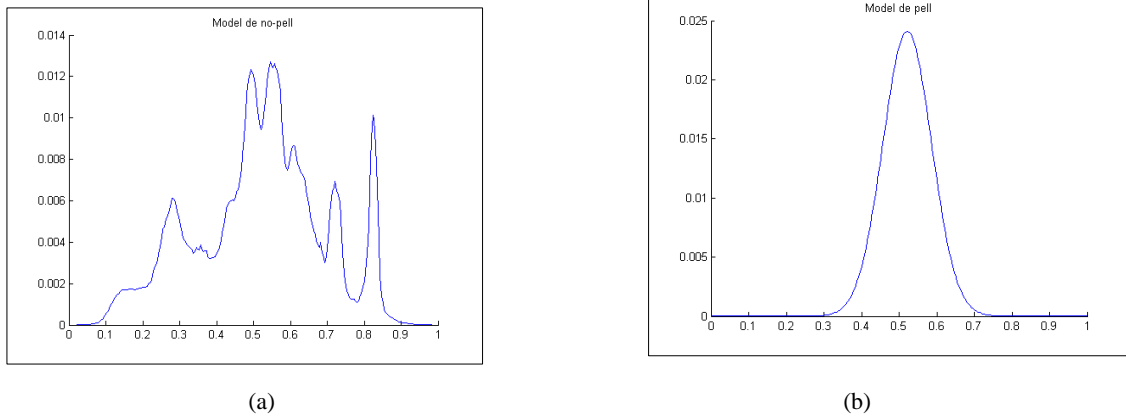


Fig. 10 (a) Model de no-pell. (b) Model de pell.

A partir dels dos models anteriors i de les observacions, podem determinar la probabilitat de que un píxel sigui de pell i la de que sigui de *no-pell*. Començarem explicant com determinar la probabilitat de que un píxel sigui de pell i, a continuació, explicarem com determinar la probabilitat de que sigui de *no-pell*.

Per calcular la probabilitat de que un píxel sigui de pell disposem de la mitjana del píxel μ_{pixel} , la desviació típica del píxel σ_{pixel} i de la distribució gaussiana que modela la pell. Les distribucions gaussianes estan definides per dos paràmetres: la seva mitjana μ i la seva desviació típica σ . Per tant, també disposem dels paràmetres μ_{pell} i σ_{pell} . D'altra banda, cal destacar que μ_{pixel} i σ_{pixel} modelen una segona gaussiana. A partir d'aquesta informació, la probabilitat de ser un píxel de pell ve determinada per l'àrea d'intersecció entre ambdues gaussianes (Fig. 11). En el nostre cas particular, fem una aproximació d'aquesta àrea.

⁶ Recordem que una f.d.p. és una funció que representa una distribució de probabilitat i que descriu la densitat de la probabilitat a cada punt de l'espai.

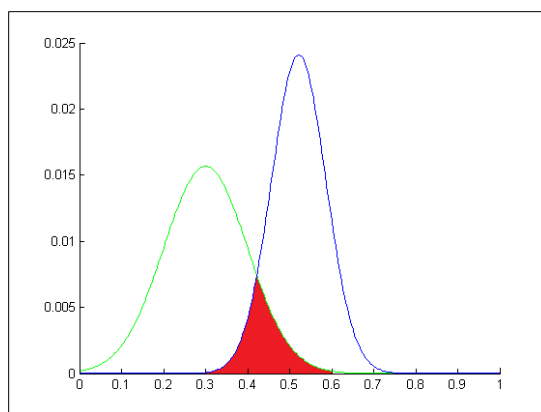


Fig. 11 Probabilitat de pell d'un píxel.

De la mateixa manera, per calcular la probabilitat de *no-pell* d'un píxel, calculem l'àrea d'intersecció entre la funció que modela el píxel a classificar i la que modela tot el que és *no-pell* (Fig. 12). En aquest cas disposem de la mitjana del píxel μ_{pixel} , la desviació típica del píxel σ_{pixel} i de la f.d.p de *no-pell*. Cal recordar que μ_{pixel} i σ_{pixel} modelen una gaussiana. En aquest cas també fem una aproximació de l'àrea entre ambdues funcions.

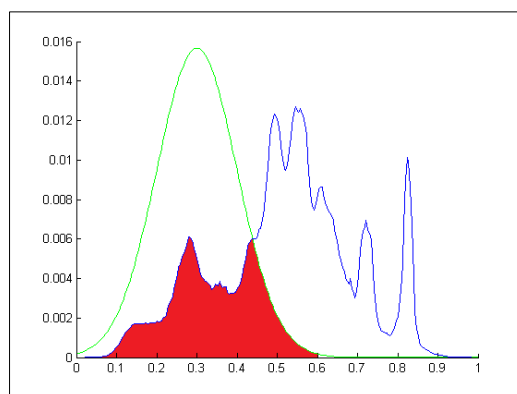


Fig. 12 Probabilitat de no-pell d'un píxel.

Ara sabem calcular la probabilitat de ser pell P_{pell} i la probabilitat de ser *no-pell* $P_{no-pell}$ de cada píxel. Com hem vist, aquestes probabilitats es dedueixen a partir les distribucions de pell i *no-pell* i de les observacions de cada píxel (μ_{pixel} i/o σ_{pixel}). A partir d'aquest últim càlcul, ens preguntem com trobar les distribucions de probabilitat anteriors.

3.2. Aprenentatge

En aquest apartat exposarem com trobar les distribucions de probabilitat que modelen els píxels de pell i de *no-pell*. Per tal de trobar les distribucions de probabilitat anteriors duem a terme un aprenentatge supervisat. En un aprenentatge supervisat, el conjunt d'entrenament està format pel conjunt de mostres i el seu corresponent valor desitjat, és a dir, la classe a la qual pertany cada mostra. Disposem d'un conjunt de 346 imatges amb condicions d'il·luminació variants i en les quals apareixen 27 cares diferents. Per generar el conjunt d'entrenament, seleccionem dues regions de pell i *no-pell* successivament de 63 imatges. La resta d'imatges ens serveixen únicament per la classificació i la validació dels resultats.

3.2.1. Model de pell

Fins aquest punt hem assumit que modelem la pell mitjançant una funció gaussiana. En aquest apartat es justifica per què fem servir una gaussiana i s'exposa com determinar els paràmetres⁷ d'aquesta per tal de modelar la pell.

Si extraiem els valors d'un conjunt de píxels de pell, podem observar que es concentren al voltant d'un punt (*Fig. 13*). A més, cal recordar que una funció gaussiana descriu com es dispersen els punts al voltant d'una mitja.

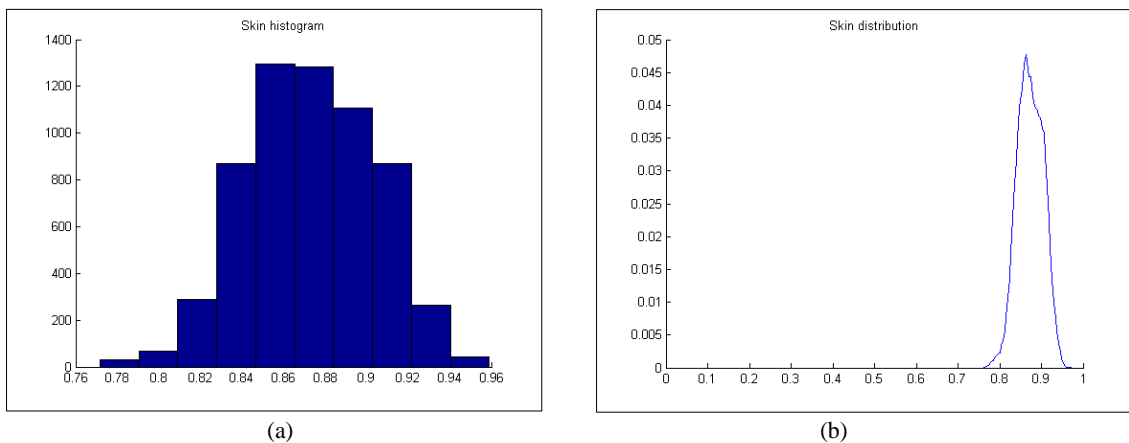


Fig. 13 (a) Histograma de pell (b) Funció de densitat de pell.

Per tant, una funció de densitat gaussiana ens permet modelar adequadament la pell (*Fig. 14*). A continuació, veurem com determinar els paràmetres de la gaussiana que modela la pell.

⁷ Recordem que els paràmetres de la funció de densitat gaussiana són μ i σ , on μ és la mitjana i σ és la desviació típica de la distribució.

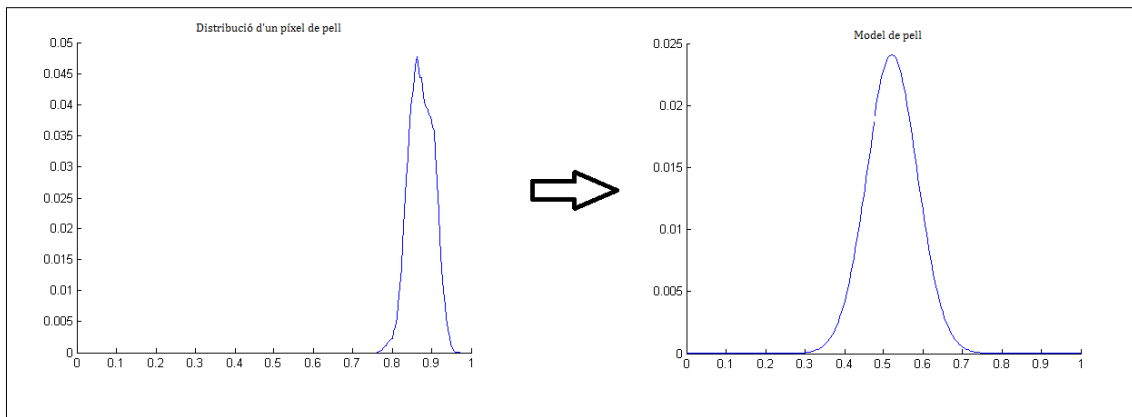


Fig. 14 Modelatge de la pell.

Com hem vist a la introducció, la majoria de models de color existents depenen de les condicions d'adquisició de la imatge. L'article d'Álvarez, Gevers i López [2] proposa un mètode per aconseguir invariància de color respecte les condicions d'adquisició d'una imatge. El mètode es basa en l'anàlisi de la correlació entre diferents espais de color per aconseguir ser invariant a les variacions de llum, és a dir, combina i dona pesos a diferents espais de color per obtenir un model de color invariant. En general, combinar múltiples classificadors (descriptors de color) és una bona tècnica per millorar el rendiment de classificadors individuals [2]. Per tal d'aconseguir aquesta invariància, el mètode té en compte múltiples vistes de l'objecte a classificar, en el nostre cas, té en compte múltiples vistes de diverses cares per poder modelar la pell. Utilitzem el terme vista per fer referència a unes determinades condicions d'adquisició de la imatge [2].

El mètode modela la pell mitjançant la seva mitjana μ i la seva desviació típica σ i assigna pesos a cada component de cada espai de color. Per modelar la pell, seleccionem regions de cara sota diferents condicions (diferents vistes). Cada regió seleccionada conté múltiples píxels. A continuació, separem els valors dels píxels de la regió seleccionada en els diferents canals de color. En el nostre cas particular, utilitzem els següents 12 canals de color: R, G, B de l'espai RGB; r, g de l'espai RGB normalitzat; O1, O2 de l'espai oponent; L, a, b de l'espai Lab i H, S de l'espai HSV. Després, calculem la mitjana de cada canal, és a dir, donat el valor de cada píxel per a cada canal, calculem la mitjana de cada canal per obtenir informació referent a una vista determinada. Repetim el procés amb diferents regions de diferents vistes.

A partir dels valors dels canals de color obtinguts sota diferents vistes i assumint que cada vista té la mateixa probabilitat d'aparèixer [2], modelem la pell. Una combinació adequada dels canals de color sota diferents vistes genera un model per al qual la variància⁸ de la pell es minimitza [2]. Per tant, la combinació anterior disminueix les desviacions degudes a les

⁸ La variància és una mesura de dispersió d'una variable respecte el seu valor mitjà.

condicions d'adquisició variants respecte el valor esperat (μ). Existeixen tècniques d'optimització que proporcionen solucions òptimes per minimitzar la variància d'una distribució. La tècnica utilitzada a l'article s'anomena frontera eficient⁹. La frontera eficient conté diferents conjunts de valors formats per una mitjana de l'objecte a classificar (la pell) i el conjunt de pesos associats als components (canals de color) per als quals es minimitza la variància del darrer objecte [2]. El conjunt més apropiat de la frontera eficient és el que conté la ràtio mitjana - variància més elevada ja que proporciona el rendiment més alt [2]. Els pesos pertanyents al darrer conjunt s'utilitzaran, posteriorment, per mantenir la invariància respecte les condicions d'il·luminació durant el modelatge de *no-pell* i la classificació de píxels.

Per tant, aplicant la frontera eficient, trobem la mitjana i la variància que modelen la pell així com els pesos que s'han d'aplicar als diferents components de color per tal d'obtenir la mitjana i la variància anteriors. Notem que a partir de la variància és fàcil obtenir la desviació típica ja que la primera és el quadrat de la segona.

A la Fig. 15, podem trobar un resum del procés d'obtenció de la mitjana μ , la desviació típica σ i el conjunt de pesos ω de l'etapa d'aprenentatge del model de pell.

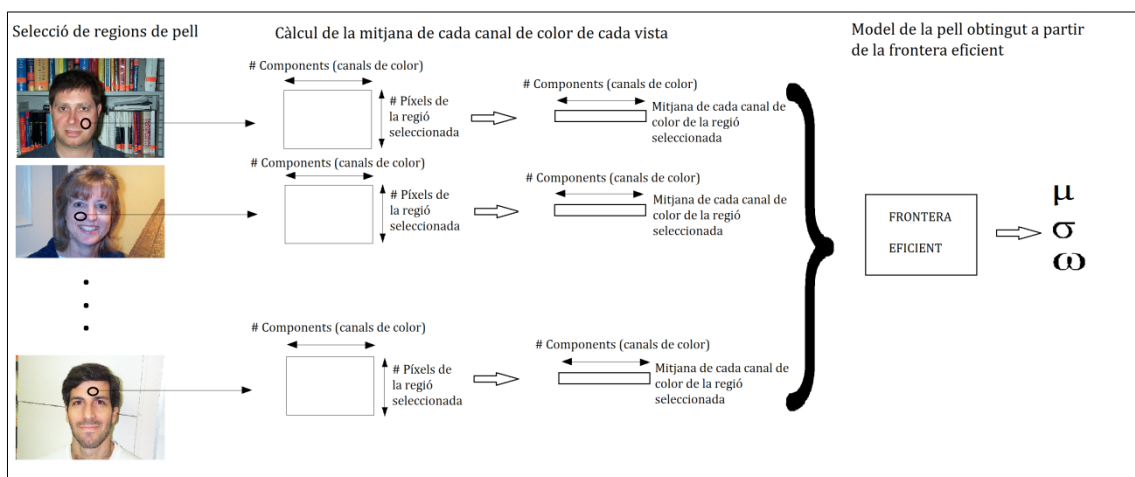


Fig. 15 Procés per obtenir el model de pell.

μ : mitjana de la pell

σ : desviació típica de la pell

ω : pesos associats a cada canal de color.

⁹ Donat un conjunt de components (canals de color), és possible combinar-los formant un nombre infinit de solucions (models de color diferents).

3.2.2. Model de no-pell

Fins aquest punt hem assumit que utilitzem una funció de densitat de probabilitat (f.d.p.) per determinar la probabilitat de *no-pell* de cada píxel. Cal remarcar que els píxels de *no-pell* poden representar una gran varietat d'objectes i d'espais sense característiques comunes. En conseqüència, considerem inviable obtenir un model de *no-pell* suficientment acurat per representar aquesta diversitat. Com hem esmentat a 3.2.1, a l'hora de buscar un model de *no-pell*, hem de tenir en compte els pesos ω obtinguts durant el modelatge de la pell per tal de mantenir la invariància respecte les condicions d'il·luminació. Per això, primer transformem les imatges d'entrada aplicant els darrers pesos. Recordem que un dels avantatges del mètode de Álvarez, Gevers i López [2] era el de treballar en un espai d'una única dimensió. Per tant, les imatges transformades seran imatges en tons de gris. A continuació, seleccionem regions de *no-pell* de diferents imatges transformades. A partir dels valors d'aquestes mostres, busquem una f.d.p. que les modeli (Fig. 16, Fig. 17).

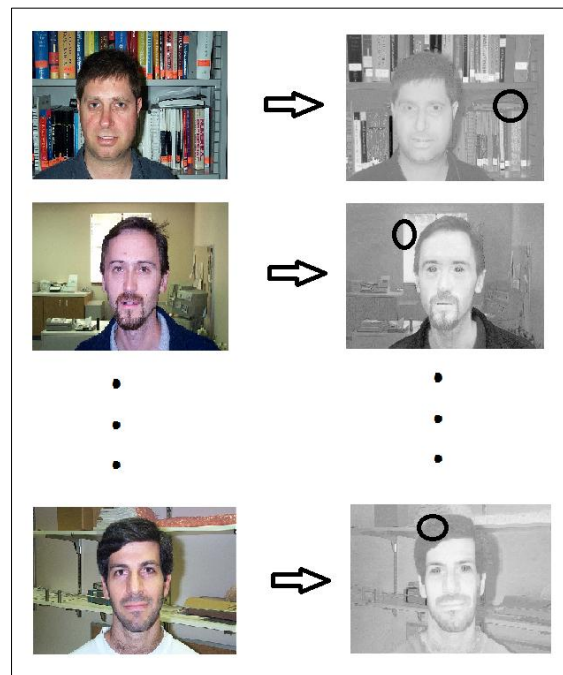


Fig. 16 Transformació imatge i selecció de regions de *no-pell*.

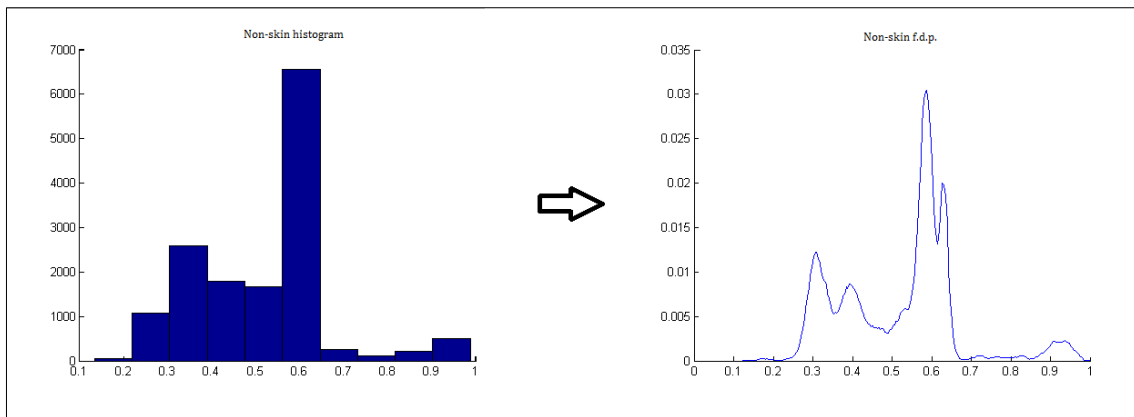


Fig. 17 Modelatge de no-pell.

Evidentment, les mostres seleccionades no poden contemplar de manera exhaustiva tot el que és *no-pell*.

3.3. Esquema general

Finalment, com a resum, veurem un mapa conceptual de tot el que hem exposat en aquest capítol (Fig. 18).

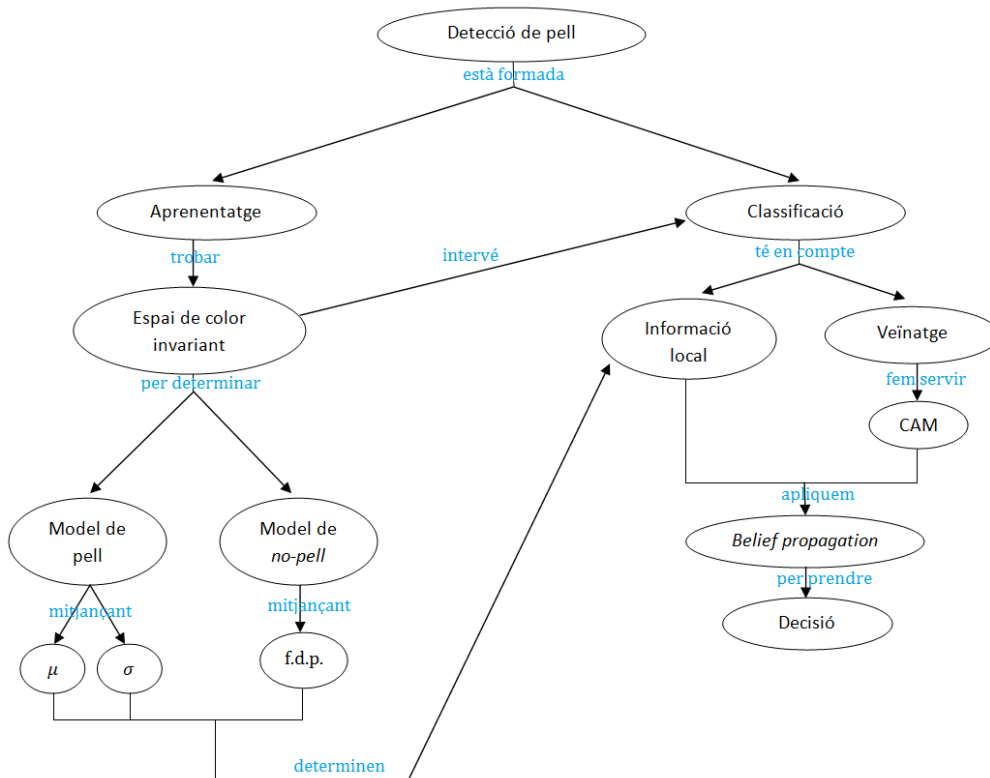


Fig. 18 Mapa conceptual de l'algorisme de detecció de pell proposat.

4. Disseny

En aquest capítol definirem l'esquema general del projecte a nivell de disseny i exposarem el seu desenvolupament. L'objectiu d'aquest capítol és obtenir un disseny que ens faciliti la implementació del projecte.

4.1. Disseny general

En aquest apartat veurem com dividir el projecte en diferents tasques per tal de simplificar-ne la implementació. Per començar, separarem el projecte en dos grans blocs: aprenentatge i classificació. L'aprenentatge es du a terme una única vegada i el seu resultat s'utilitza durant la classificació. En canvi, la classificació es du a terme tantes vegades com imatges vulguem classificar. Per tal d'avaluar els resultats obtinguts a l'acabar la classificació, necessitem un tercer bloc encarregat de calcular valors significatius per validar la classificació.

Posteriorment, a l'apartat 4.2, dividirem els blocs esmentats en diferents mòduls i estudiarem la interacció entre aquests darrers.

4.1.1. Disseny general del bloc d'aprenentatge

El reconeixement de cares pot ser vist com un problema de classificació binari, és a dir, de dues possibles classes: pell i *no-pell*. El bloc corresponent a l'aprenentatge ha de trobar les f.d.p. de les dues classes del nostre problema. Com hem vist al capítol anterior, utilitzem una funció gaussiana per modelar la pell. Recordem que una funció gaussiana ve determinada per dos paràmetres: la mitjana μ i la desviació típica σ de la distribució. Per tant, tenir la mitjana de la pell μ_{pell} i la seva desviació σ_{pell} equival a tenir la funció gaussiana que la modela. D'altra banda, necessitem la f.d.p. de *no-pell* i els pesos ω per poder dur a terme la classificació. Per tant, l'aprenentatge ha de ser capaç de retornar-nos μ_{pell} , σ_{pell} , ω i la f.d.p. de *no-pell* $f_{dp_{nopell}}$, a partir del conjunt d'imatges de l'entrenament (Fig. 19).

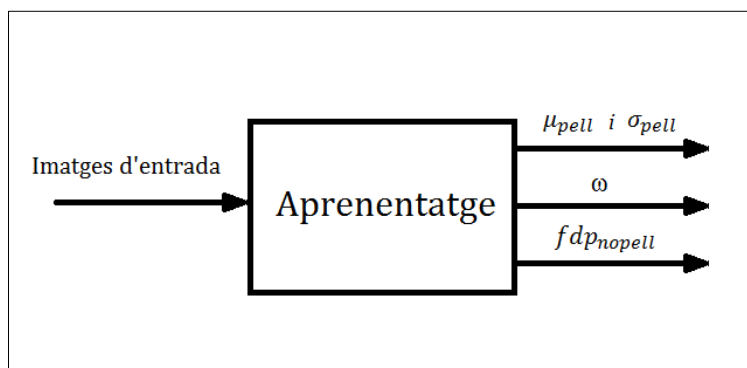


Fig. 19 Diagrama d'aprenentatge.

4.1.2. Disseny general del bloc de classificació

El bloc corresponent a la classificació és l'encarregat de determinar quins píxels de la imatge són píxels de pell i quins són de *no-pell*. Com hem vist al capítol 2, el sistema de detecció de pell ha de ser capaç de, donada una imatge d'entrada, retornar una imatge binària en la qual les regions blanques corresponguin als píxels de pell i les regions negres als píxels de *no-pell*. El bloc de classificació ha de retornar-nos la darrera imatge binària a partir de les dades retornades per l'aprenentatge (μ_{pell} , σ_{pell} , ω i fdp_{nozell}) i la imatge que volem classificar (Fig. 20).

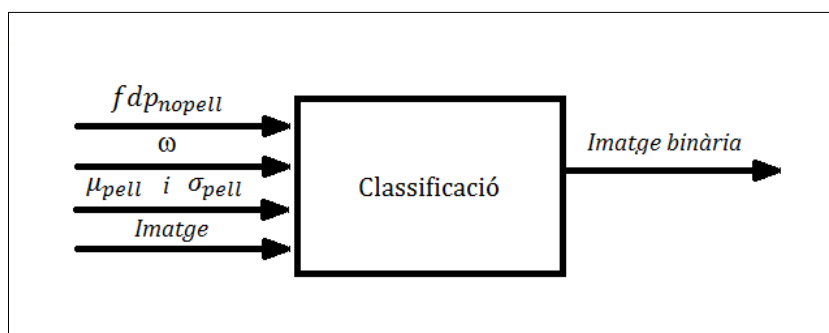


Fig. 20 Diagrama de classificació.

4.1.3. Disseny general del bloc avaluador

El bloc avaluador s'encarrega de valorar els resultats de la classificació. Per a cada imatge a classificar disposem d'una imatge binària corresponent al resultat de classificació esperat, és a dir, una imatge binària que conté informació sobre la ubicació de regions de pell dins de la imatge a classificar (Fig. 21). Anomenem la darrera imatge màscara. El bloc avaluador compara la imatge binària resultant de la classificació amb la màscara de la imatge original i retorna mesures quantitatives per avaluar els resultats de la classificació (Fig. 22).



Fig. 21 Imatge a classificar i màscara de la imatge a classificar.

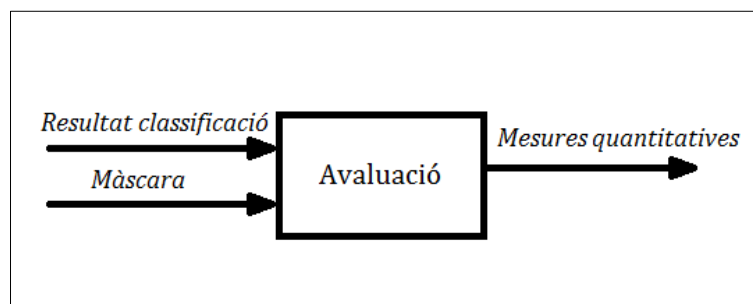


Fig. 22 Diagrama d'avaluació.

Les mesures quantitatives que utilitzem per avaluar la classificació són la precisió, la taxa de detecció i l'efectivitat. La precisió mesura la probabilitat de que el resultat sigui vàlid, és a dir, la ràtio entre el nombre de píxels de pell correctament classificats i el nombre de píxels de pell trobats [2]. Aquesta mesura ens indica si hi ha molts falsos positius, és a dir, si hi ha molts píxels de *no-pell* classificats com a píxels de pell. Com més elevada és la precisió, menys falsos positius tenim. D'altra banda, la taxa de detecció mesura la ràtio entre els píxels de pell correctament classificats i el nombre de píxels de pell reals. Per tant, aquest mesura ens indica si hi ha molts falsos negatius, és a dir, molts píxels de pell classificats com a *no-pell*. Com més elevada és la taxa de detecció, menys falsos negatiu tenim i, per tant, més bona és la classificació. Finalment, l'efectivitat és una mesura que compara les dues mesures anteriors [2].

4.2. Etapes

A l'apartat anterior, hem separat el projecte en tres blocs. El tercer bloc, el bloc d'avaluació, està format únicament per un mòdul que s'encarrega de generar les mesures quantitatives esmentades anteriorment a partir de dues imatges d'entrada (la imatge resultant de la

classificació i la màscara de la imatge original). Per aquest motiu, en aquest apartat ens centrarem únicament en el disseny més detallat dels blocs d'aprenentatge i classificació.

4.2.1. Etapes de l'aprenentatge

L'aprenentatge del nostre projecte es divideix en dues etapes: l'aprenentatge del model de pell i l'aprenentatge del model de *no-pell* (Fig. 23). Com hem vist al capítol anterior, l'aprenentatge del model de pell determina la mitjana μ i la desviació típica σ de la pell, així com els pesos ω que permeten aconseguir invariància respecte les condicions d'adquisició de la imatge. D'altra banda, l'aprenentatge del model de *no-pell* determina la f.d.p. de *no-pell*. Recordem que els pesos ω retornats per l'aprenentatge del model de pell també intervenen en l'aprenentatge del model de *no-pell*.

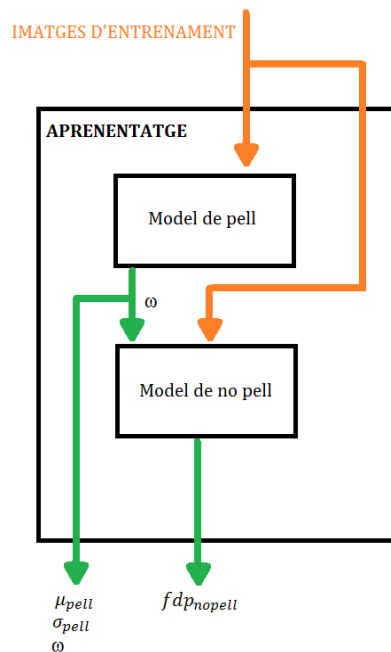


Fig. 23 Diagrama detallat d'aprenentatge.

A continuació, analitzarem pas a pas el procés de modelatge de pell mitjançant dos diagrames de flux. En el primer diagrama, veurem com extreure les dades de les imatges d'entrada per generar el conjunt de dades d'aprenentatge de pell. Posteriorment, en el segon diagrama, veurem com aprendre el model de pell i els pesos esmentats anteriorment. Recordem que per modelar la pell necessitem aprendre la seva mitjana μ_{pell} i la seva desviació típica σ_{pell} .

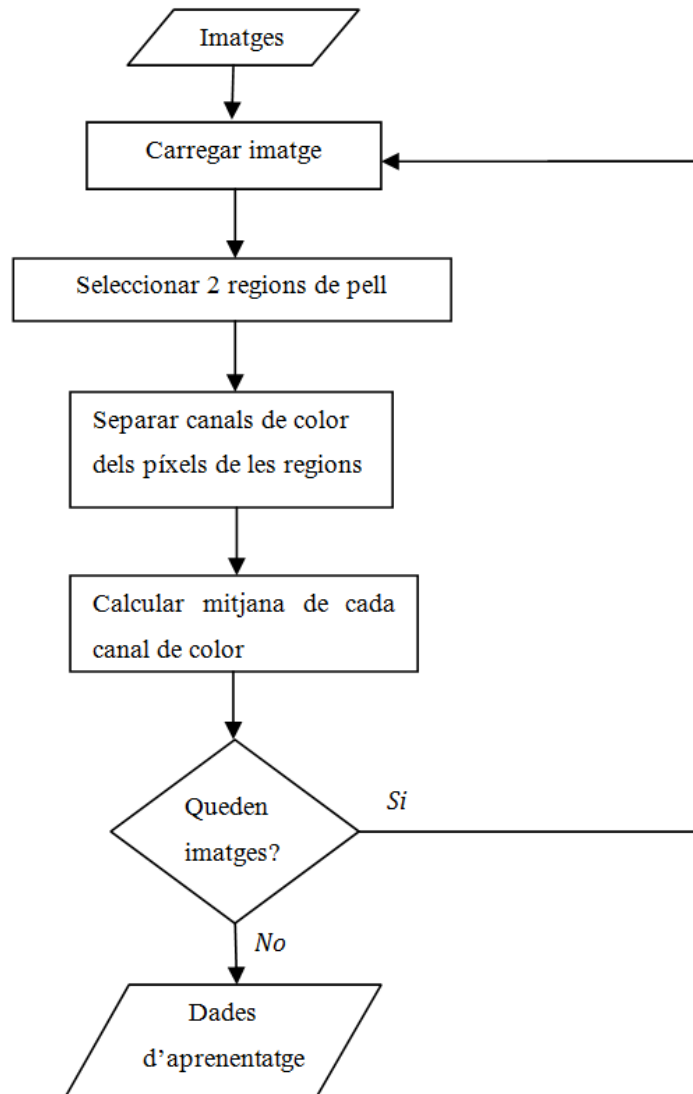


Fig. 24 Diagrama de flux de la preparació de dades de l'aprenentatge del model de pell.

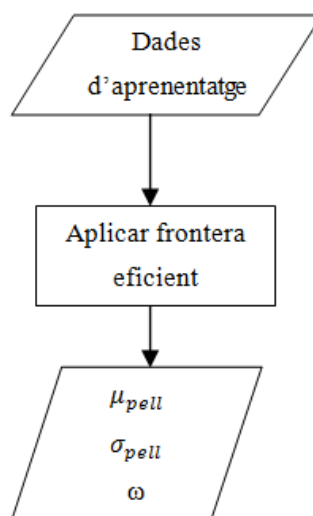


Fig. 25 Diagrama de flux de l'aprenentatge del model de pell.

A la Fig. 24, podem veure com generem les dades per determinar el model de pell. Per a cada imatge del conjunt d'aprenentatge, primer carreguem la informació referent als seus píxels. A continuació, seleccionem dues regions de pell i separem els canals de color dels píxels de les regions seleccionades. Finalment, calculem la mitjana de cada canal de color. Un cop tenim les mitjanes de cada canal de color de totes les regions seleccionades, entrem les dades a la frontera eficient (Fig. 25) que ens retorna la mitjana de la pell μ_{pell} , la desviació típica de la pell σ_{pell} i els pesos ω esmentats anteriorment.

Com hem vist al principi de l'apartat, l'aprenentatge es divideix en dues etapes. Tot seguit, analitzarem pas a pas el procés de modelatge de *no-pell* mitjançant dos diagrames de flux. En el primer diagrama, veurem com extreure les dades de les imatges d'entrada per generar el conjunt de dades d'aprenentatge de *no-pell*. Posteriorment, en el segon diagrama, veurem com aprendre el model de *no-pell*. Recordem que per modelar tot el que és *no-pell* necessitem aprendre la $f dp_{nopell}$.

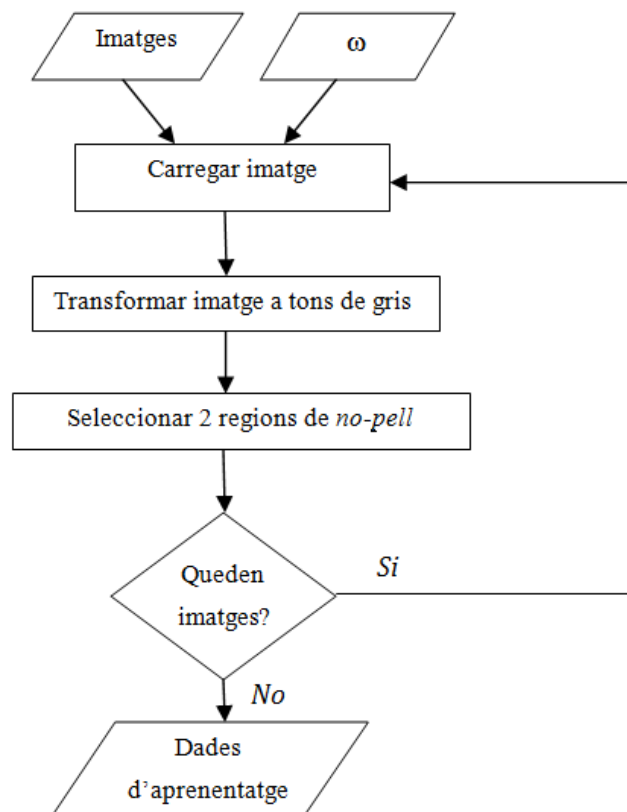


Fig. 26 Diagrama de flux de la preparació de dades de l'aprenentatge del model de *no-pell*.

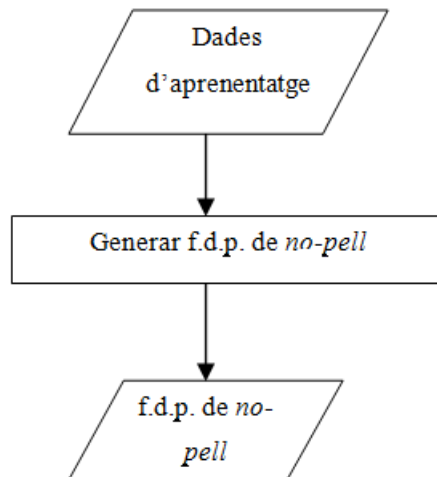


Fig. 27 Diagrama de flux de l'aprenentatge del model de no-pell.

A la Fig. 26 podem veure com generem les dades per determinar el model de no-pell. Primer carreguem la imatge i la transformem en una imatge de tons de gris mitjançant els pesos ω obtinguts durant l'aprenentatge del model de pell. Tot seguit, seleccionem dues regions de no-pell i, un cop preparades les dades de les regions seleccionades del conjunt d'aprenentatge, determinem la f.d.p. de no-pell (Fig. 27).

4.2.2. Etapes de la classificació

La classificació del nostre projecte es divideix en dues etapes: la generació de dades d'entrada del CAM i la classificació dels píxels (Fig. 28). Com hem vist al capítol anterior, la classificació ha de tenir en compte el veïnatge. Per tant, aquesta segona etapa és l'encarregada de crear el CAM i d'aplicar l'algoritme *Belief propagation* per determinar les probabilitats de cada píxel de ser de pell i no-pell. A l'hora de crear el CAM necessitem conèixer les observacions dels píxels, és a dir, la mitjana i la desviació típica de cadascun d'ells. Recordem que el resultat de la classificació és una imatge binària en la qual els píxels blancs corresponen a les regions de pell i els píxels negres a les de no-pell.

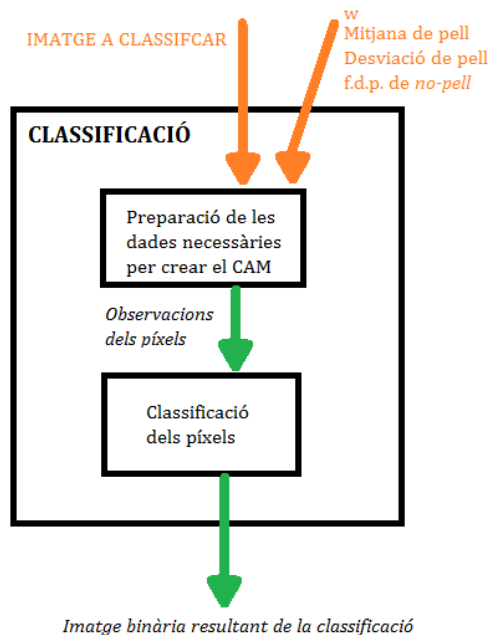


Fig. 28 Diagrama detallat de classificació.

A continuació, analitzarem pas a pas el procés de classificació mitjançant dos diagrames de flux. Primerament, veurem com calcular les observacions de cada píxel per, posteriorment, crear el CAM i, tot seguit, veurem la creació d'aquest darrer i la resta d'etapes de classificació.

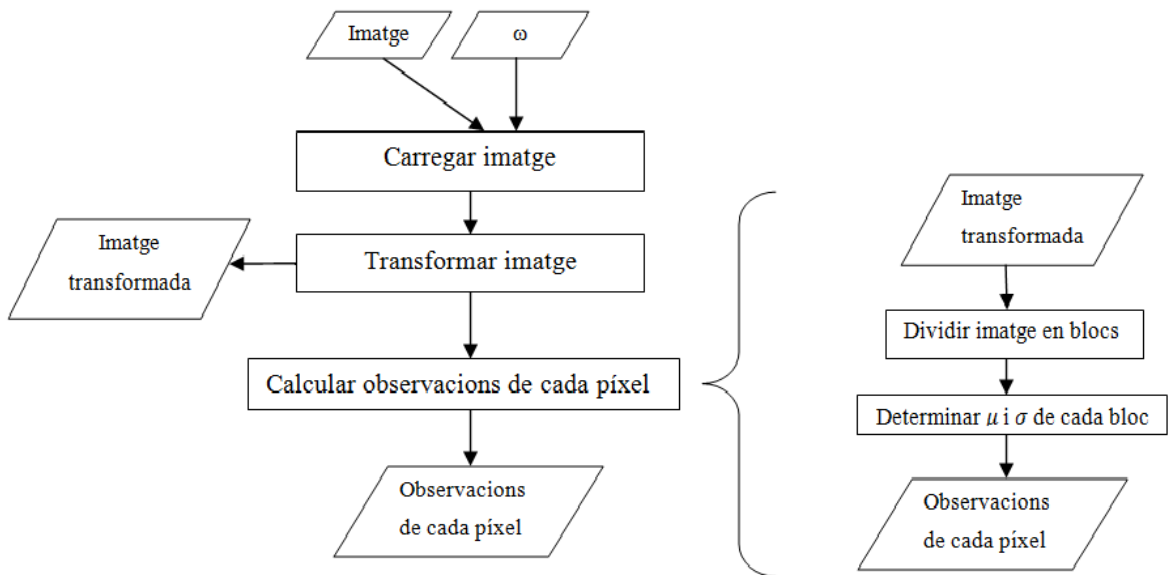


Fig. 29 Diagrama de flux de preparació de dades per crear el CAM.

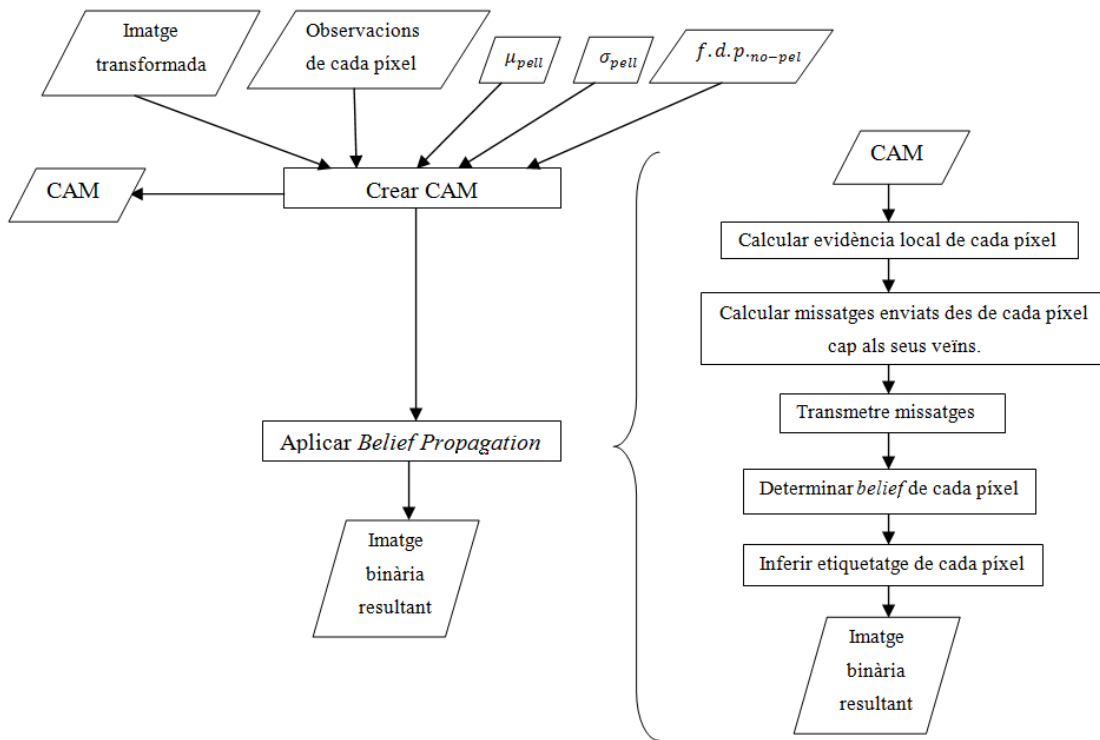


Fig. 30 Diagrama de flux de la classificació de píxels.

A la Fig. 29, podem veure com preparam les dades per crear un CAM. Després de carregar la imatge i de transformar-la donant pesos als diferents canals de color, calculem les observacions de cada node del CAM, en el nostre cas particular, les observacions de cada píxel. Com hem vist a 3.1.5, les observacions corresponen a la mitjana i la desviació típica dels píxels. Per determinar aquestes observacions, tenim en compte els valors dels píxels veïns. Per tant, generem blocs de 25 píxels per a cada píxel de la imatge i calculem la mitjana i la desviació típica de cada bloc. A partir d'aquestes observacions, passem a crear el CAM. Finalment, apliquem l'algoritme *Belief propagation* per obtenir la imatge binària resultant de la classificació.

Com podem veure a la Fig. 30, a l'hora d'aplicar el *Belief propagation*, primer determinem les evidències locals de cada píxel. Tot seguit, calculem els missatges que han d'enviar els píxels als seus veïns, els transmetem, determinem els *beliefs* de cada píxel a partir dels missatges entrants i les evidències locals i, finalment, inferim l'etiquetatge dels píxels a partir dels seus *beliefs*. Si la probabilitat de pell supera la probabilitat de *no-pell*, suposem que es tracta d'un píxel de pell. En cas contrari, suposem que es tracta d'un píxel de *no-pell*. L'etiquetatge dels píxels ens genera la imatge binària resultant de la classificació.

5. Resultats

L'objectiu d'aquest capítol és presentar els resultats obtinguts i comparar-los amb els ja existents per tal de veure si el fet de tenir en compte el veïnatge millora la detecció de pell.

Aplicarem el procediment exposat al capítol 0 a la base de dades de cares Caltech [6] per obtenir els resultats del nostre projecte. Aquesta base de dades conté imatges de diferents persones preses sota diferents il·luminacions i diferents escenaris (interiors i exteriors).

Cal destacar que existeixen diversos paràmetres susceptibles de variar els resultats de la detecció de pell durant la classificació de píxels. Estudiarem la influència d'aquests darrers paràmetres mitjançant les proves descrites al següent apartat.

5.1. Proves

En aquest apartat veurem les proves realitzades per obtenir millors resultats i estudiarem la influència de determinats paràmetres sobre els possibles resultats.

5.1.1. Veïnatge

Fins ara hem estat treballant amb un veïnatge de 4 per a cada píxel. No obstant, pot ser interessant veure com varien els resultats de la classificació si augmentem el veïnatge dels píxels. És d'esperar que augmentant el veïnatge dels píxels es consumeixi més memòria i que augmenti el temps necessari per la classificació. Per aquest motiu, haurem d'avaluar si les millores aportades per la modificació d'aquest paràmetre compensen els problemes que en puguin derivar.

A continuació, analitzarem els guanys obtinguts augmentant el veïnatge respecte el consum de memòria i el temps de classificació (*Taula 3, Fig. 31, Fig. 32*).

	Precisió	Taxa de detecció	Efectivitat
4 veïns	0.868 ± 0.12	0.693 ± 0.23	0.753 ± 0.18
8 veïns	0.881 ± 0.15	0.721 ± 0.20	0.773 ± 0.19

Taula 3 Influència del veïnatge sobre els resultats de la classificació.

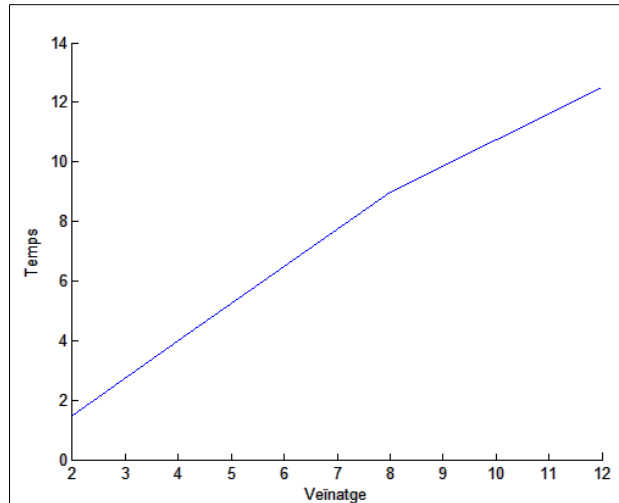


Fig. 31 Augment del temps de classificació en funció del veïnatge.

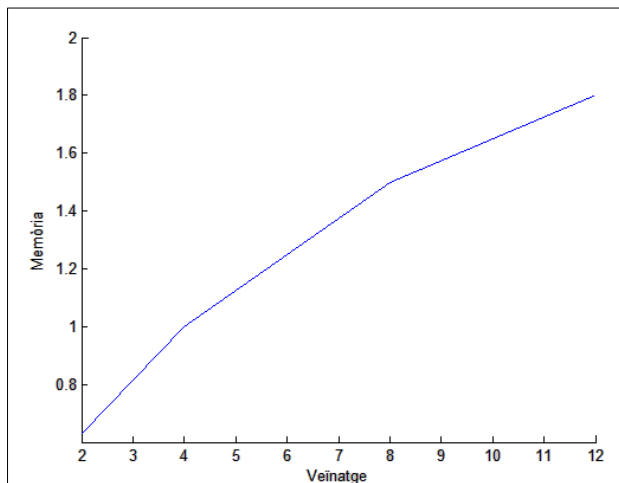


Fig. 32 Augment del consum de memòria en funció del veïnatge.

A partir dels resultats anteriors, observem que a l'incrementar el nombre de veïns d'un píxel, els resultats de classificació milloren sensiblement. Nogensmenys, aquesta millora ve acompanyada d'un increment notable del temps de classificació i de la memòria necessària. Per aquest motiu, utilitzarem únicament quatre veïns per generar els resultats finals del projecte.

5.1.2. Número d'iteracions del Belief Propagation

Com hem vist a l'apartat 3.1.4, el *Belief propagation* és un algoritme iteratiu. Podem determinar el número d'iteracions en funció dels canvis produïts entre dues iteracions consecutives, és a dir, quan els canvis entre dues iteracions consecutives són inferiors a un llindar, l'algoritme s'atura. De la mateixa manera, podem determinar el nombre d'iteracions de forma empírica per tal d'obtenir millors resultats. Donat que volem estudiar la influència del nombre de iteracions de l'algoritme sobre els resultats obtinguts, intentarem determinar-lo

empíricament per tal de millorar els resultats de la classificació. A continuació, analitzarem l'impacte del nombre d'iteracions del *Belief propagation* sobre la classificació de la primera imatge de la base de dades (Fig. 33).

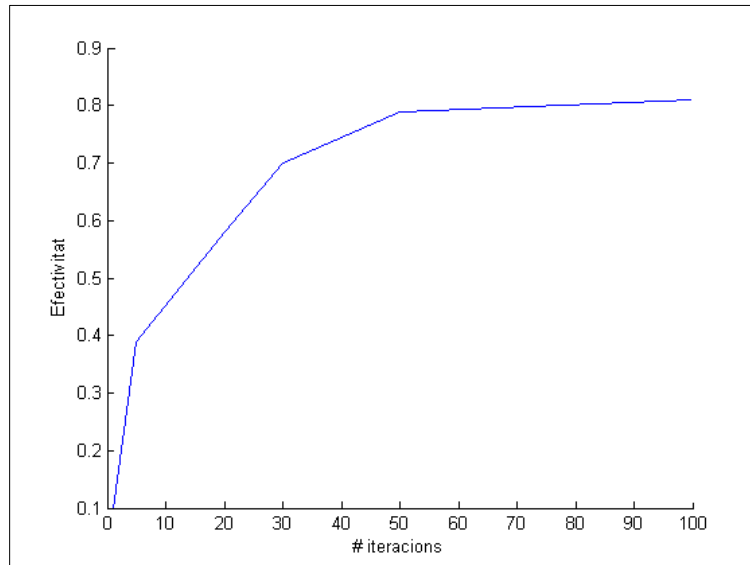


Fig. 33 Augment de l'efectivitat en funció del nombre d'iteracions del *Belief propagation*.

La introducció d'iteracions addicionals al *Belief propagation* millora sensiblement la classificació de píxels. Nogensmenys, cada iteració aporta una millora menys notable respecte l'anterior. A més, cal remarcar que aquest augment d'iteracions comporta inevitablement un augment del temps de classificació requerit (Fig. 34). Per tant, podem determinar un nombre d'iteracions per al qual les millores introduïdes siguin prou significatives i el temps empleat per la classificació sigui raonable.

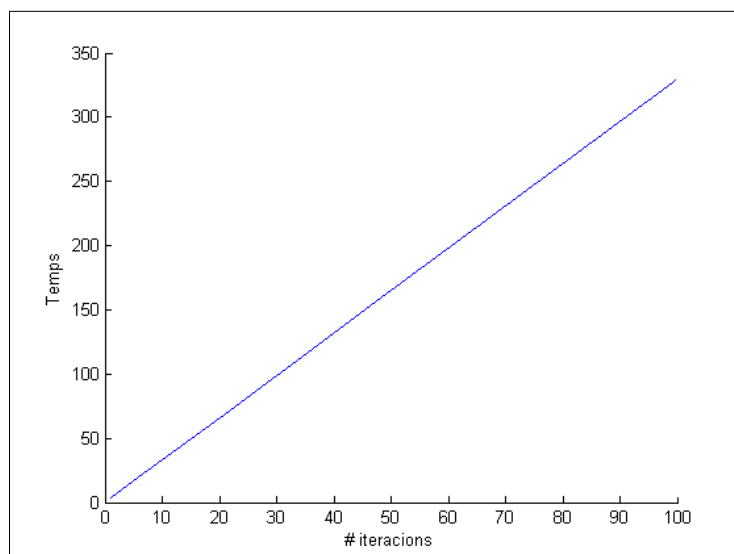


Fig. 34 Augment del temps de classificació en funció del nombre d'iteracions del *Belief propagation*.

5.1.3. Paràmetres de la propagació de missatges

Com hem vist a l'apartat 3.1.4, els nodes del CAM estan connectats a través d'una matriu de propagació ψ_{ij} que defineix la relació entre els estats d'ambdós nodes. Aquesta matriu de propagació s'utilitza a l'hora de calcular els missatges que es transmeten entre nodes. Per tant, podem dir que té una certa influència sobre els resultats obtinguts. En aquesta secció veurem com varien els resultats de la classificació en funció dels valors de la matriu de propagació. A continuació, analitzarem l'exemple de tres matrius de propagació concretes (Taula 4, Fig. 35).

	Precisió	Taxa de detecció	Efectivitat
$\psi_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$	0.868 ± 0.12	0.693 ± 0.23	0.753 ± 0.18
$\psi_2 = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$	0.736 ± 0.12	0.605 ± 0.21	0.654 ± 0.17
$\psi_3 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.71 \end{bmatrix}$	0.58 ± 0.18	0.51 ± 0.11	0.53 ± 0.15

Taula 4 Influència de la matriu de propagació sobre els resultats de la classificació.

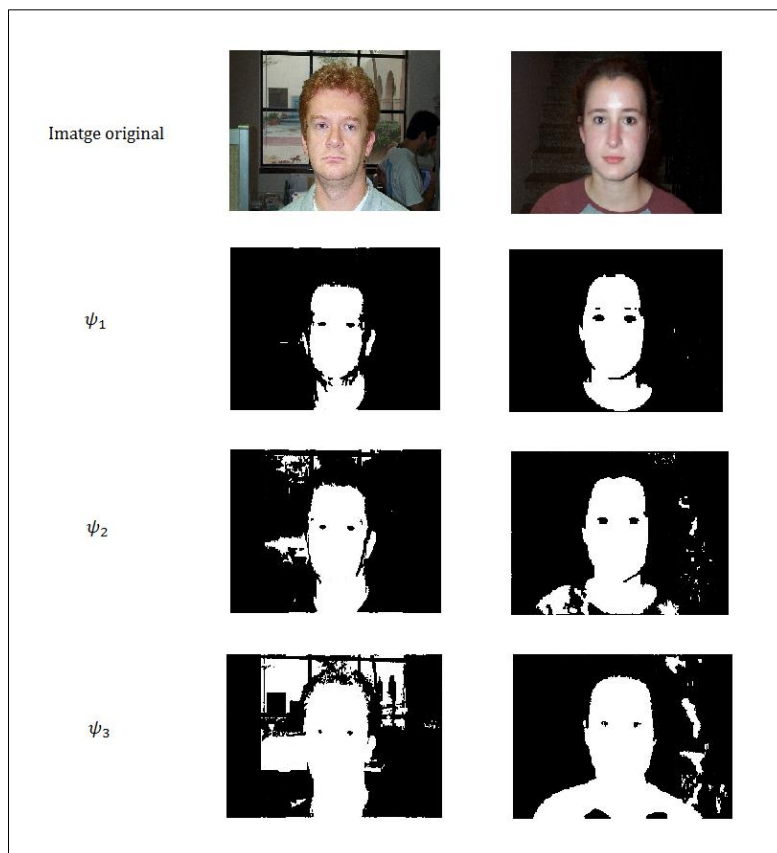


Fig. 35 Detecció de cares per diferents matrius de propagació.

Les regions blanques corresponen als píxels de pell i les regions negres als píxels de no-pell.

Com podem observar a la [Taula 4](#), la matriu de propagació que genera els millors resultats és la matriu $\psi_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$. En conseqüència, utilitzarem la darrera matriu per generar els resultats finals.

5.1.4. Pesos

Fins ara hem estat classificant els píxels tenint en compte uns pesos obtinguts a partir d'un conjunt d'imatges de diferents persones sota diferents vistes. Seria interessant analitzar els resultats de la classificació de les imatges d'una determinada persona utilitzant uns pesos obtinguts a partir d'un conjunt d'imatges d'aquesta mateixa persona. Per fer això, tornem generar les dades d'aprenentatge seleccionant regions de pell de la primera persona de la base de dades. A continuació, classifiquem diverses imatges en les quals apareix aquesta persona sota diferents vistes. A la següent taula exposem els resultats obtinguts:

	Precisió	Taxa de detecció	Efectivitat
Pesos generals	0.868 ± 0.12	0.693 ± 0.23	0.753 ± 0.18
Pesos per persona	0.870 ± 0.13	0.750 ± 0.24	0.777 ± 0.18

Taula 5 Influència dels pesos sobre els resultats de la classificació.

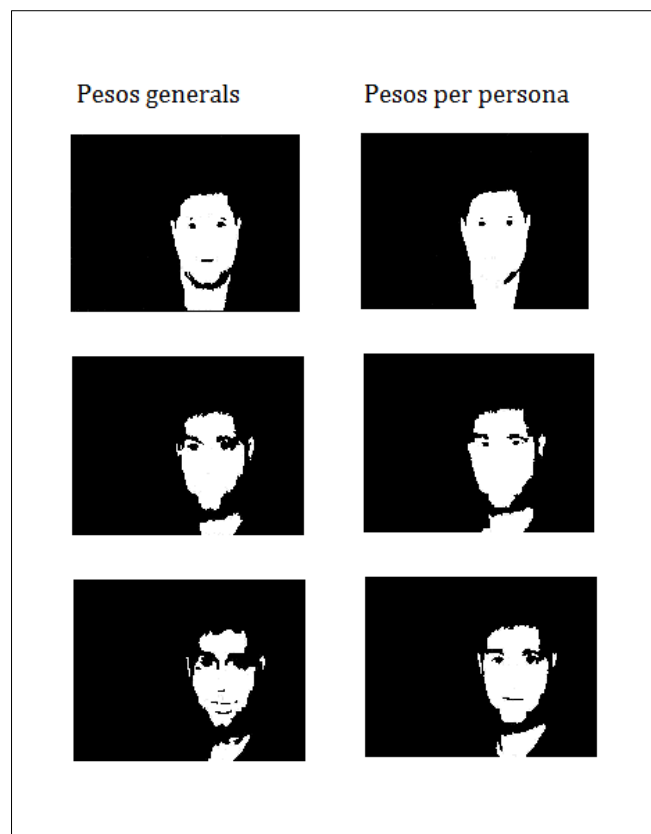


Fig. 36 Detecció de cares amb diferents pesos.

Com podem observar a la *Taula 5* i a la *Fig. 36*, si utilitzem els pesos per persona obtenim millors resultats. No obstant, com que volem classificar imatges de diferents persones, utilitzarem els pesos generals per generar els resultats finals del nostre projecte.

5.2. Resultats finals

En aquest apartat veurem quins són els resultats finals obtinguts. Per obtenir aquests resultats hem tingut en compte un veïnatge de quatre píxels i uns pesos calculats de manera general, és a dir, a partir de diferents vistes d'imatges de diferents persones. A més, hem utilitzat la matriu de propagació ψ_1 de l'apartat 5.1.3 a fi d'obtenir els millors resultats possibles.

Tot seguit, compararem els resultats obtinguts en el nostre projecte amb els resultats exposats a l'article [2] (*Taula 6*).

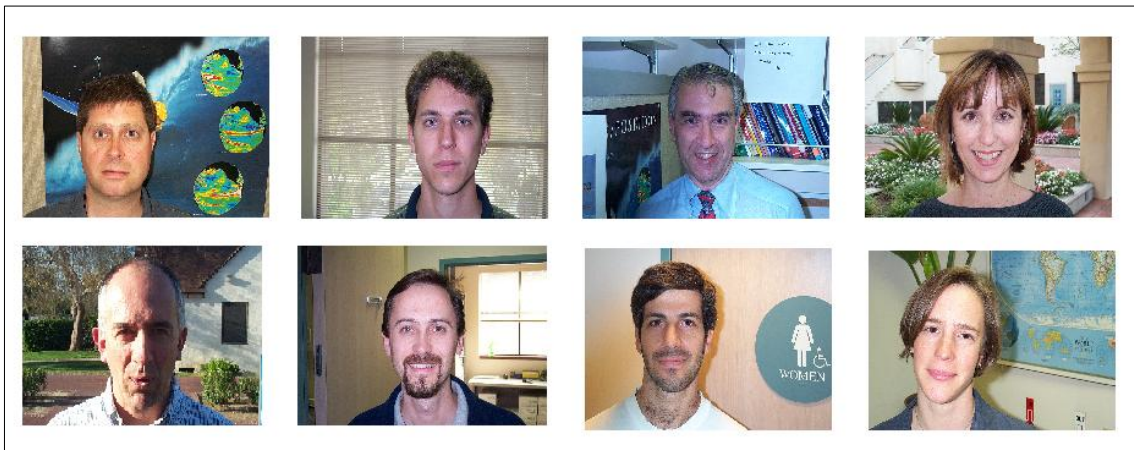


Fig. 37 Algunes imatges de la base de dades Caltech utilitzades per testejar.

	Precisió	Taxa de detecció	Efectivitat
Resultats del projecte	0.870 ± 0.13	0.750 ± 0.24	0.767 ± 0.18
Resultats de l'article [2]	0.756 ± 0.22	0.718 ± 0.11	0.713 ± 0.17

Taula 6 Resultats de detecció facial.

Com podem observar, el fet de tenir en compte el veïnatge millora els resultats de la detecció facial. A l'article [2], es compara el mètode proposat amb set altres mètodes de detecció de pell existents. Tot seguit, podem trobar un resum dels resultats obtinguts mitjançant els mètodes precedents (*Taula 7*).

	Precisió	Taxa de detecció	Efectivitat
Mètode RGB	0.694 ± 0.20	0.884 ± 0.17	0.761 ± 0.17
Mètode CbCr	0.309 ± 0.21	0.548 ± 0.31	0.379 ± 0.23
Mètode HS	0.514 ± 0.21	0.807 ± 0.28	0.585 ± 0.21
Mètode RGB estadístic	0.635 ± 0.23	0.723 ± 0.28	0.643 ± 0.22
Variància Mín.	0.195 ± 0.03	0.190 ± 0.02	0.318 ± 0.05
Vista única	0.365 ± 0.26	0.636 ± 0.34	0.430 ± 0.27
Múltiples vistes	0.703 ± 0.18	0.497 ± 0.20	0.550 ± 0.15
Mètode article [2]	0.756 ± 0.22	0.718 ± 0.11	0.713 ± 0.17
Mètode del projecte (article [2] + veïnatge)	0.870 ± 0.13	0.750 ± 0.24	0.767 ± 0.18

Taula 7 Rendiment de diversos mètodes de detecció facial extrets de [2] i del mètode emprat en el projecte.

Si analitzem els resultats de la [Taula 7](#), veiem que obtenim resultats significativament millors en quant a precisió i resultats sensiblement millors en quant a efectivitat. Tanmateix, obtenim resultats notablement pitjors pel que fa la taxa de detecció. La taxa de detecció del mètode RGB està per sobre de la del mètode proposat en aquest projecte. No obstant, aquesta taxa de detecció tan elevada ve acompanyada d'una precisió inferior. Com hem vist a l'apartat [4.1.3](#), una precisió elevada implica pocs píxels de *no-pell* mal classificats. D'altra banda, una taxa de detecció elevada implica pocs píxels de pell mal classificats. Per tant, el mètode RGB presenta una taxa de detecció elevada en detriment d'una *sobredetecció* de píxels de pell. Òbviament, volem trobar un resultat que presenti un bon compromís entre ambdues mesures. Recordem que l'efectivitat és la mesura que ens retorna aquest compromís. Per tant, busquem un resultat amb la millor efectivitat possible. El mètode exposat en el nostre projecte presenta l'efectivitat més elevada de la [Taula 7](#) i, per tant, podem dir que millora el rendiment de la resta de mètodes.

Si comparem alguns resultats del mètode de l'article [2] amb els del mètode del nostre projecte, observem que la introducció del veïnatge disminueix progressivament les regions de píxels mal classificats ([Fig. 38](#)) i, en conseqüència, augmenten la precisió, la taxa de detecció i l'efectivitat.



Fig. 38 (a) Classificació sense veïnatge. (b) Classificació amb veïnatge.

5.3. Planificació final

En aquest apartat presentem la taula i el diagrama de terminis de la planificació final (Taula 8, Fig. 39).

Com podem observar, hem hagut de dedicar més temps del previst a revisar l'estat de l'art dels camps aleatoris de Markov ja que hem trobat més dificultats de les previstes a l'hora d'inferir l'etiquetatge. A més, hem trobat errors que ens han portat a modificar algunes dades de l'aprenentatge durant l'etapa de test. No obstant, aquestes petites desviacions han pogut ser compensades i, en general, s'han pogut complir els terminis establerts.

Name	Begin date	End date
Revisar l'estat de l'art (setmana 1)	10/13/09	10/20/09
Revisar l'estat de l'art (setmana 2)	10/20/09	10/30/09
Revisar l'estat de l'art (setmana 3)	12/16/09	12/22/09
Revisar l'estat de l'art (setmanes 4 i 5)	12/22/09	1/8/10
Revisar l'estat de l'art (setmana extra)	1/8/10	1/15/10
Analitzar els requisits i dissenyar solució general (setmana 1)	10/30/09	11/6/09
Analitzar els requisits i dissenyar solució general (setmana 2)	1/15/10	1/22/10
Realitzar l'aprenentatge (setmanes 1-3)	11/6/09	11/26/09
Realitzar l'aprenentatge (setmanes 5-9)	1/22/10	2/19/10
Realitzar la classificació (setmanes 1-1.5)	2/19/10	3/2/10
Realitzar la classificació (setmanes 1.5-2)	3/2/10	3/12/10
Realitzar la classificació (setmanes 3-6)	3/18/10	4/15/10
Testejar (setmanes 1)	3/10/10	3/18/10
Testejar (setmanes 2-4)	3/24/10	4/18/10
Realitzar l'aprenentatge (setmana extra)	3/18/10	3/26/10
Escriure la memòria del projecte	4/29/10	6/3/10

Taula 8 Terminis finals.

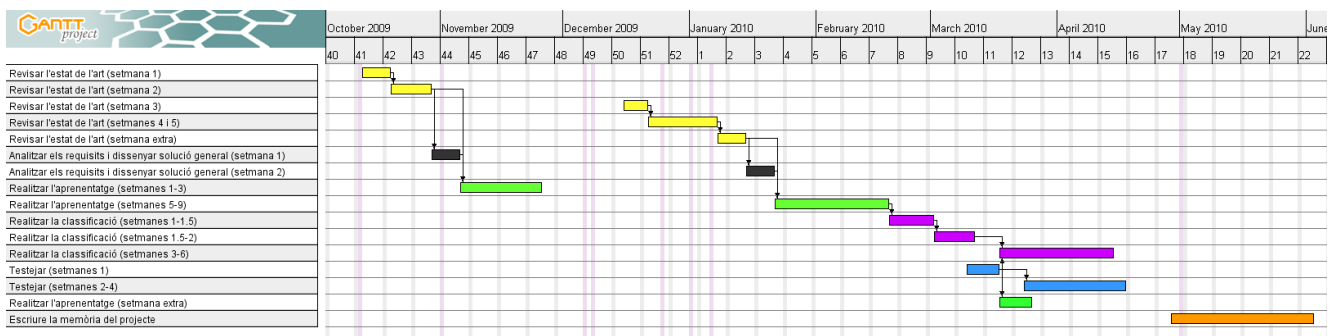


Fig. 39 Diagrama de Gantt de la planificació final.

6. Conclusions i treballs futurs

En aquest últim capítol de la memòria exposarem les conclusions a les que hem arribat a l'acabar el projecte i proposarem possibles treballs futurs que podrien continuar la feina ja presentada.

6.1. Conclusions

Al llarg del projecte hem fet una revisió detallada de l'estat de l'art dels algoritmes de detecció de pell, hem analitzat els seus punts febles i hem proposat, implementat i validat un nou algoritme que té en compte el veïnatge a l'hora de classificar.

Per aconseguir-ho, hem revisat exhaustivament l'estat de l'art dels mètodes que treballen amb el veïnatge. Com a resultat, hem implementat un algoritme que es basa en els camps aleatoris de Markov. Per validar el nostre algoritme, hem dut a terme diversos experiments sobre una base de dades d'imatges exteriors i interiors i hem comparat els resultats amb els de diferents mètodes existents a l'estat de l'art. Finalment, a partir dels resultats obtinguts, podem concloure que la inclusió del veïnatge en el procés de decisió millora la qualitat dels algoritmes de detecció de pell.

6.2. Treballs futurs

En aquest apartat, veurem alguns treballs futurs que podrien continuar la feina ja presentada. Com hem vist a l'apartat 3.2.2, considerem inviable obtenir un model de *no-pell* suficientment acurat. En conseqüència, la probabilitat de *no-pell* d'un píxel mai serà del tot fiable. Una línia de continuïtat d'aquest projecte podria ser el disseny d'un camp aleatori de Markov en el que no es contemplés la probabilitat de *no-pell*, és a dir, un CAM en el que només fos necessari calcular la probabilitat de pell per determinar els missatges entre píxels i, posteriorment, inferir l'etiquetatge de cadascun d'ells.

D'altra banda, cal remarcar que en el nostre projecte, tots els píxels estan influenciats pels seus veïns. No obstant, hi ha píxels de pell que són veïns de píxels de *no-pell* i viceversa. En aquests casos, seria millor que els píxels no es veiessin afectats pels seus veïns. Una segona línia de continuïtat del nostre projecte seria realitzar una detecció de contorns prèvia que trenqués les relacions entre píxels. Entenem per detecció de contorns la detecció de punts en els quals la intensitat canvia bruscament. Aquest mètode ens permetria trencar les dependències d'un píxel

amb els veïns que presentessin intensitats significativament diferents. Per tant, ens permetria trencar el veïnatge entre píxels de pell i *no-pell*.

7. Bibliografia

1. *A survey of skin-color modeling and detection methods.* **Kakumanu, P., Makrogiannis, S. i Bourbakis, N.** 2006, Pattern Recognition.
2. *Learning photometric invariance for object detection.* **Álvarez, Jose M., Gevers, Theo i López, Antonio.** 2010, International Journal of Computer Vision.
3. **Mathworks.** Matlab. [En línia] <http://www.mathworks.com>.
4. **Microsoft.** Microsoft. [En línia] <http://emea.microsoftstore.com/es/es-ES/>.
5. **Li, Stan Z.** *Markov Random Fields Modeling in Image Analysis.* s.l. : Springer, 2009.
6. **M.Webers.** The Caltech Frontal Face Dataset. [En línia] <http://www.vision.caltech.edu/html-files/archive.html>.
7. *Human Skin Colour Clustering for Face Detection.* **Kovac, Jure, Peer, Peter i Solina, Franc.**
8. **Zaiats, V., Calle, M.Luz i Presas, Rosa.** *Probabilitat i estadística.* s.l. : Eumo Editorial, 1998.
9. *Markov Random Field Models: A Bayesian Approach to Computer Vision Problems.* **Kamberova, Gerda.** University of Pennsylvania : s.n., 1992.
10. **Zhao, Wen-Yi i Chellappa, Rama.** Image-Based Face Recognition: Issues and Methods.
11. *Optimal dynamic portfolio selection: multi-period mean-variance formulation.* **Li, Duan i Ng, Wan-Lung.** 1998.
12. **Sourceforge.net.** Gantt Project. [En línia] <http://ganttproject.biz>.

Firmat: Adriana Romero Soriano

Bellaterra, Juny de 2010

Resum

En aquest projecte es proposa un algoritme de detecció de pell que introdueix el veïnatge a l'hora de classificar píxels. Partim d'un espai de color invariant après a partir de múltiples vistes i introduïm la influència del veïnatge mitjançant camps aleatoris de Markov. A partir dels experiments realitzats podem concloure que la inclusió del veïnatge en el procés de classificació de píxels millora significativament els resultats de detecció.

Resumen

En este proyecto se propone un algoritmo de detección de piel que introduce la vecindad para clasificar píxeles. Usamos un espacio de color invariante aprendido a partir de múltiples vistas e introducimos la influencia de la vecindad mediante campos aleatorios de Markov. A partir de los experimentos realizados podemos concluir que la inclusión de la vecindad en el proceso de clasificación de píxeles mejora significativamente los resultados de detección.

Abstract

In this project, we propose a skin detection algorithm which introduces a neighborhood system to classify pixels. We use a color invariant model learnt from different views and we add the influence of the neighborhood using Markov random fields. From the experiments it is concluded that the inclusion of a neighborhood system in the classification process improves significantly the detection results.