

Detecció automàtica de punts facials característics mitjançant un algoritme inspirat en el còrtex cerebral

Ferran Vidal Marginet*
Universitat Oberta de Catalunya

David Masip Rodó†
Universitat Oberta de Catalunya (Rambla del Poble Nou 156, 08018, Barcelona)

En aquest article presentem l'aplicació d'un algoritme inspirat en el còrtex cerebral anomenat HMAX, com a part central d'un procés de detecció automàtica de punts facials característics en imatges. El procés parteix d'un conjunt de mostres d'entrenament de les quals s'extreuen unes característiques mitjançant un algoritme inspirat en els processos que tenen lloc al cervell en l'acte visual. A partir d'aquestes característiques es construeix un classificador mitjançant *Support Vector Machine* que serà utilitzat per detectar punts facials característics en imatges noves. Els resultats obtinguts ens mostren que l'algoritme HMAX té la capacitat d'obtenir de forma correcta punts facials característics. No obstant, els nivells de correctesa a què arriba aquest algoritme, amb els paràmetres que s'han explorat en aquest treball, disten molt d'altres algoritmes actuals.

1. INTRODUCCIÓ

Cada dia és més senzill trobar càmeres fotogràfiques digitals capaces de detectar rostres en temps real. La investigació en detecció automàtica de cares ha arribat a resultats robustos, ràpids i molt precisos, com és el cas de l'algoritme presentat per Viola i Jones l'any 2001[1]. No es pot dir el mateix si el què es vol detectar són punts particulars de la cara. Tot i que s'ha investigat molt en aquesta àrea, encara avui no hi ha algoritmes prou ràpids i precisos. Detectar-los de forma acurada ens obriria les portes a analitzar l'expressió de la cara i poder determinar, per exemple, l'estat d'ànim d'una persona, saber si menteix o no, etc. En aquest treball, proposem reduir les imatges a vectors numèrics a través d'un algoritme inspirat en els processos que el fa cervell alhora de processar imatges. Aquestes vectors numèrics serveixen per entrenar, mitjançant tècniques de *machine learning*, un classificador que ens permetrà detectar punts facials característics en imatges noves. Hem utilitzat una base de dades d'imatges de rostres per validar aquest procés.

2. ESTAT DE L'ART

Al llarg dels últims anys s'han aplicat molts mètodes per optimitzar el reconeixement de punts facials. És comú classificar aquests mètodes en mètodes basats en la forma i mètodes basats en la textura tot i que també hi ha exemples de combinació de les dues metodologies. Entre els mètodes basats en la forma hi destaquen especialment els que utilitzen ASM (Active Shape Models) i els que utilitzen AAM (Active Appearance Models). Aquests mètodes apareixen l'any 1998 de la ma de Tim Cootes i Chris Taylor ambdós professors de la Manchester University[2, 3]

Pel què fa als mètodes basats en la textura, té especial rellevància el treball de Vukadinovic i Pantic[4] en que presenten un mètode per la detecció automàtica de 20 punts facials. Utilitzen filtres de Gabor per extreure punts característics d'un conjunt de zones prèviament determinades de forma heurística. Valstar et.al. van encara una mica més enllà i presenten un procediment[5] que detecta 22 punts característics (s'hi afegeixen les retines). El procediment en qüestió és una combinació de *Support Vector Regression* i camps aleatoris de Markov. Al treball de Belhumeur et al.[6] proposen un model Bayesià que combina la sortida d'un detector local amb un consens de models globals no paramètrics. El seu algoritme és considerat el més acurat i eficaç a dia d'avui, capaç de localitzar de forma precisa els punts característics en imatges complicades com les de la base de dades *Labeled Face Parts in the Wild*[7].

En un marc una mica diferent, també cal destacar l'estudi de Xiangxin Zhu i Deva Ramanan[8] de la universitat de Califòrnia on proposen un model unificat per la detecció de cares, estimació de postures i localització de punts característics. Argumenten que un model unificat fa el problema més senzill. Per exemple, si ens centrem en detectar punts característics, hem d'haver utilitzat abans un detector de cares que ja ens donarà un cert error que acumulem i que ens suposa ja un important biaix.

Però, des del punt de vista biològic, com ho fa el cervell per reconèixer un objecte? Di Carlo et. al.[9] creuen que cada dia hi ha més evidències que demostren que l'habilitat de reconèixer objectes ràpidament malgrat aparents variacions substancials es resol al cervell a través d'una cascada de càlculs 'feedforward' (en xarxes no recurrents) reflexius que culminen en una representació neuronal al còrtex temporal inferior. Malgrat tot, l'algoritme que produeix aquesta solució continua molt poc conegut. Tot i que molts dels models de reconeixement d'objectes s'inspiren en la biologia humana, n'hi ha pocs que siguin fisiològicament consistents, és a dir, que tinguin una certa correspondència amb els models teòrics que segueix el sistema visual. Encara menys són els casos d'aplicació d'aquests models a bases de dades d'imatges reals.

Poggio però, responsable del *Center for Biological & Com-*

*Electronic address: fvidalmar@uoc.edu

†Electronic address: dmasipr@uoc.edu

putational Learning del MIT¹ ha publicat diferents treballs juntament amb el seu equip on sí que s'intenta preservar la consistència a nivell biològic. El què es presenta en aquests treballs, com per exemple[10–12], és un model jeràrquic que segueix l'organització del còrtex visual i construeix una representació invariant de la imatge que després podrà ser utilitzada en diferents tasques de reconeixement d'objectes. Poggio, es va inspirar en el model presentat, l'any 1980, per Fukushima[13] on proposava una simulació del còrtex visual mitjançant una xarxa neuronal jeràrquica. Fukushima, va batejar aquest model amb el nom de Neocognitron. Aquest model, s'inspira alhora en les investigacions que l'any 1968 van publicar Hubel and Wiesel[2] per les quals serien guardonats amb el premi Nobel de medicina. Aquests dos neurobiòlegs van descriure una jerarquia de cèl·lules al còrtex visual primari. Breument el seu model comença amb un conjunt de cèl·lules radials simètriques que responen a un estímul de llum i s'alternen amb cèl·lules simples que responen a estímuls de barres o vores en una orientació particular. El proper estadi de la jerarquia són les cèl·lules complexes, que responen bé a barres orientades o vores dins dels seus camps receptius i que no són sensibles ni a la ubicació ni a la fase de les barres. L'estadi final de la jerarquia es compon d'unes cèl·lules hipercomplexes que no són només invariants a la posició i a la fase de les barres si no que també són selectives pel què fa a la seva llargada.

En general les investigacions actuals estan cada dia més aprop de trobar un algoritme de detecció de punts facials que compleixi amb les expectatives que se li imposen. Malgrat tot de forma subjacent sempre hi ha la aspiració a construir uns processos que no només funcionin computacionalment bé i siguin ràpids si no que siguin també coherents amb els models acceptats de funcionament del còrtex visual. Creiem que tot i que a priori obtinguem algorismes més lents o menys precisos, perseverar en la consistència fisiològica acabarà portant beneficis globals i robustos.

Tornant als treballs de Poggio i el seu equip, cal destacar la creació del model HMAX que ha demostrat molt bons resultats en diferents proves de reconeixement d'objectes. És per això que hem centrat el nostre treball en l'aplicació d'un model avançat de l'HMAX al reconeixement de punts facials característics en imatges de rostres.

A la secció següent presentem detalladament els mètodes que hem utilitzat en el treball. Al tercer apartat descrivim l'entorn computacional en què hem fet les proves. Tot seguit fem una descripció detallada de l'aplicació dels mètodes descrits al reconeixement de punts facials característics en rostres. Al següent apartat presentem els resultats obtinguts i per acabar, a la última secció, fem una valoració crítica d'aquests resultats.

3. MÈTODES

Per tal d'avaluar la bondat de l'algoritme HMAX en el reconeixement de punts facials, hem utilitzat diferents fons amb llicències lliures. S'ha utilitzat el marc de treball CNS, la llibreria LibSVM i l'implementació en OpenCV de l'algoritme de Viola and Jones. Tots ells els detallem a continuació.

3.1. Cortical Network Simulator

Els models computacionals inspirats en el còrtex són cada dia més populars, però presenten dues dificultats importants. Per una banda requereixen una potència computacional molt alta i, per una altre, el codi necessari per mantenir el control de les connexions en xarxa pot complicar-se molt i esdevenir complex i difícil tant de mantenir com de modificar. És per això, que des del *Center for Biological and Computational Learning* del MIT, Mutch i Poggio han creat un marc de treball anomenat CNS ('Cortical Network Simulator')[14] que permet definir i executar models inspirats en el còrtex de forma ràpida i senzilla. El CNS està disponible sota llicència *GNU General Public Licence*, cosa que permet la seva utilització per part de tota la comunitat científica. Per 'model del còrtex' entenem un model en xarxa consistent en un cert nombre de capes n-dimensionals de cèl·lules on cada capa codifica alguna característica de l'espai n-dimensional. Totes les cèl·lules d'una capa han de ser del mateix tipus i la connectivitat entre les capes pot ser completament arbitrària. Entre els models que es poden simular amb aquest sistema hi trobem el model HMAX mencionat en l'apartat anterior.

3.2. Model HMAX

Es creu que el reconeixement d'objectes al còrtex cerebral es porta a terme a través de la via visual ventral[15] que va des de l'escorça visual primària V1, passant per les àrees visuals V2 i V4 i arribant al còrtex inferotemporal, IT. El model HMAX, segueix aquest recorregut i intenta modelitzar, de forma quantitativa, un conjunt de fets àmpliament acceptats en referència a la franja ventral dins de l'escorça visual primària. El model, en la seva versió més simple, consisteix en una jerarquia de quatre capes d'unitats computacionals on les unitats simples, anomenades S, s'alternen amb les unitats més complexes, anomenades C. Les unitats S combinen les entrades amb una funció d'ajust en forma de campana per tal d'augmentar la selectivitat i les unitats C seleccionen les entrades a través d'una operació de màxim per tal d'incrementar la invariància. Aquest model és qualitativa i quantitativament consistent amb diverses propietats de les cèl·lules al llarg de la via visual ventral[16]. Mutch i Lowe, presenten a [17] una versió del model HMAX consistent en les 5 capes següents:

Capa d'imatge. A partir d'una imatge en blanc i negre es crea una piràmide de 10 escales, cadascuna en un factor d'augment de $2^{1/4}$ de l'anterior.

¹ <http://cbcl.mit.edu/>

Capa de filtre de Gabor(S1). La capa S1 es calcula a partir de la capa d'imatge anterior centrant filtres de Gabor amb un cert rang d'orientacions a cadascuna de les possibles posicions i escales. Els filtres de Gabor són de tamany 11x11 i segueixen la fórmula:

$$G(x, y) = \exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} X\right) \quad (1)$$

on $X = x\cos\theta - y\sin\theta$ i $Y = x\sin\theta + y\cos\theta$ i x i y varien entre -5 i 5 i θ varia entre 0 i π . Els paràmetres γ , σ i λ s'han fixat a 0.3, 4.5 i 5.6 respectivament tal i com es suggereix a [18]

Capa d'invariància local(C1). Per cada orientació la piràmide S1 es convoluciona amb un filtre 3D de màxim a través d'unitats 10x10 al llarg de la posició i amb 2 unitats de profunditat en l'escala. Una unitat C1 acaba sent el valor màxim (d'aquesta orientació) que cau dins del filtre de màxim.

Capa de característiques intermèdies(S2). A cada posició i escala de la capa C1, calculem la coincidència entre els patrons de C1 centrats en aquella posició i escala i cadascun dels patrons de prototip. Aquests patrons prototip representen les característiques intermèdies del model.

Capa d'invariància global(C2). Finalment, es crea un vector d-dimensional, cada element del qual és la màxima resposta (sigui on sigui de la imatge) a un dels patrons prototip del model.

3.3. Support Vector machine

Segons va definir McCarthy al 1956, 'la intel·ligència artificial és la ciència i enginyeria dedicada a fer màquines intel·ligents, especialment programes d'ordinador intel·ligents'[19]. Una branca de la intel·ligència artificial és l'aprenentatge automàtic (o *machine learning*) que està dedicada al disseny, anàlisi i desenvolupament d'algoritmes i tècniques que permetin aprendre als ordinadors. De forma més concreta, es centre en crear algoritmes capaços de generalitzar comportaments a partir del reconeixement de patrons. Aquests algoritmes es poden classificar en algoritmes d'aprenentatge supervisat i algoritmes d'aprenentatge no supervisat segons disposem o no d'un conjunt de dades ja classificades. És en el marc dels algoritmes d'aprenentatge supervisat que cal situar el *Support Vector Machine*. Es tracta d'una tècnica que, a partir d'unes dades d'entrenament ja classificades, ens permet classificar noves dades. En el cas particular del reconeixement d'objectes permet, donada la representació numèrica d'una imatge decidir si pertany a un cert conjunt o no. En aquest treball hem utilitzat una implementació del *Support Vector Machine*, anomenada LibSVM [20] amb llicència 'modified BSD'.

3.4. Algoritme Viola and Jones

Viola i Jones, l'any 2001, van proposar un marc de treball per la detecció d'objectes en temps real que esdevingué molt popular. En particular, el seu algoritme està considerat un dels més ràpids i precisos en la detecció automàtica de cares. En aquest treball, n'hem utilitzat la implementació que se n'ha fet a la llibreria OpenCV, sota llicència BSD[21].

4. PREPARACIÓ DE L'ENTORN COMPUTACIONAL

El marc de treball CNS, està preparat per utilitzar la tecnologia CUDA. Aquesta tecnologia és un model de programació en paral·lel creat per NVIDIA i implementat pels processadors GPU que ells produeixen. L'avantatge d'utilitzar els processadors GPU en lloc de la CPU, rau en la possibilitat d'utilitzar una quantitat molt gran de fils diferents de forma simultània. Així, si una aplicació està dissenyada per utilitzar tots aquests fils, el rendiment és molt més alt. Per fer els testos dels quals es deriva aquest article hem utilitzat dos ordinadors de sobretaula amb les següents característiques:

1. Processador Intel(R) Core(TM)2 Quad CPU Q9400 @ 2.66GHz, amb 4Gb de RAM i tarja gràfica GeForce 8600 GT, amb sistema operatiu Ubuntu 12.10 de 64 bits i drivers CUDA 4.1.
2. Processador Intel(R) Core(TM)i5-3450 @ 3.10GHz, amb 8Gb de RAM i tarja gràfica GeForce GT 630, amb sistema operatiu Windows 7 Home de 64 bits i drivers CUDA 4.1.

5. APLICACIÓ

A continuació fem una descripció detallada del procés seguit per tal d'avaluar la bondat de l'algoritme HMAX en el reconeixement de punts facials a partir d'imatges de rostres. Com a primera aproximació i per motius de temps, ens hem centrat en reconèixer un punt característic, el centre de l'ull dret. Hem creat una aplicació que, a partir d'un subconjunt d'una base de dades d'imatges de cares ens construeix un model estadístic capaç de reconèixer l'ull dret en les imatges del subconjunt complementari.

5.1. Base de dades

Hem utilitzat la base de dades BioID², que conté 1521 imatges en blanc i negre en format PGM i amb una resolució de 384x286 pixel. Cadascuna de les imatges conté una vista frontal de la cara d'una de les 23 persones de las quals consta la

² Disponible a <http://www.bioid.com/downloads/software/bioid-face-database.html>

base de dades. També es disposa d'un fitxer on hi ha les coordenades de 20 punts facials per cadascuna de les imatges.



Figura 1: Punts característics etiquetats a la base de dades BioID

El primer que hem fet és separar la base de dades en dos subconjunts, un d'entrenament i l'altre de test, de 1000 i 521 imatges respectivament.

5.2. Entrenament

Obtenció de mostres Per cadascuna de les 1000 imatges d'entrenament hem obtingut 20 mostres de 40x20 pixels, 10 que hem etiquetat com a 'ull positiu' i 10 que hem etiquetat com a 'un ull negatiu'. D'entre les que representen l'ull positiu una imatge és el requadre 40x20 centrat en l'ull dret i les 9 restants s'han obtingut aplicant un escalat a la imatge i una petita rotació (s'han aplicat rotacions de 5,10 i 15 graus i escalats de 1.1, 1.2 i 1.3 vegades la imatge original). Les que representen l'ull negatiu s'han obtingut retallant 10 requadres de 40x20 pixels centrats en l'ull però a una distància de 20 pixels i amb una rotació aleatòria. En total, hem obtingut 10000 imatges d'ulls positius i 10000 imatges d'ulls negatius. Per tal de corregir els efectes de les diferents il·luminacions hem restat, a cada imatge, la mitjana dels valors totals dels seus pixels.



Figura 2: Exemples de mostres d'ulls utilitzades per l'entrenament

Processament de mostres El processament de les mostres és un dels passos claus i, d'alguna manera, té com a objectiu definir, de la millor manera possible i de forma numèrica, què entenem per 'ull positiu' i què entenem per 'ull negatiu'. Hem processat les mostres d'ulls amb l'algoritme HMAX i, paral·lelament, ens hem quedat amb els pixels directament, així, tindrem la possibilitat de comparar els resultats entre ambdós mètodes.

Construcció del classificador Mitjançant la llibreria ja esmentada LIBSVM, hem creat dos classificadors no lineals³, un a partir de les mostres processades amb HMAX i un altre a partir de les mostres obtingudes directament amb els pixels. El classificador ens dona, a partir d'una mostra, la probabilitat de pertànyer a la classe 'ull positiu' i la probabilitat de pertànyer a la classe 'ull negatiu'.

5.3. Test

El test es porta a terme amb el segon subconjunt d'imatges i el què pretén és avaluar l'efectivitat dels classificadors obtinguts en l'entrenament.

Finestra lliscant Donada una imatge del subconjunt de 521 imatges de test, hem utilitzat l'algoritme de *Viola and Jones* per detectar el requadre que engloba la cara en la imatge. Llavors hem marcat un requadre més petit que té per costats el 60% de la longitud del requadre gran i és per aquesta zona per on hem passat una finestra lliscant de 40x20 pixels, d'esquerra a dreta i de dalt a baix. Per cada pas que fa la finestra lliscant, obtenim dos vectors, un resultat de passar l'algoritme HMAX a la imatge i l'altre corresponent al valor dels pixels. Aquests vectors són guardats en dos fitxers separats i així, al final, un cop la finestra lliscant arriba a l'extrem inferior dret del requadre petit, tenim dos fitxers amb els resultats d'aplicar HMAX o d'agafar solament els pixels de tots els requadres de 40x20 pixels per on ha passat la finestra lliscant.

Aplicació del classificador En aquest punt tenim 521 fitxers referents a la resposta per HMAX de la finestra lliscant i 521 fitxers amb vectors de pixels. És en aquest moment que apliquem el classificador que hem obtingut amb SVM tant per uns fitxers com pels altres. Obtenim, per cada imatge i per cada línia del fitxer la probabilitat de cada finestra de ser un ull o no.

Decisió de la posició de l'ull Finalment, donades les probabilitats de cada finestra de ser ull o no ser-ne, cal decidir quina de les finestres escollim com a ull predit. Una opció és agafar la finestra que tingui la probabilitat més alta de ser

³ Hem utilitzat un nucli no lineal RBF i els paràmetres òptims s'han seleccionat mitjançant l'eina *grid.py* pertanyent a la mateixa llibreria.

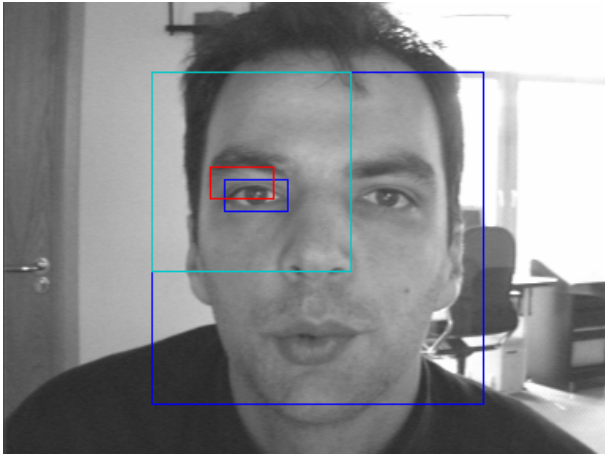


Figura 3: Viola and Jones ens dona el requadre blau exterior. El petit requadre blau interior ens indica la posició correcta de l'ull. El requadre blau cel que emmarca una subregió de la cara ens indica la zona on passar la finestra lliscant(en vermell) per tal de buscar l'ull.

ull. Una altra opció és agafar la posició mitjana de totes les finestres que tenen una probabilitat més alta que un cert llindar.

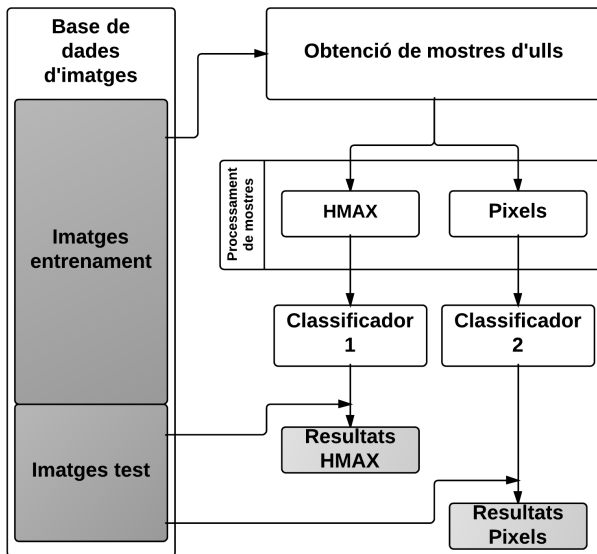


Figura 4: Esquema bàsic que segueix l'aplicació. Des de la obtenció de mostres del subconjunt d'imatges d'entrenament fins a l'aplicació del classificador obtingut al subconjunt d'imatges de test.

6. RESULTATS

6.1. Consideracions prèvies i definicions

L'algoritme HMAX, implementat sobre el marc de treball CNS, ofereix la possibilitat de variar molts paràmetres. Entre ells, podem variar el nombre d'escales a tractar a la capa

d'imatge, els paràmetres dels filtres de Gabor a la capa S1, el nombre de patrons de prototip, etc. El marc de treball CNS està pensat per ser aplicat a problemes de reconeixement d'objectes que requereixin obtenir vectors de característiques d'un nombre petit d'imatges. La nostra finestra lliscant fa que el nombre d'imatges a processar, per cadascuna de les cares, sigui d'entre 1 i 3 milers. L'algoritme calcula el vector de característiques per cadascuna de les imatges de forma independent i això fa que es dupliquin molts dels càlculs de forma innecessària. Això ha fet que els testos a fer hagin sigut molt costosos a nivell computacional limitant així el nombre de proves.

Per tal de presentar els resultats obtinguts, hem escollit una criteri força comú en altres treballs d'aquestes característiques a fi de que els resultats siguin fàcilment comparables. Es tracta de calcular 3 indicadors, la ràtio de classificació, l'error mitjà i la desviació típica d'aquest error. Definim, primer de tot, l'error:

Error(e) Sigui d_e la distància entre el punt predit pel classificador i el punt real i sigui d_{iod} la distància interocular de cada imatge, definim l'error e com la relació entre d_e i d_{iod} , és a dir, $e = \frac{d_e 100}{d_{iod}}$.

Ràtio de classificació(ràtio) Percentatge d'imatges de test en què la posició de l'ull s'hagi encertat. Considerem que la posició de l'ull ha estat encertada sempre i quan l'error e sigui menor al 10%. La ràtio doncs, ve expressada per la següent fórmula:

$$ràtio = \frac{\sum \frac{d_e 100}{d_{iod}} < 0.1 d_{iod}}{\# \text{mostres de test}} \quad (2)$$

Error mitjà(E_μ) Mitjana aritmètica de l'error e .

Desviació típica (E_σ) Desviació típica de l'error e .

6.2. Resultats comentats

Els resultats obtinguts es poden dividir en dos grups diferents. Per una banda hem fet l'execució normal de l'aplicació tenint en compte 50 patrons prototip i tenint-ne en compte 800. A continuació presentem una taula i un gràfic amb els resultats obtinguts.

	Ratio	E_μ	E_σ
Pixels	80.61%	8.63 ± 0.61^a	7.09
HMAX ^b	77.2%	8.92 ± 0.96^a	9.42
HMAX ^c	48.94%	17.03 ± 1.44^a	16.80

^aInterval de confiança calculat al 95%

^bHMAX amb 800 patrons de prototip

^cHMAX amb 50 patrons de prototip

En segon lloc, per tal de donar consistència estadística als resultats, hem fet una validació creuada de 3 passos fixant el nombre de patrons prototip a 800 per ser aquest el valor que

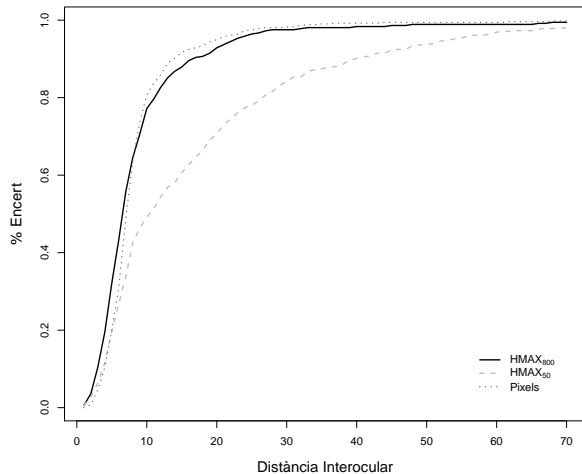


Figura 5: Gràfic que il·lustra la proporció d'encert en relació al percentatge de distància interocular.

havia donat millors resultats. Degut a l'alt cost computacional del procés hem optat per fer-ho amb una mostra de 300 imatges escollides a l'atzar entre les 1521 totals de la base de dades. La següent taula mostra tots els resultats obtinguts.

Algoritme	Pas	Ratio	E_{μ}	E_{σ}
Pixels	1	78.80%	10.50 ± 2.86^a	14.58
	2	62.00%	16.81 ± 3.57	18.22
	3	57.00%	14.38 ± 3.23	16.50
HMAX	1	58.00%	15.07 ± 3.74	19.06
	2	54.00%	15.71 ± 3.04	15.53
	3	59.00%	17.12 ± 4.41	22.48

^aInterval de confiança calculat al 95%

A vistes d'aquests resultats, cal destacar dos aspectes: primer que la ràtio d'encert de l'HMAX no arriba als nivells d'encert dels Pixels i segon, que la validació creuada acaba brindant uns resultats amb una ràtio molt més baixa que quan l'experiment el fem amb la base de dades complerta. Pel què fa al primer punt, cal tenir en compte que l'algoritme HMAX té la possibilitat de variar molts paràmetres i que, tot i que el nombre de patrons prototip és un dels més importants, la variació d'altres paràmetres podria influir de forma substancial i brindar uns resultats millors. Pel què fa al segon aspecte, la conclusió és que la nostra aplicació és especialment sensible al nombre de mostres d'entrenament.

7. CONCLUSIONS I TREBALL FUTUR

En aquest article hem mostrat el procés de construcció d'un classificador capaç de reconèixer automàticament punts característics de cares en imatges. En particular, ens hem centrat en l'aplicació de l'algoritme HMAX i hem presentat diverses proves que demostren que aquest algoritme ofereix uns resultats satisfactoris. Malgrat tot, amb els paràmetres que s'han explorat en aquest treball, no arriba a tenir uns nivells de correctesa tant alts com els que ofereixen altres algoritmes actuals.

Cal destacar, que el sistema implementat aquí requereix un temps de comput molt alt. Aquest fet, ha dificultat els testos i l'exploració de les variables. En aquest sentit, en futurs treballs, seria interessant millorar dos aspectes:

a. *CNS amb finestra lliscant* Per tal d'augmentar la velocitat dels testos caldria modificar lleugerament el marc de treball CNS per tal d'evitar els càlculs duplicats en el cas de les finestres lliscants.

b. *Exploració de paràmetres* Fet lo anterior, hi hauria la possibilitat, d'explorar a fons un bon rang de paràmetres variables de l'algoritme HMAX de tal forma que es podria arribar a obtenir la combinació òptima i més satisfactòria.

-
- [1] P. A. Viola and M. J. Jones, *Rapid Object Detection using a Boosted Cascade of Simple Features* (2001).
- [2] D. H. Hubel and T. N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*, vol. 160 (1962), URL <http://jp.physoc.org/content/160/1/106.short>.
- [3] T. F. Cootes, G. J. Edwards, and C. J. Taylor, *Active Appearance Models* (Springer, 1998).
- [4] D. Vukadinovic and M. Pantic, *Fully Automatic Facial Feature Point Detection Using Gabor Feature Based Boosted Classifiers* (2005).
- [5] M. F. Valstar, B. Martinez, X. Binefa, and M. Pantic, *Facial Point Detection using Boosted Regression and Graph Models* (San Francisco, USA, 2010), URL <http://ibug.doc.ic.ac.uk/media/uploads/documents/CVPR-2010-ValstarEtAl-CAMERA.pdf>.
- [6] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, *Localizing Parts of Faces Using a Consensus of Exemplars*, vol. 99 (IEEE Computer Society, Los Alamitos, CA, USA, 2013).
- [7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*, 07-49 (2007).
- [8] D. Ramanan, *Face detection, pose estimation, and landmark localization in the wild*, CVPR '12 (IEEE Computer Society, Washington, DC, USA, 2012), ISBN 978-1-4673-1226-4, URL <http://dl.acm.org/citation.cfm?id=2354409.2355119>.
- [9] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, *How does the brain solve visual object recognition?*, vol. 73 (2012), URL <http://download.cell.com/neuron/pdf/PIIS089662731200092X.pdf?intermediate=true>.
- [10] M. Riesenhuber and T. Poggio, *Hierarchical models of object recognition in cortex*, vol. 2 (1999).
- [11] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio,

- Robust object recognition with cortex-like mechanisms*, vol. 29 (2007).
- [12] C. Theriault, N. Thome, and M. Cord, *Extended coding and pooling in the HMAX model*. (2012).
- [13] K. Fukushima, *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*, vol. 36 (1980).
- [14] J. Mutch, U. Knoblich, and T. Poggio, *CNS: a GPU-based framework for simulating cortically-organized networks*, MIT-CSAIL-TR-2010-013 / CBCL-286 (Cambridge, MA, 2010).
- [15] L. G. Ungerleider and J. V. Haxby, 'What' and 'where' in the human brain., vol. 4 (1994).
- [16] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, T. Poggio, T. Serre, M. Kouh, C. Cadieu, U. Knoblich, et al., *A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex* (2005).
- [17] J. Mutch and D. G. Lowe, *Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields*, vol. 80 (Kluwer Academic Publishers, Hingham, MA, USA, 2008), URL <http://dx.doi.org/10.1007/s11263-007-0118-0>.
- [18] T. Serre, L. Wolf, and T. Poggio, *Object Recognition with Features Inspired by Visual Cortex*, CVPR '05 (2005), ISBN 0-7695-2372-2.
- [19] J. McCarthy, *What is artificial intelligence?* (2007), URL <http://www-formal.stanford.edu/jmc/whatisai/node1.html>.
- [20] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, vol. 2 (2011), software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [21] G. Bradski, *The OpenCV Library* (2000).