

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

Aleix Torra Palomas

Grau en Enginyeria en Tecnologies Industrials

Juny 2016

**Aplicació d'un mètode per a la detecció
d'ícones que s'adapti a les necessitats de la
validació automàtica de dispositius
Infotainment**

MEMÒRIA

Autor:
Director/s:
Convocatòria:

Aleix Torra Palomas
Lluís Pérez Vidal
Juny 2016



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquest projecte de fi de grau consisteix en la creació d'una nova eina de visió artificial, que permeti detectar una icona determinada en una pantalla, amb l'objectiu de millorar la validació automàtica dels dispositius Infotainment.

S'ha realitzat aquest projecte dins de l'empresa SEAT, més concretament en el Centre Tècnic de SEAT. El departament en el que estava designat era el d'Elèctrics i, dins d'aquest, en el departament d'Automatització, Connected Car & Infotainment.

Degut a que l'eina utilitzada fins al moment per a la detecció d'ícones en pantalles en la validació donava molts problemes, tant a la hora d'utilitzar-la com en el mateix funcionament, s'ha decidit implementar-ne una de nova que sigui més fiable i a la vegada augmenti les seves capacitats. Per exemple, l'eina antiga només trobava les ícones en el cas de que tots els píxels fossin exactament iguals, mentre que la nova permet identificar la icona malgrat que la trobada en pantalla varii una mica en la forma o en el color.

El que fa l'eina és a partir d'una icona donada, buscar-la en una altra imatge que representa la pantalla dels dispositius Infotainment que es volen validar, retornant la posició del vèrtex superior esquerra i del vèrtex inferior dret de l'espai on l'ha trobada, en forma de coordenades (x, y).

Per a facilitar la integració de l'eina en el sistema de validació instaurat dins de SEAT, s'ha decidit implementar-la mitjançant el llenguatge de programació lliure Python, i mitjançant la llibreria OpenCV.

S'han realitzat una gran quantitat de proves, utilitzant diferents ícones en diferents pantalles, amb l'objectiu de tindre un gran ventall de resultats que permetin veure si l'eina funciona correctament i més tard poder-ne extreure conclusions.

Com a conclusió principal s'ha vist que el mètode funciona correctament en la gran majoria de casos. Aconseguint, d'aquesta manera, els objectius marcats a l'inici.

Sumari

El següent sumari es genera i s'actualitza automàticament amb l'opció “*actualizar campos*” que surt quan es prem el botó dret del ratolí a sobre d'ell.

RESUM	1
SUMARI	3
1. GLOSSARI	5
2. PREFACI	7
2.1. Contextualització.....	7
3. INTRODUCCIÓ	9
3.1. Motivació	9
3.2. Objectius del projecte	9
3.3. Abast del projecte	9
4. INFOTAINMENT	11
4.1. Elements bàsics que componen Infotainment	11
3.1.1. Unitat principal (MU: Main Unit)	11
3.1.2. Llista de funcions de que disposa Infotainment.....	13
5. ANTECEDENTS	15
6. NUCLI DE LA MEMÒRIA	18
6.1. Visió artificial	18
6.1.1. Captura.....	18
6.1.2. Imatges.....	19
6.1.3. Preprocessament	22
6.1.4. Segmentació.....	30
6.1.5. Reconeixement	42
6.2. Eines estudiades per a aplicar el mètode	43
6.2.1. Idea inicial.....	44
6.2.2. Segona idea	47
6.2.3. Classes	51
6.2.4. Altres funcions	52
7. RESULTATS	57
7.1. Estructura dels resultats	57
7.2. Primeres proves.....	59

7.3. Proves amb l'aplicació integrada en el sistema Seat	73
7.4. Nou algoritme.....	77
7.5. Proves amb l'algoritme	79
8. ANÀLISI ECONÒMIC	85
8.1. Planificació temporal:.....	85
8.2. Costos	86
8.3. Impacte en el medi ambient	87
9. CONCLUSIONS	88
9.1. Fiabilitat del mètode:.....	88
9.2. Optimització del mètode:	88
9.3. Nou algoritme:.....	90
9.4. Conclusió general	91
10. PROPOSTES DE MILLORA	92
11. VALORACIÓ PERSONAL	94
12. AGRAÏMENTS	95
13. BIBLIOGRAFIA	96
13.1. Referències bibliogràfiques	96
13.2. Bibliografia complementaria	99

1. Glossari

- Eina 'standalone': una eina que només depengui d'ella mateixa, que no es necessiti tenir instal·lats altres programes o llibreries.
- Display: és com es coneix a la pantalla on es mostra el que està succeint dins la Main Unit.
- Broadcast: és una forma de transmissió d'informació, on hi ha un emissor i múltiples receptors.
- Tuple: és una llista ordenada d'elements.
- Berkeley Software Distribution: sistema operatiu derivat del sistema Unix, amb reduïdes restriccions pel que fa a la seva distribució.
- Array: és una matriu
- String: és un conjunt de lletres entès com a text en programació
- Màster: s'usa força vegades aquesta paraula durant el treball, equival a la paraula en anglès 'Master'. Fa referència a la imatge que s'obté de la pantalla, on es busquen les icones.

2. Prefaci

2.1. Contextualització

Aquest projecte està molt relacionat amb les pràctiques realitzades a l'empresa i més concretament amb la validació automàtica que s'hi duu a terme.

Començant pel principi, els dispositius Infotainment cada vegada evolucionen més en la direcció de la tecnologia mòbil com podrien ser els Smartphones o les tauletes, pel que fa a:

- Major connectivitat.
- Major personalització.
- Major resolució de les pantalles.
- Major complexitat gràfica, podent incloure animacions.

Pel que fa a l'àrea de la validació automàtica d'Infotainment, això es tradueix en la necessitat de la millora continua dels mètodes utilitzats, per tal de poder adaptar-se a aquesta evolució. Especialment en el cas del processat d'imatge, ja que juga un paper molt important pel que fa a la navegació pels menús i a la identificació dels elements gràfics que trobem en pantalla.

En l'escenari actual on ens trobem, els dispositius Infotainment que s'estan validant presenten gràfics sempre estàtics, amb petites animacions en les transicions entre els estats, però que no afecten o no es tenen en compte a l'hora de validar. Les icones que trobem són sobre fons negres i no es poden personalitzar, de manera que sempre són iguals (o haurien de ser-ho). Per tant els mètodes utilitzats fins ara són força senzills i són els tres següents:

- OCR (Optical Character Recognition): És el Reconeixement de caràcters, es gestiona des de VW i no tenim suport, funciona malament.
- OSR (Optical Symbol Recognition): És el reconeixement de símbols, funciona de manera ràpida i fiable, cosa que permet que a vegades fins i tot es reconeguin textos com a símbols per a evitar l'OCR. Tot i això és molt sensible a qualsevol canvi per petit que sigui.
- Image Difference: És la comparació de dues imatges píxel a píxel, molt eficaç, però només en casos molt concrets.

Tot sembla indicar que de cara a la pròxima generació de dispositius, prevista per al 2018, hi haurà canvis força importants, els comentats a l'inici, i això tindrà un efecte directe pel que fa al processat d'imatge en l'aspecte de la validació.

- Major resolució de les pantalles: implica major càrrega computacional, i en conseqüència major temps d'execució.
- Personalització: possibilitat per l'usuari de modificar elements gràfics, com ara colors, temes...
- Major complexitat gràfica: elements o fons de pantalla animats i dissenys gràfics amb transparències entre d'altres possibilitats.

3. Introducció

3.1. Motivació

He decidit fer aquest projecte perquè he estat de becari a Seat i se m'ha donat aquesta oportunitat des de dins un cop he tingut les pràctiques curriculars acabades. A part, el processat d'imatges em sembla un tema força interessant i ple de possibilitats.

3.2. Objectius del projecte

Per tant, l'objectiu d'aquest projecte és el d'estudiar els mètodes de processament d'imatge existents, per a decidir i més tard implementar, el mètode que millor permeti trobar una icona determinada en una pantalla.

3.3. Abast del projecte

Aquest projecte ha de complir les següents condicions:

- Ha de ser integrable en el sistema actual UTIL – EXAM, això implica la utilització del llenguatge de programació lliure Python.
- Que el compromís entre la efectivitat, la càrrega computacional del sistema i el temps sigui ajustable.
- Que sigui implementat en una pràctica interfície gràfica que inclogui les funcionalitats més bàsiques.
- Que el mètode sigui implementat en una eina 'standalone'.

4. Infotainment

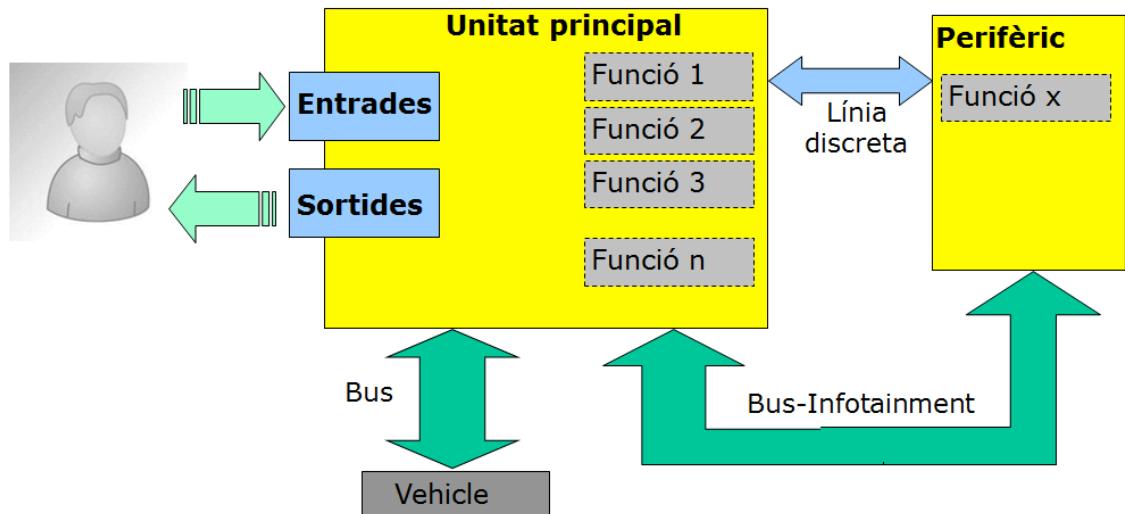
Anteriorment s'ha parlat força d'Infotainment i de que aquest projecte està encarat a millorar la seva validació automàtica, donat que s'ha estat realitzant dins d'aquest departament, per tant a continuació es profunditzarà una mica sobre el tema [1].

S'entén com a sistema Infotainment al sistema que engloba totes les funcions encarregades de proporcionar, tant al conductor com als seus acompañants, entreteniment i informació o comunicació aliena a la conducció.

Com a sistemes d'entreteniment s'entén la Radio i Media, compostos per música, vídeo i connectivitat.

El sistema de comunicació i informació engloba la navegació, telefonia i internet.

4.1. Elements bàsics que componen Infotainment



Il·lustració 1: Elements que componen Infotainment

3.1.1. Unitat principal (MU: Main Unit)

S'anomena unitat principal a la centraleta que realitza les funcions principals del sistema i gestiona la resta de components. D'alguna manera es podria dir que és el 'cervell' del sistema d'Infotainment.

Funcions que realitza:

- Interactua amb les entrades i sortides que dona o rep l'usuari.
- Es comunica amb el vehicle mitjançant un bus de dades.
- Conté l'HMI, que és l'aplicació d'interfície de l'usuari.
- Integra la gran majoria de les funcions de usuari, rebent senyals dels sensors existents.
- Interactua amb els perifèrics que contenen funcions d'usuari.
- Realitza el diagnòstic de ella mateixa.
- Realitza el diagnòstic dels perifèrics.

Entrades de l'usuari:

És el sistema que està conformat per una sèrie de dispositius o mecanismes que permeten a l'usuari interactuar amb el sistema seleccionant alguna funció disponible. En aquest procés rep els inputs de l'usuari i els passa a la unitat principal. Pot ser un perifèric a la MU.

Diferents tipus d'input:

- Tàctils: tant poden ser les pantalles tàctils, com els botons físics, com els comandaments del volant.
- Acústics: funciona mitjançant un micròfon i la funció de reconeixement de veu.

Sortides de l'usuari:

És el sistema compost per un conjunt de dispositius o mecanismes encarregats de posar a disposició de l'usuari la informació. Al rebre la informació de la MU la fan visible o audible de cara a que l'usuari rebi la resposta. També poden ser perifèrics.

Tipus d'output:

- Visual: mitjançant el Display o pantalla.
- Acústics: mitjançant els altaveus.

Elements perifèrics:

Són les centraletes externes a la MU que realitzen funcions que aquesta no té integrades i serveixen per a transmetre-li informació. A mesura que una funció opcional va agafant força dins del mercat, es contempla la possibilitat d'integrar-la dins de la MU.

Exemples de perifèrics:

- Centraletes d'entrades o sortides: es poden trobar els displays, la botonera o els 'joysticks'.
- Funcions opcionals:
 - Centraleta Bluetooth (requereix d'una antena).
 - DAB, SDARS, (també requereix antena).
 - Reconeixement per veu (requereix micròfon).
 - Amplificadors d'àudio (requereix línies d'àudio).

4.1.2. Llista de funcions de que disposa Infotainment

- **Recepció de serveis Broadcast** (ràdio, TV): recepció i sintonització de les senyals emeses per els proveïdors de serveis de Broadcast. Es té tant Àudio, analògic (AM i FM) i digital (terrestre i satèl·lit), com vídeo, analògic(TV) i digital (DVB (Digital Video Broadcasting)), DMB. Està format per una antena, cablejat i un sintonitzador.
- **Reproducció Media:** consisteix en la descodificació i reproducció de diferents formats de Media obtinguts a través de diferents suports.
 - Tipus de formats:
 - Àudio: CDDA, mp3, wma, AAC...
 - Vídeo: DVD, Blue-Ray
 - Tipus de suports:
 - Àudio:
 - Analògic: Cassette, entrada Auxiliar

- Digital: lector de CD, port USB, iPod, lector de targetes SD
 - Vídeo:
 - Digital: lector de DVD, lector de Blue-Ray
 - Connexió per radiofreqüència: Bluetooth AD2P, AVRCP
 - Digital: lector de DVD, lector de Blue-Ray
- **Telefonia i connectivitat:** és el servei de comunicació amb l'exterior integrat dins del vehicle. Inclou serveis actuals dels mòbils més enllà del que podria ser el servei telefònic, com per exemple accés a internet o serveis online, utilitzant el mòbil com a interfície entre l'usuari i l'exterior. Es pot integrar el mòbil mitjançant connexió Bluetooth, mitjançant un cable o a partir de la targeta SIM directament.
- **Navegació:** és el sistema de guiat des de una posició coneguda fins a un destí seleccionat a través d'indicacions tant visuals com acústiques. Dins es troben diferents funcions com navegació dinàmica, navegació predictiva o sistema de posicionament per a vehicles robats entre d'altres.
- **So:** és el sistema que s'encarrega del tractament de les senyals elèctriques de so per adaptar-les a l'entorn del vehicle, mitjançant amplificadors, i convertir-les en senyals acústiques, mitjançant altaveus. Hi ha múltiples tipus de configuracions.
- **HMI (Human Machine Interface):** l'HMI és la interfície entre l'usuari i Infotainment, i té com a objectiu presentar les funcions a l'usuari d'una manera senzilla i comprensible, es serveix de 4 formes per a fer-ho: visual, acústica, tàctil i veu.
- **Informació o ajustos de vehicle:** cada vegada més el món d'Infotainment inclou ajustos i informacions de confort de la conducció.
- Les informacions inclouen els paràmetres de bon funcionament del vehicle i a la vegada notificacions de les anomalies que es puguin detectar.
 - Els ajustos permeten a l'usuari personalitzar des de els comandaments i amb el display de la MU, els paràmetres relatius al confort de la conducció.
- **Reconeixement de veu:** és la funció que habilita a l'usuari a controlar les funcions mitjançant la seva veu. Es poden controlar d'aquesta manera les funcions de navegació, telèfon, ràdio i Mèdia.

5. Antecedents

En el sector de la visió artificial s'ha investigat força ja, i es poden trobar diferents aplicacions.

Sense sortir del sector de l'automoció, es pot veure com ja s'ha integrat la visió artificial en els vehicles més nous. S'utilitza, per exemple, per a la detecció de senyals mitjançant càmeres de vídeo integrades dins del cotxe.

D'aquesta manera s'aconsegueix que si el conductor del vehicle no es fixa en alguna senyal o no recorda quina era la senyal en vigor en aquell moment, pot saber de totes formes quines normes ha de complir en aquest tram de carretera, ja que t'ho recorda en tot moment.

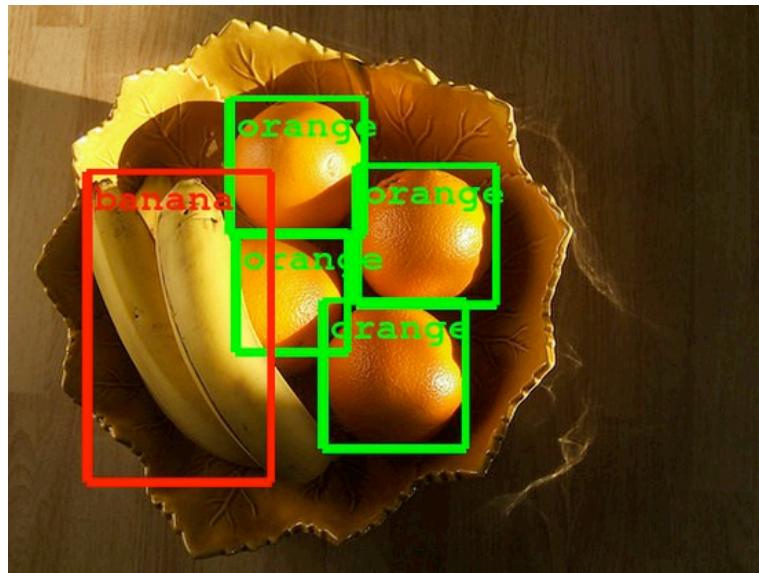


Il·lustració 2: Vehicle Volvo amb detector de senyals de trànsit [2]

Empreses de cotxes com per exemple BMW o Volvo han investigat i desenvolupat bastant en aquesta direcció.

Una altra empresa que aposta molt fort en aquest sector es Google, que ara mateix hi està dedicant molts recursos [3].

Utilitzen xarxes neuronals per a analitzar les imatges. S'entenen com a xarxes neuronals a una sèrie d'ordinadors i processadors que aconsegueixen crear una xarxa d'intel·ligència artificial o cervell virtual capaç d'aprendre per ella mateixa sense cap tipus de pauta o supervisió humana.



Il·lustració 3: Detecció de peces de fruita amb l'eina de Google

En la imatge anterior es pot veure com, mitjançant la visió artificial, s'han reconegut els objectes, en aquest cas 4 taronges i dos plàtans, malgrat que uns tapin en part els altres.

De fet, han aconseguit anar més lluny encara, creant un cotxe autònom que funciona sense conductor, anomenat 'Google Car' [4], capaç de conduir autònomament per ciutat i per carretera, detectant altres vehicles, senyals de tràfic, vianants...



Il·lustració 4: Imatge del 'Goolge Car'

Aquest cotxe funciona mitjançant diferents sistemes [5]:

- LIDAR: és l'aparell que es veu sobre el cotxe, detecta els objectes mitjançant llum làser, i va combinat amb el GPS.
- Radar: disposa de 4 radars per a detectar la proximitat dels objectes que té al voltant.
- Càmera: és la part que ens interessa, s'utilitza per a detectar les senyals de trànsit, els semàfors, les línies dels carrils...

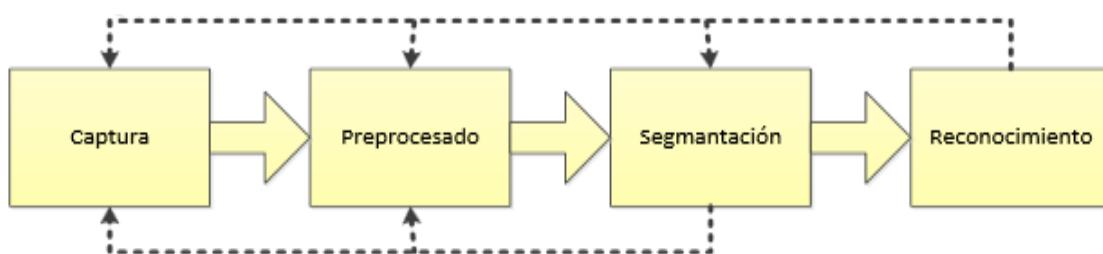
Hi ha més aplicacions encara, però aquestes semblen les més interessants en l'aspecte de que malgrat no sigui exactament la mateixa funció, aquest projecte també s'utilitzarà en el sector de l'automoció.

6. Nucli de la memòria

6.1. Visió artificial

La visió artificial és la branca de la intel·ligència artificial que s'encarrega de modular matemàticament els processos de la percepció visual dels éssers vius per a poder simular-los després en un ordinador. Consisteix en adquirir, processar, analitzar i comprendre les imatges del món real per a poder produir informació numèrica o simbòlica per a que més tard sigui tractada [6].

Etapes d'un sistema de visió artificial:



Il·lustració 5: Esquema de les etapes de la visió artificial

- **Captura:** és l'etapa que consisteix en la adquisició de les imatges digitals mitjançant algun tipus de sensor com per exemple una càmera digital.
- **Preprocessat:** prepara la imatge eliminant les parts no útils o realçant les parts que poden ser més interessants.
- **Segmentació:** aïlla els elements d'interès per a millorar després la seva interpretació.
- **Reconeixement:** destria els objectes segmentats en funció de les característiques que se n'obtenen.

6.1.1. Captura

Generalment és molt important el procés de captura i digitalització de la imatge, ja que si la imatge no té bona resolució, o està mal digitalitzada, treballar-hi serà més complicat. Més si es té en compte que s'han de definir els contorns de les figures o objectes que es busquen.

S'entén que la imatge digital és la representació bidimensional d'una imatge a partir d'una matriu numèrica de bits, que són la unitat mínima d'informació, composta per uns i zeros.

Aquesta pot presentar dos tipus de resolució, estàtica o dinàmica, podent tractar-se d'una imatge matricial, que és un mapa de bits on es representa un conjunt de píxels o punts de color, i és el més utilitzat, o d'un gràfic vectorial, imatge producte de objectes geomètrics independents. La principal diferència entre ambdós resolucions és que a diferència de la resolució estàtica, la dinàmica no perd l'escala al ampliar la mida de la imatge.

Pel que respecta a aquest projecte no es profunditzarà en la adquisició de les imatges, ja que les imatges venen donades com a captures de la pantalla de la MIB. Aquestes captures es realitzen mitjançant un aparell anomenat 'frame grabber'. Aquest és un dispositiu electrònic que realitza captures de pantalla donada una senyal de vídeo.



Il·lustració 6: Imatge d'un 'frame grabber' [7]

6.1.2. Imatges

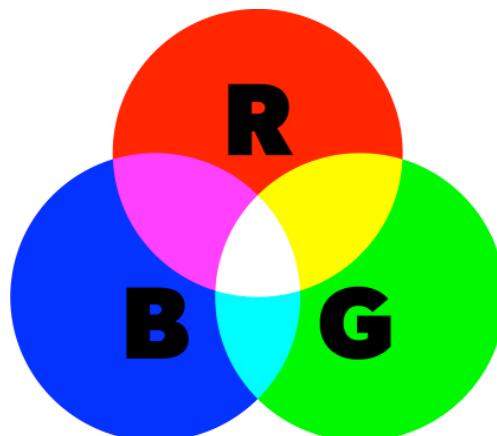
Models de color:

S'entén el model de color com a una representació matemàtica abstracta que descriu la forma en que els colors es defineixen com a vectors de nombres. Normalment es defineixen com a 'tuples' compostes de tres o quatre valors, i permet la reproducció en format digital [8].

Com a models de color es pot trobar per exemple, RGB, CMYK i HSV.

RGB:

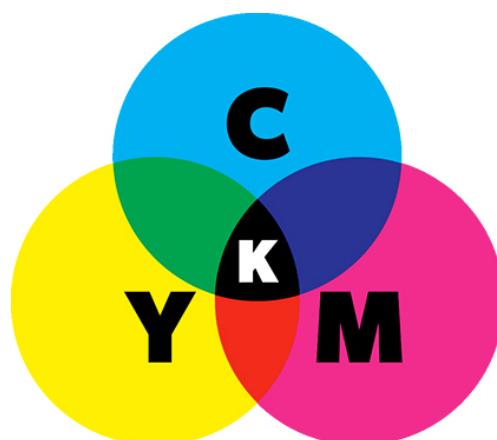
RGB és un model de color que es basa en la síntesi additiva, permet representar un color mitjançant la barreja dels tres colors de llum primaris, vermell (Red), verd (Green), i blau (Blue). És el model més indicat a hora de representar els colors a una pantalla de monitor. A partir d'aquests tres colors es poden generar fins a 16,7 milions de colors diferents. Aquest model assigna un nivell d'intensitat diferent a cada color primari, i aquests valors oscil·len entre el 0 (negre) i el 255 (blanc).



Il·lustració 7: Representació del model de color RGB

CMY o CMYK:

El CMY o CMYK és el model de color que s'utilitza per a impressions. La diferència entre aquests dos és que al segon se li afegeix el color negre. Està compost pels següents colors primaris: cian(Cyan), magenta(Magenta), groc(Yellow) i apareix la possibilitat d'afegir el negre (Black). La principal diferència amb l'anterior és que mentre que en el RGB els colors són additius, en el CMY són sostractius, essent complementaris un model i l'altre. Cada parell de colors additius en crea un de sostractiu i viceversa. Per entendre-ho millor, en el RGB al fer la suma de totes les llums obtenim el color blanc, mentre que en el CMYK sumem pigments, i de la suma de tots ells apareix el negre. Per a obtenir el color blanc en aquest model es fa a partir de l'absència de tots els colors (0%C, 0%M, 0%Y, 0%K), la resta de colors es generen a partir de percentatges.

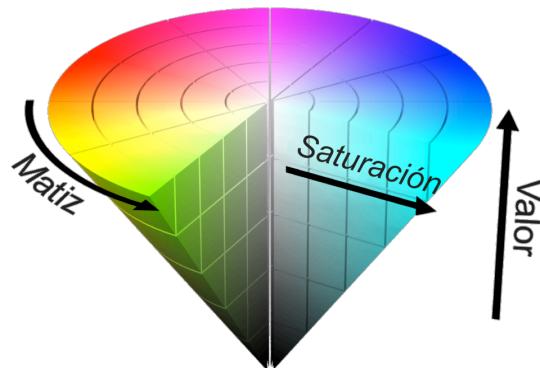


Il·lustració 8: Representació del model de color CMYK

HSV:

És la representació tridimensional del color, basat en tres components diferents: matís o

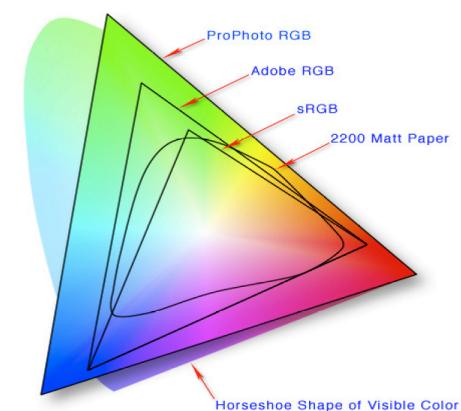
tonalitat (Hue), saturació (Saturation) i valor (Value) [9]. Es diferència del model RGB en l'aspecte de que si bé en el model RGB les coordenades són euclidianes, en aquest són de tipus cilíndriques, es pot representar mitjançant un con, on el valor es considera l'altura, el ton es considera l'angle al voltant de la vertical i la saturació la distància de la vertical o centre del con a la superfície externa. A més a més és una manera de representar els colors que s'assembla molt a la forma en que els humans els percepren.



Il·lustració 9: Model de color HSV

Espais de color:

Existeix una funció de mapatge entre l'espai de color i el seu model, anomenada gama de color, que defineix cada espai de color [10]. Es considera un espai de color com a un sistema d'organització específica que permet la interpretació del color. L'espai de color depèn del model de color que s'utilitza juntament amb els dispositius físics que el reproduïxen i les seves pròpies limitacions. Com a exemple podem trobar l'Adobe RGB, sRGB, ProPhoto RGB o 2200 Matt Paper, basats en el model RGB entre d'altres espais.



Il·lustració 10: Espai de color

6.1.3. Preprocessament

La gran majoria de les vegades, les imatges originals que es volen tractar, contenen més informació de la que és realment rellevant a la hora de tractar-les. És llavors quan és recomanable simplificar-les, al fer-ho, suprimim tota aquella informació que és irrelevante. El fet de simplificar-les ens permet controlar la quantitat d'informació i la naturalesa d'aquesta. Amb això s'aconsegueix posteriorment una segmentació més fàcil i fiable [11].

Es filtra reduint la complexitat de textures, eliminant soroll o eliminant objectes molt petits.

Tipus de soroll:

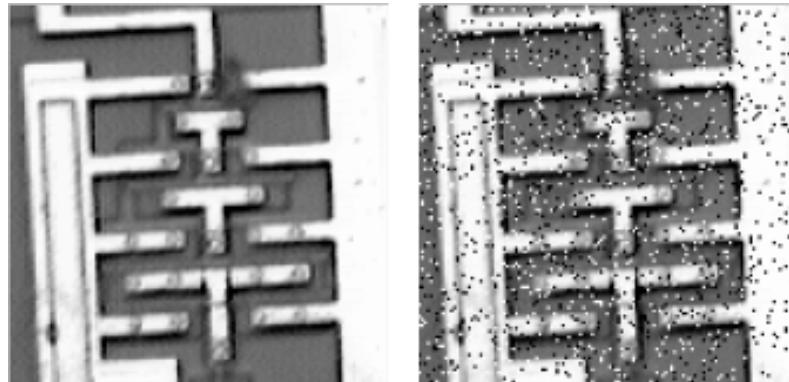
S'entén com a soroll la informació no desitjada que contamina la imatge, generalment es manifesta en píxels aïllats. Hi ha diferents tipus de soroll, el gaussià, l'impulsional i l'uniforme, aquest últim pot ser freqüencial o multiplicatiu.

- **Soroll gaussià:** En el soroll gaussià el valor final de cada píxel és l'ideal més una certa quantitat d'error, i es podria descriure com a una variable gaussiana que segueix una distribució normal. Aquest produceix petites variacions de la imatge, veient-se alterada la intensitat dels píxels, i sol ser degut a components electrònics, com ara sensors o digitalitzadors.



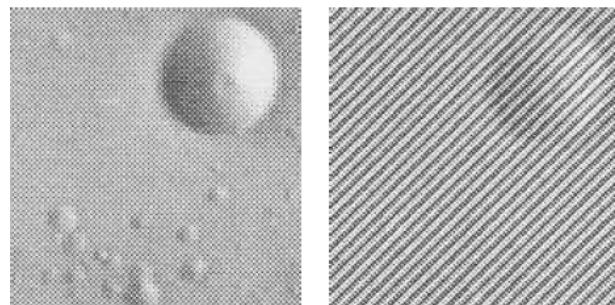
Il·lustració 11: Imatge original i a la seva dreta amb soroll gaussià

- **Soroll impulsional:** El soroll impulsional, més conegut popularment com a 'sal i pebre', consisteix en que el valor del píxel no varia una mica respecte al valor ideal com passa en el soroll gaussià sinó que pren un valor molt alt que és el blanc (anomenat 'sal') o molt baix, que és el negre (anomenat 'pebre').



Il·lustració 12: Imatge original i a la seva dreta amb soroll impulsional

- **Soroll uniforme:** Es coneix així perquè el soroll que afecta a la imatge en qüestió segueix una distribució uniforme, en el sentit de que la probabilitat de prendre un valor qualsevol de gris dins d'un interval definit és constant. Es poden trobar dos tipus diferents de soroll dins del soroll uniforme: el soroll uniforme freqüencial, en que la imatge que s'obté és la suma de la original més una interferència de senyal periòdica (dues imatges superiors), i per altra banda es té el soroll uniforme multiplicatiu, on la senyal obtinguda s'obté a partir de la multiplicació de dues senyals (dues imatges inferiors).



Il·lustració 13: Imatge original i a la seva dreta amb soroll freqüencial



Il·lustració 14: Imatge original i a la seva dreta amb soroll uniforme

Eliminació del soroll:

Es pot eliminar soroll mitjançant diferents tipus de filtre:

- Dins del domini espacial:

- **Filtres de ordre:** els filters d'ordre el que fan és substituir el valor del píxel que s'està estudiant per un nou valor que s'obté dels píxels veïns. Es pot calcular de tres formes diferents:
 - Fent la mitjana dels valors dels píxels veïns
 - Agafant el valor més repetit dels píxels veïns
 - Agafant el valor màxim o el valor mínim dels píxels veïns

Té avantatges i inconvenients, funciona molt bé per a atenuar el soroll de tipus impulsional, elimina efectes enganyosos i preserva els contorns de la imatge, per contra, perd detalls com punts o línies fines, arrodoneix les cantonades dels objectes i pot haver-hi desplaçament dels contorns. Veiem una foto com a exemple, primer agafant una submatriu de 3x3 i després amb una submatriu de 13x13:



Il·lustració 15: A l'esquerra filtrat de mínims 3x3, a la dreta filtrat de màxims 3x3



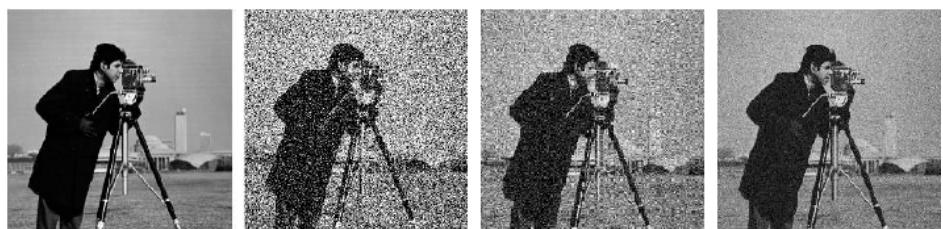
Il·lustració 16: A l'esquerra filtrat de mínims 13x13, a la dreta filtrat de màxims 13x13

- Promig de les imatges: es fa el promig de diferents imatges d'una mateixa escena, entenent l'escena com a la imatge original o ideal, i les diferents imatges com a la imatge original amb soroll. Es podria dir que es 'sumen' les diferents imatges amb la intenció de reduir el soroll i amb l'objectiu d'apropar-se el màxim a la imatge original. Quan major és el nombre d'imatges sumades, més s'aproxima la nova imatge a l'original $f(x,y)$.

$$g(x,y) = f(x,y) + \eta(x,y)$$

$$\bar{g}(x,y) = \frac{1}{M} \sum_{i=1}^M g_i(x,y)$$

A la continuació es mostra la imatge original, la imatge amb soroll gaussià, la suma de 5 imatges amb soroll i finalment la suma de 25. Veiem com quan major és el nombre d'imatges sumades més s'assembla a l'original.



Il·lustració 17: Imatge original seguida de la imatge amb soroll gaussià de sumes d'imatges amb soroll

- Filtres de mitjanes:**

- Lineals

- Mitja (passa baixa espacial): disminueix el guany de les altes freqüències. Fa el promig dels píxels de la imatge original.

$$Ma = \frac{1}{nm} \sum_{(x,y) \in W} f(x,y)$$

entenent nm com a la suma de píxels veïns escollits dins la finestra W (entenem com a finestra els píxels que es decideixen agafar per a treballar, que envolten el píxel que s'està estudiant).

- Mitja geomètrica: consisteix en fer el producte dels valors dels píxels que es troben dins la finestra.

$$Mg = \prod_{(x,y) \in W} [f(x,y)]^{\frac{1}{nm}}$$

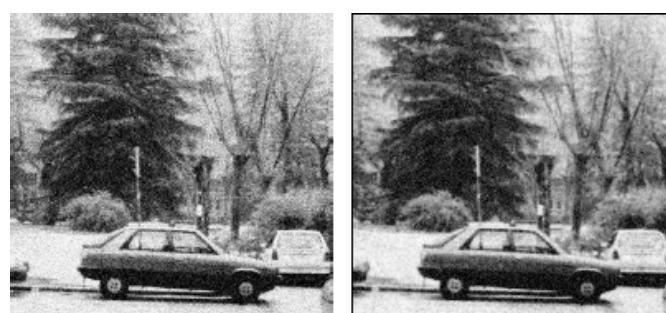
Funciona prou bé a la hora d'eliminar el soroll de tipus Gaussià, però no funciona bé pel que fa a l'impulsional.

- Mitja harmònica: s'enten com a la divisió de la mida de la finestra entre la suma de la inversa del valor dels píxels que conté.

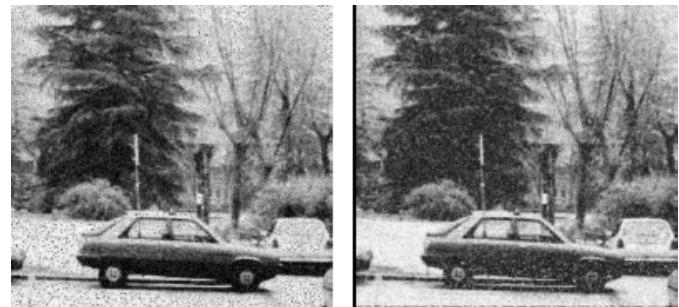
$$Mar = \frac{nm}{\sum_{(x,y) \in W} \frac{1}{f(x,y)}}$$

Funciona bé per a eliminar tant el soroll Gaussià com per a el soroll impulsional del tipus 'pebre', mentre que per a eliminar el soroll de tipus 'sal' falla.

A continuació es poden veure quatre imatges, la primera imatge amb soroll gaussià, a la de la seva dreta se li ha aplicat un filtre passa baixa espacial, a la tercera un filtre amb mitja geomètrica i a la última un filtre amb mitja harmònica.



Il·lustració 18: Imatge amb soroll gaussià, a la dreta imatges amb filtra passa baixa



Il·lustració 19: Imatge amb soroll tipus sal, i a la dreta amb filtre amb mitja geomètrica

- Mitja contra-harmònica: ve definida per la següent fórmula:

$$Mca = \frac{\sum_{(x,y) \in W} f(x,y)^{R+1}}{\sum_{(x,y) \in W} f(x,y)^R}$$

R és la tendència del valor final, i té les següents propietats:

- R > 0: elimina soroll del tipus 'pebre'
- R < 0: elimina soroll del tipus 'sal'
- R = 0: realitza la mitja aritmètica

A la imatge a continuació es veu una imatge amb soroll de tipus sal i a continuació un filtrat contra-harmònic amb R = 3.



Il·lustració 20: Imatge amb soroll tipus salt, i a la dreta amb un filtre amb mitja contra-harmònica

- De Gauss: els filtres de Gauss són màscars de convolució que imiten la campana de Gauss. Com a propietats es pot dir que té simetria rotacional (té el mateix efecte en totes

direccions), a la hora de ponderar el valor de cada píxel, decreix la ponderació amb la distància al centre, preserva baixes freqüències però elimina les altes. S'utilitza com a filtre previ a la detecció de contorns que s'usa en aquest projecte. La imatge de l'esquerra té soroll gaussià i a la de la dreta se li aplica un filtre de submatriu 5×5 i $\sigma = 1,6$.



Il·lustració 21: Imatge amb soroll gaussià, a la de la dreta se li aplica un filtre de Gauss

- No lineals:
 - Outlier: es calcula la diferència entre el valor del píxel i la mitjana dels veïns que conformen la màscara. Si es dona el cas de que la diferència entre els dos valors és major a un valor preestablert, el píxel passa a ser considerat soroll, i es substitueix el seu valor per la mitjana. Gràcies a això funciona molt bé davant soroll impulsional, millor que els filtres comentats anteriorment.

- Dins del domini freqüencial:

Malgrat que normalment s'acostuma a treballar en el domini freqüencial quan s'està treballant en el processat d'imatges, a la hora de filtrar es sol fer en el domini espacial, degut a la gran càrrega computacional que comporta filtrar en el domini freqüencial degut a la transformada de Fourier.

La transformada de Fourier és una transformació matemàtica que s'utilitza a l'hora de transformar senyals entre el domini del temps, o continu, i el domini freqüencial, o discret. D'aquesta manera permet facilitar càlculs. S'utilitza en anàlisis de senyals, en disseny de controladors i en processat d'imatges.

A continuació es poden veure els perfils més comuns en el filtrat en domini freqüencial:

- Passa baixa:



Il·lustració 22: Forma del filtre passa baixa

- Passa alts:



Il·lustració 23: Forma del filtre passa alts

- Passa banda:



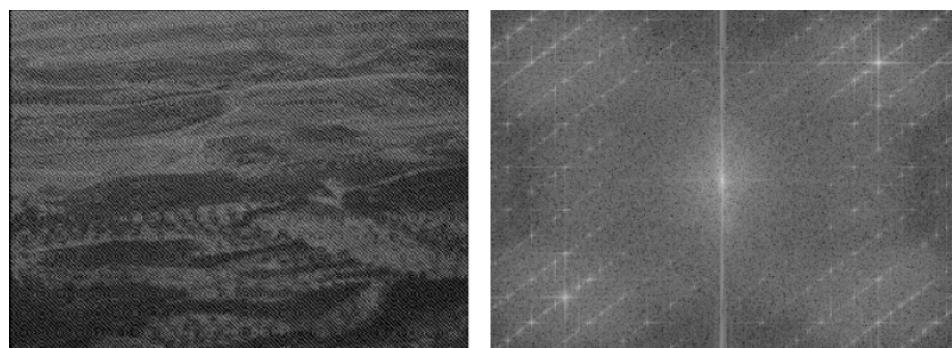
Il·lustració 24: Forma del filtre passa banda

- Rebuig banda:



Il·lustració 25: Forma del filtre rebuig banda

Aplicant la transformada de Fourier a una imatge amb soroll obtenim el següent:



Il·lustració 26: Imatge amb soroll a l'esquerra, filtrada a la dreta

6.1.4. Segmentació

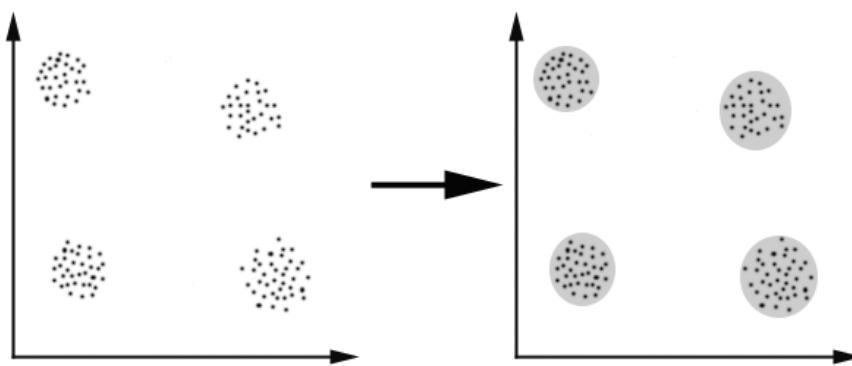
La segmentació de dades en imatges digitals consisteix en dividir una imatge en parts més petites (grups de píxels) o objectes, i té com a objectiu simplificar o seleccionar les parts que més interessin de la imatge per a fer més senzill el seu anàlisi. Això es fa etiquetant cada píxel de la imatge, de forma que els que tinguin la mateixa etiqueta tindran característiques semblants (intensitat, textura o color) [12].

És aplicable a la medicina en diagnòstics, detecció d'empremtes digitals, reconeixement de cares, sistemes de control de tràfic (detecció de matrícules amb radars), reconeixement de cares...

Mètodes de segmentació:

- Mètodes d'agrupament o 'clustering':

Entenem aquest mètode com el procés d'organitzar objectes en grups, els membres dels quals són similars en algun o alguns aspectes [13].



Il·lustració 27: Mètode d'agrupament o 'clustering'

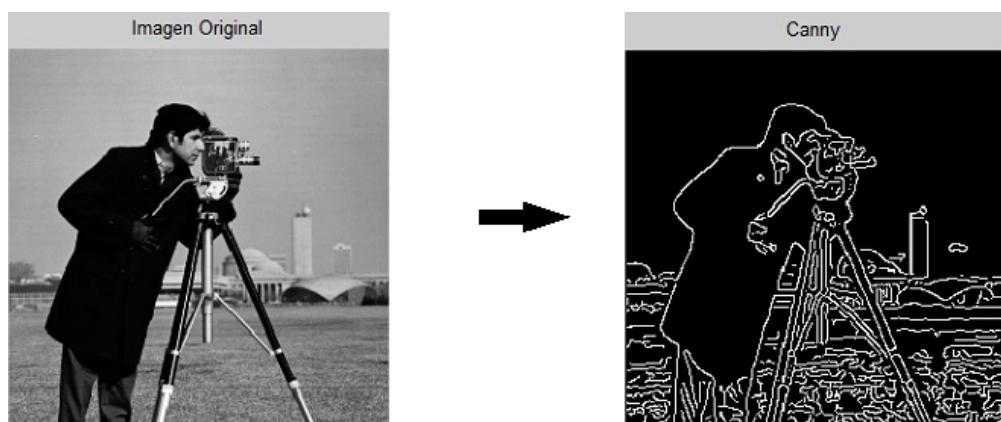
Algoritmes que s'utilitzen:

- Algoritme de 'k-means' [14]: el primer en utilitzar-lo va ser James MacQueen l'any 1967, i consisteix en seleccionar un nombre k de centres de clústers de forma aleatòria o heurística, i es divideix la imatge assignant cada píxel a un clúster, buscant una variància mínima entre ambdós. Un cop feta la primera divisió es tornen a calcular els centres dels clústers de nou a partir dels píxels assignats a cada clúster, i es tornen a assignar els píxels de nou. Es repeteix el procés fins a aconseguir la convergència. Es considera que s'ha arribat a la convergència quan els punts que intervenen ja no canvien de clúster, o quan els seus centres deixen de canviar de posició.
- Algoritme de Lloyd: aquest algoritme és una heurística per a resoldre l'algoritme de k-means d'una forma més ràpida.
- Altres: Fuzzy C-means, hierarchical clustering, mixture of gaussians.

La solució que s'obté no és una solució òptima, ja que depèn molt de les k que s'agafen i dels clústers inicials, obtenint un resultat amb múltiples interpretacions. Apart, no s'aborden tots els requisits a la vegada, i la eficàcia depèn de la definició de distància que s'estigui utilitzant. També afecta la quantitat d'elements, com més elements, més complexitat tindrà i major serà el temps de processament.

- Detecció de contorns:

Es poden definir els contorns com a transicions entre dues regions de diferents nivells de gris, els contorns ens permeten definir les fronteres dels objectes [15].



Il·lustració 28: Mètode de detecció de contorns

Tècniques utilitzades:

El operadors utilitzats es basen en la primera derivada, bàsicament mira les diferències entre els nivells de grisos. Al calcular la primera derivada en dos dimensions obtenim un vector anomenat gradient que apunta la direcció de la màxima variació, amb el mòdul proporcional a aquesta.

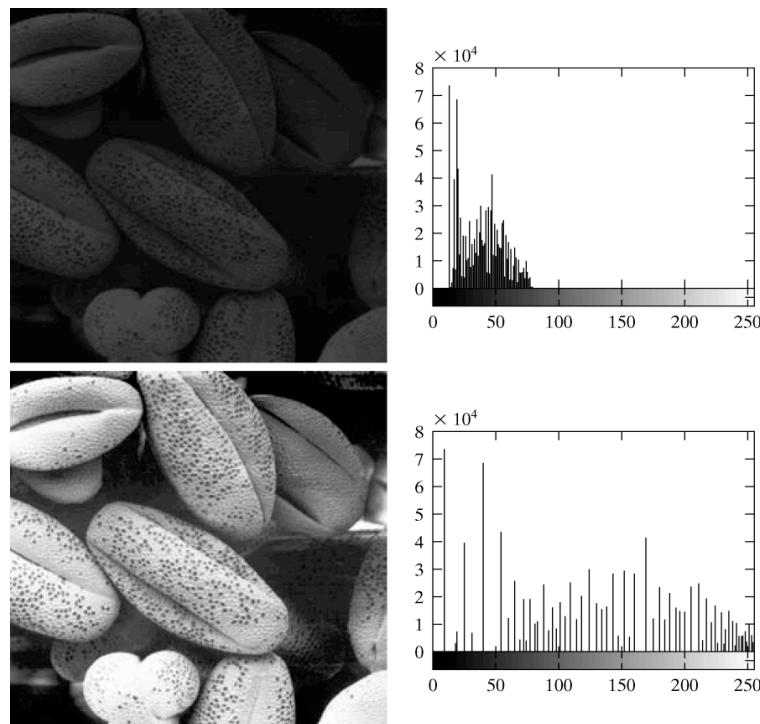
$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}; \text{ amb mòdul: } |\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

Operadors que s'utilitzen:

- Operador de Roberts: és un dels operadors més utilitzats, obté una bona resposta pel que fa a les vores que es troben en diagonal. És força fiable en quan a la localització, però és molt sensible al soroll i això en disminueix la fiabilitat.
- Operadors Prewitt, Sobel i Frei-Chen: són tres operadors força utilitzats també, es poden formular de forma conjunta ja que les màscares que utilitzen només varien entre elles en un determinat nombre.

- Mètodes basats en l'histograma:

Es pot considerar l'histograma d'una imatge com a una funció discreta que representa la freqüència relativa dels nivells de gris de la imatge, és a dir, el nombre de píxels de la imatge en funció dels nivells d'intensitat del seu gris [16]. No necessàriament ha de ser l'histograma només sobre el nivell de grisos, també es pot treure l'histograma d'una imatge en color, obtenint-ne l'histograma de cada color, vermell, verd i blau, (RGB), per separat. A continuació es pot veure, a la primera imatge, com varia l'histograma de grisos variant la il·luminació de la mateixa imatge. Mentre que a la segona veiem l'histograma de grisos, seguit del de vermell, el del verd i el del blau (RGB).



Il·lustració 29: Histogrames de dues imatges iguals amb diferents nivells d'il·luminació



Il·lustració 30: Histograma de gris i de tres colors d'una imatge a color

La comparació d'imatges a partir del seu histograma és un mètode molt eficient en comparació amb d'altres, ja que es calcula a partir de tots els píxels de la imatge.

Entenent les següents variables:

- g : són els possibles nivells d'intensitat que pot prendre un píxel.
- $P(g)$: és la probabilitat de que aparegui un determinat nivell d'intensitat g :

$$P(g) = \frac{N(g)}{M}$$

- M: és el nombre de píxels de la imatge.
- N(g): és el nombre de píxels que es troben en un determinat nivell d'intensitat g.

Propietats de l'histograma:

- Mitjana: s'entén com a mitjana el valor mig dels nivells de gris, i dona informació sobre la brillantor general de la imatge. Definida per la següent fórmula, on tenim que L és 256, el nombre total de nivells de gris que estan compresos entre el 0 i el 255:

$$\bar{g} = \sum_{g=0}^{L-1} g \cdot P(g)$$

- Variància: la variància mesura la dispersió al voltant del valor de la mitjana. Una variància alta es correspon a una imatge amb un contrast elevat, mentre que si la variància obtinguda és baixa es correspon a una imatge amb poc contrast. Ve definida per la següent fórmula:

$$\sigma^2 = P(g) \cdot \sum_{g=0}^{L-1} (g - \bar{g})^2$$

- Asimetria: mesura la simetria que té l'histograma respecte a la mitjana dels nivells de gris. Si el valor absolut que s'obté és elevat indica que tenim una asimetria elevada i viceversa. Ve descrita per la següent fórmula:

$$u = P(g) \cdot \sum_{g=0}^{L-1} (g - \bar{g})^3$$

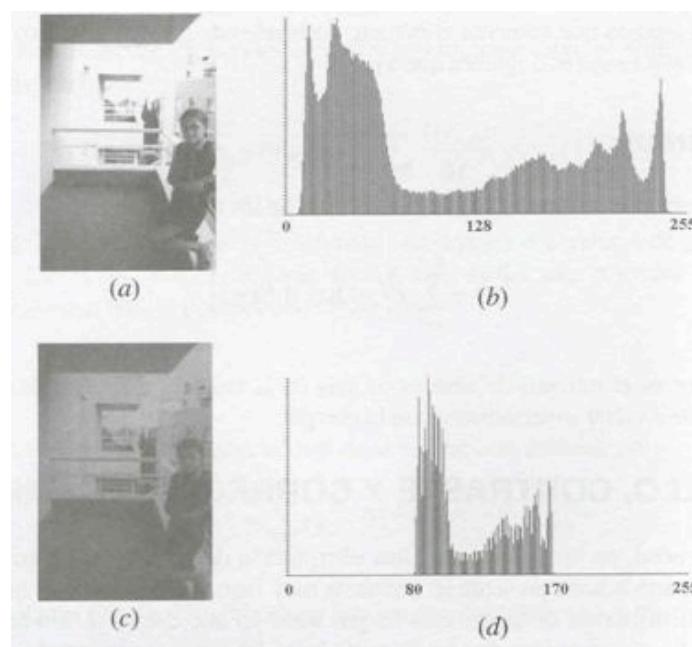
- Energia: l'energia ens dona informació sobre la distribució dels nivells de gris, com més nivells diferents de gris trobem, més petita serà l'energia de la imatge. Està compresa entre el 0 i l'1, essent l'1 el cas en el que només tenim un nivell de gris. Es dedueix de la fórmula a continuació:

$$E = \sum_{g=0}^{L-1} (P(g))^2$$

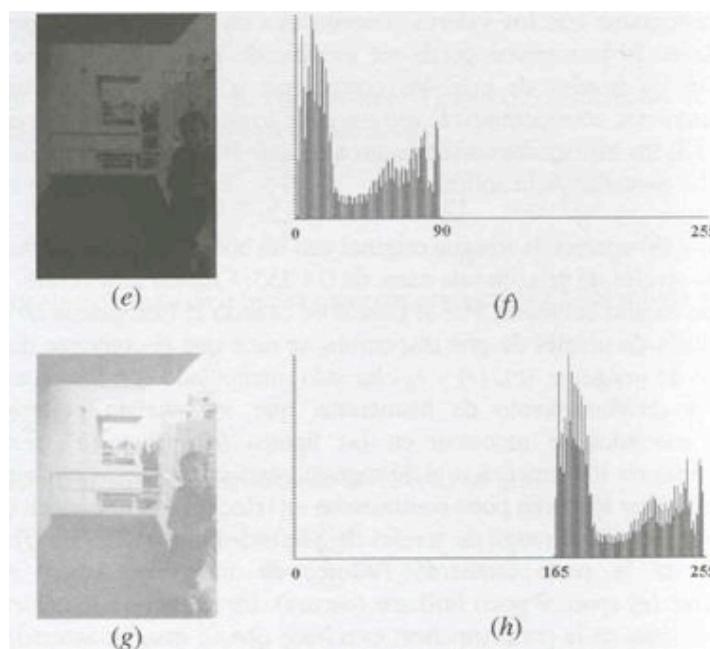
- Entropia: ens dona informació sobre la distribució de grisos com ho fa l'energia també, però de forma inversa, a major nombre de nivells de gris major serà l'entropia. S'obté de la següent fórmula:

$$e = - \sum_{g=0}^{L-1} P(g) \cdot \log_2[P(g)]$$

L'histograma es pot manipular, d'aquesta manera s'aconsegueix variar les seves propietats. Això és molt útil de cara al tractament i el processat d'imatges.



Il·lustració 31: Histogrames d'una imatge amb diferents contrastos



Il·lustració 32: Histogrames d'una imatge amb diferent il·luminació

Es pot veure que les quatre imatges anterior són la mateixa modificant aspectes com seria el contrast o la brillantor.

La primera de totes (a), és la original, es pot veure que l'histograma comprèn pràcticament tot l'interval de grisos. Es pot veure com a les tres imatges que la segueixen s'hi ha canviat tant el contrast com la brillantor. El contrast s'ha vist reduït degut a que l'amplada de l'histograma ha disminuït (imatge (c)), perdent varietat de grisos, i a menor varietat de grisos menor és el contrast. Pel que fa a la brillantor, com més a la dreta es troba la concentració de píxels (imatge(e)) més clara és la imatge i viceversa (imatge (g)).

Per a la manipulació de l'histograma, com s'ha vist a les imatges anteriors, existeixen diferents mètodes:

- Contracció de l'histograma: al contraure l'histograma es produeix una disminució del contrast.
- Expansió de l'histograma: és la operació contrària a l'anterior, al expandir l'histograma es produeix un augment del contrast.
- Desplaçament de l'histograma: s'utilitza aquesta operació quan es vol augmentar o disminuir la brillantor de la imatge, però mantenint les relacions dels nivells de gris dels píxels entre ells. Es realitza afegint o sostraient un valor fix a tots els nivells de gris de l'histograma de la imatge.

- Realçat per equalització: és una de les tècniques més utilitzades per a la millora del contrast. Al augmentar el contrast augmenta amb ell la qualitat de la imatge. Consisteix en trobar una funció tal que realci el contrast general de la imatge expandint els nivells de gris, però que a la vegada mantingui el valor de la intensitat, per això la funció ha de ser el menys agressiva possible.
 - Equalització uniforme: l'histograma es modifica de tal manera que tots els nivells de gris es reparteixen, de forma equitativa, el mateix valor dins del rang de valors establert.
 - Equalització exponencial: consisteix en distribuir els valors de l'histograma de manera que segueixin una funció exponencial determinada.
- Realçat adaptatiu del contrast: es basa en altres valors de la imatge que no sigui exactament l'histograma, com podria ser el valor de la mitjana d'intensitat dels píxels o la seva variància. Obtenint el contrast a partir de la variància, i la brillantor a partir de la mitja.
- Combinació de tècniques: consisteix en combinar les operacions sobre l'histograma comentades anteriorment amb altres operacions, com podrien ser el filtrat i la sostracció d'imatges (afegint un filtre de passa baixa).
- Èmfasis en la alta freqüència: consisteix en el mateix que l'operació anterior, combinació de tècniques, però amb un filtre de passa alts. Ambdós mètodes tenen com a objectiu principal realçar els contorns de la imatge original.

- Mètodes de creixement de regions:

Aquest mètode consisteix en definir regions, on cada regió és un objecte que es vol segmentar [17]. Per a començar el procés es defineixen unes 'llavors', entenent per llavor un píxel o petita zona que s'obté com entrada de la funció. Cada 'llavor' es fa créixer mirant els píxels que la rodegen, i agrupant els que no pertanyen a cap regió prèvia, en funció de si compleixen unes condicions determinades d'assignació al objecte en qüestió.

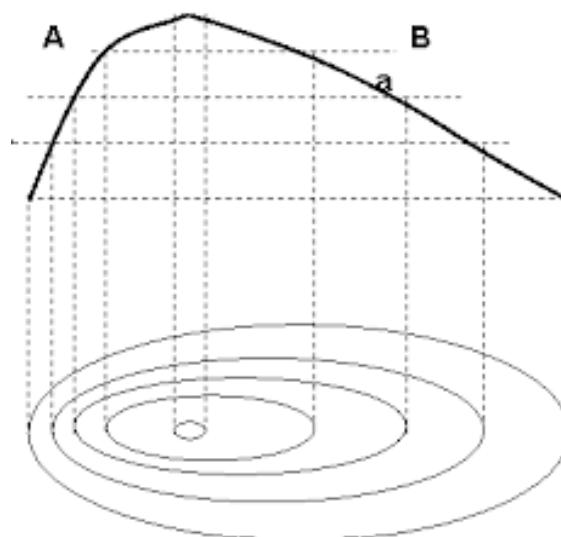


Il·lustració 33: Imatge original, i a la seva dreta aplicat el mètode de creixement de regions

Els resultats obtinguts amb aquest mètode depenen en gran mesura de la posició inicial de les ‘llavors’, però l’existència de soroll pot fer que quedin mal col·locades, obtenint una solució no òptima.

- Mètode del conjunt de nivell:

El mètode és una tècnica numèrica basada en la propagació de corbes i superfícies a partir de factors intrínsecos com la curvatura de les regions segmentades, o extrínsecos, com la intensitat o la textura de la imatge en qüestió [18]. La parametrització de la corba es duu a terme mitjançant Lagrange. Se sol aplicar en mapes per a les corbes de nivell, d'aquesta manera es pot, simbòlicament, donar-li tres dimensions a una imatge o un mapa que només en té dos.



Il·lustració 34: Representació del mètode conjunt de nivell

Es resol pel mètode de diferències finites. Aquest té diferents avantatges, el principal és que al ser implícit no té paràmetres d'entrada, a part, és una manera directa per a estimar les propietats, es pot canviar de tipologia i és intrínseca. Pot utilitzar-se per a definir un marc d'optimització.

La solució és molt convenient per a fer front a diverses aplicacions de visió artificial i s'aplica molt en anàlisis d'imatges mèdiques, les implementacions són molt eficients però tenen certes limitacions.

- Mètodes de partició gràfica:

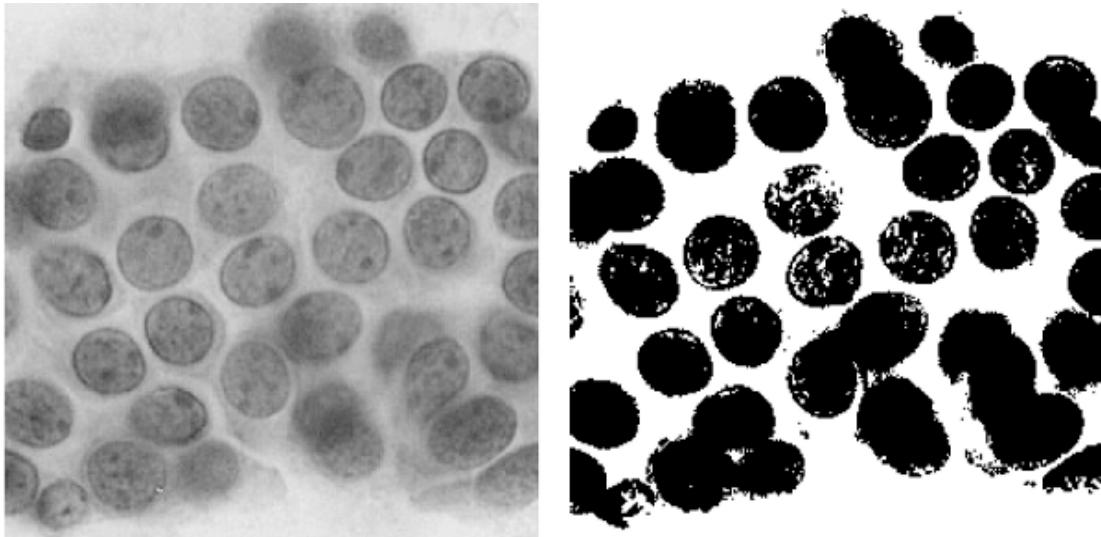
La imatge es modela com un graf ponderat no dirigit [19]. Un píxel o un grup de píxels s'associen amb nodes i els pesos de les arestes generades defineixen la similitud amb els píxels veïns. Cada partició de nodes dona com a sortida d'aquests algoritmes els objectes que la imatge pot contenir.

Alguns dels algoritmes utilitzats:

- Talls normalitzats
- Camí aleatori
- Tall mínim
- Partiment isoperimètric
- Arbres d'expansió mínima

- Mètode de transformació divisòria o conca('Watershed'):

Aquest mètode calcula les línies divisòries d'aigües, interpretant el nivell de gris d'un píxel com a una altura en un relleu, anomenat relleu topogràfic [20]. També es pot fer mitjançant el gradient, els que tenen major gradient corresponen a línies divisòries.



Il·lustració 35: Imatge d'aplicació de 'watershed'

'L'aigua' baixa cap als punts de menys intensitat de gris per un camí amb un mínim local en comú, omplint una conca que defineix un segment o objecte. Això permet la segmentació d'estructures complexes que no es poden processar mitjançant altres mètodes convencionals.

- Mètode del valor llindar:

El mètode consisteix en fixar un valor llindar o límit, de tal manera que els píxels dins del llindar formaran objecte al segmentar [21]. A la següent imatge s'està aplicant un llindar mínim de 128. Quan el valor del píxel és major es pinta de negre, sinó passa a ser blanc. Aquest procés de classificar els píxels en blanc i negre s'anomena binarització.



Il·lustració 36: Imatge aplicant el mètode del valor llindar

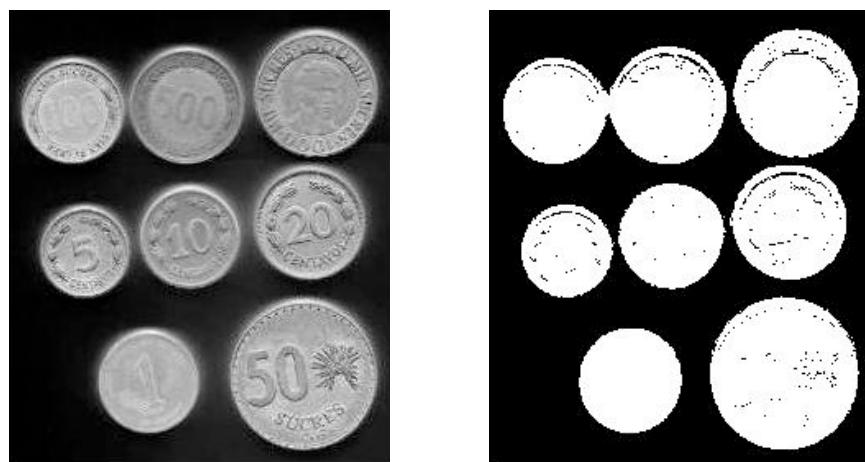
Hi ha diferents tipus de binarització:

- Llindar únic (T): només es troben dues agrupacions de píxels semblants, un objecte i un fons, si el valor del píxel és major que T passa a formar part de l'objecte, mentre que els píxels que són inferiors passen a formar part del fons.
- Llindar multinivell (T_i, T_j): com que es tenen un nombre X d'objectes es necessitaran $X-1$ intervals. Definint $T_i < \text{píxel} < T_j$, on cada interval representa una classe d'objecte.

- Segmentació basada en models:

Bàsicament aquest tipus de segmentació es basa en que els elements d'interès que s'estan buscant dins la imatge tenen tots una estructura similar en la forma o una geometria repetitiva. D'aquesta manera permet saber prèviament el que s'està buscant i acotar a aquests paràmetres la busca. Per a calcular els paràmetres s'utilitza un model probabilístic que pugui explicar les variacions que puden tenir les estructures que s'estan buscant. S'han de seguir els següents passos:

- Selecció dels exemples d'entrenament a partir dels quals es realitzaran les proves per a veure si funcionen correctament els models.
- A partir dels exemples d'entrenament s'extreu un model probabilístic de les variacions que es poden trobar en les estructures estudiades.
- Provar la inferència que té el model en la imatge.



Il·lustració 37: Imatge aplicant la segmentació basada en models

- Altres tipus de segmentació:

- Segmentació multi-escala:

La segmentació multi-escala funciona per sectors. Consisteix en ajuntar les zones semblants en funció de l'escala de l'element que les separa. Es pot anar augmentant l'escala fins a simplificar suficientment la imatge.

- Segmentació semi-automàtica:

En aquest tipus de segmentació, el mateix usuari s'encarrega de definir les regions d'interès que es volen estudiar mitjançant el ratolí.

- Xarxes neuronals de segmentació:

Es basen en segmentar regions molt petites de la imatge, i guardar aquesta informació en una xarxa neuronal artificial o un conjunt de xarxes neuronals. Un cop realitzat aquest procés es construeix un mecanisme que marca les àrees d'una imatge d'acord amb els patrons trobats en la xarxa neuronal. Un tipus de xarxa dissenyada expressament per a això és el mapa de Kohonen.

6.1.5. Reconeixement

Consisteix, un cop la imatge s'ha segmentat, en captar la informació o les característiques dels objectes que s'hi han trobat, per a posteriorment descartar-los o no, simulant la visió dels éssers humans [22].

- Mètodes de reconeixement:

- Mètodes basats en l'aparença: són els mètodes que utilitzen imatges d'exemple anomenades plantilles o prototips.

- Aparellament de contorns
 - Divisió en cel·les
 - Aparellament per escala de grisos
 - Aparellament per gradient
 - Histograma de les respostes del camp d'estudi

- Col·leccions de models geomètrics
- Mètodes basats en les característiques: S'usen per a buscar aparellaments entre les característiques d'un objecte i les de la imatge de referència:
 - Arbres d'interpretació
 - Conjectura i prova
 - Consistència de la postura
 - Agrupament per postura
 - Invariància
 - Valoració geomètrica
 - Característica d'escala invariant transformada
 - Característiques principals accelerades

6.2. Eines estudiades per a aplicar el mètode

Un dels requisits que demanava l'empresa és la implementació del mètode mitjançant el llenguatge Python de programació, ja que és el que utilitzen ells mateixos i facilita la integració de l'aplicació en el seu sistema, dissenyat amb Python.

Apart d'això, utilitzar el llenguatge Python té altres avantatges, és el programari que s'utilitza a la universitat, per tant no es parteix de zero en quan als coneixements sobre el seu funcionament. És un llenguatge molt llegible i intuïtiu, a part de ser de programari lliure, que implica que és codi obert i no és de pagament. A més, hi ha molta documentació i molta informació, com poden ser llibreries i mòduls.

Al començar a cercar sobre llibreries de Python relacionades amb el tractament d'imatges apareixen dues de seguida: la PIL o PILLOW i OpenCV.

La llibreria PIL està encarada a obrir, manipular (afegir màscares, filtres, afegir text..) i guardar imatges, es tenia coneixença d'aquesta llibreria degut a que ja s'havia utilitzat prèviament dins de la universitat.

Malgrat això, si s'investiga una mica més sobre el processat d'imatges per internet de

seguida apareix la llibreria OpenCV, de la qual a tot arreu se'n parla meravelles. I ha estat l'escollida per a començar a implementar l'eina.

- Llibreria OpenCV:

La llibreria OpenCV [23] és una llibreria lliure de visió artificial, desenvolupada originàriament per Intel. S'utilitza tant per a la detecció de moviment en sistemes de seguretat com en aplicacions on és necessari el reconeixement d'objectes. És una llibreria multi-plataforma, que vol dir que hi ha versions per a Linux, Mac i Windows, i, per tant, és accessible per a tothom. Disposa de més de 500 funcions, moltes d'elles encarades a processos de visió artificial, com ara el reconeixement d'objectes.

Es publica sota la llicència BSD (Berkeley Software Distribution), que permet que sigui utilitzada lliurement per a investigació o propòsits comercials.

Aplicacions:

- Com a aplicació genèrica, s'utilitza en sistemes de vigilància de vídeo.
- Una aplicació més concreta és la seva utilització en el sistema de visió del vehicle Stanley de la universitat de Stanford.

6.2.1. Idea inicial

La primera idea era la d'intentar buscar una imatge a partir d'una icona donada seguint els següents passos:

- 1- A partir de la icona donada trobar característiques que poguessin definir-la, com per exemple nº de costats o forma (per exemple: 3 costats si és un triangle com és el cas del símbol de play), mida i color. Un cop trobades les característiques, guardar-les en una base de dades de tal manera que quedi la informació extreta assignada a cada icona, i no s'hagi de perdre temps buscant les característiques cada vegada.
- 2- Buscar un símbol a la imatge de la pantalla, que contingui les mateixes característiques o unes de similars. I que, donat un valor llindar, determini amb més o menys exactitud la icona buscada.

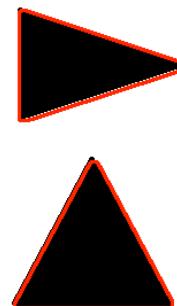
➤ Funcions implementades:

Les primeres funcions implementades, que es comentaran a continuació, apliquen totes una funció que canvia el model de color a HSV per a poder facilitar el treballar

amb les imatges. S'aplica un filtre de color aplicant el mètode del valor llindar tant per a H com per a S i per a V. D'aquesta manera s'aconsegueix segmentar o focalitzar els esforços en les figures que interessa, obviant la resta. Un cop feta la part de filtratge i primera part de segmentació s'aplica la detecció de contorns, obtenint els següents resultats.

- Detecció de triangles [24]:

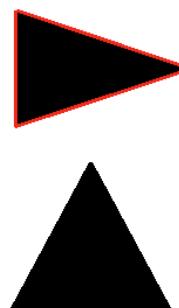
A partir d'un filtre de color, detecta les figures que tenen tres costats. Com a inconvenients es té que no sempre funciona amb exactitud, (al provar-ho amb el play en pantalla no el detectava), i si altres coses com lletres. Apart, resulta complicat determinar els intervals de funcionament.



Il·lustració 38: Triangles detectats

- Detecció de triangles segons inclinació:

Afegint una petita modificació a l'anterior es pot aconseguir que només agafi els triangles que tenen un costat vertical, d'aquesta el que s'aconseguiria és tenir un filtre previ que permeti detectar els triangles més semblants a la icona de 'Play'.



Il·lustració 39: Triangle amb costat vertical detectat

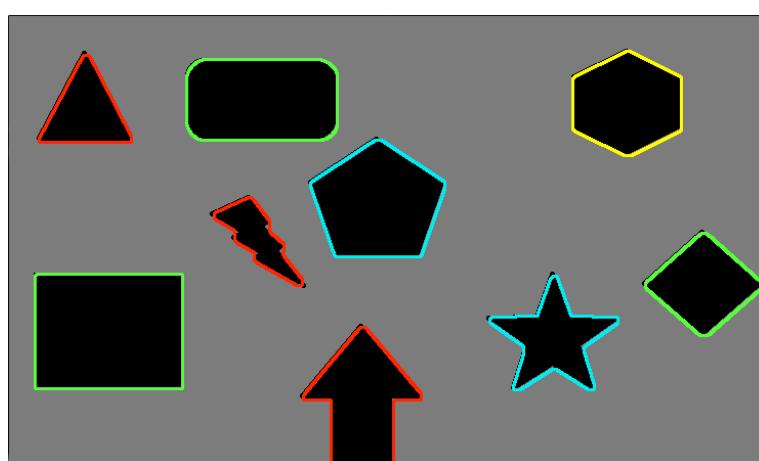
- Detecció de figures segons color i costats:

Modificant la funció anterior aconseguim trobar figures geomètriques filtrant per color i nombre de costats. Pot detectar també figures com quadrats o pentàgons, només s'ha de canviar la condició del nombre de costats. Malgrat això, els problemes són els mateixos que amb la funció anterior.

- Nombre de figures detectades i costats:

A partir de l'anterior es pot fer una funció que detecti el nombre de figures geomètriques que tenim en pantalla i el nombre de costats que té cada una. Això permetria veure tot el que troba per a després poder filtrar el que es vulgui, però els problemes trobats són els mateixos que en les dues funcions anteriors.

S'obtenen uns resultats com aquests:



Il·lustració 40: Múltiples figures amb diferent nombre de costats detectades

On la funció retorna el nombre de figures trobades, els punts dels vèrtex i marca cada figura d'un color en funció del nombre de costats (el vermell s'utilitza per triangles i per 7 o més costats).

- Detecció de cercles [25]:

Detecta el nº de cercles que apareixen en la imatge de la pantalla, però s'ha de regular la precisió de la funció que els busca, perquè dona molts errors, i no sempre detecta els mateixos cercles malgrat la imatge sigui la mateixa.

- Complicacions o dificultats trobades:

- Com es pot determinar quines característiques agafar de cada icona donada?
- Com es pot determinar el rang d'exactitud que es vol buscar o trobar?
- Es necessiten funcions diferents per a cada tipus d'ícone, que fa que sigui un algoritme molt específic i, el que es busca, és que sigui el més general possible.

6.2.2. Segona idea

Abans de tenir temps de pensar en una resposta o de trobar una solució als problemes acabats de plantejar s'ha trobat un nou mètode, utilitzant la funció 'match template' [26]. El que fa aquesta funció és directament comparar dues imatges, buscant una dins de l'altre. Agafa la imatge que busca, en el nostre cas la ícone, i la col·loca a la cantonada de dalt a l'esquerra, a partir d'aquí avança píxel a píxel cap a la dreta i cap a baix, comparant cada secció que fa de la imatge amb la de referència utilitzant 6 mètodes diferents:

- **CV_TM_SQDIFF (squared difference):** defineix la desviació entre dos colors calculant-la mitjançant la suma dels quadrats de les diferències entre els valors dels píxels pertanyents a les imatges que s'estan comparant [27].

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

- **CV_TM_SQDIFF_NORMED (normalized squared difference):** es calcula de la mateixa manera que l'anterior, però el resultat queda dividit pel seu mòdul.

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

- **CV_TM_CCORR (cross correlation):** aquest mètode s'utilitza per a veure el grau de semblança entre dues sèries. En aquest les sèries estan formades pels valors dels píxels que conformen cada imatge [28].

$$R(x, y) = \sum_{x', y'} (T(x', y') \cdot I(x + x', y + y'))$$

- **CV_TM_CCORR_NORMED (normalized cross correlation):** el mètode és el mateix que l'anterior però dividint el resultat pel seu mòdul.

$$R(x, y) = \frac{\sum_{x',y'} (T(x', y') \cdot I(x + x', y + y'))}{\sqrt{\sum_{x',y'} T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}}$$

- **CV_TM_CCOEFF (correlation coefficient):** el coeficient de relació mesura el grau amb el que els moviments de dues variables, en aquest cas la variació de les intensitats dels píxels de les dues imatges, estan relacionats entre ells [29].

$$R(x, y) = \sum_{x',y'} (T'(x', y') \cdot I(x + x', y + y'))$$

- **CV_TM_CCOEFF_NORMED (normalized correlation coefficient (NCC – Fast Normalized Cross-Correlation):** és el mateix que l'anterior dividint el resultat pel seu mòdul.

$$R(x, y) = \frac{\sum_{x',y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x',y'} T'(x', y')^2 \cdot \sum_{x',y'} I'(x + x', y + y')^2}}$$

On:

$$\begin{aligned} T'(x', y') &= T(x', y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'', y'') \\ I'(x + x', y + y') &= I(x + x', y + y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x + x'', y + y'') \end{aligned}$$

On es té que R és el resultat de comparar T ('template', imatge que estem buscant) amb la regió de la I ('Image', imatge on estem buscant el 'template') on es troba en aquest moment, píxel a píxel.

- Funcions implementades:

- Match template:

És la funció principal a partir de la qual s'ha dissenyat el mètode [30]. El que fa la funció és aplicar la funció 'match_template()' de la llibreria OpenCV utilitzant cada vegada un mètode diferent dels anomenats anteriorment. Cada mètode retorna una 'array' amb el rang de coincidència que ha trobat en cada posició. El mètode que utilitza és anar avançant per la fila, d'esquerra a dreta, i per columnes de dalt a baix, (com hem comentat

anteriorment), obtenint una llista de llistes, on la primera llista és la primera fila, la segona llista és la segona fila etc., i essent cada element de la llista una columna diferent. D'aquesta manera es pot identificar fàcilment la posició on el nivell de coincidència és més elevat, a partir dels índex de la llista.

L'explicació de perquè s'utilitzen els 6 mètodes és degut a que un cop realitzades diferents proves, s'ha detectat que no tots els mètodes troben la mateixa imatge com a la que més coincideix, apart, n'hi ha que fallen més que d'altres. D'aquesta manera s'aconsegueix una major fiabilitat.

Versió 1: En un inici aquesta funció només agafava la imatge que més s'assemblava a la original, el problema de fer-ho d'aquesta manera és que podia ser que no l'hagués de trobar, però al agafar sempre la imatge que més s'assembla, cabia la possibilitat de que la imatge més semblant trobada no fos la icona buscada, o que la icona ni tan sols aparegués en pantalla.

Versió 2: En la segona versió, es decideix aplicar un llindar, de tal manera que, quan el valor de la imatge que troba com a més semblant no és superior al llindar, no l'agafi. D'aquesta manera s'aconsegueix filtrar els resultats que siguin erronis.

Versió 3: També es pot donar el cas de que aparegui més d'una imatge igual en pantalla i només n'agafi una quan les hauria d'agafar totes. Per això s'ha decidit en aquesta nova versió que a partir d'un rang, es recopilin totes les imatges trobades, i després es passaran a comprovar una per una, comparant-les directament amb la imatge de referència, ('template').

El resultat es dona en forma d'un diccionari, com a clau es tindrà el nom del retall, i com a resultat es tindrà una llista de retalls de la imatge original o màster. Es donarà també la posició de la part superior esquerra i la posició de la part inferior dreta per a, més endavant, poder saber la posició de la icona, d'aquesta manera s'aconsegueix facilitar la integració del mètode en el sistema actual. A part, es calcularà la semblança amb la que ha trobat cada retall en tant per 1, d'aquesta manera es podran comparar tots els retalls en la mateixa escala. Per a calcular el tant per 1 s'ha d'estudiar cada mètode prèviament, ja que els tres normalitzats et donen el tant per 1, però els tres restants et donen el valor absolut calculat com s'explica en la funció extreure rang.

A partir d'aquesta funció s'han dissenyat dues funcions més amb petites variacions:

- Match template one method:

Aquesta funció fa el mateix que la anterior, però només utilitza un dels mètodes, que ve indicat per l'usuari.

- Multiple match template:

Aquesta funció el que fa és molt semblant a la primera, però està dissenyada d'una forma diferent. En comptes de calcular el rang a partir de l'extreure rang i utilitzar una fórmula diferent per a cada funció per a calcular si es troba o no dins el llindar establert, ho fa a partir d'un valor límit i de l'array obtinguda inicialment.

- Extreure rang:

Per a poder calcular els tants per u de cada mètode, i per a poder establir el valor que haurà de superar cada retall per a ser seleccionat, s'ha implementat la següent funció.

Cada mètode retorna un valor. En els mètodes on es busca la diferència entre les dues imatges, SQDIFF i SQDIFF_NORM, es vol que el resultat sigui mínim, mentre que en els altres quatre, es busca que la semblança sigui màxima. En el cas dels mètodes normalitzats són fàcils de definir, ja que el seu valor es trobarà entre el 0 i l'1. Per als mètodes sense normalitzar, s'obtenen valors molt diferents. Per a determinar els valors límit amb que comparar l'array de valors de coincidència que s'ha comentat abans, s'ha implementat una funció anomenada extreure_rang_pattern(). El que fa aquesta funció és determinar els màxims valors de coincidència que es poden trobar a la pantalla, que es donarien en el cas de que en pantalla aparegués la imatge idèntica. Això s'aconsegueix comparant la imatge amb ella mateixa utilitzant el match_template. Per a determinar el valor mínim que es vol complir i poder determinar el rang, es posarà com a condició que el valor obtingut al comparar les imatges sigui major o igual que p^*r . (essent p la proporció o nivell d'exactitud que volem i r el resultat obtingut de la funció extreure_rang).

En els casos en que es busca la diferència entre les dues imatges, es vol que el resultat sigui mínim. Es troba un problema en el cas de SQDIFF, i és que al comparar-la amb ella mateixa el resultat és 0, ja que mesura la diferència entre les dues imatges i aquestes són iguals. Per tant es perd la referència de sobre quins valors ha de voltar el resultat quan es compara amb una

imatge que no sigui exactament igual. S'ha optat per a crear dues imatges de la mateixa mida que la imatge referència, una tota blanca i l'altra tota negra, d'aquesta manera s'obté el valor de la màxima diferència que es pot trobar entre dues imatges, i a la hora de comparar, en comptes de buscar que sigui com a mínim una proporció p (ja que al multiplicar $p \cdot 0 = 0$, i es seguirà sense tindre una referència), es buscarà que sigui com a màxim una proporció d' $(1-p) \cdot r$ (essent r el resultat de la màxima diferència).

- Comparar imatges histograma màxim:

Una vegada es tenen el seguit de retalls que es troben dins del diccionari obtingut amb el ‘match template’, on es troba la llista de retalls que compleixen que es troben dins del rang determinat anteriorment, el que es fa és comparar cada retall amb la imatge de referència mitjançant el càlcul i posteriorment la comparació dels seus histogrames [31]. Per a dur a terme aquesta tasca s'utilitza la funció compareHist() [32]. Un cop comparades totes les imatges amb la imatge de referència, la funció et retorna la imatge que més s'assembla, així com la seva posició.

6.2.3. Classes

A aquestes alçades del programa, s'arriba a la conclusió que generar-ho tot en forma de classes facilitarà la seva utilització.

- **Primera classe:** Per a començar s'establirà la classe Imatge, que s'utilitzarà per a fer operacions sobre la icona i per extreure'n informació. Està composta per les següents funcions:
 - Funció show:
S'utilitza per a mostrar la imatge en pantalla.
 - Funció save:
S'utilitza per a guardar la imatge.
 - Funció load grey:
S'utilitza per a carregar la imatge dins el programa en escala de grisos.

- Funció extreure rang:

S'utilitza per a extreure els rangs que s'han comentat anteriorment d'una imatge.
- Funció retallar imatge:

S'utilitza per a agafar la part de la imatge que queda dins les posicions donades per l'usuari, x_1, x_2, y_1, y_2 .
- Funció copiar imatge:

S'utilitza per a fer una copia de la imatge donada.
- Funció copiar imatge grey:

S'utilitza per a realitzar una còpia de la imatge en escala de grisos.
- Funció forma imatge:

S'utilitza per a extreure de la imatge les mesures d'amplada i alçada.
- **Segona classe:** Un cop acabada la primera classe es procedeix a crear la segona, anomenada Master, composta per les següents funcions que ja s'han comentat anteriorment:
 - Funció múltiple match template
 - Funció match template all methods
 - Funció match template one method
 - Comparar imatges histograma màxim

6.2.4. Altres funcions

Un cop implementades les classes es comencen a llançar diferents tipus de proves i a pensar com es podria millorar el programa per a que funcioni millor.

Un cop es comencen a realitzar proves, apareix un nou problema:

Es dona un cas concret quan s'està estudiant el 'dropdown' o desplegable de Media, on es busca quina de les fonts és la que està activa, on la diferència entre estar activa o no és

realment molt petita, prou perquè no es pugui detectar amb el mètode anterior.



Il·lustració 41: 'Dropdown' de media desplegat

Com es pot veure a la imatge, en aquest cas estaria activa la font 'TARJETA SD2', i la única diferència és que està envoltada per una fina línia vermella, però al passar a escala de grisos, la diferència és mínima, i l'eina no la detecta. Per tant, es decideix reestructurar les classes de tal manera que puguin actuar tant amb color com sense, ja que fins ara s'ha estat treballant en escala de grisos per a facilitar els càlculs.

- Primera idea:

La primera idea que apareix, és la modificar la classe Imatge per a poder treballar amb imatges en color afegint la següent funció:

- Funció load color:

S'utilitza per a carregar la imatge dins el programa en color.

Un cop modificada la classe Imatge la idea és afegir a la classe Master una funció que compari els histogrames de les imatges en color, però de seguida es complica la cosa, ja que s'haurien de calcular quatre histogrames per cada imatge, l'histograma de grisos i tres histogrames de color per a vermell, verd i blau, i, un cop calculats, comparar-los respectivament un a un. Amb aquesta solució, s'augmentaria significativament la càrrega computacional, i amb aquesta el temps de processat.

- Segona idea:

Seguint amb la idea de comparar les imatges amb color, es troba una funció a la llibreria OpenCV anomenada cv2.absdiff() [33], que directament resta dues imatges píxel a píxel, i un cop restades, s'agafa la que sigui menys diferent. Això es durà a terme implementant una nova funció anomenada:

- Funció restar imatges

Aquesta funció compara la llista de retalls amb la imatge de referència restant les imatges. S'ha de tenir en compte que per a restar dues imatges han de ser de la mateixa mida exacta.

D'aquesta manera, s'obté un nou mètode per a realitzar les comprovacions. És força interessant utilitzar-la en el cas de que es necessiti realment molta exactitud o una fiabilitat molt elevada.

Basada en aquesta s'ha generat una altre funció anomenada:

- Funció restar imatges dropdown:

Aquesta funció és una adaptació de l'anterior però amb una funcionalitat molt més concreta, que seria la de veure si en el 'dropdown' la font que es busca està activa o no. Això es farà comparant la imatge trobada amb dues de referència, una marcada i l'altra no, de tal manera que la que més s'assembla serà la que és real. En el cas de que s'assembli més el retall a la marcada voldrà dir que la font està activa i viceversa. A continuació es pot veure les dues icones amb les que es compararien els retalls obtinguts en el cas en que s'estigués treballant amb l'exemple anterior.



Il·lustració 42: Font SD2 activa



Il·lustració 43: Font SD2 inactiva

- Nova funció:

Apart de les funcions anteriors, totes basades en el OSR (Optical Symbol Recognition), també s'ha investigat una mica sobre l'OCR. L'OCR és el que actualment es coneix com a reconeixement òptic de caràcters (Optical Character Recognition). Aquest és un procés que té com a objectiu digitalitzar textos automàticament detectats a partir d'una imatge. Ha pres molta importància en els últims anys en el tema de la indústria, ja que al reconèixer textos a partir d'imatges es redueix el nombre de recursos humans necessaris. Es pot posar com a exemple els radars de les carreteres, on a partir de la imatge es reconeix i llegeix automàticament els nombres i les lletres de la matrícula del vehicle.

S'ha creat la següent funció, a partir de la llibreria Tesseract-OCR [34].

- Funció pytesseract ocr:

Aquesta funció retorna un 'string' amb el que reconeix com a text dins de la imatge donada. No acaba de funcionar del tot correctament com es veurà en l'exemple a continuació, tot i així pot ser força interessant de cara a veure si apareix un text determinat en una imatge. A continuació es pot veure un exemple i el que s'ha obtingut en aplicar la funció:



Il·lustració 44: Font SD2 activa en el 'Dropdown'

En aquesta imatge s'ha llegit el següent:

@ CD \$3 TARJETA SD 1

2*3 TARJETA SD Z ~&~ USBI

~g~ USB 2 9 AUDIO BT

Aux AUX

Es pot veure com no llegeix la part que està més fosca, en part això és bo, ja que d'aquesta manera s'alleugera el resultat. Per altra banda es pot veure com apareixen símbols en la imatge com a lletres en la resposta. Aquest pot ser un problema en el cas de que interessi que el text aparegui idènticament en pantalla, però en el cas de que busquem si per exemple apareix 'TARJETA SD1' funciona prou bé. En el cas de que en un futur prengués més rellevància el text que es pot trobar en la imatge fóra bo profunditzar més en aquesta funció, de moment queda com a una funció creada, que funciona, però pendent de d'ajustar.

7. Resultats

7.1. Estructura dels resultats

Un cop s'han implementat les classes que permetran buscar una icona en una pantalla es procedeix a fer una sèrie de proves per a estudiar la fiabilitat d'aquest mètode.

Per començar s'han triat una sèrie d'ícones amb les que es vol comprovar si apareixen en una sèrie de màsters definits. La intenció és a partir d'un script comprovar si detecta cada icona en cada màster per a cada mètode diferent. D'aquesta manera es pot analitzar l'efectivitat de cada mètode, o si un mètode funciona millor en un tipus de icona i l'altre en un altra.

El nom dels màsters s'ha pensat de forma que contingui dins el nom de les icones que s'haurien de trobar, ve definit de la següent manera per tal de facilitar l'anàlisi posterior:

MEDIAMAIN_MIXOFF_REPEATTITLE_PLAY_SKIPFORWARD_SKIPBACKWARD_MUTE

En aquest màster s'haurien de trobar les següents icones:

- MEDIAMAIN
- MIXOFF
- REPEATTITLE
- PLAY
- SKIPFORWARD
- SKIPBACKWARD
- MUTE

REAR_WHEEL_CAR_STATE_BATTERY_4_COBERT_2_MUTE_CHECK_WITHOUT_BOX

I en aquest les següents:

- REAR_WHEEL_CAR_STATE

- BATTERY_4
- COBERT_2
- MUTE
- CHECK_WITHOUT_BOX

S'ha pensat de tal manera que cada icona surti com a mínim una vegada al llarg de tots els màsters, per a poder-les estudiar totes. Per a cada icona buscada en cada màster s'obté un valor entre el 0 i l'1, que ens indica el grau de semblança que ha trobat entre la icona original i la trobada finalment. Per a fer l'anàlisi s'ha d'establir un líindar mínim a partir del qual es considera que la icona trobada és prou semblant a l'original. Un cop definit el líindar la idea és anar-lo modificant i mirar si els errors es minimitzen o augmenten per a trobar el valor del líindar òptim.

És important tenir clar el líindar que s'utilitzarà, ja que si és molt elevat i les icones no es troben idèntiques en el màster pot no trobar icones que hauria de trobar, però tanmateix, com més petit és el líindar, major és el temps de processat del mètode. Un dels objectius d'aquest anàlisi de resultats és intentar trobar el valor del líindar que permeti trobar les icones que busquem amb un nombre d'errors molt baix i a la vegada intentar minimitzar-ne el temps de processat.

Per a facilitar l'anàlisi de resultats aquests es representaran en excels de les següents maneres:

- Una taula de resultats per a cada mètode on s'indica el valor de semblança amb el que ha trobat la icona en el màster si la ha trobat, i en el cas de que no l'hagi trobat retorna 0.
- Una taula amb la mateixa estructura que l'anterior, però en comptes de retornar els valors de si s'ha trobat o no, retorna 'PASS' o 'FAIL' dependent de les següents condicions:
 - Si l'ha trobat i l'havia de trobar: 'PASS'
 - Si l'ha trobat però no l'havia de trobar: 'FAIL'
 - Si no l'ha trobat i no l'havia de trobar: 'PASS'
 - Si no l'ha trobat i l'havia de trobar: 'FAIL'

D'aquesta manera es pot obtenir una idea de la fiabilitat que té el mètode.

- Una nova taula on ens indica per cada valor del llindar, de mínim a màxim(que és quan pren valor 1), el nombre de 'FAILS' corresos per cada mètode. A partir d'aquesta taula es pot extreure per quins valors del llindar és més efectiu el mètode.
- Una sèrie de carpetes anomenades 'nom de la icona_en_nom del màster', per exemple:'PAUSE_in_MEDIAMAIN_MIXOFF_REPEATALBUM_PAUSE_SKIPFORWARD_SKIPBACKWARD', on dins de cada carpeta es troba el resultat obtingut per cada mètode representat com a una imatge. Per tant en cada carpeta es podran arribar a tenir fins a 6 imatges, seria el cas en el que es troben amb tots els mètodes, o cap si no s'ha trobat amb cap. En el cas de trobar-se, el resultat apareix com a la imatge del màster i el lloc on ha trobat la icona encerclat. Per exemple en el cas de buscar la icona 'PAUSE' en el màster 'MEDIAMAIN_MIXOFF_REPEATALBUM_PAUSE_SKIPFORWARD_SKIPBACKWARD', es generaria una carpeta amb el nom comentat anteriorment, i en el cas de trobar la icona 'PAUSE' en el màster, apareixeria una imatge com la següent, on es pot veure encerclada la icona del 'PAUSE' de color blau.



Il·lustració 45: Context Mèdia amb icona PAUSE trobada

7.2. Primeres proves

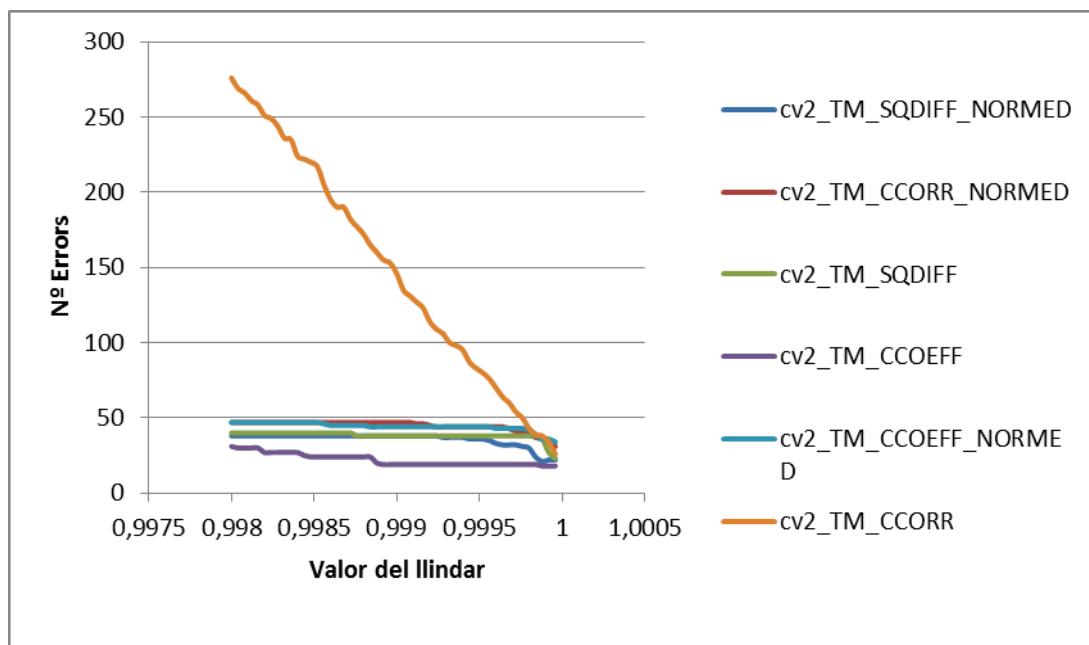
- Primera execució i anàlisi dels resultats:

El primer anàlisi que s'ha realitzat ha estat a partir de 53 icons en 28 màsters. Al ser les icons extretes directament d'aquests màsters se sap segur que es poden trobar dins del màster idènticament, per tant es proposa un valor llindar molt proper a 1, el

0,998 (havent fet alguna prova prèvia més per a corroborar-ho).

Un cop aplicat el mètode en totes les icones i tots els màsters, 3 hores i 6 minuts després, s'ha decidit fer un anàlisi del valor del llindar, observant com es modifica el nombre d'errors en funció d'aquest.

S'han recopilat mostres agafant 50 valors entre el 0,998 i l'1 i aquest és el gràfic que s'ha obtingut:



Il·lustració 46: Gràfic de representació de resultats dels 6 mètodes

D'aquest gràfic es pot extreure que a major valor del llindar més fiabilitat obtenim ja que el nombre d'errors és menor. Apart també depèn del mètode utilitzat, mentre que el CCOEFF o el SQDIFF_NORMED mostren poca variació en el nombre d'errors, no és el cas de la resta de mètodes. Es tenen 53 icones en 28 màsters, comprovant cada icona en cada màster, s'obté un total de $53 \times 28 = 1484$ resultats per a cada mètode, això implica un total de 8904 comprovacions. Per a realitzar tots aquests càlculs s'han invertit un total de 3h i 6 min.

Ara es pot fer el càlcul de com evoluciona el percentatge d'error al variar el llindar, a partir de la següent fórmula:

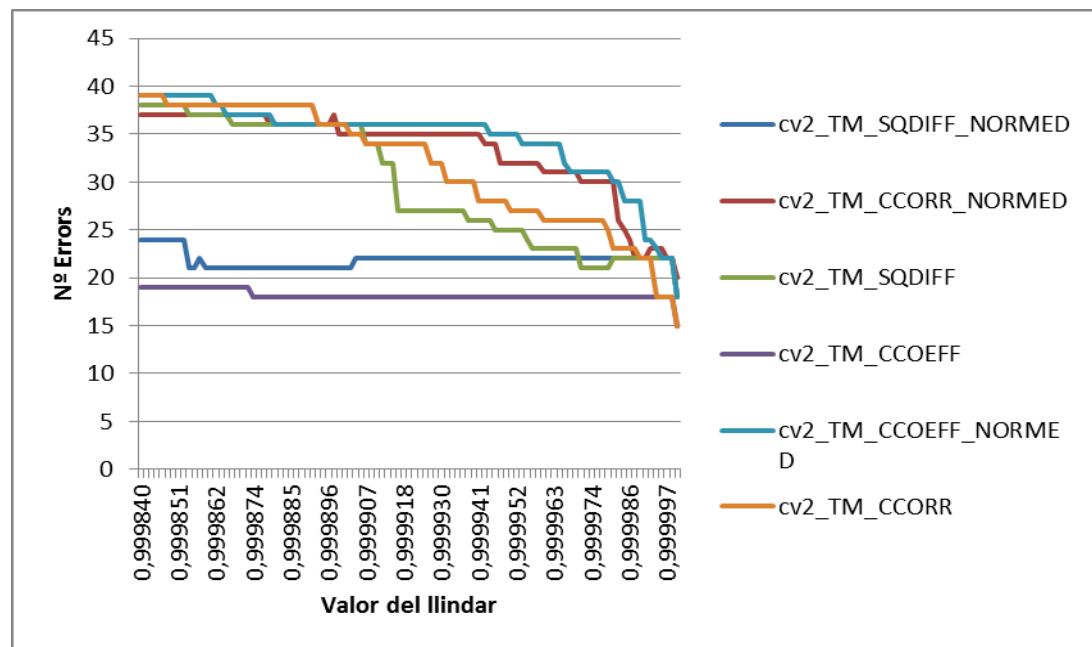
$$P = \frac{n^{\circ} \text{ errors}}{n^{\circ} \text{ resultats}} \cdot 100$$

	LLINDAR = 0,998		LLINDAR = 0,99996		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
SQDIFF	40	2,70	23	1,55	1,14
SQDIFF_NORMED	38	2,56	21	1,42	1,14
CCORR	276	18,60	26	1,75	16,85
CCORR_NORMED	47	3,17	31	2,09	1,08
COEFF	31	2,09	18	1,21	0,88
COEFF_NORMED	47	3,17	34	2,29	0,88

Taula 1: Resultats primera prova

Vist l'anterior gràfic, s'ha decidit tornar a fer un estudi, aquest cop agafant encara més intervals, i concentrant els esforços en l'últim tram, per a intentar deduir com es comporta.

S'han agafat 100 valors entre el 0,99984 i l'1 i aquests són els resultats obtinguts:



Il·lustració 47: Gràfic de representació de resultats dels 6 mètodes

Es veu com en aquest nou estudi es segueix comportant com en l'anterior, veient que a mesura que s'augmenta el llindar disminueix el nº d'errors.

Es torna a fer una taula com l'anterior i aquests són els resultats obtinguts:

	LLINDAR = 0,99984		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	Nº ERRORS	PERCENTATGE	
SQDIFF	38	2,56	18	1,21	1,35
SQDIFF_NORMED	24	1,62	18	1,21	0,40
CCORR	38	2,56	15	1,01	1,55
CCORR_NORMED	37	2,49	20	1,35	1,15
COEFF	19	1,28	15	1,01	0,27
COEFF_NORMED	39	2,63	18	1,21	1,42

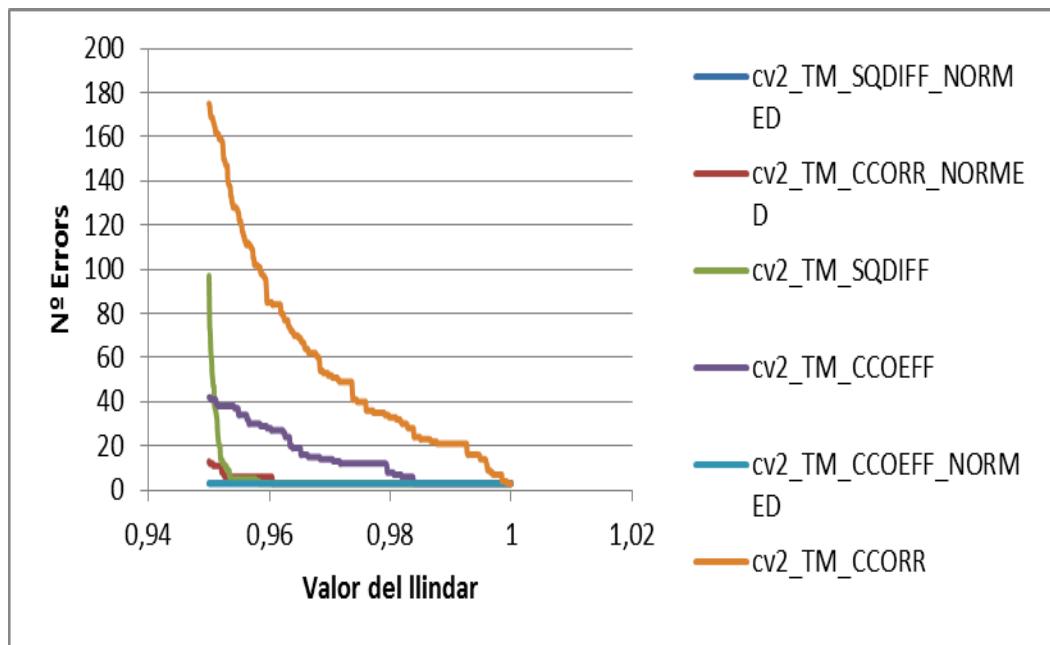
Taula 2: Resultats primera prova

A partir dels resultats anteriors es poden extreure diferents conclusions:

- Com a norma general, al augmentar el valor del llindar augmenta la fiabilitat del mètode, la qual cosa té sentit ja que les icones que s'estan buscant es troben idènticament dins dels màsters, en el cas de que sigui idèntica el resultat del càlcul que fan tots els mètodes és de 1. Per tant quan major serà la fiabilitat serà quan el llindar sigui 1 ja que voldrà dir que només troba la icona si és idènticament igual a la que es troba dins del màster.
- Com es veu en els gràfics, no tots els mètodes tenen la mateixa fiabilitat, mentre que el COEFF sempre es troba per sota quan es parla de nombre d'errors, el mètode de CCORR funciona bastant malament. Això ja s'havia detectat en alguna prova prèvia, malgrat que només es detecten 15 errors quan el llindar val 1 i està de l'ordre de la resta de mètodes, de seguida que el llindar es baixa el nombre d'errors incrementa molt. És un mètode que falla bastant, i en el cas d'haver-lo d'utilitzar després en l'aplicació s'ha de tenir en compte.

➤ Segona execució i anàlisi dels resultats:

Per a la segona prova s'ha decidit provar 12 icones en els mateixos 28 màsters anteriors, que comporta una quantitat de comprovacions de 336, invertint un temps de 57 minuts. La diferència en aquest cas rau en que ara les icones són extretes de la primera generació de ràdios en comptes de la segona, que és la que s'ha estudiat anteriorment, amb la intenció de veure si al variar una mica les icones segueix trobant-les, i estudiar el llindar en el que funciona millor.



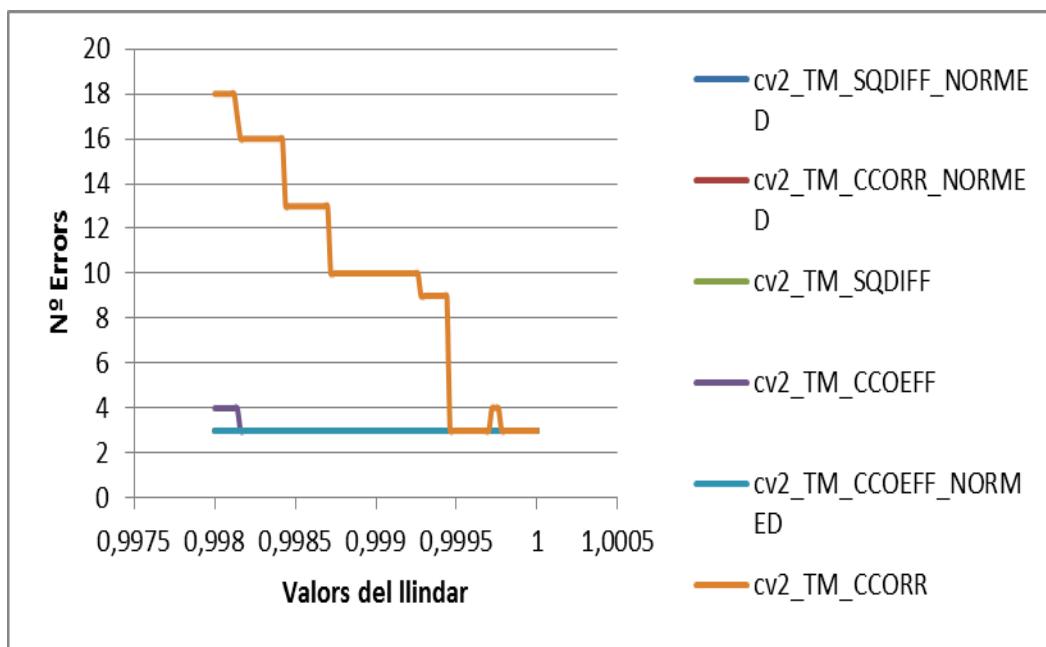
Il·lustració 48: Gràfic de representació de resultats dels 6 mètodes

Es segueix el gràfic amb la taula de percentatges:

	LLINDAR = 0,95		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
SQDIFF	97	28,87	3	0,89	27,98
SQDIFF_NORMED	3	0,89	3	0,89	0
CCORR	175	52,08	3	0,89	51,19
CCORR_NORMED	13	3,87	3	0,89	2,98
COEFF	42	12,5	3	0,89	11,61
COEFF_NORMED	3	0,89	3	0,89	0

Taula 3: Resultats segona prova

Com en el cas anterior s'ha decidit fer una segon estudi concentrat en valors propers a 1. Aquests han estat els resultats:



II·lustració 49: Gràfic de representació de resultats dels 6 mètodes

Veient la taula de percentatges:

	LLINDAR = 0,998		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
SQDIFF	3	0,89	3	0,89	0
SQDIFF_NORMED	3	0,89	3	0,89	0
CCORR	18	5,36	3	0,89	4,46
CCORR_NORMED	3	0,89	3	0,89	0
COEFF	4	1,19	3	0,89	0,30
COEFF_NORMED	3	0,89	3	0,89	0

Taula 4: Resultats segona prova

A priori sembla que aquests resultats són molt bons, només apareixen 3 errors de 336 resultats, per tant una probabilitat d'error de 0,89%. Però si s'analitzen els resultats amb detall es pot veure com molts d'aquests, malgrat hagin sortit com a 'PASS' a la fulla de l'Excel mencionada anteriorment, no tots són 'PASS' verdaders, sinó que molts d'ells són errors, la majoria apareixen degut a que el límít que s'ha agafat per la part inferior és bastant petit, de 0,95 , això permet que es trobin una sèrie de resultats quan no els hauria de trobar o no els ha trobat correctament. Per això es fa l'estudi dels valors del límít i es farà també una ullada als resultats obtinguts en imatge.

Es pot veure que per exemple en aquests casos el resultat obtingut és 'PASS' ja que ha trobat la icona, i ha fet bé perquè l'havia de trobar, però si es profunditza més en la imatge obtinguda en els resultats es descobreix que no és la que havia de trobar. Analitzant millor algunes imatges més es troba que n'han fallat moltes.



Il·lustració 50: Ícona 'BACK' mal trobada en el context de CAR

A la imatge de l'exemple el mètode ha agafat la 'R' com a resultat en comptes de la icona 'BACK' que és la que s'hauria d'haver trobat. Probablement aquest error sigui degut a que les icones de generació 1 tenen una resolució diferent a les icones de generació 2, degut a que la resolució de les ràdios actuals de generació 2 s'ha vist incrementada respecte les ràdios de generació 1. Per això ha reconegut la corba de la part superior de la 'R' com si fos la corba del 'BACK'.

Si es comparen les dues icones es pot veure que les escales són diferents, i que la trobada 'R' s'apropa més a la de generació 1 que a la que hauria de trobar de generació 2 :



Il·lustració 51: 'BACK' de generació 1



Il·lustració 52: Ícona trobada al màster



Il·lustració 53: 'BACK' de generació 2

I aquest problema s'ha passat en altres casos també. Una possible solució a aquest cas podria ser la de variar la mida de les icones o variar la mida dels màsters de tal manera que s'acabin ajustant les mides fins a encaixar entre ells. De moment no es

profunditzarà en aquest tema ja que la previsió de cara a la tercera generació de ràdios és que pugui variar una mica el color o la forma de les icones però no està previst que varii molt la mida de les icones, per tant serà un problema, a priori, poc freqüent.

➤ Tercera execució i anàlisi de resultats:

Degut del problema anterior, s'ha decidit provar d'escalar les icones de la generació antiga per a que s'assemlin una mica més a les noves malgrat no siguin exactament iguals. Per exemple:



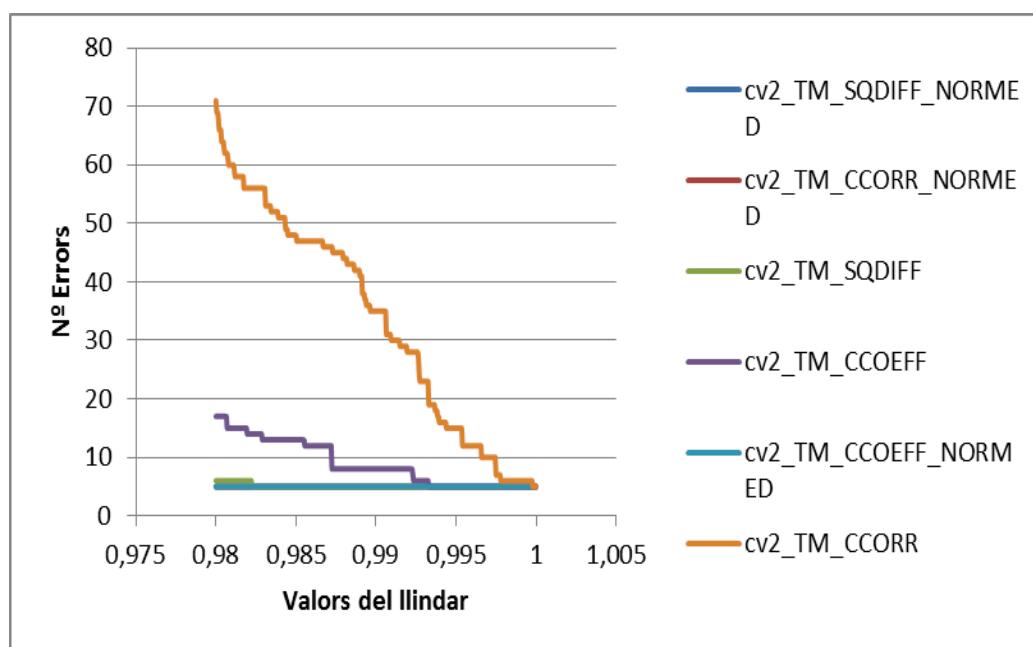
Il·lustració 54: Icona generació 1 'BACK'

Il·lustració 55: Icona generació 1 'BACK' escalada



Il·lustració 56: Icona generació 2 'BACK'

S'han modificat 7 de les 12 icones que es tenen de la generació 1, i s'han provat en els 28 màsters dels que es disposa, utilitzant un valor del llindar de 0,98, entenent que si no són idèntiques es necessitarà un marge. 1 hora i 14 minuts més tard aquests són els resultats als que s'ha arribat:



Il·lustració 57: Gràfic de representació de resultats dels 6 mètodes

Estudiant el nombre d'errors segons el llindar, agafant 400 valors entre mig s'obté el següent:

	LLINDAR = 0,98		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	Nº ERRORS	PERCENTATGE	
SQDIFF	6	0,40	5	0,34	0,07
SQDIFF_NORMED	5	0,34	5	0,34	0
CCORR	71	4,78	5	0,34	4,45
CCORR_NORMED	5	0,34	5	0,34	0
COEFF	17	1,15	5	0,34	0,81
COEFF_NORMED	5	0,34	5	0,34	0

Taula 5: Resultats tercera prova

Veient algun exemple de com ha funcionat. Buscant la icona 'MANUAL BAR' en la pantalla 'RADIOMAIN_MANUAL_BAR_MANUAL_BAR' on es veu pel nom que sí que l'ha de trobar, només s'ha trobat per dos mètodes, el COEFF i el CCORR, amb valors de 0,985724864351 i 0,999866846731 respectivament, per tant la resta que no l'han trobat es consideren errors.



Il·lustració 58: Icona 'MANUAL BAR'

Il·lustració 59: Context Ràdio amb icona trobada

Es tornen a donar casos de que apareixen falsos 'PASS'. En aquest cas s'estava buscant la icona 'SKIPFORWARD' en la pantalla 'RADIOMAIN_SKIPFORWARD_SKIPBACKWARD', i apareix com que l'ha trobat, però al comprovar la solució es pot veure que no és així, que el que ha trobat no és la icona correcta. En aquest cas només s'ha trobat amb el mètode CCORR, amb un valor de 0,989254423954.



II·lustració 60: Ícone 'SKIP_FORWARD'

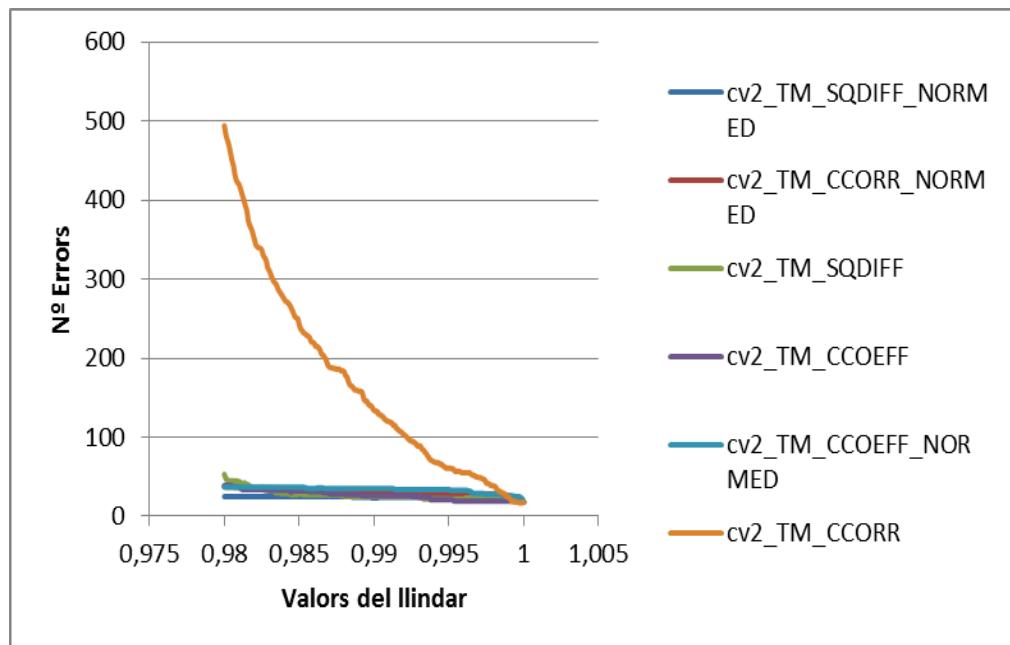
II·lustració 616: Ícone no trobada correctament

No deixa de ser curiós que el nombre d'errors al final de l'execució sigui per tots igual, malgrat que no tots han fallat en el mateix.

➤ Quarta execució i anàlisi de resultats:

La següent prova que s'ha realitzat ha consistit en buscar les 53 icones de generació 2 dins els 28 màsters que ja s'han utilitzat anteriorment, aquesta vegada però, s'han modificat una mica de diferents formes: en uns se'ls ha afegit línies, cercles o canviat colors mitjançant la eina Paint, mentre que en uns altres se'ls hi ha aplicat diferents tipus de filtres com podrien ser filtres gaussians, efectes d'ones, efectes de llums o pixelats i distorsions utilitzant la eina GIMP.

S'ha establert un líindar mínim de 0,98, ja que, com que no seran exactament iguals, es necessitarà disposar d'una mica més de marge. S'han generat un total de 1484 comprovacions, cada una pels 6 mètodes diferents. Aquest nombre de comprovacions amb aquest líindar ha tardat en realitzar-les un total de 5 hores i 14 minuts. Aquests són els resultats obtinguts:



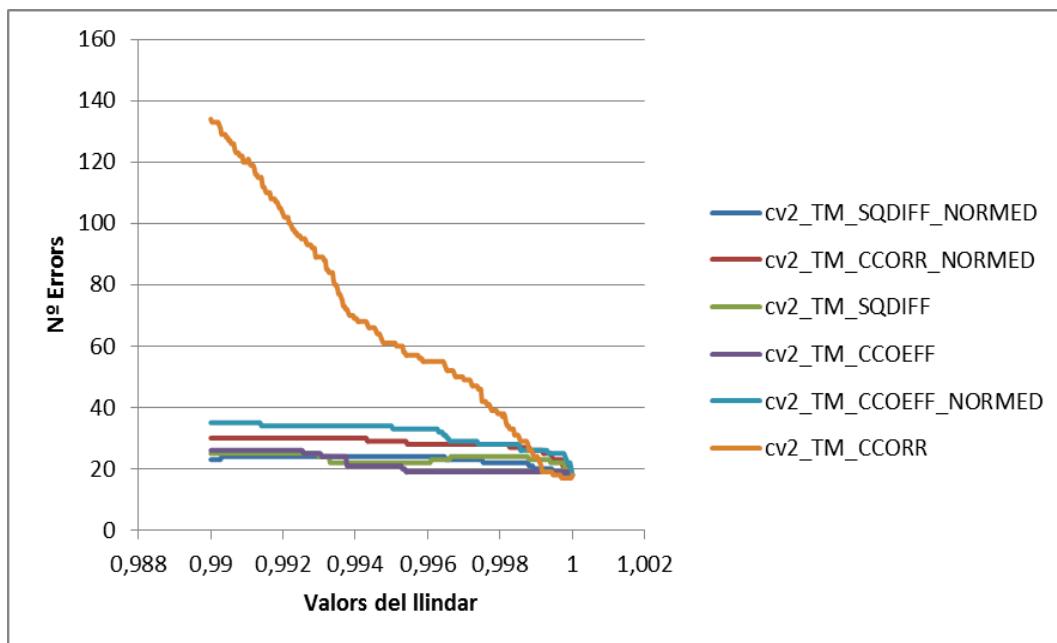
Il·lustració 62: Gràfic de representació de resultats dels 6 mètodes

Es fa la taula de resultats, prenent 200 valors entre el 0,98 i l'1 i s'obté el següent:

	LLINDAR = 0,98		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	Nº ERRORS	PERCENTATGE	
SQDIFF	53	3,57	18	1,21	2,36
SQDIFF_NORMED	25	1,68	18	1,21	0,47
CCORR	494	33,29	17	1,15	32,14
CCORR_NORMED	38	2,56	18	1,21	1,35
COEFF	39	2,63	18	1,21	1,42
COEFF_NORMED	37	2,49	18	1,21	1,28

Taula 6: Resultats quarta prova

Com que es vol concretar més en l'última part del gràfic es fa un segon estudi, aquest cop prenent 300 valors entre el 0,99 i l'1:



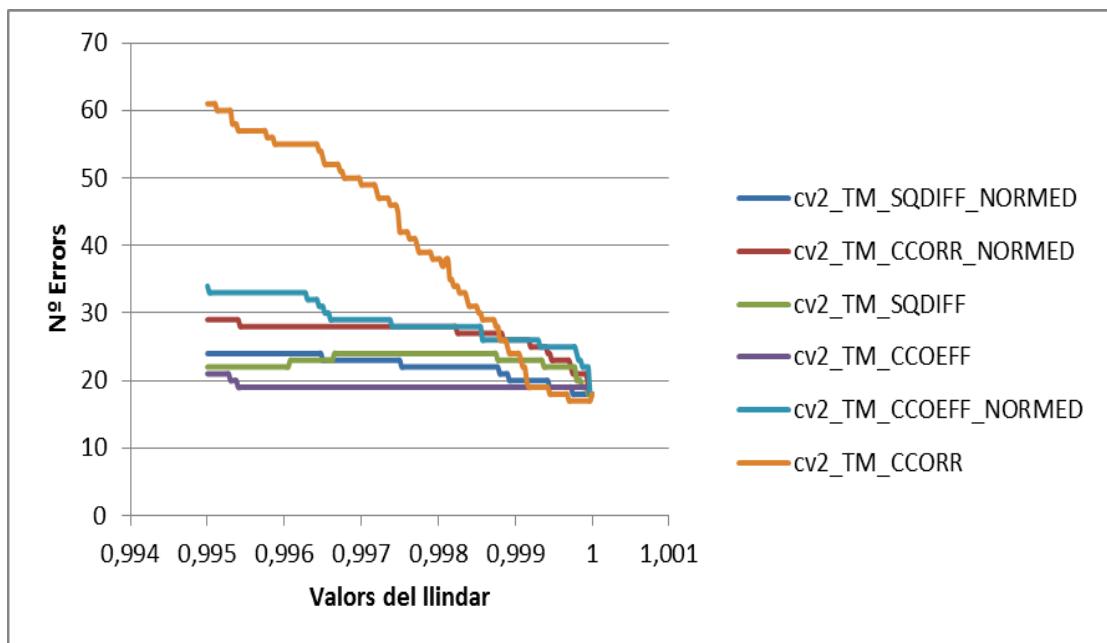
II·lustració 63: Gràfic de representació de resultats dels 6 mètodes

Càcul de la nova taula de errors:

	LLINDAR = 0,99		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
SQDIFF	25	1,68	18	1,21	0,47
SQDIFF_NORMED	24	1,62	18	1,21	0,40
CCORR	134	9,03	17	1,15	7,88
CCORR_NORMED	30	2,02	18	1,21	0,81
COEFF	26	1,75	18	1,21	0,54
COEFF_NORMED	35	2,36	18	1,21	1,15

Taula 7: Resultats quarta prova

Es torna a fer un tercer estudi, aquest cop concretant més encara. S'han pres 200 valors entre el 0,995 i l'1:



Il·lustració 64: Gràfic de representació de resultats dels 6 mètodes

Es torna a calcular la taula pels últims valors:

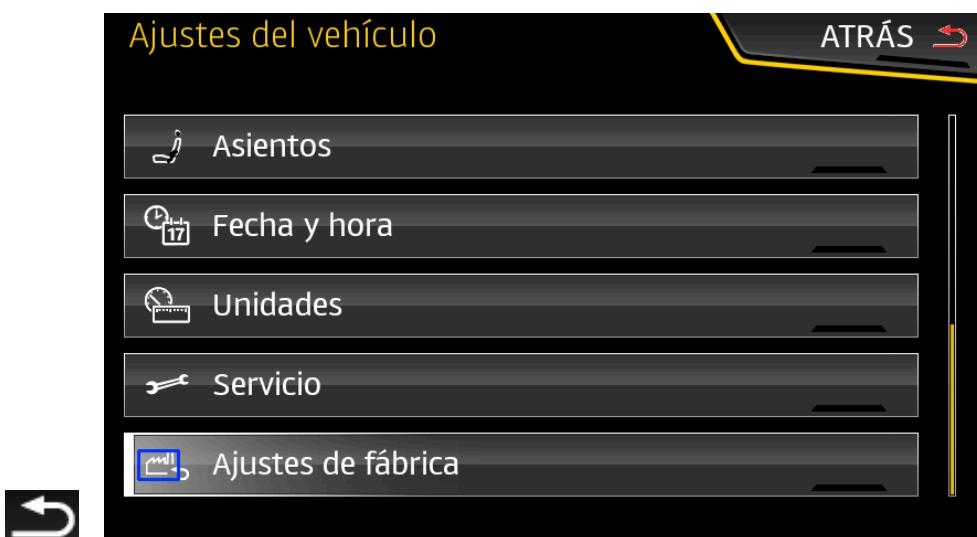
	LLINDAR = 0,995		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
SQDIFF	24	1,62	18	1,21	0,40
SQDIFF_NORMED	24	1,62	18	1,21	0,40
CCORR	61	4,11	17	1,15	2,96
CCORR_NORMED	29	1,95	18	1,21	0,74
COEFF	21	1,42	18	1,21	0,20
COEFF_NORMED	34	2,29	18	1,21	1,18

Taula 8: Resultats quarta prova

S'han realitzat més estudis en aquest cas, perquè s'està buscant un patró que és el que més sentit tindria, que és una paràbola de convexitat positiva o semblant. La idea és que si es busquen les icones en una imatge on en teoria no es troben idèntiques quan el valor del llindar és igual a 1, haurien d'augmentar el nombre d'errors. També amb un llindar de valor baix s'hauria de tenir un nombre d'errors alt. La idea d'aquest estudi és trobar un valor del llindar entre 0,98 i 1 en el que sigui mínim el nombre d'errors. Malgrat això, en vista dels resultats, només compleix aquestes expectatives el mètode CCORR. No deixa de ser curiós que el mètode que pitjor es comporta en llindars baixos sigui el que millor es comporta en llindars alts.

En la resta de mètodes s'obtenen resultats molt semblants als obtinguts en el primer cas, en els que s'arriba al resultat que el llindar amb més fiabilitat és el de valor 1.

Això pot ser degut a que hi ha falsos 'PASS' o falsos 'FAIL' que provoquen que hi hagi errors que no són o viceversa. Sense anar més lluny aquí tenim un exemple:



Il·lustració 65: Ícone 'BACK' Il·lustració 66: Ícone 'BACK' no trobada correctament

En aquest cas s'està buscant la icona de 'BACK' en una pantalla on el color s'ha modificat una mica. El resultat obtingut és de que la troba, però realment està trobant una icona que no és. També es veritat que només s'està trobant pel mètode CCORR que és el que més falla a baixos valors del llindar, en aquest cas s'ha trobat amb un valor de 0,983100035689.



Il·lustració 67: Ícone 'FILTER'

Il·lustració 68: Ícone trobada correctament

En un altre exemple es pot veure que s'ha trobat la icona del filtre tot i haver estat modificada la pantalla afegint una llum a prop de la icona per a modificar-ne

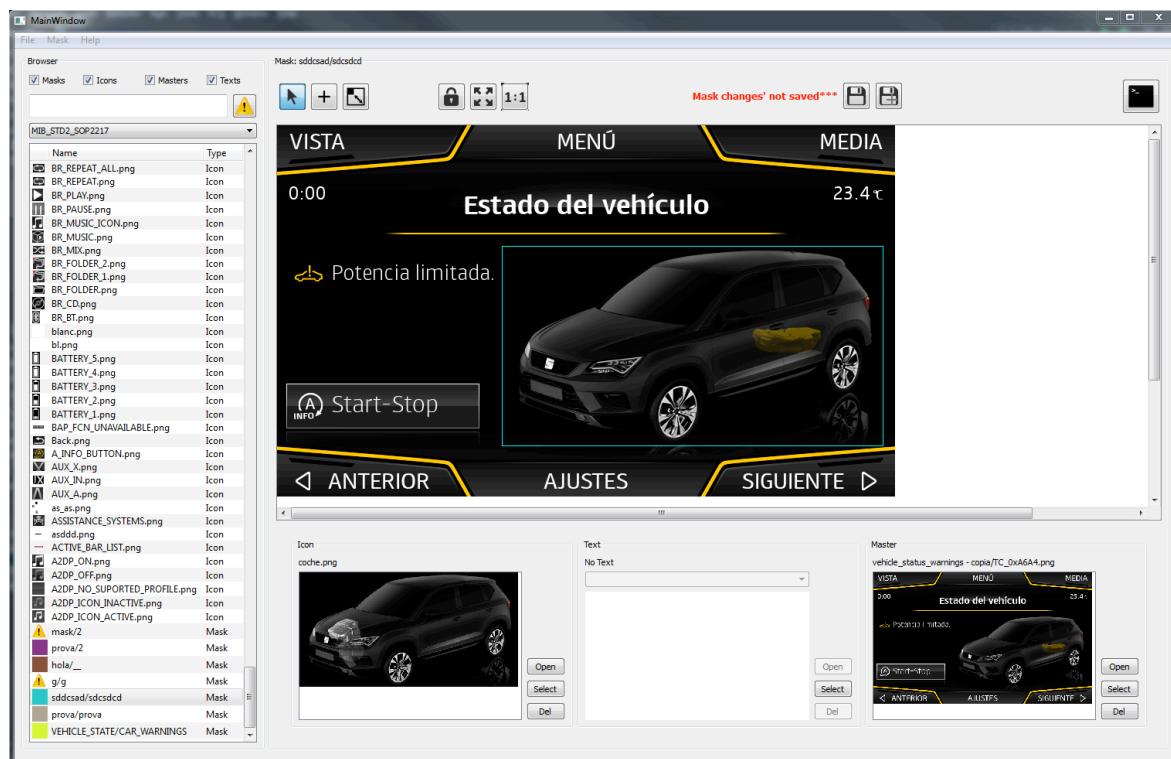
l'aspecte. En aquest cas s'ha trobat amb dos mètodes, el CCORR i el COEFF_NORMED, amb valors de 0,981525905439 i 0,995003402233 respectivament.

Es pot veure com les bateries de proves que s'han llençat no acaben de ser del tot fiables en el cas de que les icones no siguin exactes a les que es busquen en el màster, això no fa que es descartin els resultats trobats fins al moment, ja que fallen una petita part.

7.3. Proves amb l'aplicació integrada en el sistema Seat

Un cop s'ha vist que el funcionament del mètode és correcte, s'ha decidit provar d'integrar-lo a la eina dissenyada per un company de feina. Aquesta eina és la que s'utilitzarà posteriorment en les validacions, per tant interessa veure el comportament del mètode acoblat a l'eina, juntament amb les seves limitacions. D'aquestes proves finals s'espera poder determinar amb major precisió tant els rangs de treball com la seva fiabilitat.

Imatge de l'eina:



Il·lustració 69: Imatge de l'eina que s'utilitzarà a Seat

La imatge que es veu a la part inferior esquerra és la icona que es busca, mentre que la que es veu a la part inferior dreta és el màster on es busca, que és la que s'acaba representant en gran. En aquest cas es pot veure com s'ha trobat la icona en el màster, que surt encerclada amb una mascara de color blau clar. En cas de que no sigui trobada no apareix cap màscara.

El que es farà és fer una sèrie de proves buscant icons en els màsters algunes exactes i altres amb petites variacions, alternant entre els diferents mètodes i diferents valors de llindars.

➤ Exemples de proves fetes:

- 1a prova: mètode: SQDIFF; valor llindar: 0,99



Icona:

Il·lustració 70: Icona del cotxe amb motor



Il·lustració 71: Icona del cotxe trobada

Temps: 1,05 segons



Il·lustració 72: Ícone del cotxe trobada

Temps: 0,59 segons

És interessant en aquest cas que reconegui el cotxe encara que no sigui exactament igual. La idea és que d'aquesta manera a partir d'una icona de referència d'un cotxe es poden trobar tota la resta de cotxes. D'aquesta manera es pot saber si la ràdio es troba en el context 'ESTADO DEL VEHÍCULO' a partir de la mateixa icona. D'aquesta manera s'estalvia haver d'assignar una icona diferent a cada màster.

A la vegada però, s'han realitzat més proves, apujant el valor del llindar a 1, i en aquest cas només es troba quan el cotxe és idèntic, d'aquesta manera es pot classificar l'estat on es troba la MU. Per exemple si és un 'warning' del motor o si és informació sobre les rodes, entre d'altres. Per tant, jugar amb el valor del llindar dóna moltes possibilitats. També es pot agafar el text junt amb el cotxe com a icona per a determinar el tipus d'avís que apareix en pantalla. Per tant aquesta eina permet treballar a diferents nivells.

- 2a prova: mètode: CCORR_NORMED; valor del llindar: 0,98

Icona:



Il·lustració 73: Ícone de 'WARNING' vermella



Màster 1:

Il·lustració 74: Icona 'WARNING' vermella trobada

Temps: 1,07 segons



Màster 2:

Il·lustració 75: Icona 'WARNING' blanca trobada

Temps: 0,87 segons



Il·lustració 76: Ícone 'WARNING' groga trobada

Temps: 0,86 segons

Es pot veure com passa el mateix que a la primera prova, troba les ícones de 'warning' malgrat siguin de colors diferents. Això dona joc, tant podem buscar només si apareix una icona de 'warning' com podem buscar si apareix la icona d'un color determinat escollint un valor del llindar molt proper o igual a 1. Tanmateix, es pot observar que els temps de busca són força baixos, pràcticament iguals o inferiors a 1 segon.

7.4. Nou algoritme

Vistos els resultats fins al moment i a partir de les conclusions que s'han anat extraient, s'ha decidit implementar un algoritme amb la intenció d'optimitzar la utilització dels mètodes, per a reduir el temps d'execució mantenint en el possible l'efectivitat.

Com a variable independent, l'algoritme tindrà el valor del llindar, mentre que la resta de variables, els mètodes i el temps, dependran d'aquest.

A continuació es veuen una sèrie de resultats que s'han utilitzat per a extreure les conclusions:

Intervals\Mètodes	SQDIFF	SQDIFF_N	CCORR	CCORR_N	COEFF	COEFF_N
L = 1	18	18	15	20	15	18
L = 1	18	18	17	18	18	18
0,9998 < L < 1	38	24	38	37	19	39
0,998 < L < 0,9998	40	38	276	47	31	47
0,995 < L < 0,998	24	24	61	29	21	34

0,99 < L < 0,995	25	24	134	30	26	35
0,98 < L < 0,99	53	25	494	38	39	37
0,98 < L < 0,99	6	5	71	5	17	5

Taula 9: Taula de resultats segons mètode i interval

S'han utilitzat diferents colors per a visualitzar millor el nombre d'errors. Els mètodes que tenen menys errors s'indiquen de color verd, mentre els que en presenten més s'indiquen en vermell. Els de color groc es troben en un punt entremig. Òbviament, els mètodes que més ens interessen per a cada interval són els que es mostren de color verd. A la següent taula es veu, per a cada interval, l'ordre en que s'haurien d'utilitzar els mètodes.

Intervals\Ordre	1r	2n	3r	4t	5è	6è
L = 1	CCORR	COEFF	SQDIFF_N	SQDIFF	COEFF_N	CCORR_N
0,9998 < L < 1	COEFF	SQDIFF_N	CCORR_N	SQDIFF	CCORR	COEFF_N
0,998 < L < 0,9998	COEFF	SQDIFF_N	SQDIFF	CCORR_N	COEFF_N	CCORR
0,995 < L < 0,998	COEFF	SQDIFF_N	SQDIFF	CCORR_N	COEFF_N	CCORR
0,99 < L < 0,995	SQDIFF_N	SQDIFF	COEFF	CCORR_N	COEFF_N	CCORR
0,98 < L < 0,99	SQDIFF_N	COEFF_N	CCORR_N	SQDIFF	COEFF	CCORR

Taula 10: Taula de resultats amb mètodes ordenats

Com que la intenció és la de reduir la càrrega computacional i, per tant, el temps, s'agafen només els tres primers mètodes, entenent que si els tres millors mètodes per a cada interval no l'han trobat, els que pitjor funcionen tampoc ho faran. Per tant la taula final queda d'aquesta manera:

Intervals\Ordre	1r	2n	3r
L = 1	CCORR	COEFF	SQDIFF_N
0,995 < L < 1	COEFF	SQDIFF_N	SQDIFF
0,99 < L < 0,995	SQDIFF_N	SQDIFF	COEFF
L < 0,99	SQDIFF_N	COEFF_N	CCORR_N

Taula 11: Taula dels millors mètodes escollits per a cada interval

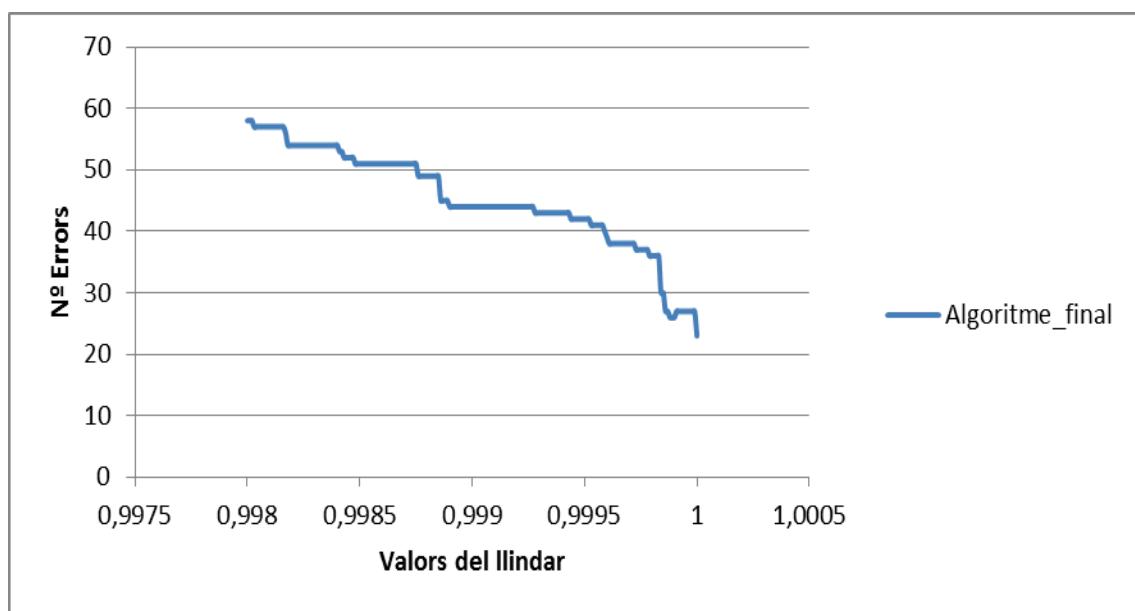
Se li dona especial importància al temps perquè a la hora de validar automàticament s'hauran de realitzar una gran quantitat de comprovacions d'imatges. Només que per anar d'un estat a un altre es necessitin fer 5 comprovacions, i que per a fer cada comprovació estigui uns 6 segons, ja són 30 segons a afegir al test. En una bateria de uns quants tests on en cada test hi hagi varies transicions entre estats podria arribar a tardar hores. Això al final acaba generant que es puguin validar menys coses.

L'Algoritme consistirà en decidir, a partir de l'interval, quin dels mètodes utilitzar, i en cas de que no es trobi, passarà a buscar el segon mètode i el mateix fins al tercer. Si amb el tercer mètode tampoc s'ha trobat s'entindrà com a que la icona no apareix en la pantalla.

7.5. Proves amb l'algoritme

- Primera prova:

La primera prova que s'ha fet ha estat la de les icones de generació 2 amb els màsters de generació 2. En aquest cas es repeteix el valor del llindar, de 0,998. Es pot veure com el temps que tarda en realitzar les 1484 comprovacions és molt inferior al que tarda en la primera prova, en la que es comprova pels 6 mètodes diferents. Mentre que en la primera prova la duració era de 3 hores i 6 minuts, en aquesta segona execució la duració ha estat de 1 hora i 13 minuts, pràcticament tres vegades inferior.



Il·lustració 77: Gràfic de representació de resultats del algoritme final

	LLINDAR = 0,998		LLINDAR = 0,99996		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
ALGORITME	58	3,91	27	1,82	2,09
Mitjana de tots els mètodes	79,83	5,38	25,5	1,72	3,66

Mitjana de tots els mètodes excepte CCORR	40,6	2,74	25,4	1,71	1,02
---	------	------	------	------	------

Taula 12: Taula de resultats primera prova algoritme

S'han pres valors sense arribar al 1, ja que per a un valor del llindar igual a 1 l'algoritme aplica uns mètodes diferents. Apart, s'han realitzat càlculs de dues maneres al realitzar la mitjana dels 6 mètodes, s'ha decidit eliminar del càlcul de la mitjana el mètode CCORR en el segon cas, ja que amb un valor de 276 errors desvia exageradament la mitjana.

Comparant l'algoritme dissenyat amb els altres 5 es pot veure com la quantitat d'errors augmenta molt poc, en percentatge es passa del 2,74% d'errors al 3,91%, variant un tan sols un 1,17%. Quan el valor del llindar val 0,99996, encara varia menys, només un 0,11%. Es consideren uns percentatges d'augment de l'error del tot assumibles veient que es redueix aproximadament per tres el temps d'execució.

➤ Segona prova:

Per a la segona prova s'ha decidit agafar les mateixes icones i màsters que en la primera execució, amb la diferència en aquest cas del valor del llindar, igual a 1, per a veure com funciona amb els mètodes triats en l'algoritme. Ha trigat 1 hora i 9 minuts.

	LLINDAR = 1	
	NºERRORS	PERCENTATGE
ALGORITME(amb un llindar de 0,998)	23	1,55
ALGORITME(amb un llindar de 1)	24	1,62
Mitjana de tots els mètodes	17,33	1,17
Mitjana de tots els mètodes menys el CCORR	17,8	1,20

Taula 13: Taula de resultats segona prova algoritme

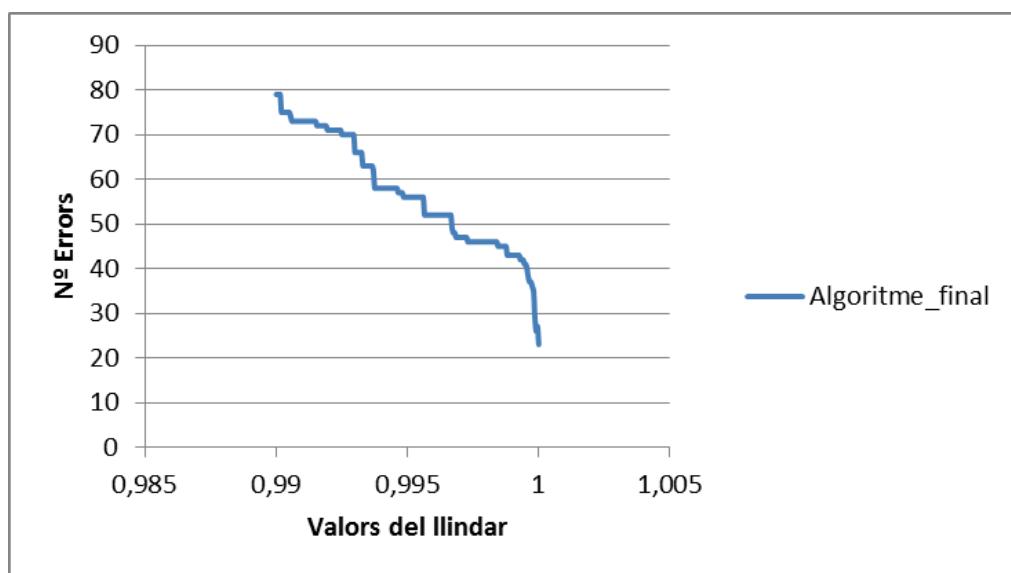
En aquest cas no és necessari eliminar el CCORR, ja que per a valors del llindar iguals a 1 és dels que millor funciona. Es pot veure com al treure'l augmenta el percentatge d'errors.

Es veu com en els dos casos de l'algoritme el resultat és pràcticament igual, pot ser que aparegui un error més en el del llindar igual a 1, degut a algun fals 'FAIL' o a algun fals 'PASS'. No és representatiu, a la vegada, mirant els resultats del principi es veu com tant el COEFF com el CCORR funcionen pràcticament igual de bé, per això surten uns resultats tan semblants. Si es comparen amb la mitjana de tots els mètodes es pot veure com la diferència del resultat obtingut és inferior a un 1%, un

0,45% concretament, apart, se segueix reduint enormement el temps.

➤ Tercera prova:

La tercera prova que s'ha realitzat ha estat la del llindar a 0,99. Per a veure com funciona l'algoritme en aquest interval. S'ha tardat en calcular un total d'1 hora i 11 minuts. Es pot veure com l'ordre de temps és més o menys semblant que els casos anteriors.



Il·lustració 78: Gràfic de representació de resultats del algoritme final

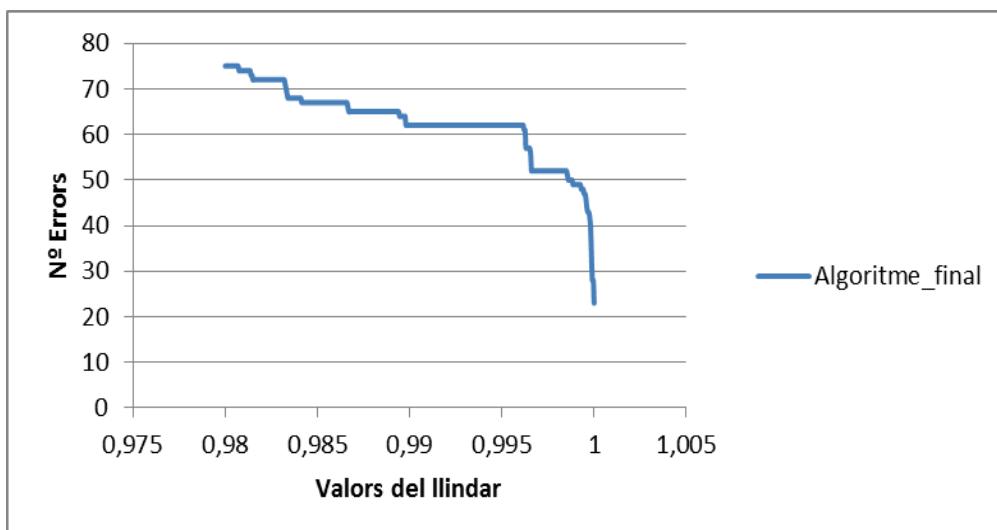
	LLINDAR = 0,99		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
ALGORITME	79	5,32	23	1,55	3,77

Taula 14: Taula de resultats tercera prova algoritme

Es pot veure com quan el valor del llindar val 0,99 el nombre d'errors és força alt, de 79 concretament, mentre que a mesura que s'apropa a un valor del llindar de 1, baixa dràsticament el nombre d'errors, fins a un nombre de 23.

➤ Quarta prova:

Com a última prova s'ha decidit provar les mateixes ícones en els mateixos màsters, provant el llindar que agafa valors inferiors a 0,99, en aquest cas s'ha agafat 0,98. Ha tardat tan sols 42 minuts i aquests són els resultats que s'han obtingut:



Il·lustració 79: Gràfic de representació de resultats del algoritme final

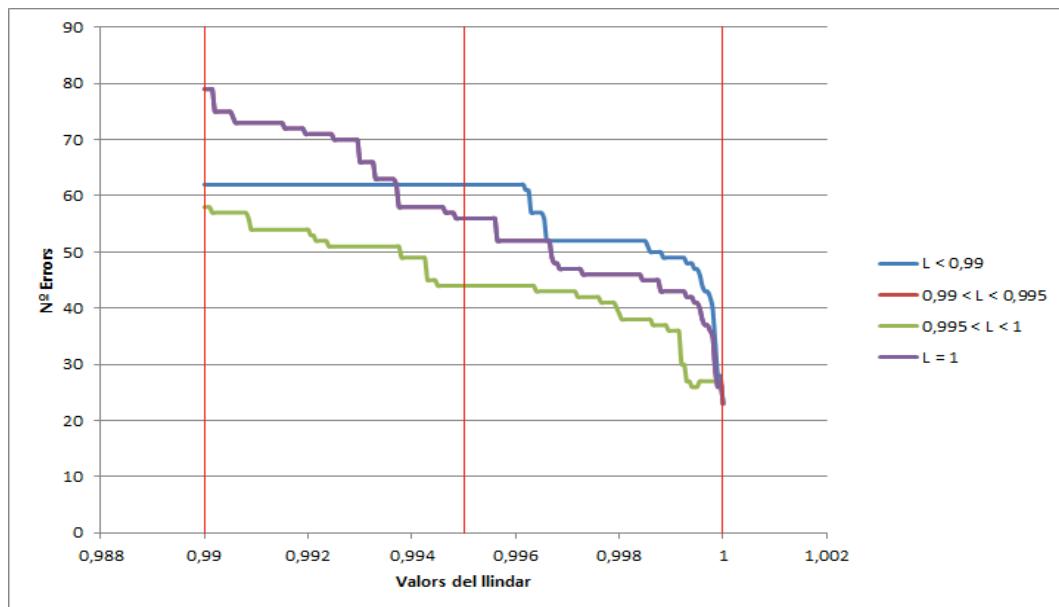
	LLINDAR = 0,98		LLINDAR = 1		DIFERÈNCIA
	Nº ERRORS	PERCENTATGE	NºERRORS	PERCENTATGE	
ALGORITME	75	5,05	23	1,55	3,50

Taula 15: Taula de resultats quarta prova algoritme

Si es compara amb la mitjana d'errors dels mètodes a un valor del llindar de 0,998, es pot veure que s'ha reduït un 0,33%, si es compara treien el mètode CCORR del càlcul de la mitjana s'obté que el nombre d'errors augmenta en un 2,31%. S'ha de pensar que és lògic que augmenti el nombre d'errors des de el moment en que el valor del llindar s'ha baixat tant, ja que pot estar trobant icones en pantalla que només s'assemblen en un 98% a la icona original, per tant són errors comprensibles.

➤ Cinquena prova:

S'ha decidit fer una nova prova, en aquest cas agafant com en les proves anteriors totes les icones de generació 2 i tots els seus màsters també, amb la diferència que s'han fet les proves agafant un valor del llindar de 0,99, i realitzant-la 4 vegades utilitzant els mètodes utilitzats en cada interval, de forma que es puguin comparar entre ells els mètodes escollits per a cada interval. Aquests són els resultats obtinguts:



Il·lustració 80: Gràfic de representació de resultats del algoritme final segons l'interval

Es poden extreure diferents conclusions a partir d'aquest gràfic:

En la zona del llindar $< 0,99$, es pot veure com la tendència de tots els mètodes és la d'augmentar el nombre d'errors conforme es redueix el valor del llindar, menys el mètode que s'ha decidit utilitzar per a aquest interval, que figura en el gràfic de color blau. La tendència que mostra és d'una major estabilitat.

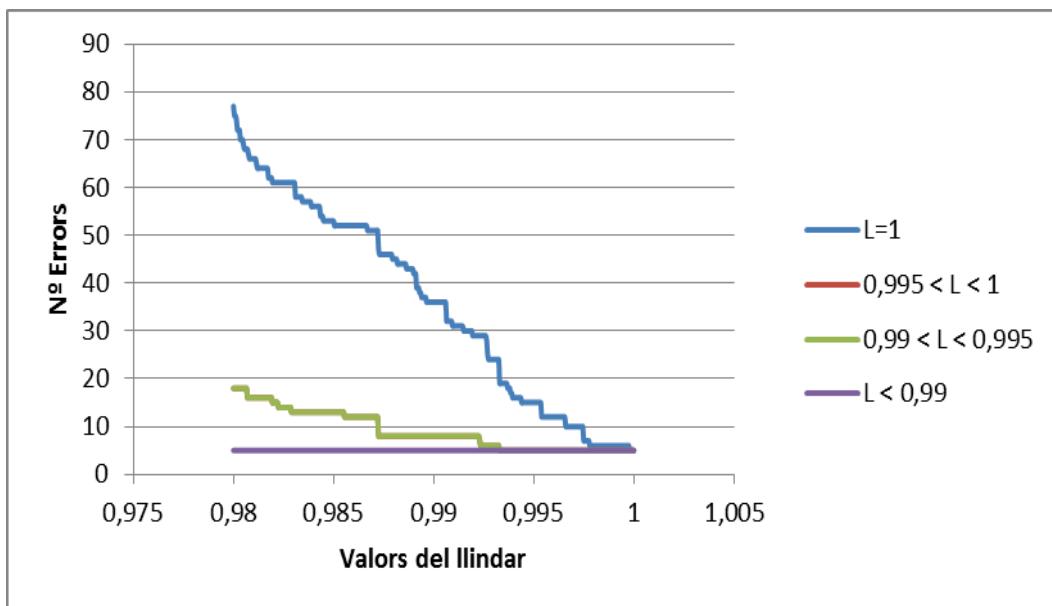
En l'interval entre 0,99 i 0,995 es pot veure com el mètode que dona menys errors és el verd en comptes del vermell (en aquest cas el vermell coincideix amb el lila). Això pot passar per motius com per exemple falsos 'PASS', que no estigui ben determinat l'interval o els mètodes a utilitzar, o la que és més probable que és que els mètodes han estat triats a partir de múltiples diferents resultats, comparant ícones amb màsters modificats, o ícones modificades en màsters sense estar-ho. Per tant aquesta només és una de les proves i pot no funcionar a la perfecció en totes les proves. En aquest cas la diferència entre els mètodes es troba aproximadament entre un 1% i un 1,5%. Per tant es considera assumible.

En el tercer tram, on es situa el valor entre 0,995 i 1 si que es pot veure com el mètode es correspon amb el millor, ja que és el de color verd qui presenta un nombre d'errors més baix en tot el sector.

Finalment es pot veure com, quan el valor del llindar val 1, el mètode corresponent al color lila mostra un valor mínim d'errors.

➤ Sisena prova:

S'ha repetit el mateix procés, aquesta vegada buscant les icones de generació 1 re-escalades dins els màsters de generació 2, generant un total de 196 resultats:



Il·lustració 81: Gràfic de representació de resultats del algoritme final segons interval

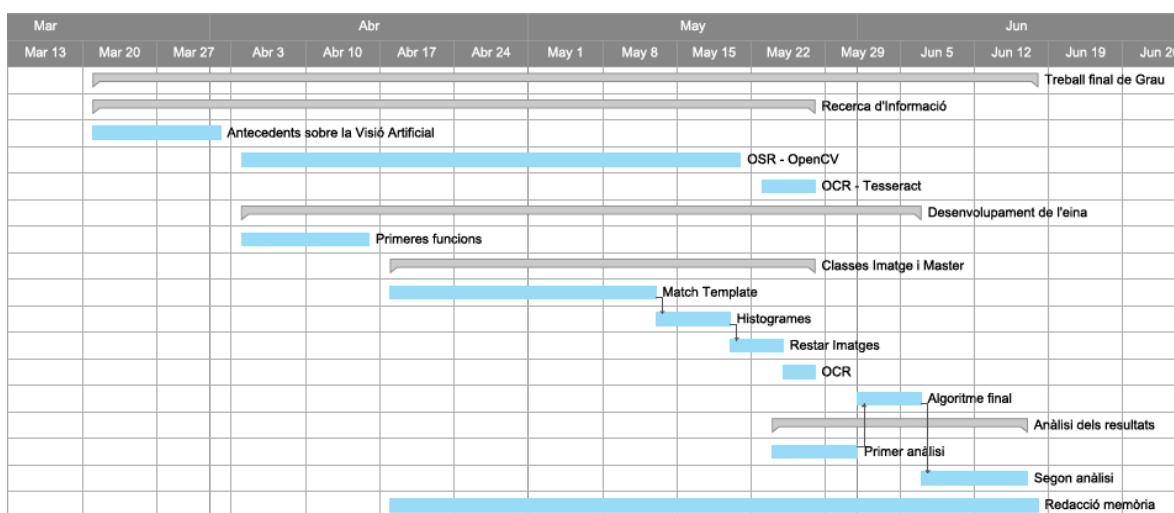
D'entrada crida l'atenció que el mètode de l'interval de color lila tingui sempre el mateix nombre d'errors indiferentment del valor del llindar, això és degut a que no ha trobat cap vegada cap icona en cap màster, per tant sempre dona el mateix nombre d'errors, que serà el nombre de vegades que hauria d'haver trobat una icona en un màster i no ho ha fet.

En aquesta ocasió el mètode de l'interval de color vermell coincideix amb el de verd, i es pot veure com funciona millor que el mètode blau on, suposadament, i com es pot veure en el gràfic, funciona bé quan el valor del llindar val 1.

8. Anàlisi econòmic

8.1. Planificació temporal:

Per a la planificació temporal s'ha decidit ajudar-se amb un diagrama de Gantt per a que la informació quedi una mica més visual.



Il·lustració 82: Diagrama de Gantt

Es podria dividir el projecte en quatre grans blocs principals: recerca d'informació, desenvolupament de l'eina, anàlisi de resultats i per últim redacció de la memòria.

El temps que a continuació es veu per a cada tasca en el diagrama no és el real. Això és degut a que hi ha tasques que s'han anat alternant. Per exemple, en el cas de la redacció de la memòria, es va començar força d'hora a documentar tant la informació que s'anava trobant com el que s'anava implementant, això no vol dir que es dediqui totes les hores del dia a això, sinó que s'hi dedica una petita part. Una mica semblant succeeix amb la recerca d'informació, malgrat s'ha anat avançant en el projecte, s'ha seguit buscant informació i mètodes per a buscar solucions als problemes que han anat sorgint.

A continuació s'explica amb més detall cada tasca:

➤ **Desenvolupament de l'eina (150 hores):**

La tasca que més hores ha comportat, veient el diagrama anterior, és la de la programació i desenvolupament del l'eina, i totes les correccions o millores que s'han anat fent.

➤ **Recerca d'informació (60 hores):**

També ha comportat força feina la recerca de tota la informació relacionada amb el projecte, ja que la visió artificial és un sector que cada vegada està agafant més força, degut a que va de la mà amb el sector de l'automatització, on s'està donant passos de gegant. Per tant hi ha una quantitat d'informació enorme, això té una part bona que és que es pot trobar pràcticament tot el que es pugui necessitar, però a la vegada s'ha de saber filtrar per a quedar-se només amb el necessari.

➤ **Redacció de la memòria (40 hores):**

La redacció de la memòria s'ha intentat compaginar amb la resta de tasques, intentant realitzar-les a la vegada, amb l'objectiu de poder reflectir el que s'ha anat fent o pensant i no deixar-se coses pel mig.

➤ **Anàlisi de resultats (30 hores):**

L'anàlisi de resultats també ha comportat el seu temps. Primerament, perquè s'ha de pensar com es pot aconseguir extreure el màxim de resultats de la manera més efectiva i menys carregosa. Apart, un cop es tenen els resultats, s'ha de pensar com es poden estructurar per a facilitar el seu anàlisis i un cop es tenen tots preparats analitzar-los i extreure'n conclusions que permetin entendre millor el funcionament de l'eina, definir-ne els patrons de funcionament i en cas de que sigui possible, pensar en possibles millores i aplicar-les.

8.2. Costos

A partir de l'apartat anterior es podria dir que el nombre d'hores total dedicat al projecte és de 280. Es poden desglossar de la següent manera:

- Hores dins de SEAT: es comença el projecte el 21 de Març de 2016, i s'acaba el 17 de Març del mateix any. Això és un total de 67 dies. Suposant que cada dia es dediquen al projecte aproximadament 3 hores i 30 minuts, això suposa un total de 235 hores aproximadament.
- Hores fora de SEAT: aproximadament el nombre d'hores dedicades fora de la feina és d'unes 45.

Si s'agafa el nombre d'hores dedicades i es multiplica pel sou mig d'un enginyer industrial, aproximadament d'uns 30€ per cada hora treballada, això suposa un total de 8.400€.

Pel que fa al cost de les llibreries utilitzades en la programació, és de 0€, ja que tant el llenguatge Python com les seves llibreries són de programació lliure i per tant es poden descarregar gratuïtament d'internet. Per tant, el cost d'aquest projecte serà directament el que cobri l'enginyer en qüestió, d'uns 8.400€.

MATERIAL	COST UNITARI (€)	QUANTITAT	TOTAL (€)
Treballador(enginyer)	30	208h	8400
Llibreries i programes utilitzats	0	-	0
			8400

8.3. Impacte en el medi ambient

Es considera que l'impacte en el medi ambient que té aquest projecte és pràcticament nul, al ser un projecte de caire informàtic. Amb això es vol dir que no es generen residus materials. L'única cosa que es consumeix és energia elèctrica, que es pot considerar una part molt petita.

9. Conclusions

Un cop s'han realitzat un considerable nombre de proves diferents, s'ha arribat a les següents conclusions:

9.1. Fiabilitat del mètode:

Per a que el mètode funcioni correctament s'ha de tenir clar com es vol utilitzar per a poder treure'n el màxim profit.

- Si es dona el cas en que la icona que s'està buscant és exacta o pràcticament exacta aplicarem un valor del llindar molt elevat, pràcticament igual a 1, com podria ser 0,998, amb el que s'han realitzat un bon nombre de proves. D'aquesta manera restringim el nombre de retalls i comprovacions a realitzar, estalviant també càrrega computacional i temps de processat.

S'ha vist en els resultats que per a un valor del llindar igual a 1 es minimitzen el nombre d'errors, tot i així sembla convenient no agafar directament 1, per si aparegués una petita variació no apreciable a simple vista. Es podria dir que és una espècie de marge de seguretat.

- En el cas de que la icona no sigui exacta del tot i varii una mica el color, o la forma el que es farà és baixar el valor del llindar, dependent de si és més diferent o menys es pot agafar des de 0,95 fins a 0,99. En el cas de baixar tant el valor del llindar augmentarà la càrrega computacional i el temps de processat.

9.2. Optimització del mètode:

Conclusions a les que s'ha arribat amb l'objectiu de millorar el temps de processat, ja que per a buscar només una icona no és tan transcendental si està 2s o 5s, però en el cas en que s'hagin de realitzar 20 o 30 comprovacions el resultat final pot ser molt diferent. Per això s'ha pensat com es pot optimitzar una mica el mètode.

- És molt important de cara a rebaixar el temps la mida de la icona que s'està buscant. Si la icona és molt petita el temps serà més elevat que si la icona és gran. Això és degut a que amb la petita ha de realitzar una gran quantitat de comprovacions, i donat que el nombre de comprovacions segueix una funció quadràtica, llavors, al ser major la mida de la icona es veurà molt reduït el nombre de comprovacions i el temps. Apart, s'agafen més retalls amb una icona petita que amb una gran ja que la

probabilitat de que trobi píxels iguals o semblants es més gran degut a que la possibilitat de combinatòria és més petita.

- En el cas de que la icona que es busca sigui pràcticament exacta ja implica una millora en el temps agafar un valor del llindar molt elevat. Tot i així encara es pot optimitzar més. Pràcticament tots els mètodes troben les ícones quan es busca un valor que sigui pràcticament igual a 1, per tant es pot buscar aplicar només un mètode, amb el que s'aconseguiria dividir el temps de processat pràcticament entre 6.

Dels mètodes estudiats en podem treure conclusions a partir de l'apartat de resultats, on es pot veure que els mètodes CCORR, COEFF o SQDIFF_NORMED són dels que millor funcionen, i més quan més a prop de 1 es troba el llindar. Per tant es pot contemplar la idea d'utilitzar només aquests tres i els altres tres no. Amb això es pot passar d'un temps dels volts de 5 segons a un temps inferior a 1 o 2 segons. I en el cas de que s'haguessin de realitzar un gran nombre de comprovacions el temps total es veuria fortament reduït.

En el cas de que es busqui més fiabilitat o es digui que la icona ha de ser exacta, es pot aplicar la funció de restar imatges. Aquesta funció és més ràpida que la funció de comparar histogrames per tant es guanya temps i fiabilitat en el mètode, ja que s'estalvien operacions com calcular l'histograma de cada retall.

- En el cas de que la imatge no sigui exacta, si que s'han d'aplicar un major nombre de mètodes. Tot i això se'n pot descartar algun. Vistos els resultats es busquen els mètodes que menys variació tinguin entre els dos valors extrems de l'interval del llindar. Es pot descartar el mètode CCORR sense cap mena de dubte, ja que de seguida que disminueix una mica el llindar apareixen una gran quantitat d'errors. La resta de mètodes més o menys es comporten igual.

Apart, en aquests casos es descarta aplicar la funció restar imatges, ja que aquesta està més encarada a comparar dues imatges iguals o pràcticament iguals, i no en seria el cas.

- Quan es parla de contextos el tema es torna una mica més delicat, donat que els contextos poden variar molt entre ells. De fet, poden variar fins al punt de que siguin més semblants entre ells el context de Car Interface i el context de Media que dos de Media entre ells o dos de Car Interface entre ells. Per aquesta raó no es recomana utilitzar aquesta aplicació per a comprovar el context on es troba, a no ser que es doni el cas de que els dos contextos que comparen siguin molt semblants. També s'ha detectat que el mètode que millor funciona en aquest cas és el descartat en el

cas anterior, el CCORR.

Una alternativa a això podria ser la de buscar detalls que només apareguin en un context determinat, i a partir d'aquí determinar en funció de si es troba o no en quin context està. En el cas que s'ha estudiat, el perfil de cada context té un color diferent, Media és vermell, Car Interface és groc... d'aquesta manera utilitzant el mètode de restar imatges amb color es podria determinar el context.

- Com a detecció de textos o OCR la funció implementada funciona prou bé si el que es busca és trobar un text concret en una zona. En el cas de que es vulgui extreure tot el text s'ha de preveure que hi ha icones que les detecta com a text i a vegades pot confondre alguna lletra. Com que de moment es té la intenció de tractar el text com a imatges o icones no és una funció prioritària. A part, com que no era un objectiu principal del projecte, sinó que s'ha proposat més tard amb la intenció de que el mètode quedí més complet, no s'ha profunditzat més en l'assumpte.

9.3. Nou algoritme:

Les conclusions extretes a partir dels resultats obtinguts de l'aplicació del nou algoritme basat en algunes de les conclusions prèvies extretes a partir dels primers resultats són les següents:

- Amb els mètodes triats per a cada interval es guanya molt temps, en calcular els resultats es triga de l'ordre de tres vegades menys aproximadament. Això és degut a que a la hora de buscar els retalls el nombre de mètodes utilitzats es mou entre l'1 i el 3. 1 en cas de que es trobi amb el primer mètode, en cas de que no s'utilitza un segon i si tampoc, fins a un tercer, si amb el tercer no s'ha trobat, es dona per segur que la icona no apareix en pantalla. Per tant, en comptes de calcular 6 vegades, es calcula entre la meitat i una sisena part. Estalviant una gran quantitat de temps, molt valorada a la optimització.
- El nombre d'errors amb aquest nou algoritme és una mica superior al nombre d'errors trobats utilitzant tots els mètodes. De l'ordre d'un 1% o un 2%. Aquest és un error que es considera assumible, veient la millora en el temps.
- Els mètodes escollits per cada interval s'han triat a partir de les primeres proves de resultats, tot i això es pot veure com a les últimes proves no acaben d'estar ben definits els intervals. De cara a seguir millorant el sistema s'hauran de seguir realitzant proves i a partir d'aquestes acabar de polir l'algoritme i els seus intervals per a seguir optimitzant el seu rendiment.

9.4. Conclusió general

Com a conclusió genèrica del treball es pot dir que utilitzant tant tots els mètodes com l'algoritme final que s'ha dissenyat, s'aconsegueixen resultats marcats en els objectius inicials, que serien trobar tant ícones idèntiques en una altra pantalla, com trobar-les malgrat tinguin petites diferències de formes o variïn alguns píxels, i seguir-les trobant independentment del color que mostren.

10. Propostes de millora

- Seguir profunditzant en la possibilitat del OCR:

En el sistema de reconeixement de caràcters OCR no s'hi ha profunditzat més donat que no era un dels objectius del projecte i que el que es necessita en el present es pot realitzar tot mitjançant el reconeixement de símbols o OSR. Tot i així es probable que arribi un moment en el futur, amb generació 3 on sí que es necessiti. En aquest cas s'haurà d'utilitzar l'algoritme que s'ha implementat però buscar la manera d'ajustar la seva precisió per a aconseguir el nivell d'exactitud buscat.

- Intentar millorar el temps de processat:

Ja s'ha realitzat aquest procés mitjançant l'algoritme que s'ha dissenyat a partir dels resultats. Tot i això, es probable que es pugui fer neteja del codi implementat amb l'objectiu d'estalviar càlculs innecessaris i d'aquesta manera estalviar temps. Un exemple era al comparar els histogrames dins la funció 'comparar_imatges_histograma_maxim()', on dins del bucle es calculava tant l'histograma de la imatge original com l'histograma del retall. Més tard es va veure que donat que l'histograma de la imatge original era sempre el mateix es podia calcular fora del bucle, el que permetia estalviar càlculs i, per tant, temps.

- Escalat de les imatges en variar la mida [35]:

Un exemple on es pot veure la necessitat d'implementar aquesta millora es pot trobar dins l'apartat de resultats, on s'ha intentat buscar les icones de generació 1 en els màsters de generació 2. En aquest cas al no tenir la mateixa resolució donava un munt d'errors, només detectables a partir de l'anàlisi humà. S'ha trobat una funció que treballa modificant la mida del màster fins que el resultat que troba és el màxim semblant a la icona de referència.

De moment s'ha decidit que no feia falta implementar aquesta funció perquè no s'esperen grans canvis de mida a generació 3, però es tindrà en compte en cas de que en un futur sigui necessari implementar-ho.

- Millorar el sistema de reconeixement:

Al comparar les imatges (original i retall) mitjançant l'histograma hi ha la possibilitat, que s'ha donat més d'un cop i que apareix dins l'apartat de resultats, de que la imatge que ha trobat considera que és prou semblant per a agafar-la com a solució final, però realment no és la que es busca. En aquests casos per a corroborar que la

imatge trobada és prou semblant a la icona original, s'ha pensat que es podria estudiar la manera d'extreure informació de la imatge de referència, com per exemple els contorns de l'objecte que es troba a dins, i comparar després amb els contorns trobats en el retall que es dona com a bo per a veure si s'hi troba un mínim de semblança. D'aquesta manera s'evitaria el que s'ha trobat en alguns resultats, que s'han denominat 'Falsos 'PASS''.

➤ Millora del sistema per a múltiples ícones iguals:

Es pot donar el cas alguna vegada que hi hagi dues o més ícones idèntiques dins la pantalla on s'estan buscant. Com a millora es podria modificar la funció que compara els histogrames dels retalls amb el de la icona original, per a que en comptes d'agafar només el retall que tingui l'histograma més semblant a l'original, en el cas de que el valor màxim de la comparació entre aquests es repeteixi, agafi tots els retalls pertinents. D'aquesta manera es podria aconseguir trobar més d'una icona en pantalla donat el cas de que aquestes són iguals.

➤ Crear una interfície:

S'havia proposat dins dels objectius inicials del projecte la opció de crear una interfície gràfica per a facilitar la utilització de la eina per part dels usuaris. Més tard, s'ha decidit que no era necessari. Tot i això, no deixa de ser una possible proposta de millora de cara a facilitar el seu ús. La interfície donaria la opció de fer dos tipus d'operacions:

- Opció 1:
 - Donada una icona i un màster, buscar aquesta dins el màster i retornar el resultat trobat, en el cas de que aparegui.
- Opció 2:
 - Donat una llista d'ícones i una llista de màsters buscar cada icona en cada màster. Una mica inspirat en com s'han obtingut els resultats de les proves.

11. Valoració personal

Com a valoració personal d'aquest projecte, puc dir que estic força orgullós de la feina feta. Realment m'ha servit per aprendre molt sobre el processat d'imatges i, més generalment, sobre la visió artificial. Pel que hem anat veient amb els treballadors i companys meus, que necessitaran l'ús d'aquesta eina, sembla que els serà molt útil en un futur molt pròxim.

A part, penso que la visió artificial és un tema amb molta força en el present i que en un futur molt pròxim en prendrà molta més, degut a la gran quantitat d'aplicacions que se li pot donar. Algunes de les aplicacions poden ser per exemple:

- L'automatització de diferents sectors de la indústria on es necessiti la comprovació visual de persones. Allà, la visió artificial oferia la possibilitat de produir 24h al dia sense que hi hagi d'haver una persona sempre allà, pel que es guanyaria en producció i s'estalviaria en sous. En el cas d'aquesta eina, l'aplicació va dirigida a donar suport als treballadors que estan validant les funcions de les ràdios. D'aquesta manera s'estalvien tests molt repetitius, i es poden deixar les validacions en funcionament tota la nit, pel que es guanya en temps treballat.
- Un aspecte molt interessant també és l'aplicació de la visió artificial en vehicles autònoms com poden ser cotxes o robots. Aquests necessiten un nivell de reconeixement d'objectes molt elevat, ja que han de ser capaços d'analitzar tot el que tenen al voltant per a evitar col·lisions o poder fer mal a persones.

Vull incloure que aquest projecte m'ha servit per a donar continuïtat a les meves pràctiques dins l'empresa SEAT, i gràcies en part a les pràctiques curriculars, i en part a aquest projecte, m'han ofert un contracte a través d'una enginyeria per a seguir treballant allà indefinidament

12. Agraïments

Com en la gran majoria de projectes, acostumen a intervenir més persones que només el que l'ha redactat, i trobo molt necessari donar les gràcies a aquesta gent, ja que sense la seva ajuda aquest projecte hagués estat una tasca molt més complicada de la que ja ha estat.

En primer lloc vull agrair tant a en Joan Heredia com a en Jonatan Rivilla, companys dins de Seat, per guiar-me en el projecte i per prestar-me la seva ajuda en tot moment, en el dubtes que he tingut tant en la programació, com en el plantejament de possibles millores i en la valoració i anàlisi de resultats .

També vull agrair a en Miguel García que accedís a que jo realitzés el projecte dins de Seat, i que accedís a ser el meu tutor allà dins.

Vull agrair a en Lluís Pérez per haver acceptat a ser el meu tutor dins la universitat per a portar-me el projecte.

13. Bibliografia

13.1. Referències bibliogràfiques

- [1] Documentació interna de Seat
- [2] MOTORPASIÓN, Volvo V40. [<http://www.motorpasion.com/volvo/volvo-v40-el-familiar-mas-deportivo>, 7 de Març del 2012]
- [3] OMICRONO, El nuevo reconocimiento de imágenes de Google puede identificar todos los objetos de una foto. [<http://www.omicrono.com/2014/09/el-nuevo-reconocimiento-de-imagenes-de-google-puede-identificar-todos-los-objetos-de-una-foto/>, 8 de setembre de 2014]
- [4] MUY INTERESANTE, Google crea su propio coche Autónomo. [<http://www.muyinteresante.es/innovacion/articulo/google-crea-su-propio-coche-autonomo-881401361097>]
- [5] MOTORPASIÓN FUTURO, Cómo funciona el coche Autónomo de Google. [<http://www.motorpasionfuturo.com/coches-del-futuro/como-funciona-el-coche-autonomo-de-google>, 12 de Maig de 2012]
- [6] UNIVERSIDAD DE SEVILLA. DEPARTAMENTO DE ARQUITECTURA Y TECNOLOGIA DE COMPUTADORES, Fundamentos de la Visión Artificial. Prof. Dr. Francisco Gómez Rodríguez
- [7] PLEORA TECHNOLOGIES, iPORT Analog-Pro External Frame Grabbers. [<http://www.pleora.com/our-products/frame-grabbers/iport-analog-pro>]
- [8] FOTONOстра, Modos de color: RGB, CMYK y sRGB. [<http://www.fotonostra.com/grafico/rgb.htm>]
- [9] DESARROLLO WEB, Modelos de color.
- [10] GUTENBERG, ¿De qué hablamos cuando hablamos de espacio de color y de un perfil de color?. [<http://www.fundaciongutenberg.edu.ar/inicio/19/284-ide-que-hablamos-cuando-hablamos-de-espacio-de-color-y-de-un-perfil-de-color->, 23 de Maig de 2013]
- [11] UNIVERSITAT DE LES ILLES BALEARS (UIB), DEPARTAMENT DE MATEMÀTIQUES I INFORMÀTICA, Tema5_Filtrado de Imágenes, Y. González

- [12] UNIVERSIDAD NACIONAL DEL SUR, BAHÍA BLANCA, Metodologías de segmentación y medición de características lineales en imágenes de sensado remoto, Marina P. Cipolletti, 2012
- [13] POLITECNICO DE MILANO , DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA, A Tutorial on Clustering Algorithms, Clustering: An Introduction. [http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/]
- [14] POLITECNICO DE MILANO , DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA, A Tutorial on Clustering Algorithms, K-Means Clustering. [http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html]
- [15] UNIVERSIDAD DE JAÉN, ÀREA DE INGENIERÍA DE SISTEMAS Y AUTOMÀTICA, Detección de bordes en una imagen, Curso 2005/2006
- [16] S.A.B.I.A., SISTEMAS ADAPTATIVOS Y BIOINSPIRADOS EN INTELIGENCIA ARTIFICIAL, El Histograma de la Imagen
- [17] ELSEVIER, Detección automatizada de microaneurismas mediante crecimiento de regiones y red neuronal Fuzzy Artmap. [<http://www.elsevier.es/es-revista-archivos-sociedad-espanola-oftalmologia-296-articulo-deteccion-automatizada-microaneurismas-mediante-crecimiento-90148090>, Septembre 2012]
- [18] PROYECCIÓN DEL MÉTODO DE SEGMENTACIÓN DEL CONJUNTO DE NIVEL EN GPU, Julián Lamas, Pablo Quesada, Francisco Argüello, Dora B. Heras, Montserrat Bóo.
- [19] UPC, EUETIT, Editor de Imágenes basado en Regiones, Aplicación en entorno Matlab. Samira Hervella Azouzi. Curs 2005/2006.
- [20] IEEE LATIN AMERICA TRANSACTIONS, Segmentación de imágenes utilizando la transformada Watershed: obtención de marcadores mediante lògica difusa, M. A. González, V. L. Ballarin, Juny 2008.
- [21] UNIVERSIDAD DE SEVILLA, INGENIERÍA INFORMÁTICA, Tema 5: Segmentación, Curs 2015/2016
- [22] GRAZ UNIVERSITY OF TECHNOLOGY, AUSTRIA, Survey of appearance-based methods for object recognition, Peter M. Roth and Martin Winter, 15 de Gener de 2008.
- [23] UNIVERSIDAD DE LA LAGUNA, OpenCV: Librería de Visión por Computador. [<https://osl.ull.es/software-libre/opencv-libreria-vision-computador/>]

- [24] ROBOLOGS, Detección de triángulos con OpenCV y Python. [<http://robologs.net/2015/01/25/deteccion-de-triangulos-con-opencv-y-python/>, 25 de Gener de 2015]
- [25] SIMPLECV, Computer Vision platform using Python. [<http://simplecv.org/>]
- [26] OPENCV, Template Matching.
[http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html]
- [27] NEWTONE, Sum squared difference. [<http://www.linguee.com/english-spanish/translation/sum+squared+difference.html>]
- [28] PAUL BOURKE, Cross Correlation. [<http://paulbourke.net/miscellaneous/correlate/>, Agost de 1996]
- [29] INVESTOPEDIA, Correlation Coefficient.
[<http://www.investopedia.com/terms/c/correlationcoefficient.asp?layout=infini&v=5E&orig=1&adtest=5E>]
- [30] OPENCV, Template Matching.
[http://docs.opencv.org/master/d4/dc6/tutorial_py_template_matching.html#gsc.tab=0]
- [31] PYIMAGESearch, How-To: 3 Ways to Compare Histograms using OpenCV and Python. [<http://www.pyimagesearch.com/2014/07/14/3-ways-compare-histograms-using-opencv-python/>, 14 de Juliol de 2014]
- [32] OPENCV, Histograms.
[<http://docs.opencv.org/2.4/modules/imgproc/doc/histograms.html>]
- [33] OPENCV, Operations on Arrays.
[http://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html]
- [34] POLYMATHE BEQUEATH, Optical Character Recognition in Python. [<https://saxenarajat99.wordpress.com/2014/10/04/optical-character-recognition-in-python/>, 4 d'Octubre de 2014]
- [35] PYIMAGESearch, Multi-scale Template Matching using Python and OpenCV. [<http://robologs.net/2014/07/02/deteccion-de-colores-con-opencv-y-python/>]

13.2. Bibliografia complementaria

A continuació s'anomenen algunes webs que s'han usat per a fer proves de programació, provant funcions diverses per a entendre millor el funcionament del que s'estava duent a terme:

[1] OPENCV, Contours: Getting started.

[http://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html#gsc.tab=0]

[2] RAPID TABLES, HTML Color Picker.

[<http://www.rapidtables.com/web/color/color-picker.htm>]

[3] ROBOLOGS, Detección de colores con OpenCV y Python.
[<http://robologs.net/2014/07/02/deteccion-de-colores-con-opencv-y-python/>]