TÍTOL: Servei de detecció, reconeixement i localització de senyals de trànsit per terminals mòbils: Software de tractament d'imatges

TITULACIÓ: Enginyeria Tècnica de Telecomunicacions, especialitat Sistemes de Telecomunicacions

Autor: Jordi Cornellà i Calsina

Director: Francesc Tarrés Ruiz

Data: 22 de febrer de 2012
Resum

Aquest projecte s'ha ideat, dissenyat i programat per solucionar un problema de la vida quotidiana. La solució permet assistir al conductor per evitar possibles accidents de trànsit. Consisteix en una aplicació per smartphones que mostra per pantalla els senyals de trànsit. El senyals mostrats són els que el conductor es trobarà metres més endavant.

S'ha tingut en consideració que els senyals poden canviar al llarg del temps. Per aquest motiu, s'ha desenvolupat una base de dades dinàmica. Aquest dinamisme ve determinat per dos processos. El primer és que l'aplicació és capaç de realitzar fotografies de la carretera i enviar-les. Tot això sense la intervenció de l'usuari. El segon procés s'encarrega de recollir les fotografies, analitzar-les i actualitzar la base de dades.

Totes les comunicacions necessàries són processades per un servidor intermediari. Aquest enllaça els smartphones amb la base de dades i gestiona l'enviament de les fotografies.

Degut a que aquest projecte abasta diferents matèries es decideix separar-les en tres volums.

Aquest volum desenvolupa com s'ha dissenyat i programat el servidor de processat d'imatge. La seva funció més important és la de mantenir actualitzada la base de dades de senyals.

El seu funcionament es basa en filtrar, detectar i reconèixer les imatges enviades pel dispositius mòbils.

El filtratge discrimina els colors no rellevants de la imatge. Els colors que es deixen passar són els vermells i els blaus. Aquests colors són els que contenen la gran majoria de senyals de trànsit.

Per fer la detecció es necessita aconseguir la imatge processada en format binari. D'aquesta manera es poden detectar objectes en la imatge. Els objectes s'analitzen per buscar possibles formes geomètriques descartant els objectes no desitjats.

El reconeixement utilitza el mètode de la correlació. Es calcula comparant uns patrons predefinits amb els objectes que provenen de la detecció. A partir del càlcul s'obté quin senyal és i s'actualitza la base de dades.

Sense aquesta part no seria possible l'actualització automàtica de la base de dades.
Overall, the project was designed and developed as a solution for a specific problem. The solution helps drivers avoid car crashes by showing traffic signs on the screen. This is an application for smartphones that allows drivers to see traffic signs beyond their visibility.

We considered the life of traffic signs, which can change over time. For this reason, we developed a dynamic database. This dynamism is achieved through two processes: the first involves the application taking photos of the road without human intervention, and the second picks up these photos, analyzes them, and updates the database.

All communications are processed by the proxy server, which connects the smartphones to the database and manages the images.

As a large project with various subjects, we decided to split it into three reports. This report explains how the image processing server was designed and developed. Its primary function is to keep the traffic sign database up to date. Its work is based on filtering, detection, and recognition of images from mobile devices.

The filtering aims to discriminate the less important colors in the image. For this project, the most important colors are red and blue, which are frequently used on traffic signs.

The detection requires a binary image from the original image for processing, making objects detectable within the image. These objects are analyzed for geometric shapes, and shapeless objects are discarded.

The recognition uses the correlation method. It compares preset patterns with the objects from the detection block. From these comparisons, the traffic sign is identified, and the database is updated.

Without this part of the project, it wouldn't be possible to maintain the database updates.
ÍNDICE

INTRODUCCIÓ ........................................................................................................... 1

CAPÍTOL 1. TRAFFIC SIGNS DETECTION & RECOGNITION (TSDR) .............. 3
  1.1. Dificultats del conductor .............................................................................. 3
  1.2. Sistema TSDR ............................................................................................. 4
  1.3. Visió Global .................................................................................................. 4
  1.4. Mercat Actual .............................................................................................. 5
  1.5. Objectius ....................................................................................................... 6
     1.5.1. Objectius Principals .................................................................................. 6
     1.5.1. Altres Objectius ....................................................................................... 8
  1.6. Resum del capítol .......................................................................................... 8

CAPÍTOL 2. IMAGE PROCESSING SERVER (IPS) .............................................. 9
  2.1. Introducció .................................................................................................... 9
  2.2. Desenvolupament ........................................................................................ 10
     2.2.1. Llibreries externes .................................................................................. 10
     2.2.2. Llibreries desenvolupades pel servidor IPS ............................................. 11
  2.3. Resum del capítol .......................................................................................... 11

CAPÍTOL 3. PROCESSAT D'IMATGE ................................................................. 12
  3.1. Introducció ................................................................................................... 12
     3.1.1. Model de color HSV ............................................................................... 12
  3.2. Filtratge .......................................................................................................... 14
     3.2.1. Introducció ............................................................................................... 14
     3.2.2. Llibreries desenvolupades pel filtratge ...................................................... 14
  3.3. Detecció ........................................................................................................... 16
     3.3.1. Introducció ............................................................................................... 16
     3.3.2. Llibreries desenvolupades per la detecció ............................................... 16
  3.4. Reconeixement ................................................................................................ 20
     3.4.1. Introducció ............................................................................................... 20
     3.4.2. Llibreries desenvolupades per la detecció ............................................... 21
     3.4.3. Patrons predefinits .................................................................................... 21
     3.4.4. Codificació dels senyals utilitzats pel reconeixement ................................ 21
  3.5. Exemples ........................................................................................................... 24
     3.5.1. Escenari 1: Fotografia d'un carrer amb tres senyals detectables ............... 24
     3.5.2. Escenari 2: Fotografia de baixa qualitat amb un senyal detectable .......... 29
  3.6. Resum del capítol ............................................................................................ 32
CAPÍTOL 4. CONNEXIONS ENTRE SERVIDORS.........................................................33

4.1. Introducció......................................................................................................................33

4.2. Configuració del servidor ..............................................................................................33

4.3. Connexió entre servidor FTP i servidor IPS...............................................................34
  4.3.1. Gestió de fitxers ......................................................................................................34

4.4. Connexió entre servidor IPS i servidor Base de Dades.............................................35

4.5. Resum del capítol ..........................................................................................................36

CAPÍTOL 5. CONCLUSIONS.......................................................................................37

5.1. Ambientalització............................................................................................................39

5.2. Línies Futures ...............................................................................................................39

BIBLIOGRAFIA ....................................................................................................................41
INTRODUCCIÓ

Els perills es presenten de diverses formes en la carretera per a un conductor; des de condicions meteorològiques adverses, passant pel mal manteniment de les vies fins la conducció nocturna. Tots aquests escenaris tenen un risc en comú: la falta de visibilitat que provoca que el conductor no es pugui anticipar.

Com a conductors ens hem trobat amb aquesta problemàtica i partint d'un treball dissenyat per nosaltres en l'optativa de Sistemes Audiovisuals va sorgir una possible solució. La idea consisteix en mostrar al conductor, per mitjà d'algun dispositiu, els senyals de trànsit que es troben més endavant. Facilitant així la conducció.

Com que actualment els smartphones disposen de GPS, connexió a Internet i a més es troben a l'abast de tothom, es decideix utilitzar aquest dispositiu per implementar la nostra idea. Tot i haver-hi moltes aplicacions pels smartphones no hi ha cap aplicació que ofereixi quelcom semblant a la nostra idea. És per això que hem dissenyat i programat una aplicació des de zero.

Al no disposar d'accés a una possible base de dades governamental, on estiguin enregistrats tots els punts en que es troben els senyals de trànsit, decidim crear la nostra pròpia base de dades. Per evitar que l'aplicació tingui un pes considerable es decideix externalitzar la base de dades.

Els senyals poden ser variants al llarg del temps, per això es pensa en una base de dades dinàmica que s'actualitza automaticament. Un mètode d'actualitzar la base de dades és utilitzant els propis usuaris de l'aplicació. Com més usuaris hi hagi, més fiable serà la informació proporcionada, ja que aquesta s'obté de fotografiar la carretera. A partir d'aquí s'envien a un servidor que analitzarà la fotografia en busca de possibles senyals. Així aconseguim la idea d'una base de dades dinàmica.

Aquest projecte no és només això, sinó que té una perspectiva de futur de millora contínua. Es pot adaptar a dispositius implementats a dins del vehicle, com per exemple, el GPS. Es podria fer servir en un futur en vehicles amb conducció automàtica.

La finalitat del projecte és per tant, desenvolupar un sistema mitjançant una aplicació que faciliti la conducció vial.

Aquest sistema tracta d'adquirir, detectar, reconèixer, localitzar i gestionar els senyals de trànsit. El sistema està basat en una aplicació Android que mostrarà les properes senyals de trànsit que el conductor es trobarà. L'aplicació captura, de forma periòdica, imatges de la carretera que s'envien a un servidor. L'anàlisi consisteix en identificar els nous possibles senyals, modificacions dels senyals anteriors o confirmació de les localitzacions de les existents. Aquestes imatges s'utilitzaran per actualitzar la base de dades dinàmica del servidor. A més, a partir de la posició GPS del terminal mòbil, el servidor proporcionarà informació
dels propers senyals de trànsit. Opcionalment també es pot activar avisos per radars fixos.

Aquest treball ha estat dissenyat i realitzat per tres persones. S’ha dividit la memòria en tres volums per explicar en detall cada bloc.

**Volum 1: Software de terminal mòbil.**
Desenvolupa la interfície gràfica, la funcionalitat interna de l’aplicació i les diferents connexions necessàries per tal que l’aplicació compleixi amb les expectatives proposades.

**Volum 2: Software del servidor**
Desenvolupa com establir les connexions entre el terminal mòbil, les bases de dades i el tractament d’imatges. A més de com s’ha dissenyat la base de dades dinàmica.

**Volum 3: Software de tractament de imatges**
Desenvolupa el processament i tractament d’imatge. Detecció i reconeixement de senyals de trànsit.

En resum, és un projecte que té com a finalitat ajudar als conductors a poder anticipar-se a les indicacions dels senyals de trànsit. El nostre objectiu és intentar reduir el nombre d’accidents, com també el de promoure la filosofia de compartir la informació entre tots els usuaris per el bé de tots.

Creiem en aquest projecte perquè ha sorgit d’una petita idea nostra que ens ha permès dissenyar i programar aquest sistema des de zero.
1.1 Dificultats del conductor

Els conductors han de ser responsables i tenir en compte molts factors. Cal estar pendent del què passa fora del vehicle, com per exemple aturar-nos als senyals de STOP, respectar els vianants. Alhora, també el què passa dins del vehicle, com per exemple estar al corrent dels menors o de l'acompanyant. Però hi ha molts factors que són difícils de controlar i/o preveure.

El factor meteorològic, és el menys previsible, en el cas de boira, neu o pluja dificulten la visibilitat de la via per la qual s’està circulant. Malgrat que es redueixi la velocitat i es prenguin les mesures adequades no es podrà evitar la mala visibilitat.

Un altre factor, són les carreteres secundàries que es troben en mal estat i tenen els senyals descuidats.

També ho són els accidents, el 42% dels accidents mortals es produeixen per la nit. Malgrat que en aquestes hores existeix un 60% menys de trànsit que durant el dia. La conducció nocturna implica un risc tres vegades major que la conducció diürn.

Algunes característiques de la visió nocturna són:

- Existeix una limitació de la percepció. Es produeix una disminució de l’agudesa visual, s’aprecien pitjor els obstacles i es pertorba el sentit cromàtic.

- Existeix una limitació del camp visual ajustant-se a la zona il·luminada. Els llums del vehicle il·luminen eficaçment una zona de 100m mínims amb el llum de carretera, i de 40m amb el llum d’encreuament.

- Existeixen factors que influeixen en la capacitat visual nocturna tal com: l’edat, la fatiga, l’alcohol, les drogues, les malalties, les limitacions visuals concretes, i altres.

Cal destacar que l’any 2011 van morir a Espanya com a conseqüència d’accidents en carretera 1479 persones, 14.5% menys que l’any 2010. Cada any els accidents es van reduïnt.

Per els motius anteriors esmentats, la intenció del sistema TSDR és augmentar la seguretat en el trànsit ajudant al conductor.
1.2 Sistema TSDR

El Sistema "Traffic Signs Detection and Recognition" (TSDR) es un servei de detecció, reconeixement i localització de senyals de trànsit per terminals mòbils.

Aquest sistema es desenvolupa mitjançant una aplicació per Android. Aquesta aplicació és compatible en els smartphones (mòbils intel·ligents) que tenen el sistema operatiu mòbil Android.

El mòbil s'ha de col·locar en el suport del cotxe en mode horitzontal i amb visió directa a la part dreta de la carretera. Això és així, perquè el mòbil realitza fotografies cada cert temps dins l'horari diürn. L'objectiu és captar els màxims senyals de trànsit possibles en el trajecte de l'usuari.

Tanmateix, l'usuari veu els senyals que es troben metres més endavant per pantalla. Aquest sistema cobreix la zona geogràfica de Catalunya.

1.3 Visió Global

El disseny per portar a terme aquest projecte consta de cinc elements claus: smartphone, proxy (servidor intermediari), database (base de dades), FTP i Image Processing Server (servidor de processat d'imatge, IPS).

El smartphone, dispositiu mòbil, està connectat amb el proxy i el FTP. El dispositiu mòbil envia al proxy peticions cada cert temps per saber si hi ha nous senyals de trànsit. Aquest consulta la base de dades i si hi ha algun senyal, retorna la resposta cap el smartphone de l'usuari corresponent.

Cal destacar que el sistema TSDR no emmagatzema cap dada de tipus personal. D'aquesta manera es compleix amb la llei orgànica 15/1999 de 13 de desembre de protecció de dades de caràcter personal.

Quan el dispositiu mòbil realitza una fotografia l'envia cap el servidor FTP. Després l'IPS analitza la fotografia en busca de senyals de trànsit.

Un cop analitzada la fotografia pel servidor IPS s'envia a la base de dades si es troba un senyal de trànsit. Aquesta base de dades és dinàmica. S'actualitza automàticament conforme els usuaris van realitzant més fotografies. La figura 1 mostra l’esquema global del sistema TSDR.
1.4 Mercat Actual

Actualment hi ha alguns vehicles que incorporen un sistema de visibilitat dels senyals en el quadre de comandament. Alguns dels fabricants que comercialitzen aquest sistema són: Audi, BMW, Mercedes, Volkswagen, Saab i Opel.

La figura 1.2 mostra que els preus d’aquests automòbils no estan a l’abast de tothom. A més a més, els vehicles que estan sortint al mercat només implementen el reconeixement dels senyals de velocitat, com mostra la figura 1.3.
A diferència dels sistemes que els vehicles actualment incorporen, el sistema TSDR destaca perquè:

- Abasta fins a quaranta senyals de trànsit.
- El sistema mostra fins a 4 senyals simultàniament per pantalla.
- El sistema implementa l’opció d’avisar mitjançant àudio els senyals.
- El sistema està format per una base de dades dinàmica que va actualitzant-me a mesura que els usuaris van realitzant fotografies. Aquesta base de dades és el punt fort destacable i diferenciable amb la resta de sistemes existents.

1.5 Objectius

1.5.1 Objectius Principals

Aquest sistema és capaç de detectar i reconèixer els 40 senyals de trànsit que es mostren a continuació en les figures 1.4, 1.5, 1.6 i 1.7.
De tots els senyals possibles s'ha escollit aquests 40 senyals perquè són els més habituals. S'ha seleccionat triangles i rodones per fer més senzilla la detecció i el reconeixement.

El punt fort del projecte és la base de dades dinàmica. Aquesta s'actualitza mitjançant les fotografies que són realitzades i enviades pels telèfons mòbils. Quan més usuaris utilitzen l’aplicació, més precisa serà la informació. Això fa que el sistema TSDR sigui autoalimentat pels usuaris. També cal destacar que la base de dades no està implementada dins del telèfon mòbil, sinó que està externalitzat en un servidor on es connecta l'aplicació. Això és bo per l'usuari perquè evita descarregar freqüentment les actualitzacions de la base de dades.

Un altre objectiu són les connexions amb el servidor. El servidor ha d'entendre les demandes del mòbil i retornar-li una resposta.
Finalment, l'últim objectiu és mostrar els senyals de trànsit en la pantalla del mòbil. Aquest objectiu és el més valorat per l'usuari. Es pot visualitzar simultàniament fins a quatre senyals de trànsit en grups de dos.

1.5.2 Altres Objectius

L'aplicació TSDR a més a més, incorpora la opció d'activar l'avís de la proximitat d'un radar fix i avís per excés de velocitat. Per evitar la distracció del conductor, a l'aplicació es pot habilitar el sintetitzador de parla que informa dels senyals mitjançant la veu. El conductor pot conèixer per on està circulant gràcies a la visualització en temps real de la posició del vehicle en el mapa.

Tot això està implementat en una interfície gràfica fàcil d'utilitzar per l'usuari.

1.6 Resum del capítol

En aquest capítol s'ha explicat que la finalitat del sistema TSDR és mostrar a l'usuari els senyals de trànsit que es troben més enllà del seu camp visual.
CAPÍTOL 2. IMAGE PROCESSING SERVER (IPS)

2.1 Introducció

El Image Processing Server (IPS) és fonamental en el sistema Traffic Sign Detection and Recognition (TSDR), ja que és l'encarregat d'actualitzar la base de dades a mesura que analitza les fotografies.

Aquest servidor està en comunicació directa amb el servidor FTP i la base de dades, com mostra la figura 2.1.

El IPS està concebut com un sistema autònom. Això vol dir que un cop configurat i posat en marxa, no cal la intervenció humana. El servidor és capaç de detectar si hi ha fotografies per processar, en cas contrari ell mateix es queda a l'espera de noves fotografies.

Es poden diferenciar tres grans blocs pel processat d'imatge: el bloc de filtratge, el bloc de detecció i el bloc de reconeixement. A més a més, disposa d'una eina que automàticament permet escriure el resultat de l'anàlisi a la base de dades.

Una de les seves especificacions és que permet detectar els senyals majors de 30x30 píxels. Si els senyals són menors que la mida establerta, significa que la fotografia s'ha realitzat a una distància més gran que l'esperada. Això comporta que el pictograma del senyal es confongui amb el contorn.

Tot i que el servidor s'ha dissenyat amb una interfície gràfica mínima, està pensat per a ser executat com un servei del sistema operatiu.

Aquest servidor es basa en la correlació creuada ja que no es tracta d'un sistema en temps real.

Fig. 2.1 Diagrama de xarxa del servidor IPS
2.2 Desenvolupament

El servidor s’ha programat en llenguatge C# perquè aquest incorpora tot el necessari per desenvolupar una interfície gràfica. Tot i que la versió final no té una interfície gràfica, per fer les proves és necessari disposar d’aquesta per anar visualitzant els resultats. A més, com que el servidor proxy i el sistema de gestió de base de dades estan programats en C#, s’ha utilitzat el mateix llenguatge per facilitar la implementació.

2.2.1 Llibreries externes

Un dels inconvenients d’utilitzar C# és la falta de llibreries dedicades al processament d’imatge dins del paquet Visual Studio 2010. No obstant això, existeixen llibreries de processament d’imatge compatibles amb C#.

- **Emgu CV:** És una llibreria de processat d’imatge per C# basada en OpenCV. OpenCV només suporta llenguatges de programació C++, C i Python. Emgu CV és una interfície entre OpenCV i C#. Les crides a OpenCV es fan a través de Emgu CV. [20]

  De Emgu CV s’utilitzen les rutines de conversió de color, conversió del tipus de dades de les imatges, redimensionament i carga d’imatges en variables. Aquestes rutines s’apliquen al filtratge i al condicionament de la imatge pel reconeixement.

- **AForge:** És una llibreria dissenyada per desenvolupadors i investigadors en els camps de Visió Artificial i Intel·ligència Artificial. A diferència de OpenCV, el llenguatge natiu de AForge és C#. [29]

  La rutina que s'utilitza de AForge és la separació d'objectes d'una imatge. Aquesta rutina s'aplica en la detecció dels possibles senyals de trànsit.

Per comparar entre senyals detectats i els patrons predefinits s'utilitza una llibreria matemàtica externa:

- **Alglib:** És una llibreria de processat de dades i càlcul numèric compatible amb C#. La funció matemàtica que s'utilitza és la correlació creuada, aplicada als càlculs de reconeixement.

Al necessitar accés a la base de dades, el IPS requereix d'una altra llibreria externa.

- **Oracle.Data.Access:** Es tracta d'una llibreria externa proporcionada per Oracle i compatible amb C#. Aquesta llibreria permet l'accés a les bases de dades d'Oracle mitjançant un protocol propietari anomenat Transparent Network Substrate (TNS). S'encarrega d'obrir les connexions, enviar el comandament SQL, rebre la resposta i finalment tancar la connexió.
2.2.2 Llibreries desenvolupades pel servidor IPS

Aquestes llibreries s'han creat específicament pel projecte. El nom assignat a cada llibreria està relacionat amb l'algoritme que implementa. Això facilita l'entorn de programació. La taula 2.1 mostra les llibreries programades i el seu propòsit. Aquestes llibreries són detallades en capítols posteriors.

**Taula 2.1 Llibreries pròpies específiques del projecte**

<table>
<thead>
<tr>
<th>Nom</th>
<th>Propòsit</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP2Filter.dll</td>
<td>Traspasar fitxers entre servidors.</td>
</tr>
<tr>
<td>ImageLoader.dll</td>
<td>Convertir formats d'imatge.</td>
</tr>
<tr>
<td>RedFilter.dll</td>
<td>Filtrar el color vermell.</td>
</tr>
<tr>
<td>BlueFilter.dll</td>
<td>Filtrar el color blau.</td>
</tr>
<tr>
<td>Labeling.dll</td>
<td>Buscar objectes en la imatge.</td>
</tr>
<tr>
<td>ShapeDetection.dll</td>
<td>Detectar la forma geomètrica dels objectes.</td>
</tr>
<tr>
<td>Decision-maker.dll</td>
<td>Decidir a quina forma geomètrica s'assembla.</td>
</tr>
<tr>
<td>ExtractRois.dll</td>
<td>Extreure els objectes de la imatge.</td>
</tr>
<tr>
<td>Normalization.dll</td>
<td>Normalitzar els valors de la imatge.</td>
</tr>
<tr>
<td>2D-2-1D.dll</td>
<td>Convertir la matriu a vector.</td>
</tr>
<tr>
<td>Patterns.dll</td>
<td>Comparar els objectes amb patrons.</td>
</tr>
<tr>
<td>Sign2DB.dll</td>
<td>Insertar informació a la base de dades.</td>
</tr>
</tbody>
</table>

2.3 Resum del Capítol

En aquest capítol s'ha vist que el servidor IPS s'encarrega de recollir les imatges del servidor FTP, processar-les i escriure la informació a la base de dades.
3.1 Introducció

En aquest capítol s'explica en detall com està dissenyat el bloc del filtratge, el bloc de detecció i el bloc de reconeixement. En la figura 3.1 es mostra la interconnexió dels blocs amb les seves llibreries.

Fig. 3.1 Diagrama dels blocs del processat d'imatge amb les llibreries desenvolupades per cada bloc.

3.1.1 Model de color HSV

Les fotografies són rebudes en model de color RGB (Red, Green & Blue). Amb aquest model de color no és pràctic realitzar el filtratge perquè el valor del píxel es pot interpretar erròniament. Un exemple és quan els valors RGB són molt alts. Malgrat que el valor vermell superi el valor de verd i de blau, la combinació de tots tres dona com a resultat el color blanc.
Per aquest motiu s'utilitza el model HSV (Hue, Saturation & Value) que separa el píxel en matís (color), saturació i valor. Com es mostra en la figura 3.2 i 3.3. D'aquesta manera és més fàcil l'extracció del color perquè és més proper a la percepció de l'ull humà.

El model HSV defineix un model de color en termes dels seus components constituents en coordenades cilíndriques:

- **Matís**: El tipus de color (com vermell, blau o groc). Es representa com un grau d'angle els valors possibles del qual van de 0° a 360°. Cada valor correspon a un color. Exemples: 0° és vermell, 60° és groc i 120° és verd.

- **Saturació**: Es representa com la distància a l'eix de brillantor negre-blanc. Els valors possibles són del 0 al 100%. Quan disminueix la saturació d'un color, augmenta la tonalitat grisenca i s'atenua el color.

- **Valor**: La brillantor del color. Representa l'altura a l'eix blanc-negre. Els valors possibles són del 0 al 100%. 0 sempre és negre. Depenent de la saturació, 100 pot ser blanc o un color més o menys saturat.

**Fig. 3.2** Cilindre HSV

**Fig. 3.3** Distribució del matís i percentatge de valor (brillantor)
3.2 Filtratge

3.2.1 Introducció

És el bloc encarregat de la discriminació de colors. Utilitzant la llibreria externa Emgu CV, s’aconsegueix que en la imatge de la figura 3.4 només resti en blanc els píxels de color que volem obtenir. La resta de la imatge en negre.

![Imatge de prova abans de passar pels filtres.](image)

**Fig. 3.4** Imatge de prova abans de passar pels filtres.

El sistema TSDR reconeix 40 senyals. Els colors a filtrar són el vermell i el blau. Per això, s’ha programat dos filtratges, un per destacar els píxels vermells, figura 3.5 i l’altre per destacar els píxels blaus, figura 3.6. Per poder discriminar els colors no desitjats s’estableixen uns llindars, un pel blau i un pel vermell.

![Sortida del filtre vermell](image)

**Fig. 3.5** Sortida del filtre vermell

![Sortida del filtre blau](image)

**Fig. 3.6** Sortida del filtre blau
En els dos casos s'ha fet servir la mateixa estructura. Primer, s'obté la imatge en format BGR (Blue, Green i Red). Aquesta imatge es transforma a un format del model de color HSV amb l'ajuda de la llibreria externa Emgu CV i es separa per Hue, Saturation i Value. Per cada component es comprova que el valor del píxel està per sobre del llindar establert. Si supera el llindar, el píxel manté el seu valor, si no s'assigna com a negre. Amb el resultat dels tres components es realitza una operació AND per assegurar que el píxel supera els tres línidars. D'aquesta manera es pot seleccionar els colors desitjats. Finalment, es binaritza la imatge resultant, figura 3.7.

**Fig. 3.7** Diagrama del bloc de filtratge

### 3.2.2 Llibraries desenvolupades pel filtratge

- **RedFilter.dll**: És la llibreria encarregada de filtrar el color vermell. El primer pas és convertir el model de color RGB a model HSV.

  Tot seguit s'aplica la separació per components i es comprova que els valors dels píxels estiguin per damunt dels llindars establerts.

  La imatge resultant és una imatge en blanc i negre. El color blanc representa tot el que s'ha trobat de color vermell dins dels línidars del fíltre.

- **BlueFilter.dll**: És la llibreria encarregada de filtrar el color blau. El primer pas és convertir el model de color RGB a model HSV.

  Tot seguit s'aplica la separació per components i es comprova que els valors dels píxels estiguin per damunt dels llindars establerts.

  La imatge resultant és una imatge en blanc i negre. El color blanc representa tot el que s'ha trobat de color blau dins dels línidars del fíltre.
3.3 Detecció

3.3.1 Introducció

És el bloc encarregat de trobar en quina posició de la imatge hi ha un possible senyal i extreure'l a una imatge nova.

Quan el procés del filtratge finalitza, la imatge es recull en el bloc de detecció. Les dades obtingudes es convergeixen al format Bitmap (Image Data Conversion). Aquest format és natiu de C# i és el que utilitza la llibreria AForge. Aquesta llibreria facilita la recerca d'objectes en la imatge. Un cop detectat tots els objectes de la imatge (Connected-Component Labeling) es discriminen per la seva mida (Cleaner). Tot seguit, s'analicza el contingut dels objectes (Shape Detection) i es decideix a quina forma geomètrica pertanyen (Shape Decisor). Finalment, s'obté la posició de l'objecte per extreure la regió d'interès de la imatge original (ROI Extraction) i aquesta s'envia al bloc de reconeixement. El diagrama de la figura 3.8 reflecteix els passos esmentats.

Dos conceptes bàsics a tenir en compte per entendre com funciona el bloc de detecció són:

- **Connected-component**: Conjunt de píxels del mateix valor contigus en una imatge binaria, figura 3.9 i 3.10. [37]
- **Object**: Cadascun dels connected-components etiquetats. Inclouen la informació per la localització en la imatge, figura 3.11.
- **ROI**: Extracció d'una part de la imatge en la que es vol realitzar alguna operació amb ella, figures 3.16 i 3.17.

3.3.2 Llibraries desenvolupades per la detecció

- **Labeling.dll**: És la llibreria encarregada de cercar els components connectats entre si com en la figura 3.9. Aquests components connectats són un conjunt de píxels que estan contigus en una imatge binaria.
(connected-component). Identifica cada component connectat assignant a cadascú una etiqueta única, com mostra la figura 3.10. A més a més, introduceix la informació necessària per poder situar el Object en la imatge com són la posició, el centre i el rectangle que forma el Object. La figura 3.11 mostra un exemple de Object.

De tots els possibles Objects es descarten aquells que són menors de 30x30 píxels i els que estan dins d'un altre Object.

![Connected Components](image1.png)  ![Labeled Connected Components](image2.png)

**Fig. 3.9** Components connectats  **Fig. 3.10** Components connectats i etiquetats

![Exemple de Object](image3.png)

**Fig. 3.11** Exemple de Object

- **ShapeDetection.dll**: S'encarrega de detectar la forma geomètrica del Object. S'aprofita que el fons té un valor zero i la possible figura un valor diferent a zero.

A partir del centre del Object, es calcula la distància horitzontal en píxels des del centre fins que troba un valor diferent a zero. Això es realitza tant a la dreta com a l'esquerra.

Finalitzat el càlcul de la distància d'aquesta posició, es repeteix el mateix procés fins arribar al punt més alt de la figura. Un cop assolit, es torna al centre i es repeteix el procés fins arribar el punt més baix de la figura.
Conclòs el procés s'obté la figura escanejada horitzontalment com en la figura 3.12. Les distàncies calculades formen una representació gràfica.

![Fig. 3.12 Formes geomètriques escanejades](image1)

- **Decision-maker.dll**: S'encarrega de decidir si la gràfica obtinguda en el ShapeDetection correspon: a un cercle, a un triangle o a un triangle invertit.

Aquestes es comparen amb les gràfiques ideals i si no s'ajusta a cap, és definida com a desconeguda.

![Fig. 3.13 Gràfica ideal d'un cercle](image2)

En la gràfica de la figura 3.13 s'observa que del semicercle superior s'obté la primera corba fins el valor mínim. El valor inicial és el diàmetre del cercle. A mesura que s'escaneja cap el punt més alt de la figura, la gràfica va decreixent fins arribar al punt mínim. Després d'arribar al punt mínim, es torna el valor inicial perquè es repeteix el procés en el semicercle inferior.

Degut a la perspectiva de la imatge no s'aconsegueix cercles perfectes. Això comporta que els valors mínims no siguin iguals. Per això, s'adapta la gràfica per compensar les possibles diferències entre ideal i real.
En la gràfica de la figura 3.14, la primera recta decreixent representa l'anàlisi de la figura des del centre fins el vèrtex superior del triangle. A mesura que s'apropa al vèrtex, la distància va disminuint. El punt mínim de la gràfica és representat pel vèrtex superior del triangle. En la segona part de la gràfica s'aprecia el salt que es produeix al tornar al centre. La recta va creixent fins arribar al punt màxim que és la base del triangle.

Degut a la perspectiva de la imatge no s'aconsegueixen triangles perfectes. Això comporta que el valor màxim no se situï a l'últim valor de la gràfica. Per això, s'adapta la gràfica per compensar les possibles diferències entre ideal i real.

La recta creixent de la gràfica, figura 3.15, representa l'anàlisi del centre cap al catet superior. En la segona part de la gràfica s'aprecia el salt que es produeix al tornar al centre. L'anàlisi del centre cap al vèrtex inferior és representat amb la recta decreixent. El punt mínim de la gràfica s'identifica amb el vèrtex inferior del triangle. Aquesta gràfica és un identificador únic d'aquest senyal, per això no es necessari la correlació creuada amb el patró del "cedeixi el pas".

En el cas que la gràfica obtinguda no correspongui a cap de les anteriors, com potser la gràfica generada pel soroll, es designa com a figura desconeguda.
- **ExtractRois.dll**: La seva tasca és obtenir els ROIs, a partir de la informació de la posició dels Objects, de la imatge original en escala de grisos o de la imatge filtrada. Així es demostra en la figura 3.16. A més a més, redimensiona el ROI extret a les mides dels patrons (30x30 píxels), com en la figura 3.17.

![Fig. 3.16 Imatge original i ROI obtingut](image)

![Fig. 3.17 ROI redimensionat a la mida del patró](image)

### 3.4 Reconeixement

#### 3.4.1 Introducció

Aquest bloc s'encarrega d'identificar a quin senyal pertany el ROI obtingut en el bloc de detecció.

Per determinar a quin patró s'assemblea més, es converteixen els bytes dels valors de la imatge en números enters (Image Data Conversion). Aquests es normalitzen (Image Normalization) i s'adapten (Image Adaptation) per poder realitzar la correlació creuada amb els patrons predefinits (Cross-Correlation). Quan s'obté el valor de la correlació creuada es decideix a quin senyal pertany (Decisor). Finalment, es retorna el codi del senyal de trànsit. El diagrama de la figura 3.18 reflecteix els passos esmentats.
### 3.4.2 Llibreries desenvolupades pel reconeixement

- **Normalization.dll**: Aquesta llibreria s'encarrega de normalitzar els patrons i els ROIs obtinguts en la part de detecció. Les imatges es normalitzen per poder adequar els valors màxims i mínims dels ROIs amb els valors del patró.

- **2D_2_1D.dll**: La llibreria externa del càlcul de la correlació creuada només treballa amb vectors d'una sola dimensió. Aquesta llibreria té la funció de convertir les dues dimensions de les imatges a una dimensió.

- **Patterns.dll**: És la llibreria que carrega i adapta els patrons en memòria per poder realitzar la correlació creuada. A més a més, també és funció seva adequar el ROI rebut. Després crida la llibreria externa i realitza els càlculs de correlació creuada per decidir a quin senyal s'assembla més. Un cop decidit retorna el codi del senyal.

### 3.4.3 Patrons predefinits

Els patrons predefinits es carreguen en memòria quan es posa en funcionament el servidor IPS. Són 40 patrons de 30 per 30 píxels agrupats en 4 grups en format blanc i negre. Aquests grups estan organitzats per color i forma del senyal. Això millora l'eficiència perquè no és necessari la correlació creuada amb tots els senyals, sinó que només amb els senyals que estan en el grup.

Els grups són:

- Rodones vermelles: Són els senyals amb orla vermella i fons blanc. Com mostra la figura 3.16.

![Fig. 3.16 Format dels patrons del grup de Rodones vermelles](image-url)
• Rodones blaves: Són els senyals amb fons blau i pictograma blanc. En la figura 3.17 mostra com són aquests patrons.

Fig. 3.17 Format dels patrons del grup de Rodones blaves

• Triangles: Són els senyals amb forma de triangle equilàter amb orla vermella. Com es mostra la figura 3.18.

Fig. 3.18 Format dels patrons del grup de Triangles

• Senyals particulars: Són els senyals que no corresponen als grups de senyals anteriors. La correlació creuada amb aquests patrons es realitza amb els ROIs de la imatge filtrada. Per aquest motiu els colors dels patrons estan invertits respecte als altres grups de senyals, com es mostra a la figura 3.19. Això permet obtenir una precisió més acurada en aquest tipus de senyals.

Fig. 3.19 Format dels patrons del grup Senyals particulars

3.4.4 Codificació dels senyals utilitzats pel reconeixement

La codificació utilitzada pel bloc de reconeixement és la mateixa que s'utilitza en l'aplicació mòbil TSDR. L'únic diferència és que la taula del reconeixement utilitza el codi NS per designar que el ROI no és un senyal.
<table>
<thead>
<tr>
<th>SIGN CODE</th>
<th>SIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS</td>
<td>No Sign</td>
</tr>
<tr>
<td>1</td>
<td>10 km/h</td>
</tr>
<tr>
<td>2</td>
<td>20 km/h</td>
</tr>
<tr>
<td>3</td>
<td>30 km/h</td>
</tr>
<tr>
<td>4</td>
<td>40 km/h</td>
</tr>
<tr>
<td>5</td>
<td>50 km/h</td>
</tr>
<tr>
<td>6</td>
<td>60 km/h</td>
</tr>
<tr>
<td>7</td>
<td>70 km/h</td>
</tr>
<tr>
<td>8</td>
<td>80 km/h</td>
</tr>
<tr>
<td>9</td>
<td>90 km/h</td>
</tr>
<tr>
<td>10</td>
<td>100 km/h</td>
</tr>
<tr>
<td>11</td>
<td>110 km/h</td>
</tr>
<tr>
<td>12</td>
<td>120 km/h</td>
</tr>
<tr>
<td>13</td>
<td>STOP</td>
</tr>
<tr>
<td>14</td>
<td>Cedeixi el pas</td>
</tr>
<tr>
<td>15</td>
<td>Intersecció amb prioritat</td>
</tr>
<tr>
<td>16</td>
<td>Semàfor</td>
</tr>
<tr>
<td>17</td>
<td>Intersecció amb circulació giratòria</td>
</tr>
<tr>
<td>18</td>
<td>Corba perillosa cap a la dreta</td>
</tr>
<tr>
<td>19</td>
<td>Corba perillosa cap a l'esquerra</td>
</tr>
<tr>
<td>20</td>
<td>Corbes perilloses cap a la dreta</td>
</tr>
<tr>
<td>21</td>
<td>Corbes perilloses cap a l'esquerra</td>
</tr>
<tr>
<td>22</td>
<td>Paviment lliscant</td>
</tr>
<tr>
<td>23</td>
<td>Vianants</td>
</tr>
<tr>
<td>24</td>
<td>Nens</td>
</tr>
<tr>
<td>25</td>
<td>Perill, doble sentit</td>
</tr>
<tr>
<td>26</td>
<td>Ressalt</td>
</tr>
<tr>
<td>27</td>
<td>Baden</td>
</tr>
<tr>
<td>28</td>
<td>Prioritat al sentit contrari</td>
</tr>
<tr>
<td>29</td>
<td>Circulació prohibida</td>
</tr>
<tr>
<td>30</td>
<td>Direcció prohibida</td>
</tr>
<tr>
<td>31</td>
<td>Prohibit avançar</td>
</tr>
<tr>
<td>32</td>
<td>Avançament prohibit per a camions</td>
</tr>
<tr>
<td>33</td>
<td>Prohibit Parar i Estacionar</td>
</tr>
<tr>
<td>34</td>
<td>Prohibit Estacionar</td>
</tr>
<tr>
<td>35</td>
<td>Sentit Obligatori a la dreta</td>
</tr>
<tr>
<td>36</td>
<td>Pas Obligat a la dreta</td>
</tr>
<tr>
<td>37</td>
<td>Sentit Obligatori a l'esquerra</td>
</tr>
<tr>
<td>38</td>
<td>Pas Obligat a l'esquerra</td>
</tr>
<tr>
<td>39</td>
<td>Enllumenat de curt abast</td>
</tr>
<tr>
<td>40</td>
<td>Intersecció de sentit giratori obligatori</td>
</tr>
</tbody>
</table>
3.5 Exemples

Aquests exemples expliquen els passos que aplica el servidor IPS per detectar i reconèixer els possibles senyals de trànsit que pot haver en les imatges. La mitjana de temps emprat pel processament d'una imatge és menor de 1000 ms amb un processador Intel i5 2430M a 2.4 GHz. Pel càlcul del temps mitjà s'ha tingut en compte imatges amb diferents situacions.

3.5.1 Escenari 1: Fotografia d'un carrer amb tres senyals detectables

El següent exemple demostra com funciona el servidor IPS amb senyals dels grups cercles blaus, cercles vermells i triangles vermells.

![Fig. 3.20 Imatge original a l'entrada del IPS](image)

En la imatge de la figura 3.20 s'observa que hi ha 3 possibles senyals a obtenir pel servidor IPS.

**Primer pas:** Filtrar la imatge pel filtre vermel·lu on s'obté la figura 3.21.
La imatge resultant mostra en blanc tots els píxels que es troben dins dels llindars establerts pel filtre. Tot i que els senyals de la imatge original no són de color vermell pur, el filtre és capaç de captar-los.

A continuació es traspassa la imatge filtrada al bloc de detecció. En aquest bloc es neteja la imatge filtrada i es marquen els diferents Objects. Aquests Objects es poden identificar perquè cadascú té assignat un color diferent, figura 3.22.

S'analitza cada Object per separat en busca de possibles formes geomètriques reconegudes pel sistema. Per analitzar el Object, s'aplica els algoritmes explicats en la llibreria ShapeDetection.
Fig. 3.22 Representació gràfica d'un Object sense forma geomètrica

Fig. 3.23 Representació gràfica d'un Object amb forma geomètrica desconeguda

Aquests Objects són detectats com a forma desconeguda, això significa que es passen a comparar amb el grup de patrons particulars. El centre dels Objects de la figura 3.22 no conté part analitzable, per això és una constant. El centre del Object de la figura 3.23 al tenir espais per escanejar, dona com a resultat una gràfica no predefinida.

Fig. 3.24 Representació gràfica d'un Object amb forma de triangle
En aquests dos últims Objects, figures 3.24 i 3.25, s'observa que les gràfiques s'ajusten a les gràfiques ideals. Per aquest motiu, passen a ser reconegudes al grup de triangles i cercles vermells respectivament.

Tot seguit, a partir de la informació dels Objects i en funció del grup, s'extreu els ROIs de la imatge original en escala de grisos o en blanc i negre. Es redimensionen a 30x30 píxels i es passen al bloc del reconeixement.

**Taula 3.2** Resultats del reconeixement dels Objects obtinguts

<table>
<thead>
<tr>
<th>ROIs d'entrada</th>
<th>Codi sortida</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="ROIs d'entrada" /></td>
<td>NS</td>
</tr>
<tr>
<td><img src="image2.png" alt="ROIs d'entrada" /></td>
<td>NS</td>
</tr>
<tr>
<td><img src="image3.png" alt="ROIs d'entrada" /></td>
<td>03</td>
</tr>
<tr>
<td><img src="image4.png" alt="ROIs d'entrada" /></td>
<td>25</td>
</tr>
<tr>
<td><img src="image5.png" alt="ROIs d'entrada" /></td>
<td>NS</td>
</tr>
<tr>
<td><img src="image6.png" alt="ROIs d'entrada" /></td>
<td>NS</td>
</tr>
<tr>
<td><img src="image7.png" alt="ROIs d'entrada" /></td>
<td>NS</td>
</tr>
</tbody>
</table>

Com a resultat, en la taula 3.2, es pot comprovar que els codis obtinguts pel IPS es corresponen amb la codificació dels senyals de trànsit de l'aplicació mòbil, taula 3.1.
Segon pas: Filtrar la imatge pel filtre blau on s'obté la figura 3.26.

La imatge resultant mostra en blanc tots els píxels que es troben dins dels límits establerts pel filtre. Tot i que els senyals en la imatge original no són de color blau pur, el filtre és capaç de captar-los.

A continuació es traspassa la imatge filtrada al bloc de detecció. En aquest bloc es neteja la imatge filtrada i es marquen els diferents Object. Aquests Object es poden identificar perquè cadascú té un color diferent com mostra la figura 3.27.

Els senyals blaus que el servidor reconeix són sempre rodones. Per aquest motiu no és necessari analitzar la forma geomètrica dels Object obtinguts, figura 3.28.
Tot seguit, a partir de la informació dels Objects, s'extreu els ROIs de la imatge original en escala de grisos i es redimensionen a 30x30 píxels. Es passen al bloc del reconeixement.

**Taula 3.3** Resultats del reconeixement dels Objects obtinguts

<table>
<thead>
<tr>
<th>ROIs d'entrada</th>
<th>Codi sortida</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>NS</td>
</tr>
</tbody>
</table>

Com a resultat, en la taula 3.3, es pot comprovar que els codis obtinguts pel IPS es corresponen amb la codificació dels senyals de trànsit de l'aplicació mòbil, taula 3.1.

Aquest anàlisi s'ha dut a terme en un temps inferior a 1500 ms. amb un processador Intel i5 2430M a 2.4 GHz.

### 3.5.2 Escenari 2: Fotografia de baixa qualitat amb un senyal detectable.

El següent exemple demostra com funciona el servidor IPS amb senyals del grup de senyals particulars.
En la imatge de la figura 3.29 s'observa que hi ha un possible senyal a obtenir pel servidor IPS.

**Primer pas:** Filtrar la imatge pel filtre vermell on s'obté la figura 3.30.

La imatge resultant mostra en blanc tots els píxels que es troben dins dels llindars establerts pel filtre. Tot i que el senyal en la imatge original no és de color vermell pur, el filtre és capaç de captar-ho.

A continuació es traspassa la imatge filtrada al bloc de detecció. En aquest bloc es neteja la imatge filtrada i es marquen els diferents Objects. Aquests Objects
es poden identificar perquè cadascú té un color diferent com es mostra en la figura 3.31.

![Fig. 3.31 Objects identificats](image)

S'analitza el Object en busca de possibles formes geomètriques reconegudes pel sistema. Per analitzar el Object, s'aplica els algoritmes explicats en la llibreria ShapeDetection.

![Fig. 3.32 Representació gràfica d'un Object amb forma geomètrica desconeguda](image)

Com que el resultat del Object no s'ajusta a cap gràfica predefinida, figura 3.32, s'identifica com a grup de senyals particulars.

A diferència dels altres grups de senyals s'extreu el ROI de la imatge filtrada, es redimensiona a 30x30 píxels i es passa al bloc del reconeixement.
Taula 3.4 Resultats del reconeixement dels Objects obtinguts

<table>
<thead>
<tr>
<th>ROIs d'entrada</th>
<th>Codi sortida</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>

Com a resultat, en la taula 3.4, es pot comprovar que el codi obtingut pel IPS es correspon amb la codificació dels senyals de trànsit de l'aplicació mòbil, taula 3.1.

Segon pas: Filtrar la imatge pel filtre blau. Aquest pas no es considera oportú representar-lo perquè la imatge a la sortida del filtre és completament negre. No hi ha res a detectar.

Aquest anàlisi s'ha dut a terme en un temps inferior a 1000 ms. amb un processador Intel i5 2430M a 2.4 GHz.

### 3.6 Resum del capítol

En aquest capítol s'ha vist que el processat d'imatge és un bloc important del sistema TSDR perquè és l'encarregat de mantenir actualitzada la base de dades. S'ha demostrat que és capaç de filtrar una imatge deixant passar el conjunt de colors que tenen els senyals. A partir de la imatge filtrada, el bloc de detecció té l'habilitat de trobar les regions on es troben els senyals. Aquestes regions són analitzades mitjançant un algoritme propi per decidir a quin grup de senyals pertany. Arribat a aquest punt, s'extreu la regió d'interés i es traspassa al bloc de reconeixement. El bloc reconeixement identifica el codi del senyal mitjançant la correlació creuada amb patrons.
CAPÍTOL 4. CONNEXIONS ENTRE SERVIDORS

4.1 Introducció

Pel bon funcionament del servidor IPS és necessari que estigui en comunicació directa amb els servidors FTP i base de dades. El primer alimenta el servidor IPS amb les fotografies i el segon s'actualitza amb les informacions dels senyals que aporta el servidor IPS, figura 4.1.

![Fig. 4.1 Connexions entre servidors](image)

4.2 Configuració del servidor

El servidor permet configurar, figura 4.2, la carpeta d'entrada de les fotografies (FTP Folder) i el directori de treball del servidor IPS (IPS Folder). També obliga a omplir les dades necessàries per poder accedir a la base de dades com són: la IP, el port, el SID, l'usuari i la contrasenya de la taula de la base de dades.

![Fig. 4.2 Configuració del servidor IPS](image)
En el momento de posar en marxa el servidor, l'aplicació comprova que no hi ha cap error en els camps. Un cop en marxa la configuració no es pot modificar. Per canviar-la s'ha de reiniciar el servidor.

### 4.3 Connexió entre servidor FTP i servidor IPS

El servidor IPS s'alimenta de les fotografies enviades per l'aplicació mòbil. És per això que es connecta directament amb el servidor FTP. El servidor IPS es pot configurar de dues maneres per accedir a les fotografies.

Opció 1: Sense el sistema de backup, totes les imatges són emmagatzemades en un directori dins el servidor FTP. El servidor IPS accedeix a aquest directori i mou la imatge més antiga al seu directori de treball. Per tant, cada imatge moguda per processar, desapareix del FTP deixant espai en disc per una altra imatge.

Opció 2: Quan el sistema de backup està activat, les imatges són emmagatzemades directament en un directori comú i compartit dins del servidor IPS. En aquest cas, el servidor FTP no utilitza el seu disc per guardar les imatges.

La informació que recull el servidor IPS és la fotografia i el fitxer de dades enviats pel telèfon mòbil, figura 4.3. El fitxer de dades aporta la informació de la latitud, la longitud, el rumb i la velocitat. Tota aquesta informació és captada en el moment que es realitza la fotografia.

Fig. 4.3 Connexió servidor FTP- servidor IPS

#### 4.3.1 Gestió de fitxers

Per gestionar quina imatge es processa abans que una altra, s'ha desenvolupat la llibreria FTP2Filter.dll. Aquesta s'encarrega de moure la imatge a processar del directori configurat al FTP Folder cap el directori de treball del IPS Folder, aquests directoris són seleccionats a la figura 4.2. Selecciona la imatge més antiga segons la data i l'hora, tot això de manera automatitzada, figura 4.4.
Quan el servidor finalitza de processar la imatge, l'esborra. Aleshores, automàticament mou una altra imatge per processar. Si no hi ha cap, es queda en espera fins que arriba alguna.

En el directori de treball del IPS s'emmagatzema la imatge que s'està processant. D'aquesta manera, si hi ha alguna incidència amb el servidor, la imatge es pot recuperar.

**Fig. 4.4** Diagrama de flux de gestió de fitxers

### 4.4 Connexió servidor IPS a servidor Base de Dades

Quan es reconeixen els senyals s'ha d'actualitzar la base de dades. La informació que s'emmagatzema és el codi resultant del reconeixement més les dades que acompanya la fotografia, com es mostra en la figura 4.5. Tota aquesta informació s'escriu en la taula Buffer de la base de dades.

Per poder inserir la informació a la base de dades s'ha desenvolupat la llibreria Sign2DB.dll. Aquesta s'encarrega d'interpretar el fitxer de dades de cada fotografia. D'aquí extreu la latitud, la longitud, el rumb i la velocitat. Un cop obtinguda tota la informació, la insereix a la base de dades.
4.5 Resum del capítol

En aquest capítol s'ha vist com el servidor IPS és un sistema autònom capaç d'adquirir imatges del directori configurat. Quan acaba de processar una imatge automàticament actualitza la base de dades amb la informació pertinent.
CAPÍTOL 5. CONCLUSIONS

Quan vam començar aquesta aplicació vam establir uns objectius mínims. D'aquests objectius tots s'han complert. A més a més, a mesura que avançàvem en el projecte, sorgien noves idees de millora que també s'han implementat amb èxit.

El inici del projecte va estar enfocat en el sistema Android. Al desconèixer per complet aquest món, es va haver d'aprendre des de zero el que comporta programar una aplicació amb el llenguatge Java per Android. Això va suposar dedicar més temps del previst per familiaritzar-se amb l'Eclipse i la SDK d'Android.

A mesura que s'avançava en l'aplicació TSDR, anaven sorgint dificultats. L'entrebanc que més temps ha necessitat ha estat la part de la interfície gràfica. Aquesta combinava mapes, imatges, missatges i la càmera.

Al finalitzar l'aplicació, es va comprovar que totes les característiques implementades funcionaven.

Després de l'aplicació d'Android, es va dedicar a desenvolupar els diferents servidors. Primer de tot, es va començar amb una base de dades molt poc evolucionada. Es va detectar que per fer una recerca quan hi hagués un gran volum de dades i usuaris, augmentava de manera considerable el temps de resposta del servidor. Aleshores, es va redissenyar la base de dades molt més estructurada. Es va dividir Catalunya en zones i es va implementar un índex d'àrees per accelerar les recerques dels senyals. Amb tot això s'optimitzà el temps de resposta.

Durant l'aprenentatge de com crear la base de dades ens vam adonar que no era trivial. S'havia d'utilitzar una eina proporcionada per Oracle. Per això es va desenvolupar un software que permet a l'administrador gestionar la base dades sense coneixements de SQL.

A la vegada s'anava implementant el servidor proxy, que permet la comunicació del telèfon mòbil amb els diferents servidors. Amb el desenvolupament del servidor vam haver d'aprendre escriure i llegir fitxers XML, xifrar dades i crear sockets asíncrons.

Després de desenvolupar el proxy i la base de dades, es va fer una prova que comunicava el telèfon mòbil amb el servidor de base dades a través del proxy. Aquesta comprovació va ser la més important perquè vam veure com el mòbil enviava una petició de senyal i com es rebia la resposta. L'aplicació interpretava la resposta transformant en imatge i so els codis rebuts.

Una prova molt satisfactoria va ser la connexió entre el telèfon mòbil i el servidor FTP. El telèfon mòbil era capaç d'enviar la fotografia amb el seu fitxer d'informació al rebre la resposta del proxy.
L’últim bloc del projecte l’hem dedicat exclusivament al servidor de processat d’imatges (IPS). El més important que hem aprés és que per analitzar els colors d’una imatge és molt útil treballar amb el mode de color HSV. Ha sigut fonamental conèixer com funciona el sistema d’etiquetatge (labelings) d’una imatge binària. Un dels problemes trobats era saber quina forma geomètrica tenia el possible senyal. Per resoldre’l vam dissenyar un algoritme propi que permet diferenciar rodones, triangles i triangles invertits.

També hem aprés que hi ha altres sistemes com el Neural Network que permet detectar i reconèixer en temps real. Degut a que és necessari emprar molt de temps per la creació i l’entrenament d’una Neural Network, no es va considerar viable en el projecte. Per això, es va decidir utilitzar en el reconeixement el mètode de la correlació creuada. Tot i no ser el mètode més ràpid, té una alta fiabilitat.

Aquest projecte ha estat un repte per nosaltres perquè a partir d’una problemàtica, hem estat capaços d’idear, dissenyar i programar una solució innovadora. La solució del projecte queda reflectida en la figura 6.1.

**Fig. 5.1** Exemple de vehicle amb TSDR i sense TSDR
El nostre objectiu és intentar reduir el nombre d'accidents mitjançant una cultura preventiva. Com també el de promoure la filosofia de compartir la informació entre tots els usuaris per el bé de tots.

La realització d'aquest projecte no hagués estat possible sense el treball en equip.

### 5.1 Ambientalització

Com que aquest projecte es tracta d'un prototip, el seu impacte ambiental és mínim.

La fase de concepció, s'ha realitzat digitalment. Això només implica un reduït consum elèctric produït pels ordinadors.

La fase de construcció com que s'ha dissenyat i programat, el impacte és el mateix que el de la fase de concepció. El disseny del projecte es pot modular i testejar per separat per trobar problemes més ràpidament en cas d’averia. En cap moment es perjudicarà el medi ambient durant la seva depuració.

En la fase d'explotació hem de tenir en compte el consum de l'aplicació i dels servidors. En el cas de l'aplicació, les pantalles són les més fosques possibles per reduir el consum en la il·luminació de la pantalla. El fet d'utilitzar màquines virtuals pels diferents servidors, reduceix el maquinari i el seu consum.

L’ús de l’aplicació genera una cultura vial a mig termini que es veurà reflectida en les emissió de CO₂.

L’ús de mapes i navegadors digitals reduceixen el consum de paper.

En el cas d’implementar en producció els servidors, s'hauria de tenir en compte que haurien d'estar encesos 24 hores al dia, 7 dies a la setmana i 365 dies l'any durant la seva vida útil.

L’última fase, la del desmantellament, el software no tindrà cap impacte ambiental. El maquinari utilitzat són els propis ordinadors i no s'ha comprat específicament pel projecte.

En el cas d’implementar en producció els servidors, s'hauria de tenir en compte les possibles averies o actualitzacions del hardware pel bon funcionament del sistema. Es podran reutilitzar components, o fins i tot, els mòduls sencers.

### 5.2 Línies Futures

Donada a la limitació de temps de la realització d’aquest projecte, han quedat obertes diferents vies de millora per futures propostes. Entre elles destaquem les més importants:
• Multi-idioma: Una opció de selecció d'idioma per adaptar-lo a les necessitats de cada usuari.

• Càlcul de rutes: L'aplicació podria oferir a l'usuari calcular una ruta i així permetre una millor eficiència en la recerca de senyals.

• Senyals: Ampliar el número de senyals de trànsit a reconèixer.

• Altres possibles alertes: Podrien ser de retencions, obres, accident, condicions meteorològiques...

• Xifrat de les dades entre servidor i mòbil: Així les dades enviades serien més segures de cara una possible interceptació de tercers.

• Projecció de la senyals rebudes en el parabrisa.

• Pantalla Mode Nocturn: Adequar la il·luminació de la pantalla a l'elecció de l'usuari.

• NFS (Network File System): Utilitzar un servidor de fitxers de xarxes per emmagatzemar les fotografies dels usuaris.

• Versions Premium: Versió de pagament que estalviar bytes de la tarifa de dades perquè no fa fotografies i no s'envien.

• Adaptació de l'aplicació: Adapta-la a altres dispositius com per exemple a navegadors GPS. Si es crea una versió per GPS, en el moment de calcular la ruta, descarregar-se les dades dels senyals que es trobarà.

BIBLIOGRAFIA

• Llibres:

Bradski, Gary i Kaebler, Adrian, Learning OpenCV Computer Vision with the OpenCV Library O'REILLY, USA, 1ª Edició, Setembre (2008).

Gargenta, Marko, Learning Android O'REILLY, USA, 1ª Edició, Setembre (2008).


• Material Docent:


- **Referències**


[18] Web Google Driving Directions URL: http://www.anddev.org/google_driving_directions_-_mapview_overlaid_t826.html


[22] Web Emgu Files
URL: http://www.emgu.com/wiki/files/2.2.1/document/Index.html

[23] Web Oracle Database XE Introduction
URL: http://www.getyourcontent.com/1/6565-0/Oracle-Database-XE----SQL.aspx

[24] Web VirtualBox
URL: https://www.virtualbox.org/

[25] Web XmlWriter Tutorial
URL: http://www.dotnetperls.com/xmlwriter

[26] Web Cryptography Microsoft
URL: http://support.microsoft.com/kb/307010

[27] Web System Security Cryptography

[28] Web Traffic Sign Detection
URL: https://sites.google.com/site/trafficsigndetection/home

[29] Web AFForge Library
URL: http://www.aforgenet.com/


[31] Web Xlightftpd
URL: http://www.xlightftpd.com/

[32] Web HTC Drivers
URL: http://driver-pro.com/related/english/hardware-drivers/?a=12226&hit=1

[33] Web HTC especificaciones
URL: http://www.htc.com/es/smartphones/htc-desire-hd/#specs

[34] Web OS Mobile
URL: http://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil

[35] Comparativa Sistemes Mòbils
URL: http://es.engadget.com/2009/03/19/la-gran-comparacion-de-los-sistemas-operativos-moviles/

[36] Web MySQL community
URL: http://dev.mysql.com/doc/refman/5.0/es/features.html
[37] Web Oracle XE

[38] Web Connected-Component
URL: http://en.wikipedia.org/wiki/Connected-component_labeling