




Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Volum I

Memòria – Pressupost- Annex

TREBALL DE FI DE GRAU



“Sistema automàtic de detecció de lesions en imatges endoscòpiques”

TFG presentat per obtenir el títol de GRAU en
ENGINYERIA BIOMÈDICA
Per **Ferran Noya Urgell**

Barcelona, 09 de Juny de 2015

Director: Raúl Benítez Iglesias
Departament de ESAII
Universitat Politècnica de Catalunya (UPC)

ÍNDEX GENERAL

Índex memòria.....	9
Resum	12
Resumen .	12
Abstract ..	13
Agraïments.....	14
CAPÍTOL 1: Introducció	15
1.2 Angiodisplàsia	17
1.2.1 Com es diagnostica?.....	18
1.2.2 Tipus d'angiodisplàsies	19
1.3 Càpsula endoscòpica	21
1.3.1 Característiques tècniques de la càpsula endoscòpica	21
1.3.2 Procediment d'exploració	23
1.3.3 Temps de diagnòstic.....	24
CAPÍTOL 2: Objectiu	25
2.1 Antecedents.....	27
2.2 Objectiu general	27
2.3 Objectius específics.....	27
CAPÍTOL 3: Metodologia	29

3.1	Preprocessat i obtenció de les imatges	31
3.1.1	Selecció dels estudis.....	32
3.1.2	Processat del vídeo	33
3.1.3	Obtenció dels fotogrames.....	35
3.2	Caracterització de les lesions	37
3.2.1	Binarització de les lesions fosques.....	38
3.2.2	Binarització de les lesions clares	49
3.2.3	Segmentació	54
3.2.4	Estudi de propietats i establiment de llindars	63
3.2.5	Classificador estadístic.....	72
3.3	Creació del vídeo	77
3.3.1	Integració al algorisme de detecció	77
3.3.2	Producció del vídeo	78
3.4	Estructura de l'algorisme final	79
3.4.1	Gestió dels fitxers	79
3.4.2	Gestió de les imatges	80
3.4.3	Integració algorisme de detecció de lesions clares i algorisme de detecció de lesions fosques.	84
CAPÍTOL 4: Resultats		87
4.1	Factors de discriminació	89
4.2	Comparació algorisme desenvolupat amb software comercial	92
CAPÍTOL 5: Conclusions		95

CAPÍTOL 6: Bibliografia	99
6.1 Referències bibliogràfiques	101
6.2 Bibliografia de consulta	101
Índex pressupost.....	105
CAPÍTOL 7: Pressupost.....	107
7.1 Desglossament de costos.....	109
7.1.1 Hores d'enginyeria	109
7.1.2 Estació de treball	109
7.1.3 Softwares utilitzats.....	109
7.2 Pressupost Total	110
Índex annex	115
CAPÍTOL 8: Annex	117
8.1 Algorisme principal	119
8.2 Algorisme per extraure fotogrames	126
8.3 Algorisme detector de lesions fosques	127
8.4 Algorisme detector de lesions clares.....	130
8.5 Algorisme semiautomàtic detector de lindars.....	133
8.6 Algorisme automàtic detecció i càlcul de lindars lesions fosques.	134


8.7	Algorisme automàtic detecció i càlcul de llindars lesions clares ...	139
8.8	Algorisme d'entrenament manual pel classificador estadístic	144
8.9	Algorisme d'entrenament per lesions fosques pel classificador estadístic	146
8.10	Algorisme d'entrenament per lesions clares pel classificador estadístic	149



Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Memòria

A faded, light-colored image of a large, classical building with a central tower and many windows, serving as a background for the title.

“Sistema automàtic de detecció de lesions en imatges endoscòpiques”

TFG presentat per optar al títol de GRAU en
ENGINYERIA BIOMÈDICA
per **Ferran Noya Urgell**

Barcelona, 09 de Juny de 2015

Director: Raúl Benítez Iglesias
Departament ESAII
Universitat Politècnica de Catalunya (UPC)

ÍNDEX MEMÒRIA

Resum	12
Resumen	12
Abstract ..	13
Agraïments	14
CAPÍTOL 1: Introducció	15
1.2 Angiodisplàsia	17
1.2.1 Com es diagnostica?.....	18
1.2.2 Tipus d'angiodisplàsies	19
1.3 Càpsula endoscòpica	21
1.3.1 Característiques tècniques de la càpsula endoscòpica	21
1.3.2 Procediment d'exploració	23
1.3.3 Temps de diagnòstic.....	24
CAPÍTOL 2: Objectiu	25
2.1 Antecedents.....	27
2.2 Objectiu general	27
2.3 Objectius específics.....	27
CAPÍTOL 3: Metodologia	29

3.1	Preprocessat i obtenció de les imatges	31
3.1.1	Selecció dels estudis.....	32
3.1.2	Processat del vídeo	33
3.1.3	Obtenció dels fotogrames.....	35
3.2	Caracterització de les lesions	37
3.2.1	Binarització de les lesions fosques.....	38
3.2.2	Binarització de les lesions clares	49
3.2.3	Segmentació	54
3.2.4	Estudi de propietats i establiment de llindars	63
3.2.5	Classificador estadístic.....	72
3.3	Creació del vídeo	77
3.3.1	Integració al algorisme de detecció	77
3.3.2	Producció del vídeo	78
3.4	Estructura de l'algorisme final	79
3.4.1	Gestió dels fitxers	79
3.4.2	Gestió de les imatges	80
3.4.3	Integració algorisme de detecció de lesions clares i algorisme de detecció de lesions fosques.	84
CAPÍTOL 4: Resultats		87
4.1	Factors de discriminació	89
4.2	Comparació algorisme desenvolupat amb software comercial	92
CAPÍTOL 5: Conclusions		95

CAPÍTOL 6: Bibliografia	99
6.1 Referències bibliogràfiques	101
6.2 Bibliografia de consulta	101

RESUM

El projecte tracta sobre el desenvolupament d'un algorisme mitjançant MATLAB, pel diagnòstic de lesions provocades per angiodisplàsies en l'intestí prim, utilitzant imatges extretes d'un estudi realitzat per una càpsula endoscòpica.

Utilitzant tècniques de tractament d'imatge, es genera un algorisme capaç de segmentar les regions d'interès d'una imatge endoscòpica i classificar-les, en lesió o no lesió.

Per la segmentació, s'utilitzen tècniques com la binarització i transformada watershed, per discriminar de la imatge d'endoscòpia les regions d'interès.

Per la classificació de les regions discriminades, s'ha realitzat un estudi de les característiques més regulars que apareixen en les lesions, pel disseny de filtres singulars per aquest tipus de lesió, juntament amb la implementació d'un classificador estadístic tipus Bayesià.

En acabar s'obté un algorisme que supera, en la majoria de prestacions, a softwares contemporanis que tenen objectius similars.

RESUMEN

El proyecto trata sobre el desarrollo de un algoritmo mediante MATLAB, para el diagnostico de lesiones dadas por angiodisplásias en el intestino delgado, utilizando imágenes extraídas de un estudio realizado por una cápsula endoscópica.

Utilizando técnicas de tratamiento de imagen, se genera un algoritmo capaz de segmentar las regiones de interés en una imagen endoscópica y clasificarlas, en lesión o no lesión.

Para la segmentación, se utilizan técnicas como la binarización y la transformada watershed, para discriminar de la imagen endoscópica las regiones de interés.

Para la clasificación de las regiones discriminadas, se ha realizado un estudio de las características más regulares que aparecen en las lesiones, para el diseño de filtros singulares para este tipo de lesión, juntamente con la implementación de un clasificador estadístico de tipo Bayesiano.

Al terminar, se obtiene un algoritmo que supera, en la mayoría de prestaciones, a softwares contemporáneos que tiene objetivos similares.

ABSTRACT

The scope of the project is to develop a MATLAB algorithm in order to diagnose angiodysplasias injuries in the small intestine. This project is carried out using images taken by an endoscopic capsule.

Using image processing techniques it has been possible to create an algorithm able to segment regions of interest of an endoscopic image and classify in injury or not.

Binarization and watershed transform techniques are used to lead to success segmentation, discriminating endoscopic image regions of interest.

For the classification of the discriminated regions, it has been developed a study of the most regular features in injuries. Hereafter it has been assigned a singular filter for this type of injury, together with the implementation of a statistical Bayesian classifier.

At the end, an algorithm that exceeds in most features the contemporary softwares that have similar goals is reached.

AGRAÏMENTS

Voldria donar gràcies al meu tutor en Raúl per l'ajuda i la guia que m'ha donat durant el desenvolupament del projecte, al doctor Marco Antonio per les hores que ha invertit facilitant-me l'obtenció de les imatges i els vídeos, als meus companys del laboratori Àlex, Xavier i Carme, per l'ajuda que m'han prestat durant el temps que he estat fent el treball amb ells i a la Martina per la seva paciència inesgotable.

CAPÍTOL 1: INTRODUCCIÓ

1.2 Angiodisplàsia

La angiodisplàsia és la malformació vascular més comuna del tracte digestiu. Es tracta d'un cúmul anòmal de vasos sanguinis dilatats, amb la paret aprimada i fràgil, que poden arribar a col·lapsar i comportar un sagnat al tracte digestiu. Es poden localitzar en múltiples zones del tracte digestiu (estomac, intestí prim, intestí gruixut). El 40-60% dels pacients tenen més d'una angiodisplàsia, típicament agrupades a la mateixa localització i un 20% d'aquests poden tenir simultàniament angiodisplàsies en altres punts del tracte digestiu.

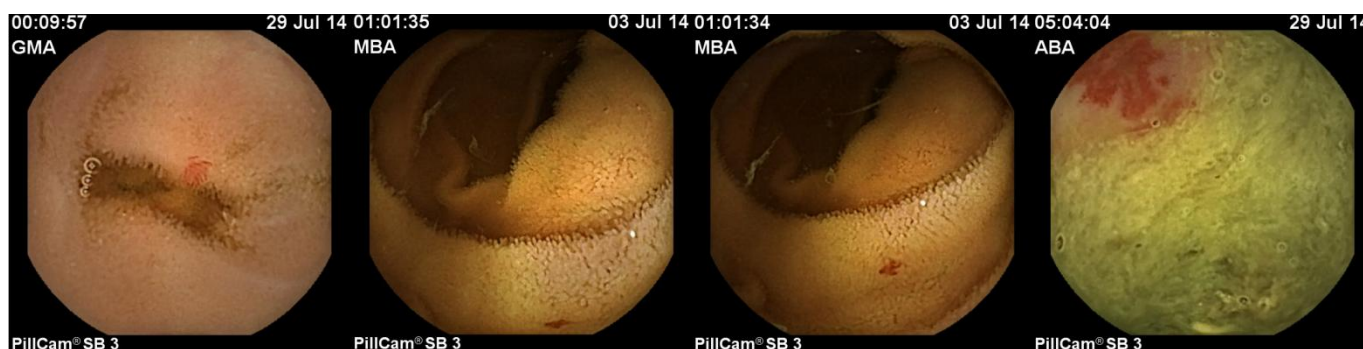


Figura 1: Exemple d'angiodisplàsies obtingudes mitjançant la càpsula endoscòpica

L'origen de les lesions no està clar. Se sol detectar en persones majors de 60 anys i estan associades principalment a malalties de la sang (ex. malaltia de von Willebrand¹), malalties de les vàlvules del cor (estenosis² aòrtica), insuficiència renal crònica terminal i, en menor mesura, cirrosi hepàtica o les malalties respiratòries greus. Es creu que estan produïdes per petites obstruccions del flux sanguini que origina una creació compensatòria anòmala de vasos dilatats i fràgils.

Els símptomes venen derivats exclusivament de l'hemorràgia i la magnitud en l'interior del tub digestiu. Quan la pèrdua de sang és escassa però mantinguda en el temps, apareix anèmia sense evidència de sagnat, també es pot detectar mitjançant proves de traça de sang en la femta del pacient. Si l'hemorràgia és abundant, a més de l'anèmia, és possible l'aparició de vòmits o bé deposicions de coloració negra o vermellosa, que en certs casos cedeixen sense tractament, però amb tendència a tornar aparèixer amb la repetició del sagnat.

És important destacar que la gravetat i la recurrència de l'hemorràgia està influenciada en molts pacients per la ingesta de medicaments que afavoreixen el sagnat, com els anticoagulants (Molina Infante y Mateos Rodríguez 2010).

¹ Malaltia de von Willebrand: Anomalia en la coagulació de la sang més comuna en els humans.

² Estenosis: Terme mèdic utilitzat per denotar constricció o aprimament d'un orifici o un conducte corporal.

1.2.1 Com es diagnostica?

Pels pacients amb hemorràgia digestiva de repetició o amb anèmia d'origen desconegut, l'endoscòpia és la prova clau per la detecció de les angiodisplàsies. Es realitzarà una endoscòpia alta, a través de la boca, per visualitzar l'estomac i el principi de l'intestí prim i/o una colonoscòpia, a través de l'anus, per avaluar l'intestí gros i els últims centímetres de l'intestí prim. En un 5% de les vegades no es troben lesions sagnants en cap d'aquestes endoscòpies, aquest fet implica una necessària exploració de l'intestí prim, un segment del tracte digestiu de 5-7 metres de longitud localitzat entre l'estomac i el còlon. La prova a realitzar en aquests casos és la càpsula endoscòpica, que consisteix en la ingesta d'una càpsula que realitza múltiples fotografies durant 8-10 hores, per l'avaluació completa de l'intestí prim. Durant l'última dècada, s'han desenvolupat diversos endoscòpis que permeten en el 75-80% dels casos l'exploració completa de l'intestí prim, tanmateix també existeixen càpsules endoscòpiques capaces de realitzar un estudi d'imatge tant del còlon com de la part alta del tracte digestiu. Estudis recents afirmen que entre el 60-70% de les hemorràgies diagnosticades en l'intestí prim són degudes a angiodisplàsies.

Quan tot i efectuar les exploracions no es diagnostica la causa del sagnat, són necessàries exploracions més invasives, com l'arteriografia o l'endoscòpia intraoperatòria (Molina Infante y Mateos Rodríguez 2010).

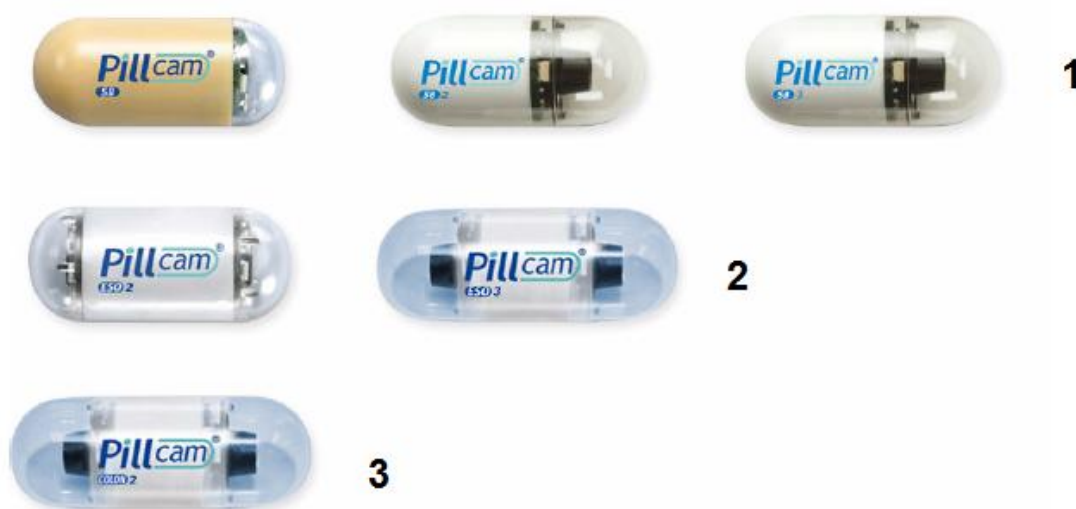
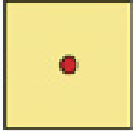
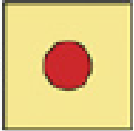

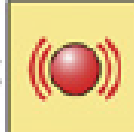




Figura 2: 1 - tres generacions de càpsula endoscòpica per l'estudi de l'intestí prim, 2 - dues generacions de càpsula endoscòpica per l'estudi de la part alta del tracte digestiu, 3 - càpsula endoscòpica per l'estudi del còlon.

1.2.2 Tipus d'angiodisplàsies

No existeixen postulats definitius sobre els diferents tipus de lesions, tot i així s'accepta com a bona la classificació proposada per un grup de metges japonesos, anomenada classificació Yano-Yamamoto.

Taula 1: Classificació Yano-Yamamoto

Classificació d'angiodisplàsies Yano-Yamamoto		
Tipus 1a		Lesió puntual (menys que 1 mm) amb supuració o sense.
Tipus 1b		Lesió clapa (més que 1 mm) amb supuració o sense.
Tipus 2a		Lesió puntual (menys que 1 mm) amb sagnat pulsatiu.
Tipus 2b		Lesió vermella sortint pulsativa sense dilatació de les venes perifèriques.
Tipus 3		Lesió vermella sortint pulsativa amb dilatació de les venes perifèriques.
Tipus 4		Altres lesions que no es poden classificar en cap dels tipus anteriors.

La classificació que es pot observar en la taula 1, està basada en un estudi on es proposava a diferents especialistes que classifiquessin una sèrie de lesions en els tipus que es detallen a la taula. Els resultats van ser que en un 82% dels casos els especialistes classificaven les lesions de la mateixa manera i en el restant 18% només un especialista va classificar de manera diferent les lesions (Yano, et al. 2008).



Figura 4: Lesió tipus 1a amb supuració.

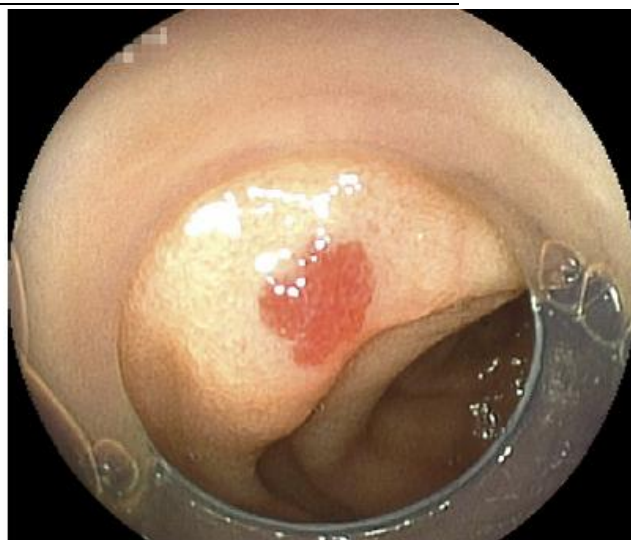


Figura 3: Lesió tipus 1b sense sagnat.



Figura 6 Lesió tipus 2a amb sagnat pulsatiu.



Figura 5: Lesió tipus 2b amb un trombe a la punta i sense dilatació de les venes perifèriques.



Figura 7: Lesió tipus 3 amb dilatació de les venes perifèriques.

1.3 Càpsula endoscòpica

La càpsula endoscòpica (CE) és l'última novetat en endoscòpia gastrointestinal i la primera tècnica capaç d'explorar tot l'intestí prim. Recentment s'han introduït nous avanços en aquest camp que inclouen la càpsula d'esòfag (Pillcam ESO) i la càpsula de còlon (Pillcam COLON), amb variacions de la càpsula inicial i que ens permet identificar lesions també en aquests trams del tub digestiu.

En la dècada dels anys 1980 es van iniciar els primers estudis amb CE (Gabriel Iddan, Israel 1981). Les primeres imatges d'estudis en animals van ser presentades pel Dr. Swain a San Diego (Digestive Disease Week 2000). A partir d'aleshores, la càpsula s'ha convertit en una de les eines d'imatge en l'estudi de l'intestí prim (González Suárez, et al. 2011).

1.3.1 Característiques tècniques de la càpsula endoscòpica

Given Imaging Ltd (Yoqneam, Israel) va ser la primera entitat encarregada de la fabricació d'aquest nou dispositiu (Vídeo-càpsula Given M2A), aprovada inicialment per la Food and Drug Administration (FDA) dels EUA per l'avaluació de trastorns a l'intestí prim el 2001. Actualment és la càpsula més utilitzada, encara que existeixen altres equips disponibles d'altres cases comercials com la endocàpsula de Olympus i la MiroCam (Intromedic).

Taula 2: Característiques dels diferents sistemes de càpsula endoscòpica més coneguts a escala mundial

Càpsula	MicroCam®	PillCam SB2®	OMOM®	Endocapsule®
Fabricant	Intromedic	Given Imaging	Jianshan Science	Olympus
Mida	10.8 x 24mm	11 x 26mm	13 x 27.9mm	11 x 26mm
Pes	3.3g	3.4g	6g	3.8g
Resolució	320 x 320 píxel	256 x 256 píxel	320 x 240 píxel	***
Processador d'imatge	CMOS	CMOS	CMOS	CCD
Freqüència	3 fps	2 fps	2 fps	2 fps
Temps de bateria	10 a 12 horas	8 horas	7-9 horas	8-9 horas
Angle de visió	150°	156°	140°	145°

Càpsula	MicroCam®	PillCam SB2®	OMOM®	Endocapsule®
TTI ³	Human body communication (HBC)	Radiofreqüència	Radiofreqüència	Radiofreqüència

El sistema CE consta d'una càpsula ingerible, una gravadora i una estació de treball on es realitzen la descàrrega i l'anàlisi de les imatges. La càpsula més extensa actualment (SB2) mesura 11 x 26 mm, pesa al voltant de 3-4g i porta incorporada una càmera de vídeo en miniatura en color, una font de llum, piles d'òxid de plata i un transmissor de ràdio-telemetria de banda d'alta freqüència. A través de radiofreqüències la càpsula transmet les imatges a uns sensors situats a la paret abdominal del pacient, i des d'aquí a una enregistradora o gravadora tipus Holter. La càpsula, que circula per l'intestí propulsada pels moviments peristàtics, realitza fotografies a una velocitat de 2 fotogrames/segon i durant un temps aproximat de 8-9 hores.

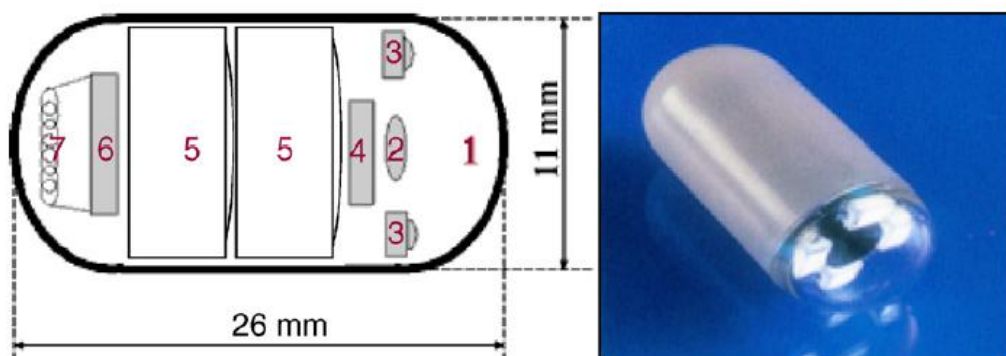


Figura 8: Càpsula endoscòpica SB2: 1) objectiu, 2) lent, 3) indicadors lluminosos, 4) captador d'imatges, 5) bateries, 6) transmissor ASCII, 7) antena.

Existeix un nou model de càpsula (SB4) en proves, que realitza 4 fotografies per segon i amb una bateria de major autonomia. Els estudis preliminars parlen d'un augment discret de la sensibilitat d'aquesta càpsula comparada amb la SB2 en la detecció de lesions en el tub digestiu. Captura més imatges de cada lesió disminuint així el percentatge de falsos positius i negatius.

Les imatges emmagatzemades a la gravadora es descarreguen en una estació de treball amb l'ajuda d'un software especial (RAPID 6) i a continuació es poden visualitzar com una pel·lícula, de manera contínua. Aquest software inclou eines d'identificació i tipificació de lesions, com un detector d'hemorràgia, la possibilitat de

³ TTI: Tecnologia de transmissió d'imatge.

realitzar cromoendoscòpia⁴ de tipus FICE i un sistema multivísó (González Suárez, et al. 2011).

1.3.2 Procediment d'exploració

L'exploració amb CE es realitza de forma ambulatoria. Quan el pacient arriba a la consulta, s'introdueixen les seves dades a l'estació de treball i es prepara la gravadora per l'estudi. Posteriorment, es col·loquen els sensors adherits a l'abdomen i s'ingereix la càpsula. Després de 2 hores de la ingesta el pacient podrà prendre la seva medicació habitual, i a les 4 hores podrà ingerir aliments lleugers. Després de 8-9 hores, el pacient haurà de tornar a l'hospital, on se li retirarà la gravadora i el cinturó, i es procedirà a descarregar l'estudi per la seva posterior revisió.

En les següents 2 setmanes, es recomana al pacient que vigili l'expulsió de la càpsula i que, en cas de no identificar-la, es posi en contacte amb el seu metge (González Suárez, et al. 2011).

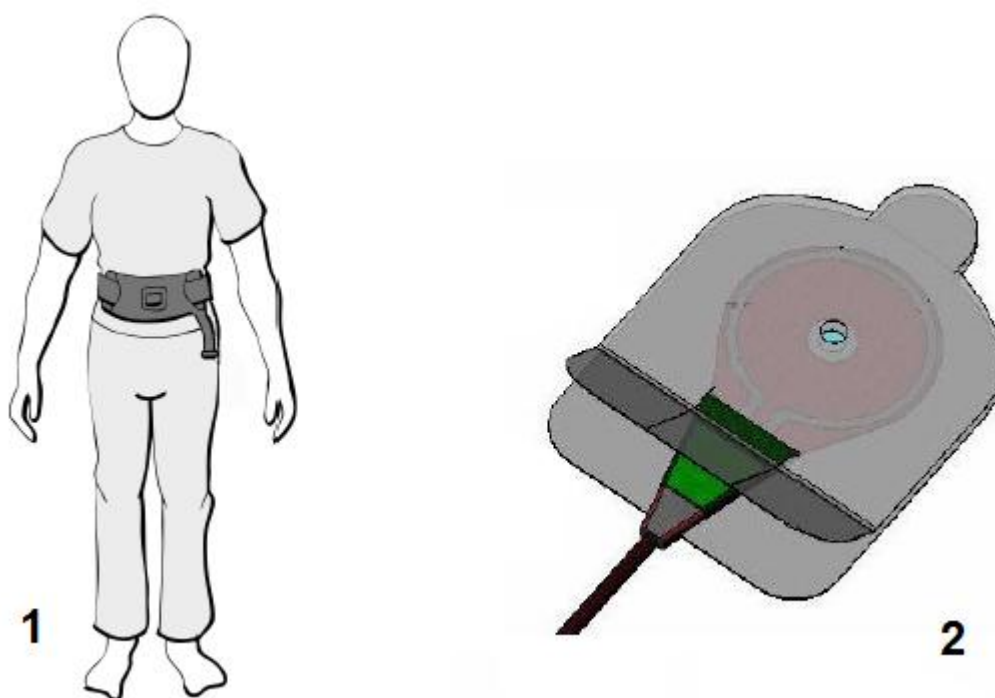


Figura 9: 1 - exemple de la posició i morfologia del cinturó que conte els sensors que registren l'activitat de la CE, 2 - exemple de sensor que es pot col·locar en parts sensibles del pacient que necessitin d'una atenció més accentuada.

⁴ Cromoendoscòpia: Tècnica diagnòstica de tinció sobre la mucosa digestiva per millorar la facilitat del diagnòstic.

Taula 3: Indicacions i contraindicacions sobre la utilització de la càpsula endoscòpica.

Indicacions	Contraindicacions
Hemorràgia digestiva d'origen obscur	Pacients amb Estenosis coneguda en el tracte gastrointestinal
Malaltia inflamatòria intestinal	Embaràs
Malaltia tumoral i síndromes polinòsics	Retenció de la Càpsula Agile Patency ⁵ o la seva expulsió deformada.
Malaltia celíaca	Pacients portadors de desfibril·ladors implantats hauran d'ésser controlats amb monitoratge intrahospitalari
Avaluació de lesions intestinals amb AINE ⁶	Pacients amb problemes de deglució com disfàgia orofaríngea o un diverticle de Zenek ⁷ .

1.3.3 Temps de diagnòstic

Un cop finalitzada l'exploració, s'obté el vídeo final que recull el recorregut realitzat per la CE. La CE enregistra des de la posada en marxa fins a esgotar la bateria, aquest fet provoca que normalment els vídeos que s'obtenen arribin a les 8 hores de duració. Mitjançant el software del fabricant, un expert pot separar les diferents zones del tracte digestiu per extreure vídeos de menor duració i poder examinar zones específiques del pacient.

El diagnòstic de les lesions es realitza mitjançant el visionat per part d'un expert del vídeo resultant. Un especialista pot visualitzar el vídeo a una velocitat de 10 fps amb certa eficiència, un augment de fps podria suposar que es passin per alt certes lesions. El visionat pel diagnòstic requereix molta concentració per part de l'especialista, es recomana que no es sobrepassin els 20 minuts de visualització seguits, ja que pot comportar una disminució de la concentració de l'expert, aquest fet és clau a l'hora d'entendre que el temps mitjà de diagnòstic és de 2 setmanes.

⁵ Càpsula Agile Patency: Càpsula soluble de prova que s'ingereix abans de realitzar l'estudi amb CE per detectar estretors en el tub digestiu.

⁶ AINE: Antiinflamatoris no esteroïdals

⁷ Diverticle de Zenek: Tipus de malformació a l'esòfag.

CAPÍTOL 2:

OBJECTIU

2.1 Antecedents

El desenvolupament de tècniques computacionals pel diagnòstic d'angiодisplàsies es va incentivar, quan la implementació de la càpsula endoscòpica com a mètode de diagnòstic va esdevenir un fet, als voltants del 2001. Aquest naixement va estar motivat per tal d'evitar un cost massa gran en temps de diagnòstic, a causa de la capacitat de les càpsules d'emmagatzemar una gran quantitat de dades.

Des d'aleshores, la iniciativa en el desenvolupament d'algorismes capaços de detectar aquestes lesions ha estat en mans de les mateixes empreses que comercialitzen les càpsules endoscòpiques, amb l'afegit que sembla no ser una prioritat a millorar, si s'observa la ràtio de falsos negatius que reporten, i la confiança en la seva fiabilitat que transmeten els especialistes que hi treballen.

Aquests fets provoquen que els antecedents per aquest treball siguin nuls pel que fa a estudis d'investigació en el camp del desenvolupament d'aquest tipus d'algorisme i la comprensible falta d'informació en les bases de funcionament dels algorismes que comercialitzen les companyies privades, que actualment subministren aquest producte.

2.2 Objectiu general

L'objectiu general del projecte és el desenvolupament en MATLAB, d'un algorisme capaç de detectar lesions en l'intestí prim provocades per una angiодisplàsia mitjançant imatges extreteres d'una càpsula endoscòpica, que superi els resultats que obtenen els softwares comercials contemporanis, tant en augment del nombre de deteccions positives com en disminució dels casos de falsos negatius en l'anàlisi de les imatges extreteres d'un estudi amb càpsula endoscòpica.

2.3 Objectius específics

Els objectius específics són metes més concretes que s'engloben dins l'objectiu general necessaris pel correcte desenvolupament del projecte:

1. Determinació de característiques específiques per l'estudi de les lesions produïdes per angiодisplàsies en imatges d'endoscòpia.
2. Determinació de llindars característics per la discriminació de les lesions respecte als elements circumdants.
3. Desenvolupament d'una metodologia per la correcta construcció d'un algorisme capaç de detectar lesions produïdes per angiодisplàsia.
4. Reducció del temps de diagnòstic d'un estudi realitzat amb càpsula endoscòpica.

CAPÍTOL 3: METODOLOGIA

En el capítol de la metodologia es descriu amb detall, els processos desenvolupats i les solucions proposades per l'assoliment dels objectius establerts.

3.1 Preprocessat i obtenció de les imatges

Pel desenvolupament del projecte s'ha utilitzat un recull de vídeos d'exploracions realitzats amb una càpsula endoscòpica model PillCam SB 3[®], cedits pel Dr. Marco Antonio Álvarez González metge adjunt del departament de digestologia de l'Hospital del Mar de Barcelona.

Taula 4: Característiques tècniques de la càpsula endoscòpica emprada.

Càpsula	PillCam SB3[®]
Fabricant	Given Imaging
Mida	11 x 26mm
Pes	2.9g
Resolució	320 x 320 píxel
Processador d'imatge	CMOS
Freqüència	AFR ⁸ de 2 a 6 fps
Temps de bateria	8 horas
Angle de visió	156°
TTI	Radiofreqüència

A causa del fet que la càpsula endoscòpica es posa en funcionament abans que el pacient l'ingereixi, les primeres imatges dels vídeos podien contenir el rostre del pacient. Una de les primeres mesures en la realització del projecte ha estat garantir el tractament anònim de les dades dels pacients. Per tal de fer-ho, el Dr. Marco Antonio Álvarez González utilitzant el software del fabricant, en aquest cas de Given Imaging, seleccionava la regió que corresponia a l'intestí prim, des del pílor fins a la vàlvula ileocecal, d'aquesta manera es descarta la possibilitat de reconèixer el pacient.

⁸ Adaptive frame rate: Varia els fps depenent de la velocitat de la càpsula, a més velocitat més fps.

Les seccions de vídeo s'han extret en format mpg. La mitjana de duració dels vídeos extrets és de 35 minuts a una reproducció de 25 fps, aquesta dada es pot considerar relativa, ja que el temps que la CE pot passar enregistrant dins de l'intestí prim varia entre 2 i 6 hores.

Un cop extreta la secció de vídeo, s'ha hagut d'obtenir els fotogrames que conformen el vídeo, ja que l'algorisme desenvolupat treballa a partir d'imatges endoscòpiques. Per fer-ho s'ha desenvolupat un codi d'extracció de fotogrames a partir d'un vídeo.

El codi sencer es pot trobar en l'annex 7.2. El codi s'ha dissenyat a tall d'eina que es pugui utilitzar en altres projectes, generalitzant tots els processos d'anàlisi al màxim.

3.1.1 Selecció dels estudis

La primera secció del codi per el processat dels estudis en vídeo, són un conjunt de funcions per preparar l'entorn del MATLAB per evitar possibles errors, definir el directori on es troben tots els estudis i entrar-hi.

L'algorisme rep el nom de "extract_frames.m" com es pot observar a la taula 5. Les 3 primeres línies del codi (close all, clear all, clc) s'utilitzen per tancar figures obertes en accions anteriors, eliminar les variables que estiguin al "workspace" de MATLAB i esborrar la finestra de comandes del MATLAB. Aquestes 3 funcions són cànons de tots els scripts que s'han desenvolupat del treball, per tal d'eliminar el risc que codis executats amb anterioritat afectin el resultat del codi amb el qual s'està treballant.

Taula 5: Primera secció del codi "extract_frames.m".

```
close all;
clear all;
clc;

tic;

folder = uigetdir;
folders=dir([folder,filesep,'*.mpg']);
cd(folder);
```

La següent línia de codi "tic", s'utilitza juntament amb la seva complementària "toc" que es col·loca al final del codi, serveix per saber el temps de computació en segons. El temps de computació, apareix escrit a la finestra de comandes del MATLAB, la informació del temps de computació s'ha utilitzat per a la gestió logística del projecte, ja que per extreure els fotogrames d'un vídeo de 35 minuts, el temps aproximat que es requereix és de 3 hores.

La següent línia de codi, s'assigna la variable "folder" a un directori que l'usuari del script determina (s'entén que es seleccionarà el directori on hi estigui emmagatzemat el vídeo o vídeos que es volen editar), a partir de la interacció amb una finestra que apareix, com es pot observar a la figura 10.

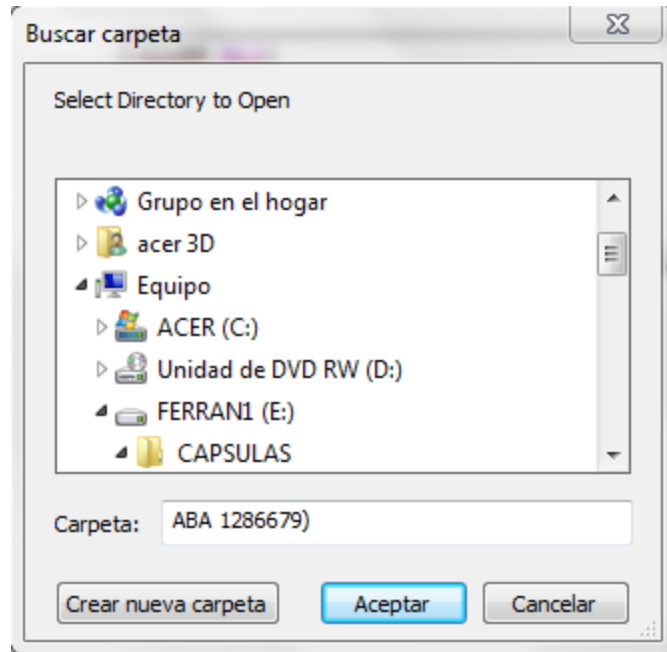


Figura 10: Finestra que apareix quan executem la comanda "uigetdir"

La sisena línia de la secció de codi, es llisten tots els arxius que estant dintre del directori seleccionat a l'apartat anterior amb format mpg i s'assignen a una variable tipus "structure" anomenada "folders". Aquest resultat s'aconsegueix amb la funció "dir". Per executar la funció "dir", s'ha d'introduir el directori que està assignat a la variable "folder" i si es vol, especificar el format d'arxiu que es llistarà, en el nostre cas els arxius en format mpg. La funció "filesep" afegeix una barra espaiadora (/) entre dos variables tipus "string", en el nostre cas "folders" i "*.mpg". El significat de posar "*" davant ".mpg" és dir a la funció que seleccioni tots els arxius en format mpg, sense haver d'especificar el directori un per un, en el cas que existeixi més d'un arxiu. Aquestes dues últimes línies de codi permeten executar el script des de qualsevol carpeta que estigui i extreure els fotogrames d'un vídeo o més, que estiguin guardats en una carpeta diferent d'on tenim el script.

I per acabar, l'última línia del codi serveix per entrar dins el directori que tenim guardat a la variable "folder" amb la funció "cd".

Aquest script està pensat per analitzar vídeos amb format mpg, però canviant l'extensió es poden executar vídeos en altres formats.

3.1.2 Processat del vídeo

La segona secció del codi, conté la lectura del primer vídeo que s'executa, la generació de la carpeta on es guardaran els fotogrames extrets i l'assignació de les variables necessàries per a l'extracció dels fotogrames.

Taula 6: Segona secció del codi.

```
for kfolder = 1:length(folders)

obj = VideoReader([folders(kfolder).name,]);

nFotogrames = obj.NumberOfFotogrames;
name=obj.name;
name = strrep(name, '.mpg', '_IMAGES');

    mkdir(folder,name);
    cd([folder,filesep,name]);
```

La primera línia del codi, és l'inici d'un bucle que ens serveix per executar totes les comandes que es troben dintre d'aquest, per cada vídeo que es tingui llistat. Aquest bucle en la primera versió del script no existia, ja que no es disposaven de molts vídeos, un cop va augmentar el nombre d'aquests es va generar una segona versió que era compatible amb l'extracció de fotogrames d'un o més vídeos continguts en una mateixa carpeta. La línia "kfolder = 1:length(folders)" genera un vector amb nom "kfolder", que té tants valors com numero de vídeos es tenen llistats. El número de vídeos llistats s'obté aplicant la funció "length" sobre la variable "folders".

A la línia següent, es llegeix el vídeo que es vol processar mitjançant la funció "VideoReader" i es guarden les seves propietats en una variable tipus "object" que s'ha anomenat "obj". La funció "VideoReader" a part de carregar com a variable el mateix vídeo, extreu d'aquest la duració en segons, el nom de l'arxiu, el directori on es troba, els bits per píxel, la velocitat en fps, la mida de cada frame (ex. 576 x 576), el número de fotogrames i el format del vídeo (ex. RGB24). La funció necessita com a variable d'entrada el nom de l'arxiu que es vol llegir, que en el nostre cas es troba dins la variable de tipus "structure" que s'ha anomenat "folders".

En les següents dues línies de codi, s'assignen els noms de "nFotogrames" al nombre total de fotogrames que té el vídeo i "name" al nom del vídeo. Per accedir a les propietats emmagatzemades utilitzant la funció "VideoReader", és necessari escriure el nom de la variable en la qual s'han guardat les dades, seguit d'un punt i el nom de la propietat (ex. nFotogrames=obj.NumberOfFotogrames).

La cinquena línia de codi es necessària per modificar la variable "name". Aquesta variable s'utilitzarà per crear més endavant una carpeta en la qual guardarem tots els fotogrames extrets. Al fer servir el nom del vídeo per crear la carpeta es produeix un error perquè dos arxius tenen el mateix nom, per això és necessari modificar la variable "name" per tal que quan es creï la carpeta no tingui el mateix nom que el vídeo que s'analitza. En el nostre cas s'ha optat per eliminar l'extensió ".mpg" que conté el nom del vídeo i afegir "_IMAGES" al final d'aquest. Aquesta operació es realitza mitjançant la funció "strrep", que a partir d'una variable tipus "string" substitueix una part d'aquesta per una de nova, es pot observar el resultat de la creació de la carpeta a la figura 11.

La sisena línia de codi és la creació de la carpeta amb la funció "mkdir". Per fer-ho li donem com a variables d'entrada "folder" que conté el directori on es vol que es crei la carpeta, i "name" que conté el nom que es vol que tingui la carpeta.

Finalment a l'última línia de codi, accedim a la carpeta que s'ha creat per tal que quan es llegeixin els fotogrames es guardin en aquesta. Per fer-ho s'utilitza la funció "cd", que necessita com a variable d'entrada el directori on es troba la carpeta, amb el nom d'aquesta. En el nostre cas el directori es troba dins la variable "folder" i s'afegeix el nom de la carpeta que es troba en la variable "name", utilitzant la funció que s'ha explicat anteriorment "filesep". Es pot comprovar el resultat de l'execució d'aquestes últimes línies de codi a la figura 11.

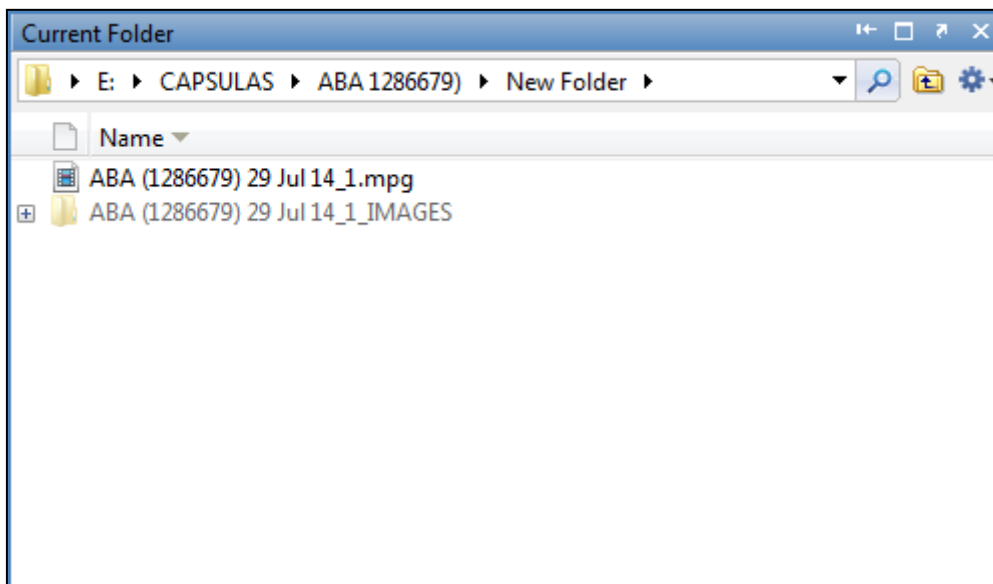


Figura 11: Carpeta creada amb la funció "extract_fotogrames"

3.1.3 Obtenció dels fotogrames

La tercera i última part del vídeo, és on s'extreuen els fotogrames del vídeo que es processa i es guarden a la carpeta que s'ha creat el l'apartat anterior.

Taula 7: Tercera secció del codi.

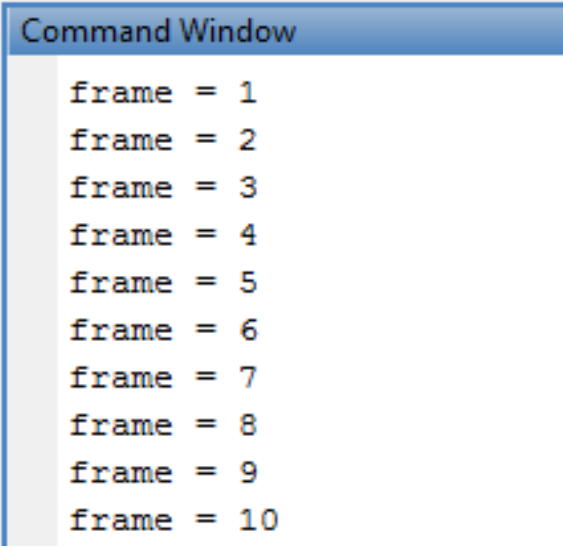
```
for k = 1 : nFotogrames
    disp(['frame = ', num2str(k)]);
    I = read(obj, k);

    filename = sprintf('IMG_%d.png', k) ;
    imwrite(I, filename);
end
cd ..
end;

toc;
```

En la primera línia, es troba l'inici del bucle que ens servirà per extreure tots els fotogrames del vídeo que s'està executant. Definim el vector "k", que tindrà la mateixa longitud que el nombre total de fotogrames i que serveix de guia per anar seleccionant els fotogrames d'un en un.

La segona línia de la secció és un recurs visual intern per l'usuari que executa el codi. Utilitzant la funció "disp(['frame=',num2string(k)]);" van apareixent a la finestra de comandos de MATLAB el número de fotogrames que es van guardant, com es pot observar en la figura 12. La funció "disp" és un diminutiu de la paraula anglesa "display" que podem traduir com mostrar i el que fa és mostrar a la finestra de comandos de MATLAB les variables tipus "string" d'entrada que li donem. Com que la funció "disp" requereix variables tipus "string", s'ha transformat el valor nominal del fotograma que s'executa a "string" mitjançant la funció "num2string".



```
Command Window
frame = 1
frame = 2
frame = 3
frame = 4
frame = 5
frame = 6
frame = 7
frame = 8
frame = 9
frame = 10
```

Figura 12: Resultat d'executar la línia de codi "disp(['frame=',num2string(k)]);" "

La tercera línia de codi és la que executa l'extracció dels fotogrames. La funció "read" llegeix els fotogrames del vídeo associat, en el nostre cas l'hem definit amb el nom de variable "obj". El valor "k" representa el número de fotograma que llegeix del vídeo associat.

Les dues següents línies de codi serveixen per crear un nom pel fotograma extret i guardar-lo a la carpeta que s'ha creat anteriorment. Assignem una primera part del nom que serà general per tots els fotogrames, i una segona part que anirà amb relació al número de fotograma que és. Per fer-ho s'ha utilitzat la funció "sprintf", que aplica el format que interressi al vector que se li dona com a variable d'entrada. En el nostre cas s'ha aprofitat la funció per afegir-li el format png al fotograma i donar un nom característic per cada fotograma. Per fer-ho s'ha donat com a format d'entrada "IMG_%d.png", la primera part "IMG" serà la part general del nom que tots els fotogrames extrets tindran, la part "%d" és una característica de la funció que escriu el número de fotograma que està essent processat,

sent aquesta la característica en el nom que diferencia cada fotograma. La següent línia assigna el nom que s'ha definit amb la funció anterior i el guarda a la carpeta que s'ha creat amb la funció "imwrite".

Les següents línies de codi són les finalitzacions dels bucles "for" indicades amb un "end", la funció "cd .." que serveix per anar al directori anterior per si és el cas d'haver-hi més d'un vídeo, que totes les carpetes que es creen per guardar els fotogrames estiguin al mateix directori i la complementaria a la funció "tic", que s'ha esmentat a la primera secció del codi, "toc".

3.2 Caracterització de les lesions

El problema fonamental del disseny del codi ha estat la caracterització de les lesions, per tal de poder discriminar-les de l'entorn i així detectar-les. En ser un projecte pioner en aquest tema no existeix cap estudi que determini patrons específics que presenten les lesions derivades d'una angiодisplàsia, aquest fet ha comportat que s'hagi hagut de fer una recerca de característiques pròpies que presenten les lesions, per poder identificar-les mitjançant un algorisme de detecció i no poder comptar amb material bibliogràfic per la resolució d'aquest problema.

El Dr. Marco Antonio Álvarez González a part de seleccionar el fragment de vídeo corresponent a l'intestí prim, també ha facilitat imatges de lesions detectades pel software del fabricant, lesions no detectades i falsos positius (imatges que el software del fabricant detectava que existia una lesió de manera errònia). La determinació de característiques s'ha fet a partir de les 19 imatges que contenen lesions, tant les detectades pel software del fabricant, com les que no, ja que un dels objectius és millorar la sensibilitat del software.

Per poder determinar les característiques de les lesions, ha sigut clau poder en primera instància segmentar les regions on apareixia la lesió. Aïllant les regions de la imatge on apareixen les lesions, es poden obtenir els píxels que fan referència a aquestes. Estudiant aquests píxels, s'han obtingut algunes característiques regulars en aquest tipus de lesió.

En la primera versió del codi que s'utilitza per a la segmentació, es va pretendre englobar en una única secció de codi, la segmentació de totes les lesions. Les diferències d'intensitat entre les lesions, van esdevenir un problema, ja que no era possible segmentar totes les lesions amb un mateix codi sense prescindir d'algunes d'aquestes, pel fet que el nivell que s'aplica a la binarització no era regular en totes elles. Un cop analitzat el problema, es va decidir classificar les lesions en dos grups. El primer grup es van anomenar lesions fosques i el segon lesions clares (figura 13). Aquesta classificació s'ha utilitzat de manera interna per poder separar en dos grans blocs, les lesions típiques d'angiодisplàsia i dissenyar per a cadascun dels grups, un codi propi per la segmentació de les lesions, solucionant així el problema anterior i millorant la precisió en la segmentació.



Figura 13: A l'esquerra, lesió clara. A la dreta, lesió fosca.

3.2.1 Binarització de les lesions fosques

Per la determinació de les lesions, en la primera versió del codi, es va fer un estudi de les propietats en gran mesura manual, utilitzant les funcions que la interfície del MATLAB proporciona. Un cop es va obtenir més informació sobre aquestes, es va poder automatitzar el procés.

El primer pas per dur a terme l'estudi manual, és carregar la imatge que es pretén estudiar a l'estació de treball del MATLAB. Per fer-ho s'executa la funció "imread".

Taula 8: Funció "imread" i les tres funcions canòniques "close all", "clear all" i "clc".

```
close all
clear all
clc
I=imread('1.png');
```

Un cop es té la imatge carregada i assignada a una variable, s'obra amb el visor de MATLAB, mitjançant la funció "imshow". Afegint la funció "figure" davant de l'anterior enumerem la imatge i evitem que quedi superposada per altres que es puguin visualitzar amb posterioritat.

Taula 9: Funció "figure" i "imshow".

```
figure; imshow(I)
```

El visor de MATLAB ofereix una sèrie d'eines per la manipulació i l'obtenció de dades de la imatge que s'ha carregat, en el nostre cas s'utilitza la funció

"data cursor". Aquesta funció et permet seleccionar un píxel de la imatge i observar el seu índex, així com els seus valors en els canals rgb⁹ (figura 14).

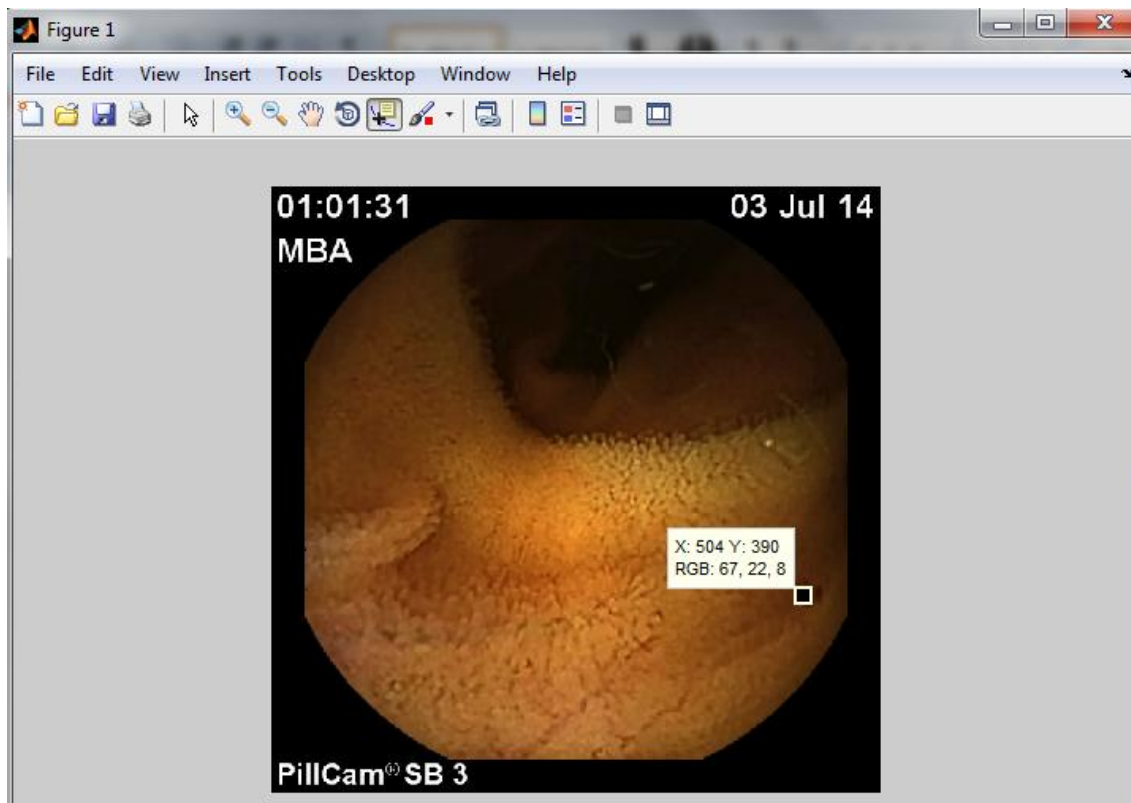


Figura 14: En la part superior de la figura es pot observar l'eina "data cursor" seleccionada i com en proporciona la informació del píxel seleccionat.

Un cop es van estudiar les propietats dels diferents canals per totes les lesions que s'havien classificat com a fosques, es va poder determinar que aquestes mantienien un rang en les intensitats del canal vermell, verd i blau molt característic. Aquesta informació s'ha utilitzat per a l'elaboració de restriccions com es pot veure en apartats posteriors.

A continuació s'assignen a diferents variables, els 3 canals per separat i la imatge en escala de grisos processada pel MATLAB amb la funció "rgb2gray". Aquesta funció, converteix les imatges en color a escala de grisos eliminant la informació sobre la tonalitat i saturació, al mateix temps que conserva la luminància.

Un cop assignades les variables, es va seguir el mètode científic d'assaig i error, per determinar quina de les 4 tenia les millors propietats per caracteritzar les lesions, executant binaritzacions amb llindars similars per les 4. Un cop acabat l'estudi, es va poder determinar que el canal que identificava amb més facilitat les lesions respecte al seu entorn era el verd. Aquest fet es deu principalment perquè els píxels de les lesions tenen l'índex del canal verd força elevat, fet que els diferencia dels píxels dels

⁹ rgb: Referència als tres canals de color: red (vermell), green (verd), blue (blau)

seus voltants. Aquesta característica no passa amb el canal vermell, ja que al ser l'intestí prim de color vermellós, la majoria de píxels de tota la imatge comparteixen la propietat de tenir l'índex del canal vermell elevat. Tot el contrari amb el canal blau, que es pot considerar el canal oposat al vermell, aquest fet comporta que la majoria dels píxels tinguin molt baix l'índex del canal blau. L'elecció s'havia de prendre entre utilitzar el canal verd com a referent per la binarització o la imatge passada a escala de grisos pel MATLAB. Aquest conflicte es va resoldre experimentalment, determinant com a millor mètode la utilització del canal verd per binaritzar.

Taula 10: Assignació a variables dels 3 canals, i conversió de la imatge a escala de grisos amb la funció "rgb2gray".

```
Ired=I(:,:,1);  
Igreen=I(:,:,2);  
Iblue=I(:,:,3);  
IR=rgb2gray(I);
```

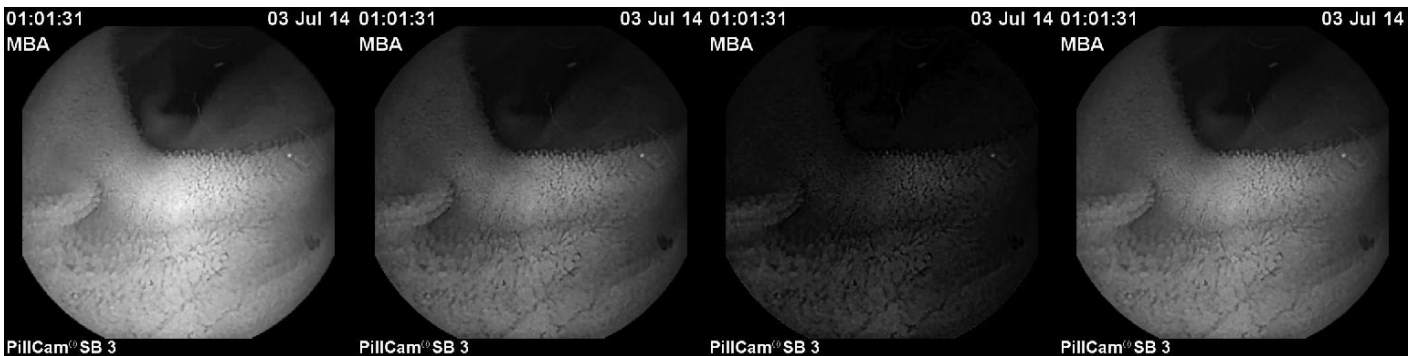


Figura 15: D'esquerre a dreta: canal vermell, canal verd, canal blau i escala de grisos.

Un altre problema que es va detectar per la binarització és la falta de contrast en la imatge. Aquest fet perjudica la determinació de les regions amb lesions, que tenen menys diferència d'intensitat respecte als seus píxels veïns. Per solucionar aquest problema, es realitza una compensació del contrast de la imatge mitjançant una rehistogramació automàtica amb la funció "adapthisteq". Aquesta funció millora el contrast mitjançant la transformació dels valors d'intensitat de la imatge que se li aplica. La funció opera en petites seccions de dades, millorant el contrast i combina els límits entre seccions mitjançant un càlcul d'interpolació bilineal, per tal d'eliminar les fronteres creades artificialment per aquesta. A causa del fet que opera en regions petites de la imatge, la transformació global de l'histograma és menys agressiva que en altres funcions que produeixen el mateix efecte, com per exemple la funció "histeq", com es pot observar a la figura 16.

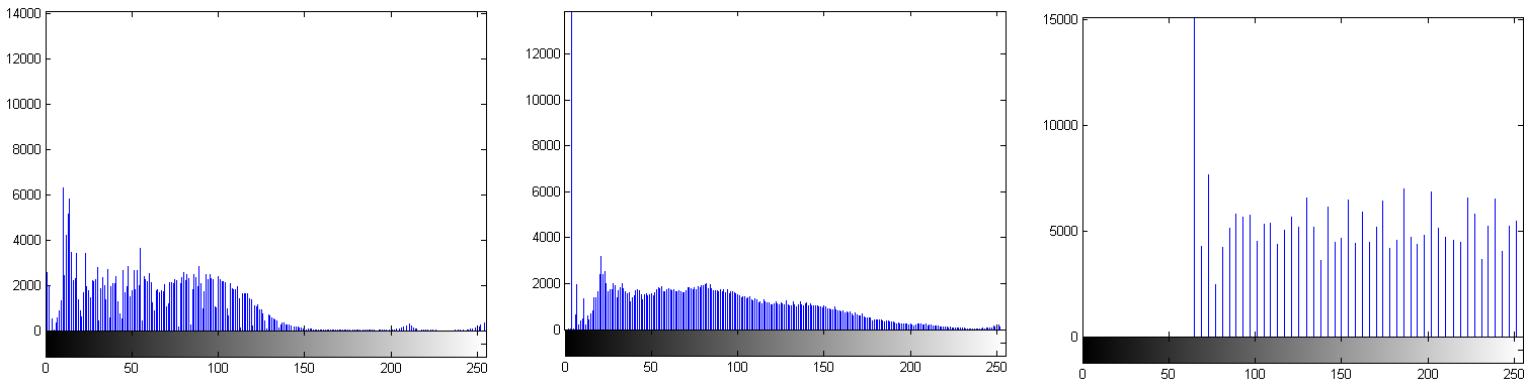


Figura 16: D'esquerre a dreta: Histograma del canal verd, histograma després d'aplicar la funció "adaphisteq", histograma després d'aplicar la funció "histeq"

El que es pretén en definitiva amb l'aplicació de la funció "adaphisteq", és un eixamplament de l'histograma per tal de millorar el contrast sense provocar un canvi molt abrupte en aquest, l'operació es coneix amb el nom d'equalització de l'histograma. Un cop aplicada la funció, s'obté una gran millora en el contrast de la imatge, com es pot observar en la figura 17. Amb aquesta transformació s'augmenta la diferència d'intensitat entre els píxels continguts a la regió on està la lesió i els píxels dels voltants, facilitant la diferenciació entre ells, aquest fet també comporta un augment del soroll, que es tractarà amb posterioritat.

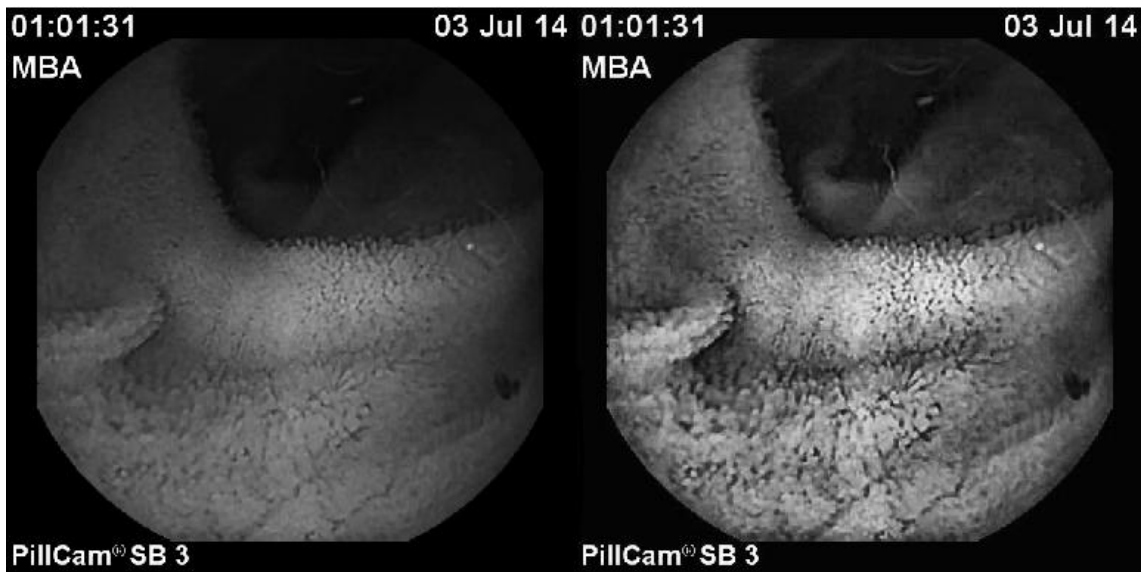


Figura 17: D'esquerre a dreta: Imatge del canal verd, Imatge del canal verd processada per la funció "adaphisteq".

Taula 11: Aplicació de la funció "adapthisteq" sobre la imatge representativa del canal verd i les funcions per visualitzar el resultat.

```
Iadpt=adapthisteq(Igreen);  
figure;imshow(Iadpt)
```

Després de realitzar la rehistogramació, una de les parts importants, no tant pel que fa a la detecció de les lesions, sinó per la reducció de possibles errors, és l'eliminació del marc de la imatge que s'analitza, que conté el temps de gravació, la data, el nom de l'estudi i el nom de la CE que s'ha utilitzat. Per fer-ho, s'aplica un filtre morfològic d'erosió a la imatge amb una alta superfície d'acció. El filtre d'erosió encongeix o aprima objectes en una imatge binària. El control de l'encongiment es pot determinar mitjançant un element estructural pel qual es propagarà l'erosió.

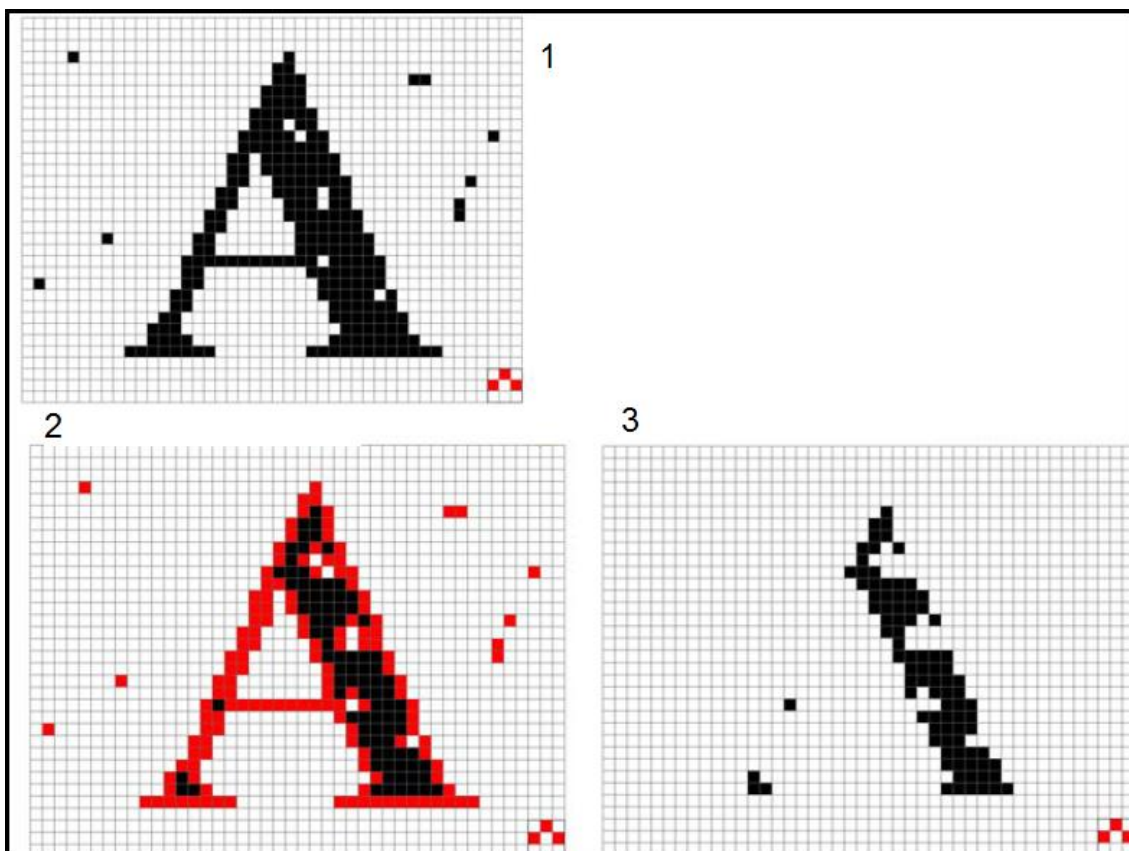


Figura 18: Funcionament del filtre erosió. A la imatge 1 es pot observar la figura original, a la imatge 2 l'aplicació de l'element estructural i en vermells els píxels que es veuran afectats per l'erosió i a la tercera imatge, es pot observar el resultat. A la cantonada dreta de cada imatge apareix en vermell, l'element estructural que s'aplica.

Per aplicar aquest filtre en MATLAB, primer s'ha d'especificar l'element estructural que s'utilitzarà per dur a terme l'erosió. Per fer-ho, s'empara la funció "strel", assignant-li quin tipus d'estructura es vol i el radi o l'àrea d'acció. La funció accepta les següents estructures: *diamond*, *disk*, *line*, *arbitrary*, *ball*, *octagon*, *pair*, *peridicline*, *rectangle* i *square*. Pel nostre cas

s'ha utilitzat l'estructura *disk* amb un radi de 5. Una vegada especificat l'element estructural, s'aplica el filtre d'erosió a la imatge anterior, mitjançant la funció "imerode", on s'ha d'assignar-li com a variables d'entrada la imatge a erosionar i l'element estructural.

Taula 12: Definició de l'element estructural amb la funció "strel" i aplicació del filtre morfològic d'erosió.

```
se=strel('disk',5);  
Ie=imerode(Iadpt,se);
```

En aplicar un filtre morfològic a una imatge no binaritzada, l'erosió afecta a tota la imatge, eliminant les zones que interessaven, però també perjudicant greument la qualitat de la imatge. Per aquest fet és necessària fer una reconstrucció de la imatge a partir de la resultant erosionada, per mantenir les parts importants de la imatge inalterades. Per fer-ho utilitzarem una funció que aplicarà una reconstrucció morfològica sobre la imatge resultant, utilitzant com a màscara la imatge prèvia a l'erosió. La funció que s'utilitza és "imreconstruct", aquesta funció necessita una imatge com a punt de partida, que en el nostre cas és la imatge erosionada, i una imatge que sigui la transformació final, que en el nostre cas és la imatge prèvia a l'erosió. El que efectua la funció és un procés de reconstrucció iteratiu sobre els píxels que no tenen índex 0 (negres) per deixar-los igual que la imatge que s'ha assignat com a transformació final.

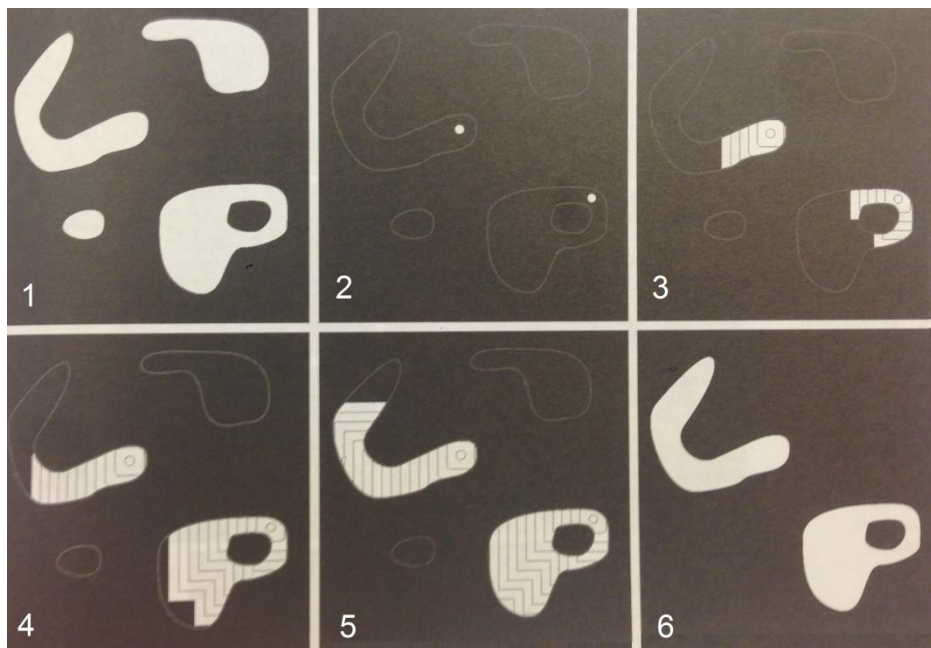


Figura 19: Funcionament del filtre morfològic de reconstrucció: 1- Imatge final. 2- Imatge inicial. 3,4 i 5 – Iteracions de reconstrucció. 6 – Imatge reconstruïda sense els píxels eliminats completament.

L'avantatge d'aquesta transformació és que els píxels blancs del marc que s'han eliminat durant l'erosió no es reconstrueixen perquè en la imatge de partida no queda cap rastre d'ells, fet que fa impossible la reconstrucció sense un píxel de sortida. Pel que fa als píxels corresponents a la imatge, la reconstrucció és gairebé total, ja que el procés d'erosió ha estat poc agressiu i només ha eliminat els píxels que estaven amb molt de contacte amb píxels d'índex d'intensitat 0. Aquest fet fa que a partir dels píxels que han quedat es pugui reconstruir tota la imatge que ens interessa deixant fora els píxels que formaven la informació del marc de la imatge (temps d'exploració, data, nom del vídeo...).

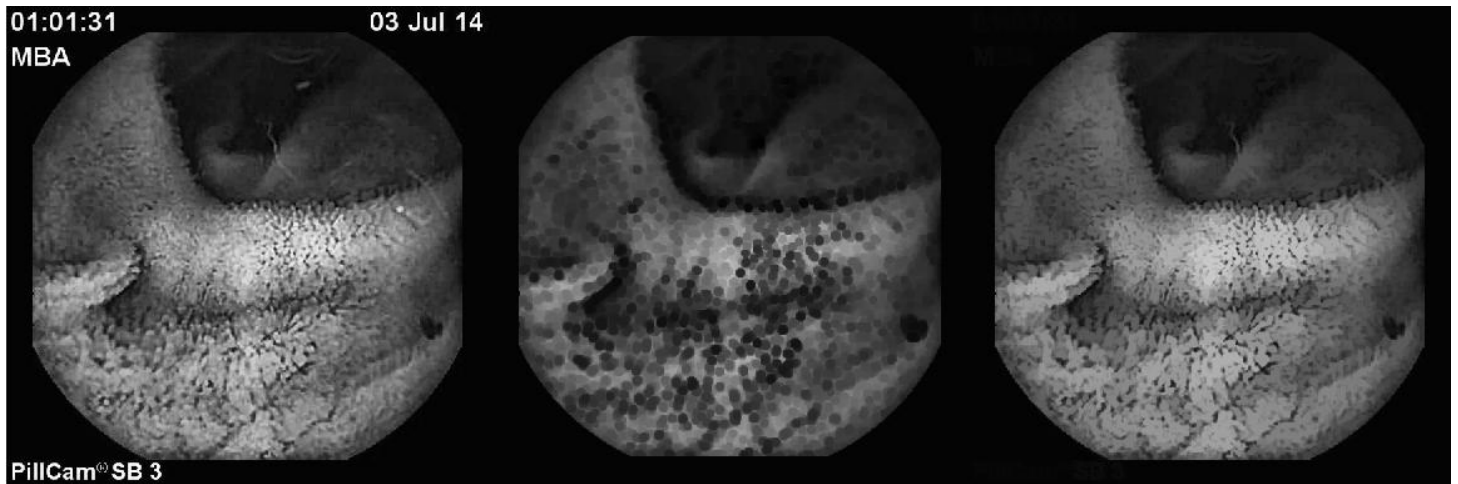


Figura 20: D'esquerre a dreta: Imatge original, Imatge erosionada, Imatge reconstruïda.

Si analitzem els histogrames (figura 21), podem observar que ha augmentat el soroll en la imatge reconstrucció, però la distribució d'aquests es manté gairebé idèntica, amb la diferència que els píxels amb índex d'intensitat més elevats han estat eliminats (com era d'esperar) i en consonància amb aquest fet, no apareixen a l'histograma de la imatge reconstruïda. La possible eliminació de píxels amb índex d'intensitat elevat, que no corresponguessin a les dades que apareixien al marc no suposa un problema, ja que les lesions no presenten píxels amb índex tan elevats, per tant no s'elimina informació valuosa per la detecció de les lesions.

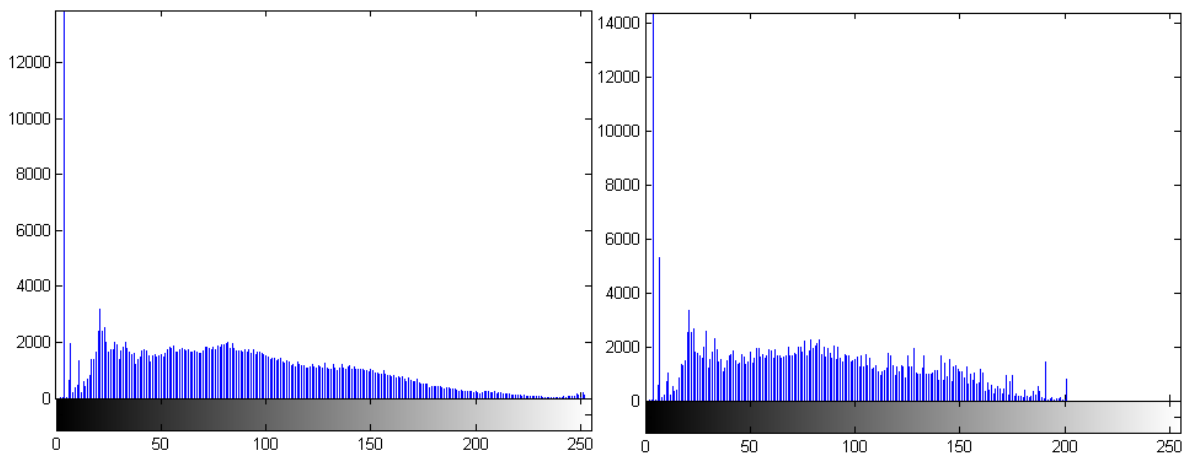


Figura 21: D'esquerre a dreta: Histograma de la imatge original, histograma de la imatge reconstruïda.

Taula 13: Aplicació del filtre morfològic de reconstrucció mitjançant la funció "imreconstruct", i la visualització d'aquest utilitzant la funció "imshow".

```
Iobr=imreconstruct(Ie,Iadpt);  
figure;imshow(Iobr);
```

Després d'eliminar els píxels del marc de la imatge i disposant només de la informació que ens interessa, s'aplica l'últim processat a la imatge abans de binaritzar. A causa de l'augment en el contrast i la reconstrucció, apareix en la imatge un alt grau de soroll tipus "pepper" que podria alterar la detecció de les lesions, per evitar-ho s'aplica un filtre de mediana a la imatge mitjançant la funció "medfilt2". Aquesta funció necessita com a valors d'entrada, la imatge que es vol filtrar i la regió d'actuació (ex. [2 2]). La funció assigna al píxel de sortida el valor de la mediana calculada a partir dels seus píxels veïns. El que determina el nombre de píxels que es tenen en compte a l'hora de calcular la mitjana, és la regió que se li assigna a la funció com a valor d'entrada.

Taula 14: Aplicació del filtre mediana.

```
I2 = medfilt2(Iobr,[15 15]);
```

L'aplicació del filtre comporta una gran disminució del soroll, però també suposa una disminució de la definició de les diferents parts de la imatge. Per la detecció de lesions, interessa eliminar el màxim nombre de regions amb píxels foscos que puguin perjudicar-la i que durant el procés, no afecti la mateixa regió que conté la lesió. El control de l'agressivitat en el filtratge es fa a través de l'àrea d'actuació que li assignes per calcular la mediana, com més gran més efecte difuminador té la funció. Com es pot observar a la taula 14, s'ha decidit utilitzar un operador elevat, que es va determinar amb el mètode científic d'assaig i error, fins a trobar la regió òptima per eliminar el màxim nombre de zones fosques sense influir en la posterior detecció de la lesió.

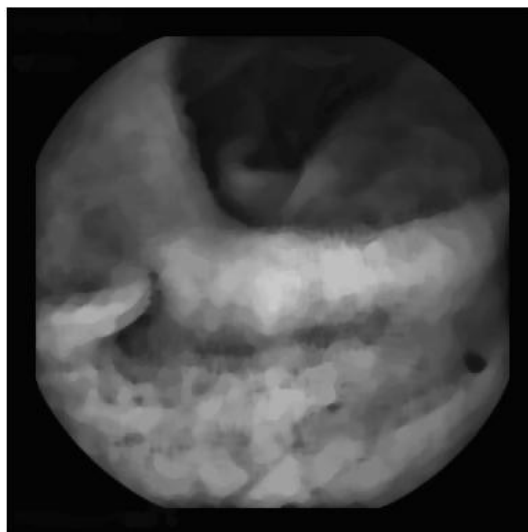


Figura 22: Resultat en aplicar el filtre de mediana

Un cop s'obté la imatge resultant d'aplicar el filtre de mediana, és necessària aplicar una binarització de la imatge, per tal de facilitar la posterior segmentació de les regions. Per fer-ho, s'utilitza la funció "im2bw", aquesta funció transforma la imatge que se li assigna com a variable d'entrada en una imatge binària (blanca i negra). Per poder realitzar la transformació s'ha d'especificar un llindar de binarització, pel qual es determinarà quins píxels passen a valdre 0 o 1. La determinació del llindar, es realitza de forma automàtica utilitzant la funció "graythresh", aquesta funció utilitza el mètode Otsu, anomenat així en honor a Nobuyuki Otsu que el va inventar el 1979. Aquest mètode calcula el valor llindar de manera que la dispersió entre valors de grisos dintre de cada segment sigui el més petit possible, però que al mateix temps la dispersió sigui la més alta possible entre segments diferents. Per fer-ho es calcula el quocient entre les dues variàncies i es busca el valor llindar perquè el quocient sigui màxim. El problema de la determinació del llindar per la binarització utilitzant aquest mètode, és que el mètode és molt sensible a les variacions de lluminositat en la imatge, com es pot veure en la figura 23.



Figura 23: Imatge amb mala lluminositat binaritzada utilitzant el mètode de Otsu.

En el nostre cas com que la càpsula endoscòpica porta incorporats uns llums led, provoca que les imatges obtingudes tinguin una diferència de lluminositat molt gran entre zones enfocades directament pels leds i zones que no. Aquest problema s'ha intentat suavitzar en els processats anteriors (equalització de l'histograma, aplicació del filtre mediana), malgrat això el llindar obtingut pel mètode Otsu, no serveix per binaritzar correctament totes les lesions, ja que el llindar obtingut té un valor massa elevat, provocant que els píxels dels voltants de les lesions passin a valdre 0, deixant aquestes sense una regió ben definida. Per aquest motiu mitjançant el mètode d'assaig i error, s'ha buscat un valor constant que provoqui que el llindar disminueixi (dividint el llindar per aquest valor), perquè valors més foscos passin a 1, deixant com a 0 les regions que presenten valors de píxel en la imatge en escala de grisos més baixos. El valor òptim resultant de l'estudi ha estat el 1.42.

Taula 15: Aplicació de funció "im2bw" i càlcul del llindar de binarització mitjançant la funció "graythresh".

```
Ibin=im2bw(I2,graythresh(I2)/1.42);
```



Figura 23: D'esquerre a dreta: Imatge binaritzada amb el mètode Otsu, imatge binaritzada disminuint el valor del llindar obtingut pel mètode Otsu amb la constant 1.42.

Algunes lesions poden estar situades als marges de la imatge (figura 24), i al binaritzar es provoca que la regió obtinguda estigui oberta per algun dels seus costats. Aquest fet promou que la segmentació no es pugui dur a terme, ja que com que no està del tot tancada, no es consideraria una regió a segmentar pel procés automàtic. Per aquesta raó, és necessari per acabar el procés de binarització correctament, aplicar un filtre morfològic de tancament i reconnectar els punts que la regió ha quedat separada.

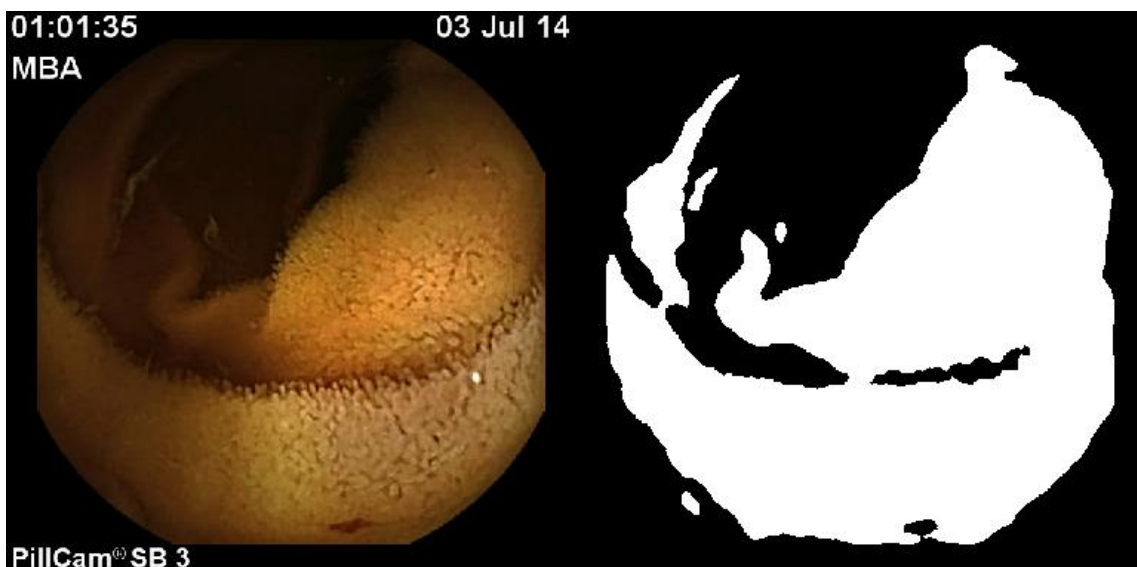


Figura 24: D'esquerre a dreta: Imatge original, Imatge binaritzada on s'observa com la regió que correspon a la lesió està oberta per un punt.

El filtre morfològic de tancament s'aplica mitjançant la funció "imclose" i de la mateixa manera que pel filtre d'erosió que s'ha comentat en l'anterior apartat, és necessari definir un operador amb la funció "strel". El filtre morfològic tancament és una combinació entre el filtre dilatació, que expandeix els píxels, i el filtre erosió que els aprima. El resultat en aplicar el filtre, és semblant al d'aplicar el filtre dilatació, amb la diferència que l'augment del soroll que provoca l'ús d'aquest filtre queda reduït per la utilització a posteriori del filtre erosió.

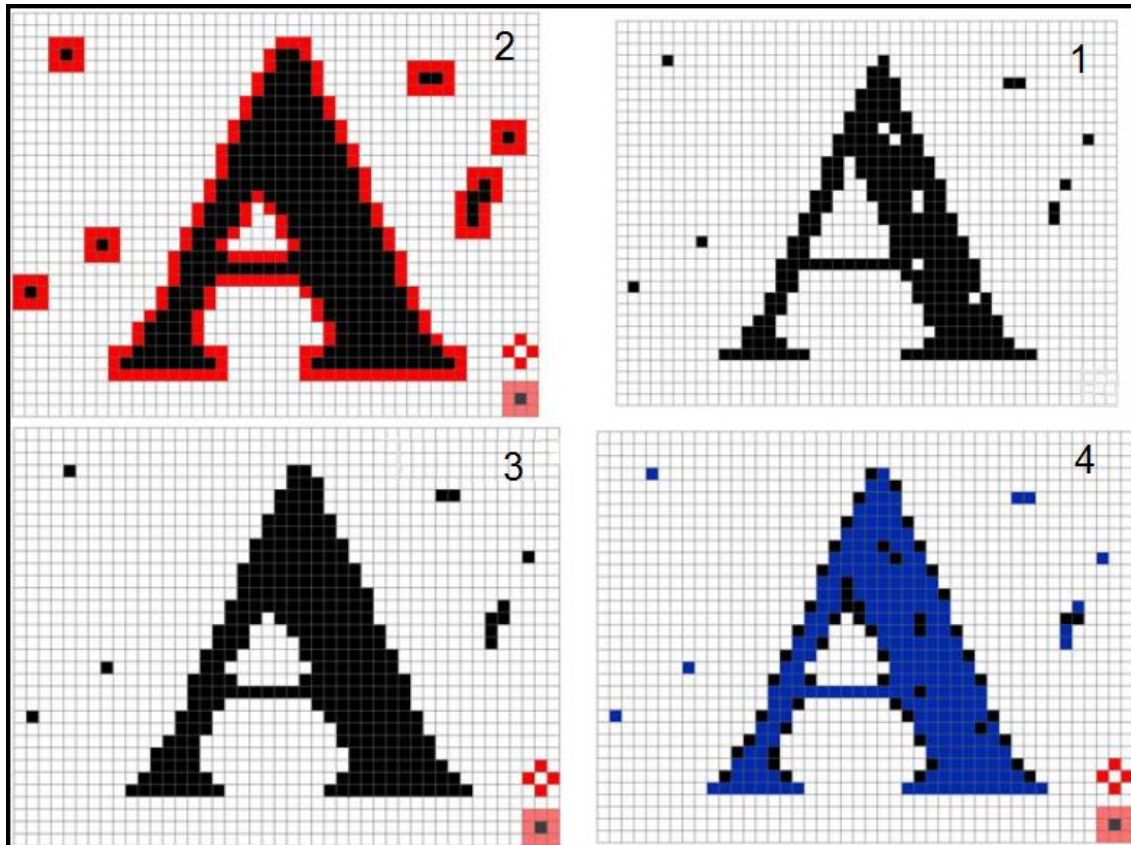


Figura 25: Exemple de l'aplicació d'un filtre tancament: 1-imatge original, 2-aplicació els operadors de dilatació i erosió, que trobem a la cantonada inferior esquerra de la imatge, 3-imatge resultant, 4-comprovació dels resultats superposant les dues imatges, en blau l'original, en negre la resultant.

Un cop aplicat el filtre de tancament, les regions obertes queden perfectament delimitades i les regions petites que podrien confondre el procés de detecció de lesions s'assimilen amb els píxels de l'entorn, eliminant-les.

Amb aquest últim processat, conclou l'etapa específica de binarització per les lesions classificades com a tipus fosc i comença el procés de segmentació, que és compartit pels dos tipus de lesions.

Taula 16: Aplicació de funció "strel" i "imclose", per aplicar un filtre morfològic de tancament sobre la imatge binaritzada.

```
si=strel('disk',3);
```



```
I4=imclose(Ibin,si);
```



Figura 26: Imatge resultant després d'aplicar el filtre morfològic de tancament.

3.2.2 Binarització de les lesions clares

La determinació de les lesions clares és un procés paral·lel a la determinació de les lesions fosques, on els dos codis parteixen de certes diferències en el processat per la binarització i comparteixen altres processos com la segmentació i l'estructura de restriccions. En l'apartat anterior s'ha explicat el mètode pel qual es binaritzen les lesions fosques, en aquest apartat es pretén explicar el mètode per la binarització de les lesions clares, però sense fer èmfasi en les funcions que ja s'han comentat en l'apartat anterior.



Figura 27: Exemple lesió clara



Figura 28: Exemple lesió clara

Les primeres 9 línies de codi de l'algorisme per la detecció de lesions clares, són exactament igual que per l'algorisme per la detecció de lesions fosques. Es comença executat les funcions "close all", "clear all" i "clc", per evitar possibles interferències d'execucions anteriors. Després es llegeix les imatges a estudiar i s'avalua mitjançant les eines d'interfície que ofereix el MATLAB, per tal de fer un estudi manual de les característiques de les lesions. Un cop s'ha estudiat les lesions, es separa la imatge en els 3 canals de color per passar la imatge a escala de grisos, també se li aplica la transformada "rgb2gray" que ofereix MATLAB i s'examina quina de les 4 lectures de la imatge afavoreix més el procés de binarització.

Taula 17: Primeres línies del codi per la detecció de lesions clares.

```
clear all
close all
clc

I=imread('IMG_645.png');
figure;imshow(I)
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);
```

A diferència del codi de detecció de lesions fosques, en aquest no s'utilitza un canal específic per realitzar els processats previs a la binarització. Amb l'estudi mitjançant les eines de la interfície de visualització del MATLAB, es va detectar que els píxels d'aquest tipus de lesió, presentaven una diferència de valors entre el canal vermell i el verd força gran, característica que els feia discriminables de la resta de píxels circumdants. Aquest fet s'aprofita per facilitar la binarització, utilitzant com a imatge base la resultant de restar el canal vermell del verd.



Figura 29: Valor dels 3 canals de color dins la lesió i als seus voltants. Es pot observar com a dins la lesió, la diferència entre el canal vermell i verd és de 104, mentre que als voltants és de 78 i 72 respectivament.

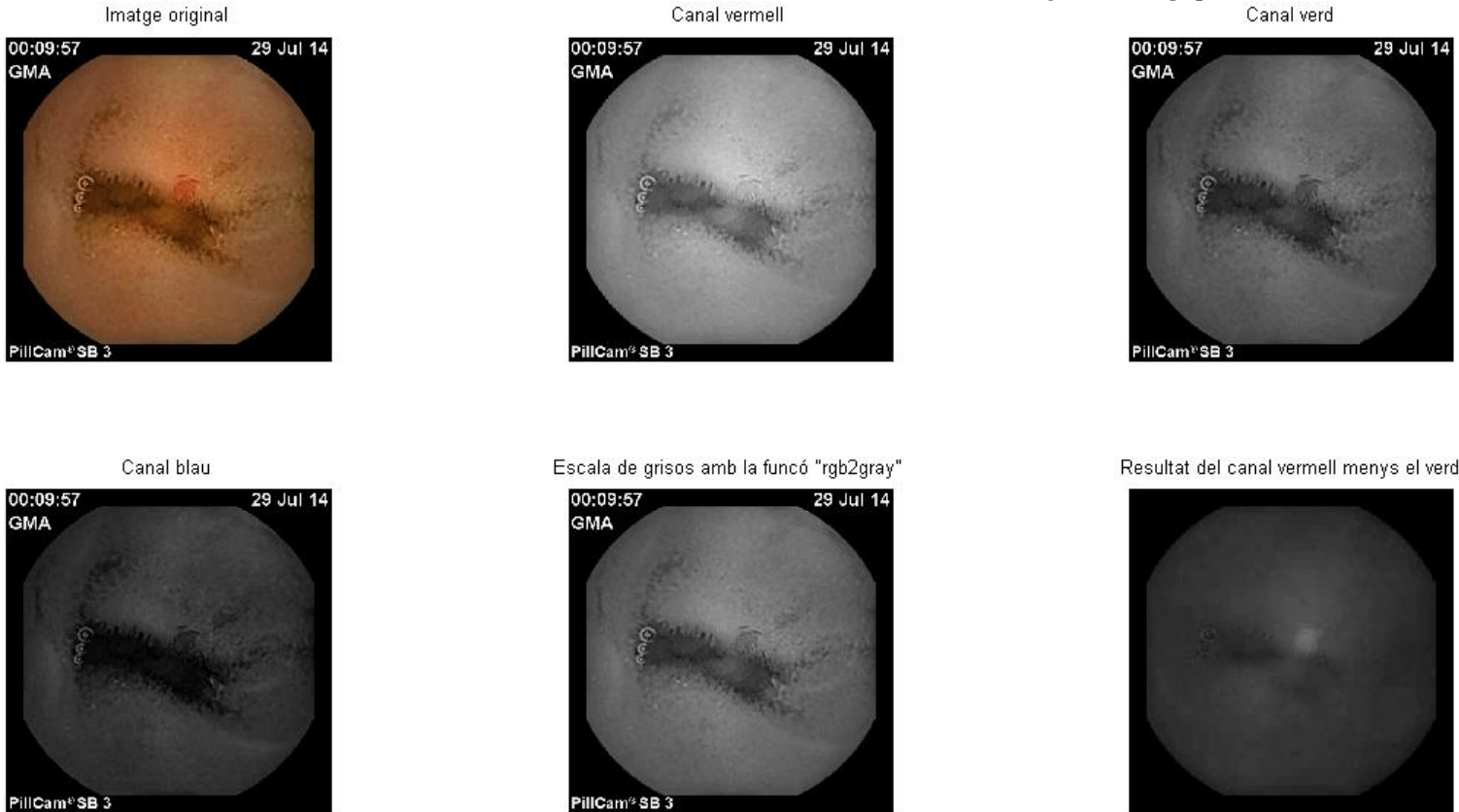


Figura 30: Representació dels diferents canals de color, la transformada a escala de grisos mitjançant la funció "rgb2gray" i el resultat de la resta entre el canal vermell i el verd.

Com es pot observar a la figura 30, en aplicar la resta entre el canal vermell i el verd, la zona on està ubicada la lesió queda ressaltada respecte al seu voltant, ja que els valors resultants són més alts, aquest fet facilita la binarització de les lesions i la seva segmentació.

Taula 18: Càlcul de la resta entre el canal vermell i el canal verd i obtenció de la figura 30 mitjançant la funció "subplot".

```
Q=Ired-Igreen;

figure;
subplot(2,3,1)
imshow(I); title('Imatge original');
subplot(2,3,2)
imshow(Ired); title('Canal vermell');
subplot(2,3,3)
imshow(Igreen); title('Canal verd');
subplot(2,3,4)
imshow(Iblue); title('Canal blau');
subplot(2,3,5)
imshow(IR); title('Escala de grisos amb la funció "rgb2gray"');
subplot(2,3,6)
imshow(Q); title('Resultat del canal vermell menys el verd');
```

Un cop s'ha obtingut la imatge que s'utilitzarà, que en el cas de les lesions fosques era el canal verd i en aquest cas és la resta del canal vermell i el verd, comença el procés de preparació de la imatge abans de binaritzar.

Taula 19: Aplicació de la funció "adapthisteq" i de la funció "medfilt2".

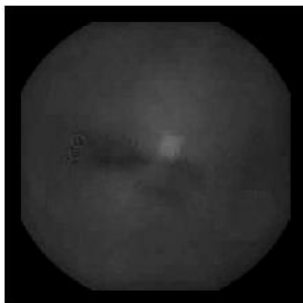
```
Iadpt=adapthisteq(Q);  
I2 = medfilt2(Iadpt,[15 15]);
```

El processament de la imatge abans de la binarització es simplifica en comparació amb les lesions fosques, ja que al restar els dos canals, els píxels que fan referència a la informació sobre el temps de gravació, el nom de la prova, la data, etc. que es pot observar als marges de la imatge, passa a valdre 0 i a formar part del conjunt de píxels negres que conformen el marc exterior de la imatge, d'aquesta manera no és necessari realitzar el filtre morfològic d'erosió per eliminar-los i el posterior filtre de reconstrucció.

Per altra banda, si que s'aplica l'augment de contrast amb la funció "adapthisteq". Amb l'equalització de l'histograma la zona on apareix la lesió està més contrastada respecte als píxels circumdants, facilitant el procés de binarització.

A l'aplicar la funció "adapthisteq" augmenta el soroll de la imatge i es necessita l'aplicació d'un filtre per suavitzar el soroll i possibles nuclis que interfereixin en la binarització mitjançant la funció "medfilt2".

Diferència entre el canal vermell i el verd



Resultat d'aplicar la funció "adapthisteq"

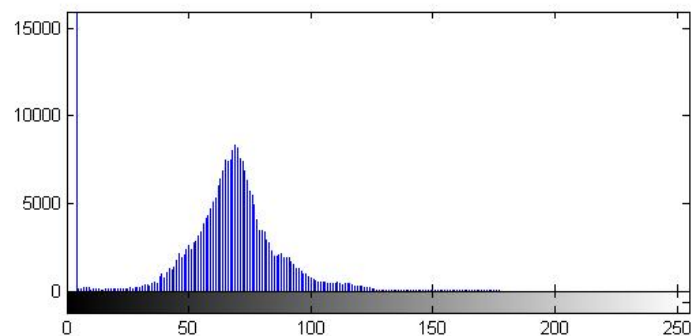
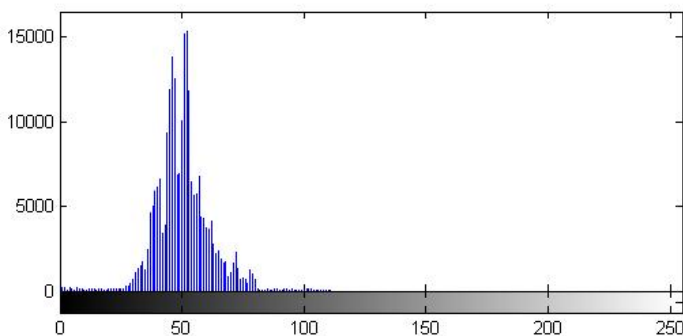
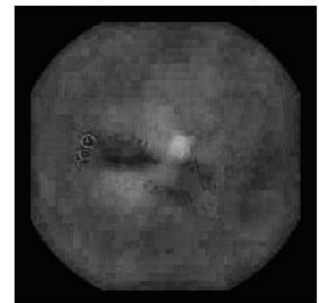


Figura 31: A l'esquerra, imatge resultant de la diferència entre el canal vermell i el verd amb el corresponent histograma. A la dreta, imatge resultant d'aplicar la funció "adapthisteq" sobre la imatge de l'esquerra amb el corresponent histograma.

Com es pot observar a la figura 31, a l'equalitzar la imatge s'obté una suavització de l'histograma resultant amb una millora de la percepció de la zona que correspon a la lesió.

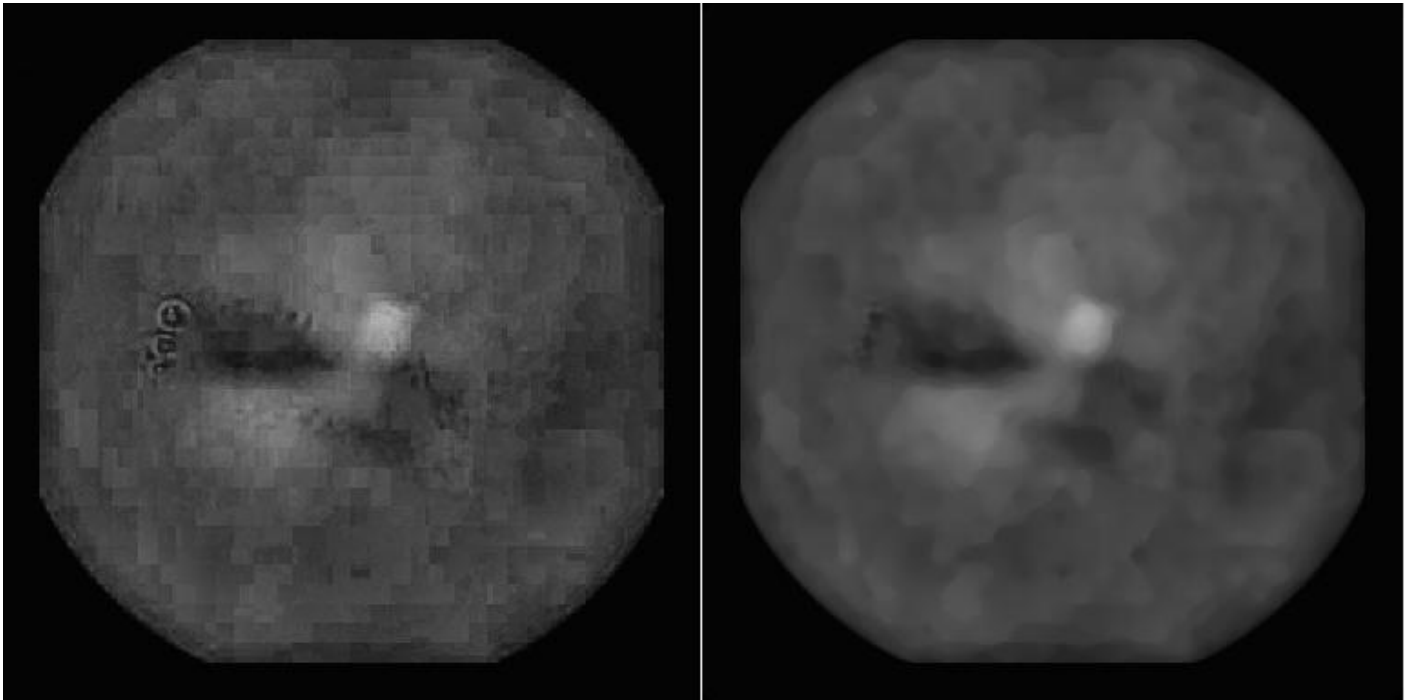


Figura 32: A l'esquerra imatge resultant d'aplicar la funció "adaphisteq". A la dreta imatge resultant després d'aplicar la funció "medfilt2".

Un cop s'obté la imatge processada pel filtre de mediana, s'aplica la binarització sobre la imatge resultant. Per fer-ho s'utilitza la funció "im2bw" com en el cas de les lesions fosques, però en aquest cas el llindar de binarització no es determina mitjançant la funció "graythresh" que utilitza el mètode Otsu per determinar-lo, es fa servir un valor constant obtingut experimentalment, que és vàlid com a llindar per a totes les lesions classificades com a clares. Aquest fet és degut perquè la utilització de la funció "graythresh" no resultava efectiva, ni fent-la més restrictiva com en el cas de les lesions fosques. Per aquest motiu es va buscar un valor llindar, que es pogués utilitzar de manera constant per totes les lesions que es tenien classificades com a clares. El llindar que es va obtenir i que s'utilitza en aquest mètode de detecció és el 0.47843137, expressat al MATLAB com 122/255.

Taula 20: Aplicació de la funció "im2bw" i la visualització de la binarització resultant amb la funció "imshow".

```
Ibin=im2bw(I2,122/255);  
figure;imshow(Ibin);
```

Com podem observar a la figura 33, la binarització resultant és molt acurada, delimitant molt bé la zona on es troba la lesió, per aquest fet no és necessari aplicar un filtre morfològic de tancament per acabar de tancar possibles regions que s'hagin quedat obertes. Aquests tipus de lesions si existeixen, queden molt diferenciades del seu entorn, fet que comporta que no apareguin regions binaritzades obertes, com passava en les lesions fosques.

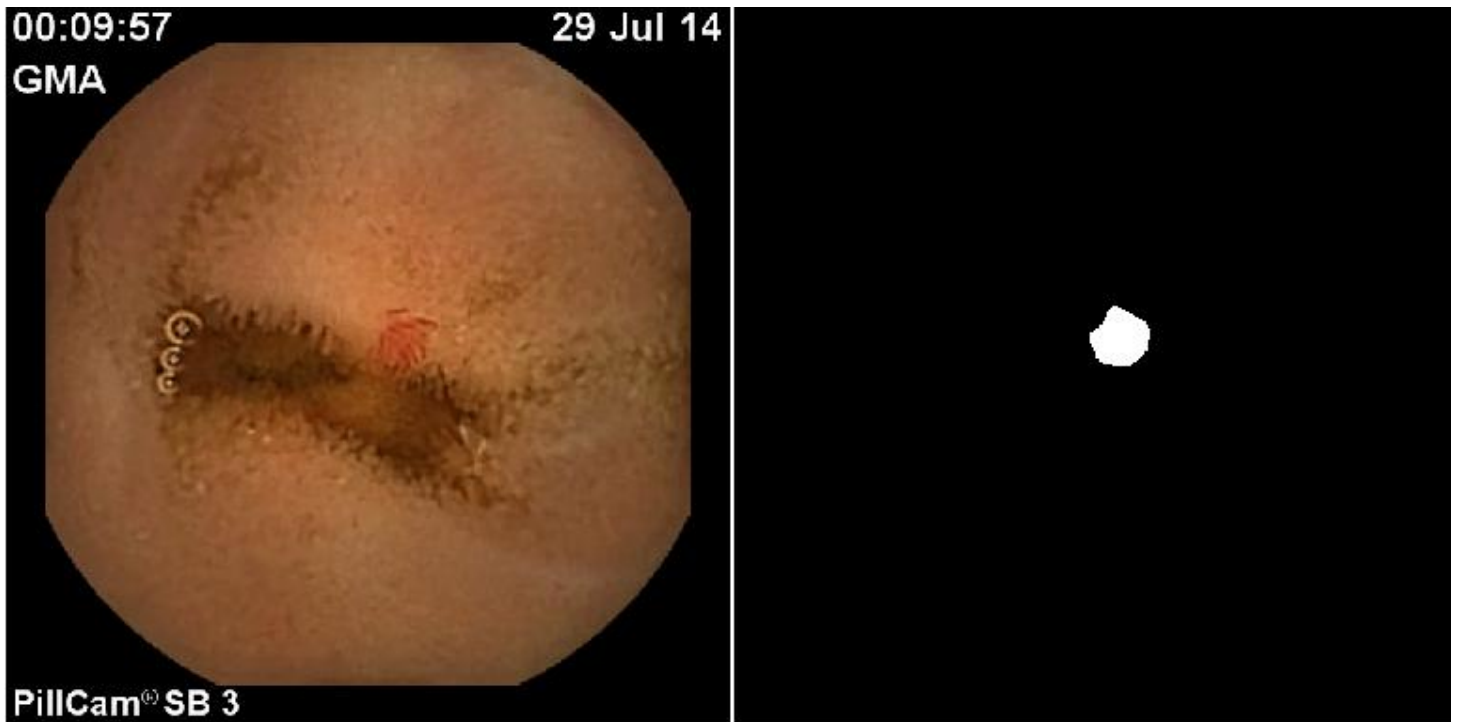


Figura 33: A l'esquerre, imatge original. A la dreta, imatge binaritzada.

Un cop s'ha obtingut la lesió binaritzada comença el procés de segmentació, que comparteixen tant el mètode de detecció de lesions clares, com el mètode de detecció de lesions fosques i que s'exposa en el següent apartat.

3.2.3 Segmentació

El procés de segmentació és clau a l'hora d'etiquetar les regions obtingudes durant la binarització, ja que és possible que apareguin diverses zones juntament amb la que correspon a la lesió. També és important en l'obtenció de propietats de les regions segmentades, pel fet que proporciona una àrea d'extracció de dades acotada a la zona que correspondria a una lesió. Per això és necessari un mètode eficaç i acurat per tal de realitzar correctament la segmentació.

En aquest procés també es realitza el primer filtratge de regions, utilitzant com a llindar l'àrea en píxels de les zones segmentades. Aquest primer filtratge és molt poc restrictiu, ja que només es pretén eliminar les regions massa grans i les molt petites.

Al finalitzar el procés de segmentació, s'obtidrà les regions que estiguin acotades dintre els paràmetres lògics que corresponen a l'àrea d'una possible lesió, diferenciades i etiquetades.

El procés que s'ha seguit per dur a terme la segmentació, ha estat l'obtenció de contorns mitjançant el càlcul de gradients i la utilització de la tècnica "watershed" per l'etiquetatge.

Pel càlcul dels gradients s'ha utilitzat la funció de MATLAB "fspecial", que serveix per crear un filtre de dues dimensions d'un tipus determinat. La funció accepta filtres *average*, *disk* (filtre *average* circular), *gaussian*, *laplacian*, *log*, *motion*, *prewitt*, *sobel*, *unsharp*. Depenent del tipus de filtre que s'especifica, és necessària la incorporació com a variable d'entrada altres factors, com poden ser la mida o el radi. Pel cas que se'ns presenta, s'ha decidit utilitzar un filtre amb operador de tipus "sobel".

G_H

-1	-2	-1
0	0	0
1	2	1

G_V

-1	0	1
-2	0	2
-1	0	1

$$G = \sqrt{G_V^2 + G_H^2}$$

Figura 34: Descripció de l'operador horitzontal i de l'operador vertical, així com el càlcul final del resultat del filtre Sobel.

L'operador Sobel tècnicament és un operador diferencial discret que calcula una aproximació al gradient de la funció d'intensitats d'una imatge. Per cada píxel de la imatge que es processa, el resultat d'aplicar l'operador Sobel és tant el vector gradient corresponent com la norma d'aquest vector.

L'operador Sobel aplicat sobre una imatge en escala de grisos o en blanc i negre, calcula el gradient de la intensitat de brillantor de cada píxel donant la direcció de l'increment més gran possible (de negre a blanc), a més calcula la suma del canvi en aquesta direcció, és a dir, retorna un vector. A la pràctica, l'operador Sobel ens mostra els píxels on detecta un canvi bruscat d'intensitat amb els píxels circumdants, aquest fet s'accentua si la imatge utilitzada és binària (blanc i negre), perquè els canvis són més abruptes que no pas en una imatge en escala de grisos (pas de 0 a 1 o en el cas d'escala de grisos de 0 a 255). Aquest fet comporta a definir el filtre Sobel com un "detector de contorns", ja que els píxels que formen els límits de les regions

que componen una imatge, tenen una marcada diferència entre els valors dels píxels veïns que comporta que es pugui diferenciar una regió d'una altra, per aquest fet l'aplicació del filtre Sobel realça aquestes zones de transició que en definitiva són les fronteres de les regions de la imatge.

Si ho definim de manera matemàtica, el gradient d'una funció de dues variables (funció intensitat d'una imatge) per a cada punt, és un vector bidimensional on els components estan donats per les primeres derivades de les direccions verticals i horitzontals. Per a cada punt de la imatge, el gradient del vector apunta en direcció de l'increment màxim possible d'intensitat, i la magnitud del gradient del vector correspon a la quantitat de canvi d'intensitat en aquesta direcció. Això implica que el resultat d'aplicar l'operador Sobel sobre una regió d'una imatge amb intensitat de brillantor constant és un vector zero, i el resultat d'aplicar-lo en un punt sobre una vora és un vector que apunta creuant la vora (perpendicular) en sentit dels punts més foscos cap als més clars.

Per aquesta raó, l'operador Sobel utilitza dos nuclis de 3x3 elements per aplicar la convolució a la imatge original per calcular les aproximacions a les derivades. És necessari un nucli pels canvis horitzontals i un altre pels verticals i els resultats de les aproximacions dels gradients horitzontals i verticals poden ser combinats per obtenir la magnitud del gradient com es pot observar a la figura 34.

Taula 21: Càlcul de gradients.

```
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ibin), hy, 'replicate');
Ix = imfilter(double(Ibin), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag);
```

Com es pot observar a la taula 21, primer es defineix l'operador que s'utilitzarà, en el nostre cas de tipus Sobel, amb la funció "fspecial". Aquesta ens retorna només l'operador vertical, per això tot seguit es calcula l'operador horitzontal fent la transposada de l'operador que s'ha obtingut, perquè com s'ha explicat abans, per tenir el resultat final és necessari el càlcul del mòdul entre l'operador horitzontal i el vertical.

Amb la funció "imfilter", que serveix per aplicar un filtre multidimensional a un vector multidimensional, apliquem els dos operadors Sobel i obtenim el resultat final calculant el mòdul dels dos vectors resultants. Afegint l'ordre "replicate", es defineix el tracte que reben els vectors que es troben fora dels límits del vector operador, que en aquest cas es consideren del mateix valor que del vector més proper dins dels límits de l'operador.

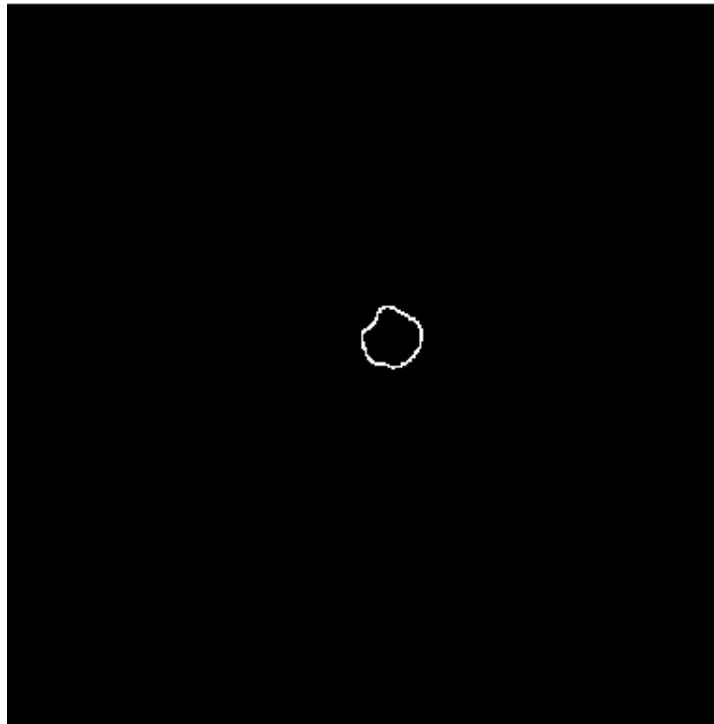
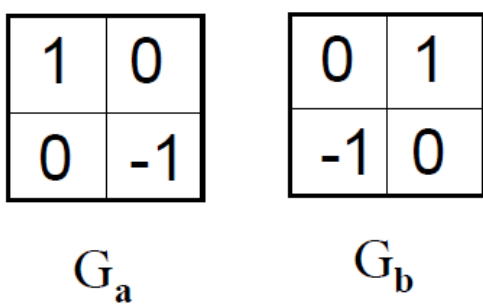


Figura 35: Imatge de gradients

La utilització com a detector de contorns l'operador Sobel es va mantenir des del principi del desenvolupament del codi pels bons resultats que s'obtenien. L'elecció en primera instància d'aquest tipus d'operador envers d'altres com poden ser el laplacian, el tipus Prewitt o Roberts ha estat que en general l'aplicació de l'operador Sobel dona més bons resultats que els altres.

Per començar, respecte al filtre Roberts, pel fet que l'operador Sobel té una regió d'actuació més gran (3x3 del tipus Sobel respecte al 2x2 de l'operador Roberts) la sensibilitat i la definició són majors en l'operador de tipus Sobel.



$$G = \sqrt{G_a^2 + G_b^2}$$

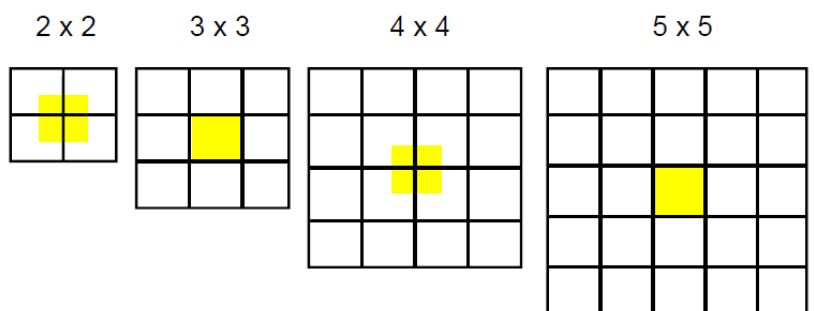


Figura 36: Operadors Roberts, a diferència del Sobel actuen sobre la imatge en diagonal.

Figura 37: Com més gran és l'operador, més píxels s'analitzen, fet que comporta més sensibilitat, més definició i més temps de computació.

Pel que fa al filtre laplacià s'ha de definir l'operador, fet que pot comportar un augment del temps de computació si s'utilitza un operador massa gran, amb això no detecta de manera directe els contorns sinó que és necessari sumar la imatge original amb la imatge filtrada per obtenir un realçament dels contorns i el filtratge utilitzant Sobel és més directe. Aquest fet és degut a que la morfologia de l'operador laplacià és diferent de la dels operadors de tipus Roberts, Prewitt i Sobel i s'aproxima més, a la morfologia d'un filtre passa alt, com es pot comprovar en les figures 38 i 39.

1D [1 -2 1]

2D $\begin{bmatrix} 0 & 1 & 0 \\ -1 & 4 & -1 \\ 0 & 1 & 0 \end{bmatrix}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

-1	0	-1
0	+5	0
-1	0	-1

Figura 39: Operador laplacià d'una i dues dimensions.

Figura 38: Operador d'un filtre passa alt.

Per últim la diferència entre Prewitt i Sobel radica en l'operador, mentre que el Prewitt es compon de zeros i uns, el Sobel incorpora un 2 com es pot veure a la figura 34, aquest fet comporta una millora en la qualitat dels contorns que s'obtenen. Amb Prewitt també és necessari el càlcul del mòdul entre l'operador vertical i l'horitzontal com es pot observar a la figura 40.

-1	-1	-1
0	0	0
1	1	1

G_H

-1	0	1
-1	0	1
-1	0	1

G_V

$$G = \sqrt{G_V^2 + G_H^2}$$

Figura 40: Operadors Prewitt.

Un cop s'obté la imatge de gradients, amb els contorns marcats, s'etiqueta la imatge amb la tècnica de watershed. Es va valorar l'opció d'etiquetar directament la imatge binaritzada sense la necessitat del càlcul de gradients, amb la funció "bwlable". La funció "bwlable" retorna una matriu de la mateixa mida que la imatge binaritzada, que conté les etiquetes de tots els valors connectats en la imatge binària amb valor 1.

Aquesta opció es va descartar, ja que aquesta tècnica requereix que les lesions estiguin contingudes en regions amb valor de píxel 1. Amb aquest mètode es necessitava una transformació morfològica prèvia de càlcul de la imatge complement amb la funció de MATLAB "imcomplement", a l'aplicació d'aquesta funció en el cas de les lesions fosques, ja que les regions quedaven en zones de píxels de valor 0, en canvi per les lesions clares no era necessari, perquè les lesions quedaven en regions amb píxels de valor 1.

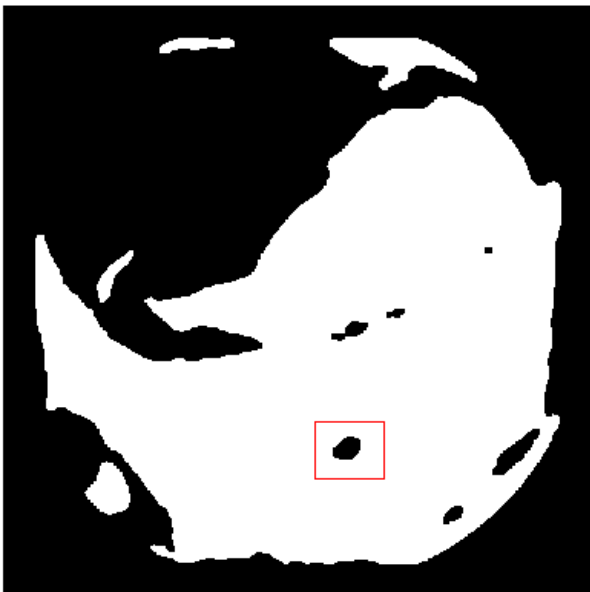


Figura 41: Dins el requadre vermell, s'observa una lesió fosca representada per píxels negres.



Figura 42: Dins el requadre vermell, s'observa una lesió clara representada per píxels blancs.

Com que d'entrada es va plantejar un model de segmentació compatible pels dos tipus de lesions, es va decidir que el càlcul dels gradients i el posterior etiquetatge amb el mètode de watershed era la millor solució per evitar errors en aquest aspecte.

Taula 22: Execució de la funció "watershed2 sobre la imatge de gradients i visualització de les regions etiquetades amb diferents colors mitjançant la funció "imagesc".

```
L = watershed(gradmag);  
  
figure;imagesc(L);
```

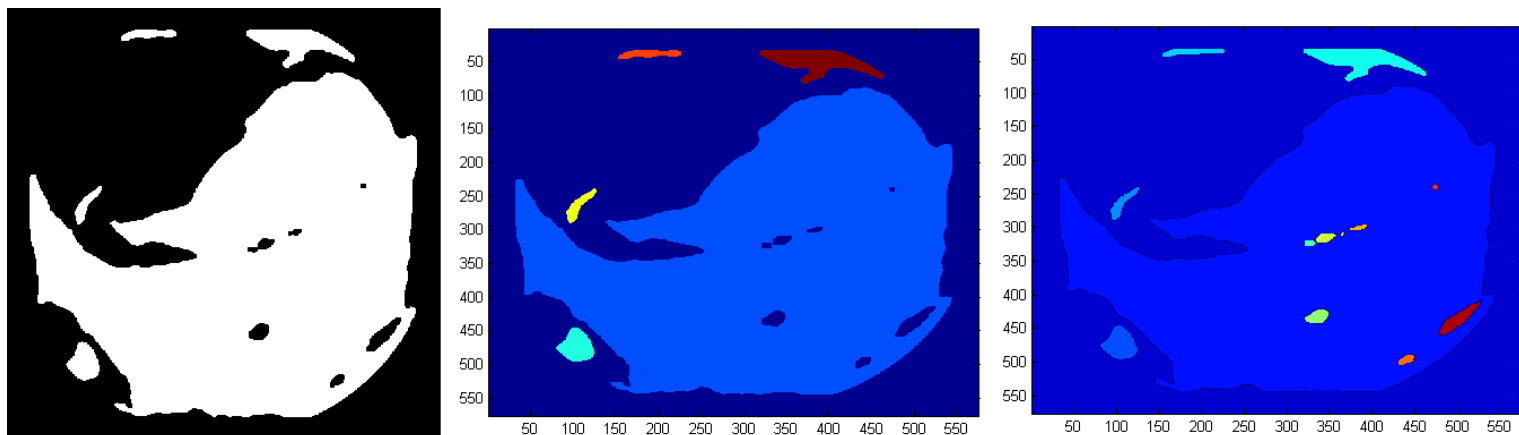


Figura 43: D'esquerre a dreta: Imatge binaritzada, etiquetatge amb "blabel" i etiquetatge amb "watershed". Es pot observar com la funció "blabel" confon les regions en negre amb el fons de la imatge etiquetant-les de la mateixa manera i com la funció "watershed" etiqueta cada regió de manera diferent.

Per entendre com funciona la funció watershed es pot imaginar una imatge en escala de grisos com un relleu topogràfic, on s'interpreta el nivell de gris dels píxels com la seva altura en el relleu (figura 44). Una gota d'aigua que cau sobre un relleu topogràfic flueix al llarg d'un camí per finalment arribar a un mínim local. La funció watershed interpreta la longitud del gradient com informació de relleu i actua com aquesta gota d'aigua, fent el recorregut entre els "pics de relleu" descrivint els límits de les regions.

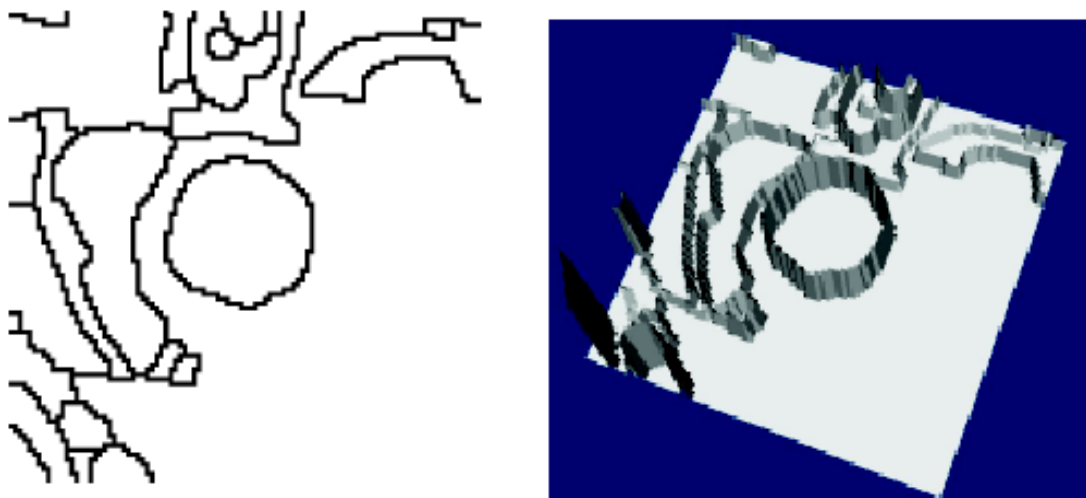


Figura 44: A l'esquerra una imatge de gradients, a la dreta com interpreta el valor dels píxels de la funció watershed si es visualitza en volum.

El resultat després d'aplicar la funció "watershed" a la imatge de gradients, és l'obtenció d'una sèrie de regions ordenades per etiquetes, com es pot observar en la figura 43 i més clarament a la figura 45.

El problema de la segmentació utilitzant el mètode watershed, és que també interpreta el fons de la imatge com una regió a segmentar, per això és necessari aplicar un filtre d'àrees, per eliminar les regions excessivament grans que normalment fan referència al fons de la imatge o a zones amb

valors de píxel constant, que és impossible que facin referència a una lesió. De la mateixa manera que es fa un filtratge de les regions massa grans, s'aprofita i també es filtren les regions massa petites per correspondre a una lesió.

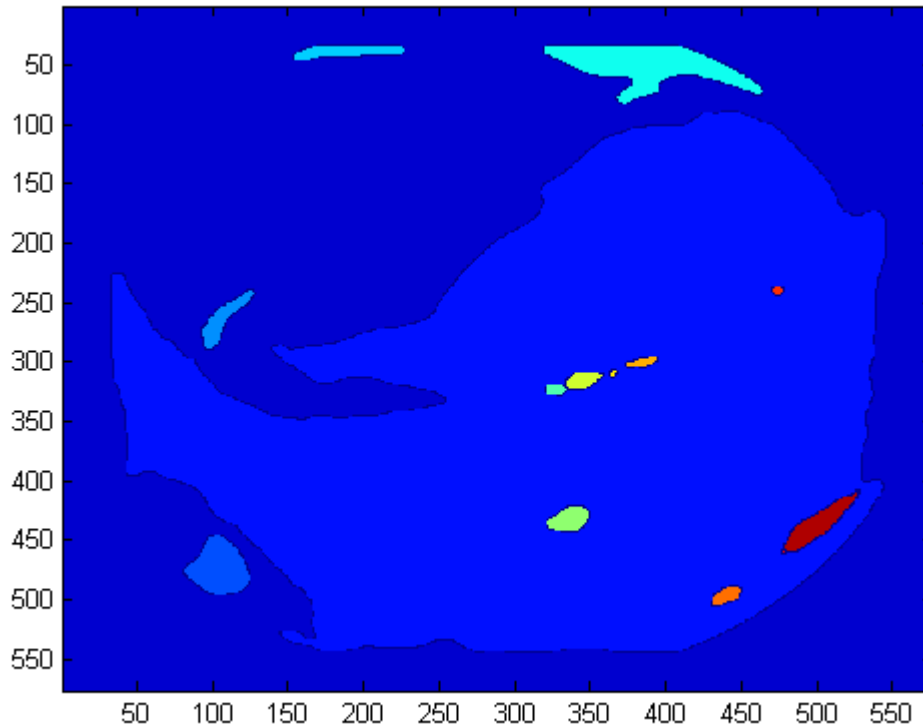


Figura 45: Regions etiquetades per la funció "watershed".

Per la determinació dels l·lindars es va consultar amb el Dr. Marco Antonio i va dictaminar que pel que fa a angiодisplàsies petites, una possible lesió continguda en una regió de 10 píxels d'àrea, ja era difícil de determinar si realment era una angiодisplàsia o una ferida temporal i en referència a les lesions més grans, va determinar que poques vegades superen el valor d'àrea màxim que s'havia obtingut de les imatges amb lesions de les quals es disposava. Així doncs es va optar per determinar l·lindars que abastessin un gran nombre de regions, essent el l·lindar màxim 10000 píxels (10 vegades superior a l'àrea màxima) i el mínim 4 píxels.

Per la determinació de l'àrea en píxels de les regions etiquetades s'utilitza la funció de MATLAB "regionprops", que pot calcular diverses propietats de les regions com el centroide, la longitud de l'eix màxim i el mínim, l'àrea, etc.

Per l'eliminació de les regions, primer es busquen les regions que no arriben al l·lindar mínim o superen el màxim amb la funció "find" i s'assignen a un vector, després mitjançant un bucle s'assigna valor 0 a totes les regions que no estan dintre dels l·lindars establerts, com es pot observar a la taula 23.

Taula 23: Determinació del valor de les àrees de les regions, assignació dels llindars i eliminació de les regions que no estan dintre els llindars amb el bucle "for".

```
Lprops = regionprops(L, 'Area');  
  
    maxlimit=10000;  
    minlimit=4;  
  
    AreasSim = find([Lprops.Area] < minlimit | Lprops.Area]>maxlimit);  
  
    E = L;  
    for etiqueta = AreasSim  
        E(L == etiqueta) = 0;  
    end
```

Al final d'aquest procés és necessari tornar a calcular la imatge gradients i aplicar la funció "watershed", per obtenir finalment les regions que han passat el filtratge com s'observa a la figura 46. Tècnicament no s'eliminen les regions que no passen el filtre, sinó que s'assigna valor 0 als píxels corresponents a les regions que no superen els llindars, per tal de no ser detectades en un segon càlcul de contorns i etiquetat.

Taula 24: Segon càlcul de la imatge de gradients i de l'aplicació de la funció "watershed" per tal d'obtenir les regions que finalment s'analitzaran.

```
hy2 = fspecial('sobel');  
  
    hx2 = hy2';  
    Iy2 = imfilter(double(E), hy2, 'replicate');  
    Ix2 = imfilter(double(E), hx2, 'replicate');  
    gradmag2 = sqrt(Ix2.^2 + Iy2.^2);  
  
    L2=watershed(gradmag2);  
    figure;imagesc(L2)
```

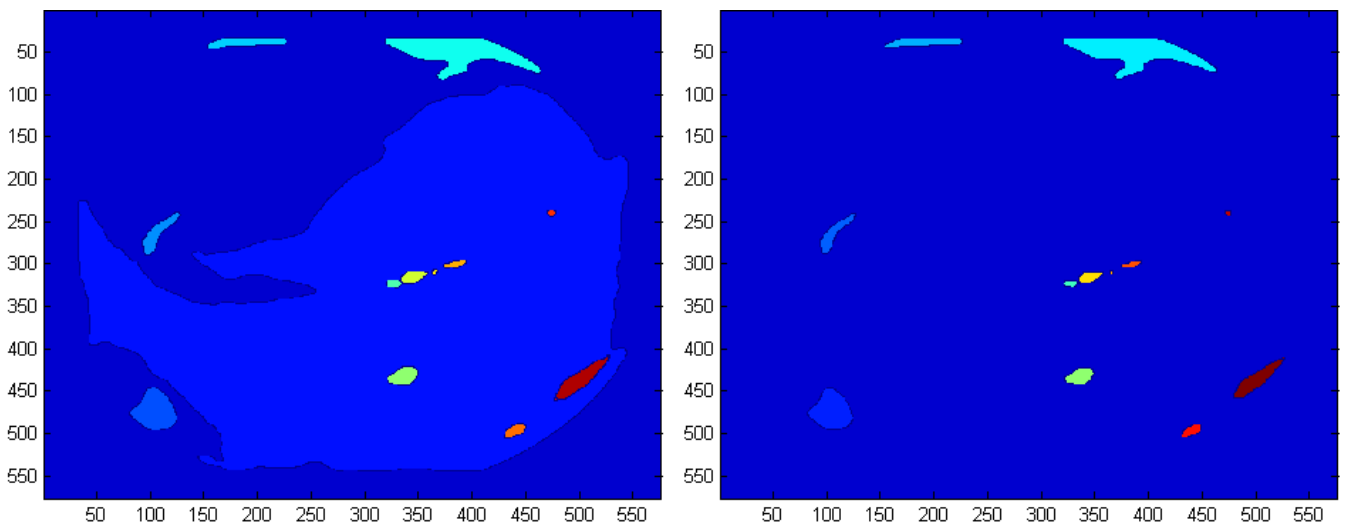


Figura 46: A l'esquerra regions abans del filtratge, a la dreta regions resultants.

El que s'obté finalment és una matriu que conté tots els píxels de la imatge de gradient etiquetats per regions. Per tal d'obtenir un vector, que calculant la seva longitud es pugui determinar el nombre de regions, es torna a aplicar la funció "regionprops" per torbar el valor de les àrees ordenat per regions, així doncs el vector que conté el valor de les àrees tindrà la mateixa longitud que el nombre de regions etiquetades.

Taula 25: Segon càlcul de les àrees de les regions obtingudes.

```
L2props=regionprops(L2, 'Area');
```

3.2.4 Estudi de propietats i establiment de lindars

Amb l'obtenció del vector que conté el valor de les àrees de les regions que s'han filtrat en última instància, acaba la part del codi de segmentació i etiquetatge de les regions d'interès.

La metodologia emprada per l'estudi de les propietats i l'establiment de lindars és la mateixa per ambdós casos de lesions, tant les fosques com les clares. La diferència radica en els valors obtinguts per l'establiment de lindars, que com ja es comentarà en aquesta secció, són diferents en els dos casos de lesió. El procés consta de diversos passos a seguir abans de l'obtenció dels lindars òptims.

Les primeres propietats que es van estudiar, estan relacionades amb els valors d'intensitat de cada canal, ja que en el primer estudi, quan s'havia de decidir quina imatge utilitzar per al procés de binarització, es va detectar que les lesions presentaven certes característiques en els nivells d'intensitat dels seus píxels que les feien específiques de les zones lesionades. També es van estudiar propietats físiques de les regions i de textura.

Com que no es disposava en tots els casos, únicament de la lesió segmentada i etiquetada com es pot observar a la figura 46, va ser necessari un estudi manual de les propietats de les lesions.

El primer pas per l'anàlisi de les regions era determinar quina etiqueta fa referència a una regió que correspongui a una lesió. Utilitzant la interfície d'usuari que el MATLAB facilita, es pot determinar el valor de l'etiqueta. Utilitzant la mateixa eina que s'ha fet servir per estudiar els valors d'intensitat dels píxels que formaven l'estructura de la lesió anomenada "data cursor", se selecciona la regió que interessa i determinem quina etiqueta la caracteritza, com es pot observar a la figura 47.

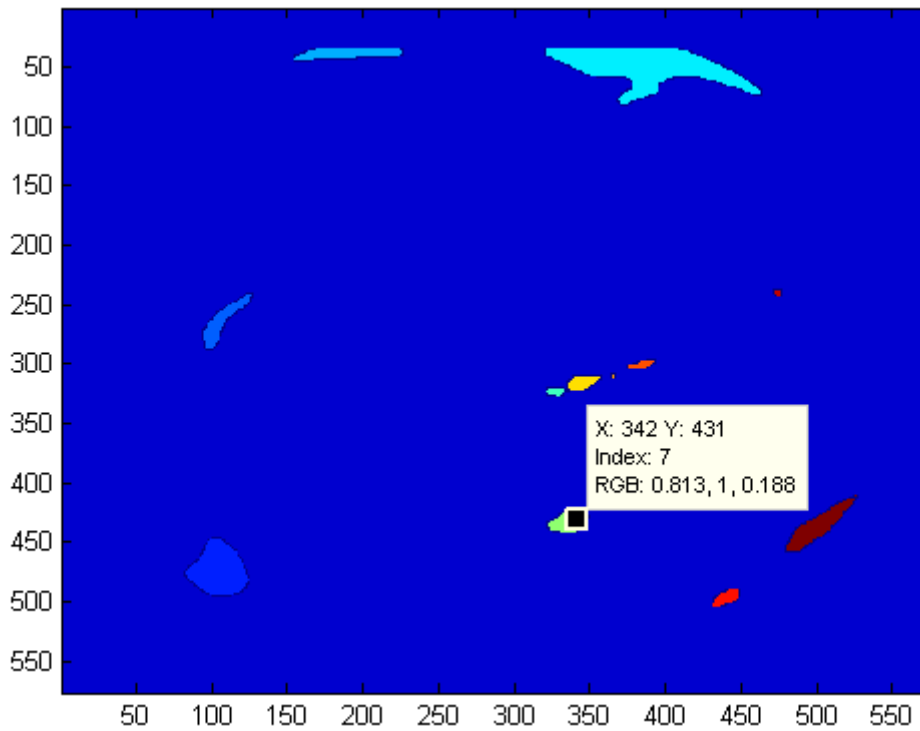


Figura 47: Obtenció del valor de l'etiqueta, en aquest cas 7.

Un cop es té determinada l'etiqueta de la lesió, s'estudia la morfologia que presenta l'histograma que representa els valors d'intensitat que tenen els diferents canals de color. A les figures 48, 49 i 50 es pot observar com els histogrames dels diferents canals, presenten una distribució semblant a una gaussiana.

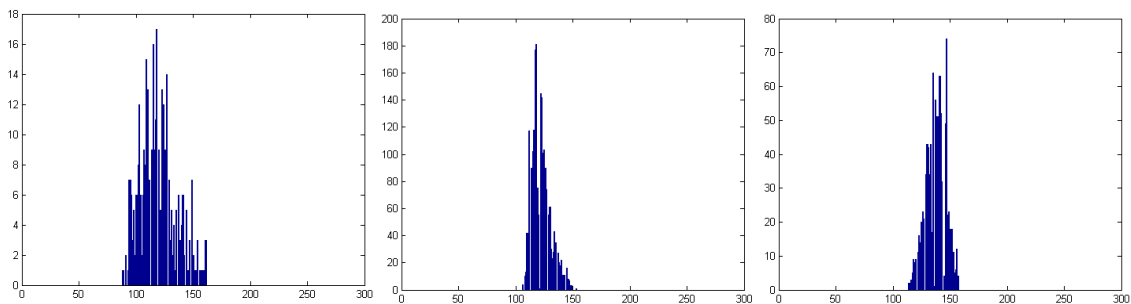


Figura 48: Histogrames canal vermell.

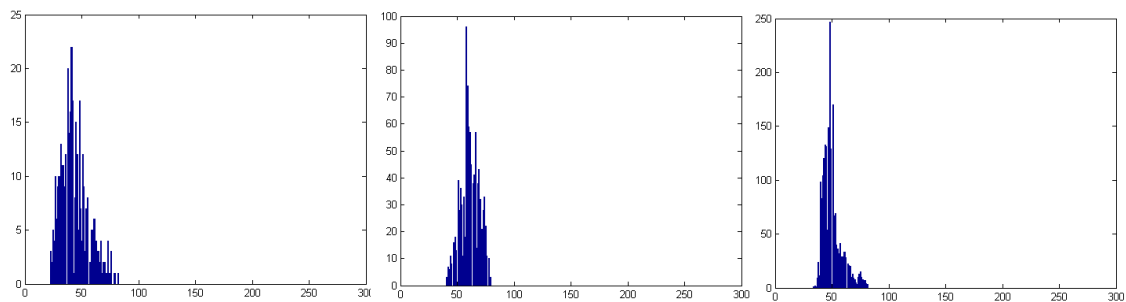


Figura 49: Histogrames canal verd.

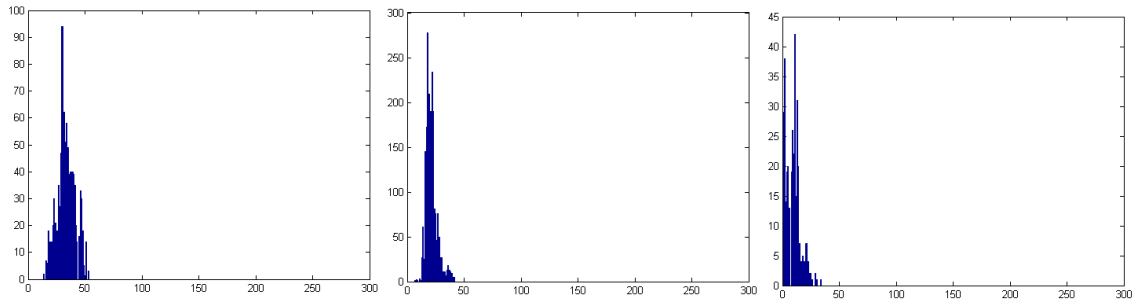


Figura 50: Histogrames del canal blau.

Es pot determinar que les intensitats de píxel dintre de les lesions mantenen un comportament similar a una distribució gaussiana, amb una desviació i una mitjana semblants a simple vista, sempre que pertanyessin al mateix grup de lesions. Amb l'estudi dels histogrames, es va fer patent la diferència entre els valors que formaven les intensitats, però es va detectar que el patró que presentava la distribució de les intensitats en el valor dels píxels era similar en els dos casos de lesions, deixant de banda el valor dels píxels.

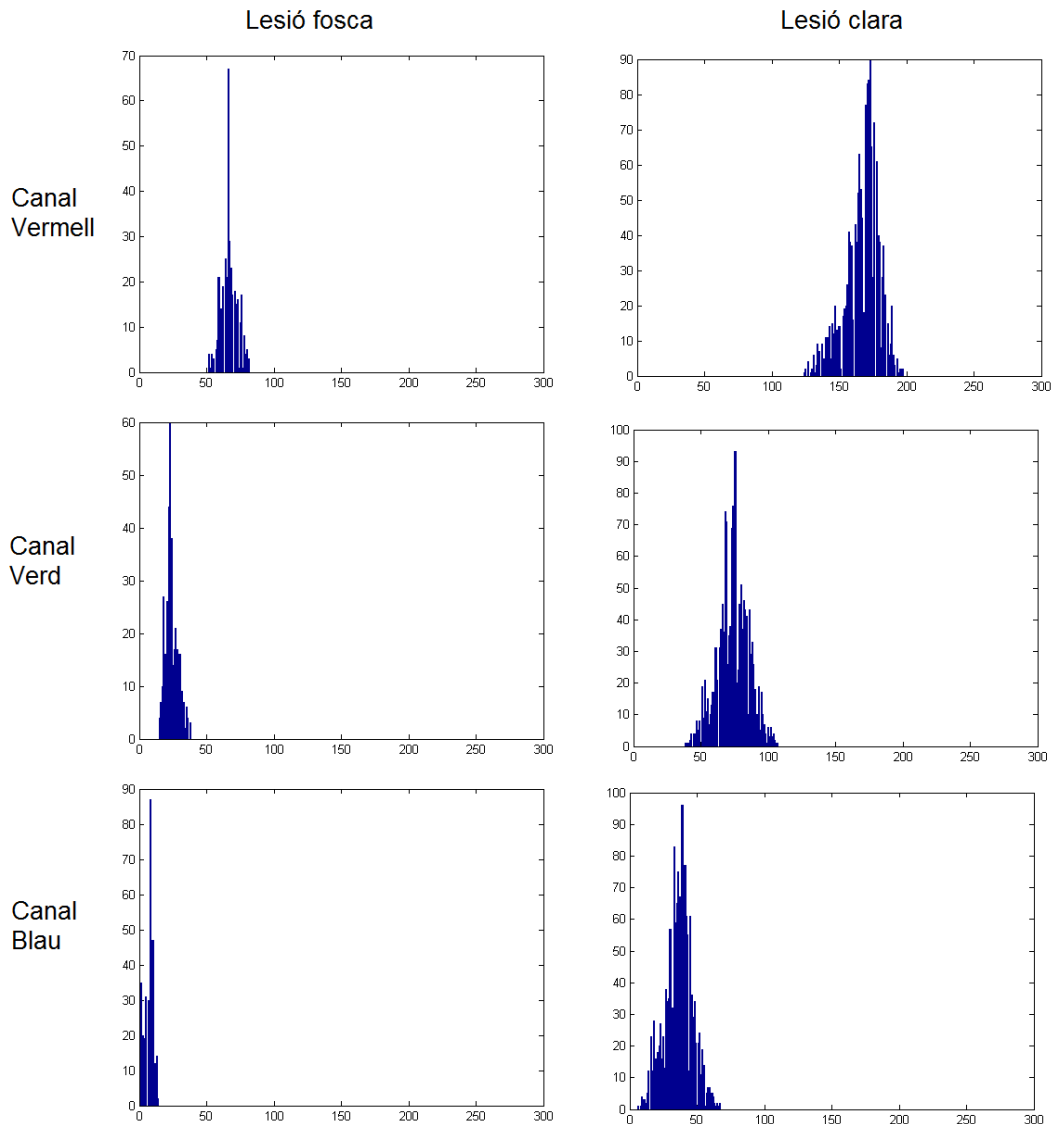


Figura 51: Comparació entre una lesió fosca i una clara.

Com es pot comprovar a la figura 51, els valors que conformen els histogrames són força diferents entre lesions clares i fosques, per aquest fet es determinen llindars distints pels dos tipus de lesions, ja que d'aquesta manera es poden obtenir llindars més restrictius per a cada tipus de lesió.

Taula 26: Codi emprat per la representació dels histogrames, on "q" és el valor de l'etiqueta de la regió que es vol analitzar.

```
vec=Ired(L2==q);
vec2=Igreen(L2==q);
vec3=Iblue(L2==q);

edg=1:1:255;

h=histc(vec,edg);
h1=histc(vec2,edg);
h2=histc(vec3,edg);

figure;bar(edg,h);
figure;bar(edg,h1);
figure;bar(edg,h2);
```

Un cop s'ha comprovat per segona vegada les diferències entre propietats de les lesions, però les semblances en el patró de distribució de les intensitats dels píxels (tant les lesions clares com les fosques presenten patrons de distribució gaussians), s'analitzen els valors que defineixen una distribució gaussiana, els seus límits obtinguts mitjançant el càlcul del màxim i el mínim, la seva desviació estàndard i la mitjana, ja que presenten patrons regulars en les regions que corresponen a lesions per a cada canal.

Taula 27: Funcions de MATLAB per calcular el màxim, la desviació estàndard, la mitjana i el mínim, d'un vector anomenat "vec" determinat.

```
A=max(vec);
B=std2(vec);
C=mean2(vec);
D=min(vec);
```

Es descarta utilitzar el nombre de píxels que tenen una mateixa intensitat o el nombre de píxels que conformen una lesió, ja que les mides de les lesions són molt variables i no és possible determinar un llindar que pugui suposar una restricció significativa.

Altres característiques que es van decidir estudiar per si apareixien patrons regulars entre les regions, van ésser les relacionades amb la textura. Així com a simple vista es pot determinar que les zones on hi apareixen lesions tenen intensitats de píxel característiques, també s'observa com la textura en la zona de la lesió sembla presentar particularitats pròpies. Les variables que es van decidir estudiar van ser l'entropia, el contrast, la correlació, l'energia i l'homogeneïtat. L'únic valor que va presentar un patró regular entre diferents lesions va ser l'entropia.

L'entropia és la mesura estadística de l'aleatorietat que es pot emprar per caracteritzar la textura d'una imatge determinada.

Taula 28: *Funció de MATLAB per calcular l'entropia en el canal vermell en una regió determinada amb valor d'etiqueta "q".*

```
Q_R=entropy(Ired(L2==q));
```

Aprofitant la descoberta feta en l'apartat 2.2.2, on es veu que la diferència entre canals és un valor significatiu en les regions que presenten lesions, també s'utilitza el resultat de la diferència entre el canal vermell amb el verd, el canal vermell amb el blau i el canal verd amb el blau, com a característiques pròpies que apareixen en les lesions. Per simplificar el càlcul s'utilitza com a valor significatiu la diferència entre el valor màxim de cada canal.

També es va utilitzar una propietat física com a valor per determinar uns certs llandars. Es va observar en els processos de segmentació, que les lesions tendien a formes ovalades o gairebé circulars, aquest fet comportava que el coeficient entre el valor de l'eix màxim i el valor de l'eix mínim hauria de tendir a valors baixos pròxims a 1. Amb la determinació d'aquest coeficient es podrien descartar lesions amb formes allargades produïdes per les ombres que projecten els plecs de l'intestí prim, que són una de les grans fonts de falsos positius.

Per la determinació dels valors de l'eix màxim i mínim, s'utilitza la funció ja comentada "regionprops", sobre l'última imatge segmentada, especificant com a valor d'entrada que es vol obtenir el *major axis length* i el *minor axis length*, com es pot observar a la taula 29.

Taula 29: *Determinació de l'eix de valor mínim i màxim d'una regió amb etiqueta "q" i càlcul del coeficient entre ells.*

```
majoraxis=regionprops(L2==q, 'MajorAxisLength');  
minoraxis=regionprops(L2==q, 'MinorAxisLength');  
  
majoraxisL=majoraxis.MajorAxisLength;  
minoraxisL=minoraxis.MinorAxisLength;  
  
mamincoef=majoraxisL/minoraxisL;
```

La mostra d'imatges amb lesions que s'utilitza per determinar els valors canònics per determinar els llandars és de 18, ja que no es disposava de més imatges on es representessin lesions.

Per determinar totes les característiques de manera automàtica, es va dissenyar un algorisme que aprofitava el coneixement que s'havien obtingut prèviament en l'estudi de les regions, de les etiquetes específiques de cada lesió per calcular els valors de les característiques que es volen utilitzar com a valors llandar.

El procediment constava de dos passos que es reiteraven per a cada imatge amb lesió: 1r executar la funció per l'anàlisi d'una imatge que tingués una lesió fosca o clara amb l'algorisme pertinent. 2n executar el codi de càlcul dels valors que s'utilitzaran com a valor llindar, aprofitant les variables obtingudes en l'aplicació de l'algorisme anterior i el coneixement del valor de l'etiqueta que representa la regió on hi apareix la lesió.

Taula 30: Càlcul dels valors que s'utilitzaran com a llindar, la "q" representa el número d'etiqueta que fa referència a la regió d'interès.

```
close all;
clc;

q=7;

vec=Ired(L2==q);
edg=1:1:255;
h=histc(vec,edg);

A=max(vec);
B=std2(vec);
C=mean2(vec);
D=min(vec);

Q_R=entropy(Ired(L2==q));
Q_G=entropy(Igreen(L2==q));
Q_B=entropy(Iblue(L2==q));

vecg=Igreen(L2==q);
Ag=max(vecg);
Bg=std2(vecg);
Cg=mean2(vecg);
Dg=min(vecg);

vecb=Iblue(L2==q);
Ab=max(vecb);
Bb=std2(vecb);
Cb=mean2(vecb);
Db=min(vecb);

dif_r_g=(A-Ag);
dif_r_b=(A-Ab);
dif_g_b=(Ag-Ab);

majoraxis=regionprops(L2==q,'MajorAxisLength');
minoraxis=regionprops(L2==q,'MinorAxisLength');

majoraxisL=majoraxis.MajorAxisLength;
minoraxiL=minoraxis.MinorAxisLength;
mamincoef=majoraxisL/minoraxiL;
```

Cada cop que s'obtenien els valors s'anotaven per obtenir una base de dades dels resultats obtinguts i poder determinar els llindars amb posterioritat.

En una primera versió del codi, es van establir llindars molt restrictius, d'acord amb els valors obtinguts de la mostra que es va emprar per calcular-los.

Per determinar el llindar del valor màxim d'intensitat es va utilitzar el valor màxim de tots els valors obtinguts, i es va establir que les regions no poguessin superar aquest valor.

Pel llindar del valor mínim, es va utilitzar el mínim dels valors obtinguts, i es va establir que les regions havien de superar aquest valor.

Pel cas de la desviació estàndard i la mitjana es van seleccionar els valors màxims i mínims i es va determinar que les regions havien d'estar entre aquests dos valors.

Amb l'entropia es va seguir el mateix procediment que pel cas de la desviació estàndard i la mitjana, seleccionant els valors màxims i mínims i determinant que les regions havien d'estar entre aquests dos valors de llindar.

Finalment amb els valors de les diferències, es va optar per utilitzar el valor mínim i determinar que les regions havien de superar-lo, ja que com es va observar, el resultat de la diferència entre canals a les regions on apareixien lesions era, en la majoria dels casos, superior a la dels píxels que no pertanyien a la lesió.

Per la implementació dels llindars a l'algorisme, s'ha optat per utilitzar una escala de bucles "if", d'aquesta manera es pretén anar filtrant les regions per fases separades. Amb aquest sistema és més fàcil la detecció d'errors, ja que et permet col·locar indicadors per saber en quin punt del filtratge es produeix una incidència.

L'escala de bucles "if" es troba inclosa en un bucle "for", que utilitzant la longitud del vector on es troben les àrees de les regions, aplica el filtratge per cada zona etiquetada, com es pot observar a la taula 31 i 32.

Taula 31: Bucle "for" i escala de bucles "if" per dur a terme el filtratge amb els llindars obtinguts pel cas de les lesions clares.

```
for q=1:length(L2props)

vec=Ired(L2==q);
edg=1:1:255;
h=histc(vec,edg);

A=max(vec);
B=std2(vec);
C=mean2(vec);
D=min(vec);

if A<225 & B>5.6 & B<18.6 & C>112.5 & C<181.4 & D>99.9

Q_R=entropy(Ired(L2==q));
Q_G=entropy(Igreen(L2==q));
Q_B=entropy(Iblue(L2==q));
```

```
if Q_R<5.9 & Q_R>3.99 & Q_G>4.3 & Q_G<6 & Q_B>3.9 & Q_B<5.8

vecg=Igreen(L2==q);
Ag=max(vecg);
Bg=std2(vecg);
Cg=mean2(vecg);
Dg=min(vecg);

if Ag<128 & Bg>5.8 & Bg<20.8 & Cg>41.2 & Cg<94.2 & Dg>29

vecb=Iblue(L2==q);
Ab=max(vecb);
Bb=std2(vecb);
Cb=mean2(vecb);
Db=min(vecb);

if Ab<80 & Bb<16.6 & Bb>4.6 & Cb>16.6 & Cb<42.3

dif_r_g=(A-Ag);
dif_r_b=(A-Ab);
dif_g_b=(Ag-Ab);

if dif_r_g>67 & dif_r_b>98 & dif_g_b>25

figure;bar(edg,h);
q
end
end
end
end
end
end
```

Taula 32: Bucle "for" i escala de bucles "if" per dur a terme el filtratge amb els llindars obtinguts pel cas de les lesions fosques.

```
for q=1:length(L2props)

vec=Ired(L2==q);
edg=1:1:255;
h=histc(vec,edg);

A=max(vec);
B=std2(vec);
C=mean2(vec);
D=min(vec);

if A<174 & B>5.92 & B<15.5 & C>65.2 & C<148.7 & D>51

Q_R=entropy(Ired(L2==q));
Q_G=entropy(Igreen(L2==q));
Q_B=entropy(Iblue(L2==q));

if Q_R<5.6 & Q_R>3.8 & Q_G>3.5 & Q_G<5.3 & Q_B>2.9 & Q_B<4.2

vecg=Igreen(L2==q);
Ag=max(vecg);
```

```
Bg=std2(vecg);
Cg=mean2(vecg);
Dg=min(vecg);

if Ag<83 & Bg>4.4 & Bg<12.1 & Cg>22.2 & Cg<51.6 Dg>10

vecb=Iblue(L2==q);
Ab=max(vecb);
Bb=std2(vecb);
Cb=mean2(vecb);
Db=min(vecb);

if Ab<36 & Bb>2.6 & Bb<6.2 & Cb>4.2 & Cb<18.9

dif_r_g=(A-Ag);
dif_r_b=(A-Ab);
dif_g_b=(Ag-Ab);

if dif_r_g>42 & dif_r_b>66 & dif_g_b>23

figure;bar(edg,h);
q
end
end

end

end
end
end
```

Aquest primer algorisme amb els llindars tan restrictius, s'implementa per poder calcular uns llindars que abastin possibles casos amb valors diferents dels que s'han obtingut mitjançant les mostres emprades, de manera automàtica. Això és degut al fet que gràcies als valors tan restrictius dels llindars, només arriben a l'últim esgraó de l'escala de filtratges les regions on apareix una lesió, d'aquesta manera es pot automatitzar el procés d'obtenció de dades sense el perill de prendre dades de regions que no són lesió.

Les dades obtingudes de l'estudi de les regions per determinar els llindars, mostren un patró compatible amb una distribució gaussiana, per aquest motiu s'ha decidit implementar el llindar òptim segons la següent equació:

$$\mu \pm 2 \cdot \sigma \quad (1)$$

El resultat d'aplicar l'equació 1 és abastar el 95.44% dels paràmetres de la distribució gaussiana que representen els valors obtinguts. Es considera un interval de confiança suficientment inclusiu per la detecció de lesions, un augment de l'interval de confiança suposaria una sobre detecció de falsos positius que no representaria rendible en vers les deteccions que es poguessin obtenir.

Per la determinació dels llindars mínims s'utilitza la variant negativa de l'equació i per determinar els llindars màxims s'utilitza la variant positiva.

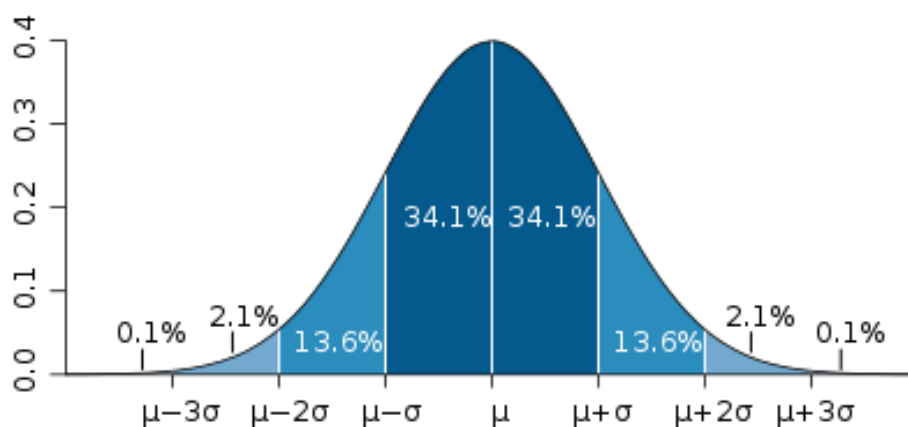


Figura 52: Distribució de probabilitat al voltant de la mitjana en una distribució normal.

Una vegada es determina l'equació per la qual es calcularan els llindars màxims i mínims, es desenvolupa un codi d'obtenció automàtic de dades i càlcul de llindars per a cada tipus de lesió.

Pel desenvolupament s'aprofita el codi de binarització i segmentació creat anteriorment, així com els llindars restrictius. Només és necessari col·locar les imatges classificades com a fosques en una carpeta determinada i les classificades com a clares en una altra. Al executar el codi, apareix una finestra on s'ha de seleccionar la carpeta on es tenen les imatges guardades i seguidament, comença el processat. El codi resultant es pot observar als apartats de l'annex 7.6 i 7.7.

Amb l'obtenció dels llindars definitius i la implementació d'aquests a l'escala de restriccions es defineix una primera versió de l'algorisme.

3.2.5 Classificador estadístic

La primera versió de l'algorisme de detecció de lesions, només disposava de les restriccions imposades en l'apartat anterior per dur a terme la detecció. En una primera prova es van obtenir molts falsos positius i es va optar per afegir un segon tipus de restricció al codi, utilitzant un classificador estadístic.

En l'aprenentatge automàtic i l'estadística, la classificació és identificar a quina sèrie de categories pertany un nou esdeveniment, sobre la base d'una sèrie de dades, que s'han utilitzat a tall d'entrenament, d'on es coneix de quina categoria formen part.

El classificador que s'escull, és el classificador de tipus Bayesià. La classificació Bayesiana proporciona una forma natural de tenir en compte

tota la informació disponible de les mides relatives de les subpoblacions associades als diferents grups dins de la població general.

Pel nostre cas s'utilitzarà un mètode Bayesa per patrons gaussians. La funció que s'utilitzarà per al càlcul, anomenada "bayesgauss", així com les que són necessàries per al seu funcionament es van extreure del *dipum_toolbox_2.0.2*.

L'aplicació del classificador estadístic consta de dues parts, la primera és l'obtenció de dades que serviran com "entrenament". Aquestes són les dades que la funció utilitza per decidir a quina subpoblació pertany la variable d'entrada. El tipus de dada que la funció "bayesgauss" utilitza per a la classificació és la matriu de covariància i el vector de mitjana, de cada regió que s'ha utilitzat com entrenament. La segona és d'implementació de la funció a l'algorisme de detecció de lesions, per determinar si realment les regions que passen l'escala de restriccions són lesions o no.

L'obtenció de dades es realitza en 2 sessions: primer s'obtenen les matrius de covariància i els vectors mitjana de les lesions fosques i clares de forma manual, després s'obtenen de falsos positius, que es poden aprofitar de la primera anàlisi que s'ha realitzat amb el codi automàticament utilitzant els algorismes anteriors.

Tant el codi manual com l'automàtic consten de les mateixes parts per al càlcul de la matriu de covariància i el vector de mitjana, però difereixen en el mètode d'obtenció de la regió que s'analitza.

El codi manual per l'obtenció de les regions s'utilitza la funció "roipoly". Aquesta funció permet mitjançant la interfície de MATLAB seleccionar manualment una regió d'interès. Un cop seleccionada la funció la guarda a l'estació de treball en forma de mascara binaria.

Taula 33: Aplicació de la funció "roipoly".

```
I1=imread('1.png');  
lesiontype1=roipoly(I1);
```

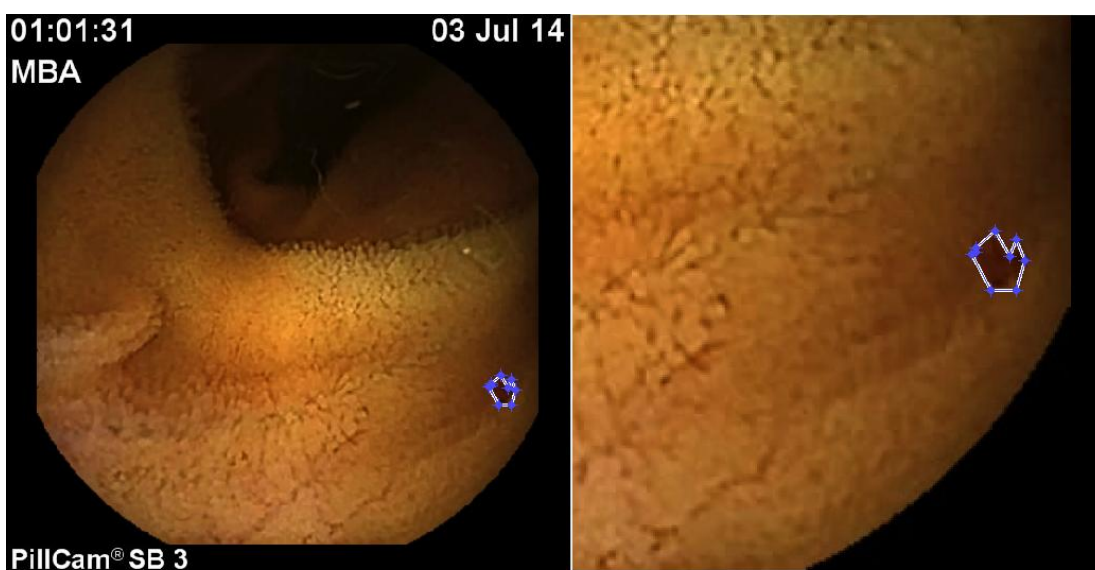


Figura 53: Aplicació de la funció "roipoly"

El codi per obtenir les regions que fan referència a una lesió és manual per tenir més precisió en l'obtenció de les dades i per evitar errors, ja que en augmentar el rang dels llindars, en alguna imatge de les que s'utilitzen com a mostra, apareixen regions que no representen cap lesió.

Per la recollida de dades de falsos positius s'ha automatitzat el procés, ja que no és necessari mirar quines regions s'analitzen i quines no, pel fet que les imatges que s'utilitzen no tenen cap regió amb lesió. El codi automàtic aprofita que els codis de detecció de lesions fosques i clares desenvolupats anteriorment, tenen com a producte final una regió binària, que es pot utilitzar com a màscara, evitant així emprar la funció "roipoly" automatitzant el procés d'obtenció de regions d'interès. És necessari aplicar per separat el codi de detecció de lesions clares i el codi de detecció de lesions fosques quan es vol obtenir regions que no representen una lesió, perquè els falsos positius que produeixen cadascuna són diferents.

Per ambdós casos, un cop s'ha obtingut la màscara binària que s'avaluarà, el procés de càlcul de la matriu de covariància i el vector mitjana és el mateix.

Primer s'utilitza la funció "imstack2vectors" que s'obté del *dipum_toolbox_2.0.2.*. Aquesta funció necessita com a variable d'entrada la imatge d'on s'ha extret la màscara i la màscara extreta. La funció extrau els píxels de la imatge original que correspondrien a la màscara i binària i els assigna a una variable de sortida.

Taula 34: Aplicació de la funció "imstack2vectors".

```
x1=imstack2vectors(I1,lesiontype1);
```

Un cop s'obtenen els píxels en color que correspondrien a la màscara extreta, s'aplica sobre la variable de sortida de la funció "imstack2vectors", la funció "covmatrix" que també es pot trobar al *dipum_toolbox_2.0.2.*. Com el seu nom indica, "covmatrix" calcula la matriu de covariància de la variable d'entrada així com el seu vector mitjana.

Taula 35: Aplicació de la funció "covmatrix".

```
[C1, m1]=covmatrix(x1);
```

Un cop es tenen les dades, es concatenen mitjançant la funció "cat" per formar el clúster de dades que serviran per a la presa de decisions. En el nostre cas s'ha utilitzat les 18 imatges amb lesió de les quals es disposava i 37 imatges detectades com a falsos positius per diverses raons. Com a resultat s'ha obtingut una matriu de 3x3x49 que conté les matrius de covariància i una de 3x49 que conté els vectors de mitjana.

Taula 36: Aplicació de la funció "cat" per a diverses matrius de covariància i vectors de mitjana.

```
CA =cat(3,C1,C2,C3,C4,C5);
```

```
MA =cat (2,m1,m2,m3,m4,m5);
```

Les matrius finals es guarden com a variables amb la funció "save", per poder ser utilitzades en l'algorisme de detecció de lesions com a valors d'entrenament. Tant el codi manual d'obtenció de dades com el automàtic es poden trobar als apartats de l'annex 7.8, 7.9 i 7.10.

Taula 37: Emmagatzematge de les dades amb la funció "save".

```
filename2='CA_MA.mat'  
save(filename2, 'CA', 'MA')
```

Un cop s'obtenen les dades que serviran com a base per la presa de decisions de la funció "bayesgauss", s'implementa el sistema de classificació estadístic a l'algorisme de detecció de lesions.

El primer pas és afegir un "path" o camí al *dipum_toolbox_2.0.2* utilitzant la funció "addpath", ja que la funció utilitza subprocessos interns que estant guardat a aquest *toolbox*.

Taula 38: Definició del camí per la utilització del "dipum_toolbox_2.0.2" emprant la funció "addpath".

```
addpath /Volumes/HUGE/CAPSULAS/dipum_toolbox_2.0.2/
```

El segon pas és carregar les dades obtingudes per l'entrenament amb la funció "imload".

Taula 39: Carreguem les variables obtingudes anteriorment a l'espai de treball de MATLAB amb la funció "load".

```
load CA_MA.mat
```

Com a tercer i últim pas, afegim un esgrao a l'escala de restricció formada pels bucles "if", on s'obtenen els píxels que corresponen a la regió que s'analitza de la imatge original amb la funció "imstack2vectors". Seguidament s'aplica la variable de sortida de la funció, juntament amb la matriu que conté les covariàncies i la matriu que conté els vectors de mitjanes a la funció "bayesgauss" per executar el procés de classificació.

Taula 40: Aplicació de la funció "bayesgauss" on "q" és l'etiqueta de la regió que s'analitza i "I" la imatge original d'on s'ha extret la regió.

```
Mask2Eval= imstack2vectors(I, (L2==q));  
dT1=bayesgauss(Mask2Eval, CA, MA);
```

La funció "bayesgauss" retorna en quin tant per un, la regió que s'analitza pertany a alguna de les regions que s'han utilitzat, separant-les per etiquetes segons l'ordre que s'han concatenat les matrius de covariància. Per això s'implementa un bucle "for" per calcular amb quines regions és més semblant la mascara que s'analitza.

Taula 41: Aplicació d'un bucle "for" per obtenir un vector amb els tant per cent de semblança que té la regió analitzada amb les dades obtingudes dels falsos positius.

```
for i =18:length(CA)
    fp1 = [fp1 (100* numel(find(dT1 == i))/length(dT1))];
end
```

Com es pot observar a la taula 41, evitant les primeres 18 etiquetes es pot obtenir un vector amb els tant per cent de semblança que té la regió estudiada, amb les regions d'entrenament que corresponen als falsos positius. D'aquesta manera aplicant un sumatori dels valors del vector, es pot saber si la regió tendeix més al grup de lesions o al grup de falsos positius.

Taula 42: Càlcul del sumatori per determinar en quin tant per cent la regió estudiada pertany al grup dels falsos positius.

```
false_positive=sum(fp1)
```

Quin dels percentatges calcular, si el de semblança al grup de lesions o al de semblança al grup de falsos positius és irrellevant, ja que són resultats complementaris. Pel nostre cas es calcula la semblança al grup de falsos positius i s'afegeix l'última restricció al codi, que es pot observar a la taula 42.

Taula 43: Restricció amb bucle "if", si la variable "falsepositive" és més petita que 50 pot seguir si no, no.

```
if false_positive<50
```

En la implementació de l'algorisme final, s'ha decidit deixar un marge de 5% en la restricció per seguretat, així doncs el límit en l'algorisme final és tenir una semblança amb el grup de les lesions d'un 45% o un 55% amb el grup dels falsos positius.

3.3 Creació del vídeo

L'últim pas en l'algorisme és la creació del vídeo utilitzant les imatges resultants de la detecció. Un dels objectius previs a la realització del treball, era la disminució del temps de diagnòstic per part dels especialistes, una manera de reduir-lo és disminuint l'extensió del vídeo que s'ha d'analitzar, creant un de nou amb només les imatges que poden contenir lesions, segons el resultat de l'aplicació del codi.

El vídeo que s'obté es pot considerar un producte d'anàlisi ràpid i dinàmic de les imatges detectades que s'han obtingut.

3.3.1 Integració al algorisme de detecció

El codi de creació del vídeo està situat en acabar el bucle de lectura i detecció de les imatges. El primer per dur a terme aquesta secció de codi, és la definició en una variable del directori on s'han guardat les imatges detectades i el seu llistat.

Taula 44: Definició del directori on es troben les imatges detectades i llistat d'aquestes.

```
ImagesFolder=strcat(Results_folder,filesep,name);  
pngFiles = dir([ImagesFolder,filesep,'*.png']);
```

Com es pot veure en la taula 44, no és necessari escriure on es troba el directori, ja que com es veurà en apartats posteriors, el directori ja està definit en una altra variable. La funció que s'utilitza per ajuntar les tres variables tipus "string" que contenen el directori, el nexa '/' i el nom amb el qual s'ha creat la carpeta de resultats, és "strcat" que com ja s'ha comentat en apartats anteriors, serveix per concatenar/ajuntar diverses variables de tipus "string".

Un cop es té definit el directori, es llisten tots els fitxers que tenen extensió ".png" amb la funció del MATLAB "dir".

Per tal que es llegeixin les imatges en ordre, és necessari ordenar el vector que conté la llista d'aquestes per la seva data de creació. Per fer-ho, es defineixen les dates de creació de les imatges en un vector i s'ordenen numèricament mitjançant la funció "sort".

Taula 45: Ordenat de les imatges.

```
S = [pngFiles(:).datenum];  
[S,S] = sort(S);  
pngFilesS = pngFiles(S);
```

3.3.2 Producció del vídeo

Un cop es té el directori determinat i les imatges que s'utilitzaran per a la constitució del vídeo, és necessari definir una variable que esdevingui el nom del vídeo.

Taula 46: Definició del nom del vídeo.

```
VideoFile=strcat (ImagesFolder, filesep, 'Final_Video');
```

Un cop s'ha fet el tràmit de definir el nom del vídeo final, s'ha de crear una variable que permeti escriure dades de vídeo, per fer-ho s'utilitza la funció "videowriter", que construeix una variable tipus "object" amb característiques típiques d'arxius de vídeo que permet la reescriptura sobre aquesta. A la variable obtinguda es poden definir les propietats com el rati de fotogrames per segon, el format del vídeo resultant, la duració en segons, el nombre de canals de color que es vol que tingui el vídeo resultant, l'altura de cada frame de vídeo en píxels, la qualitat del vídeo, els bits per píxel, el mètode de compressió i l'amplitud en píxels dels fotogrames de vídeo, com es pot observar a la taula 47. Pel nostre cas s'ha decidit aplicar una ràtio de fotogrames per segon de 4 perquè facilita l'anàlisi d'aquests.

Taula 47: Creació de la variable "object" i definició dels fotogrames per segon del vídeo final.

```
writerObj = VideoWriter (VideoFile);  
fps= 4;  
writerObj.FrameRate = fps;
```

Un cop tenim definits la variable i els paràmetres clau, s'obre la variable que contindrà el vídeo amb la funció "open" i s'executa un bucle "for" per a cada imatge. Dins el bucle es llegeix una imatge i es converteix en un fotograma del vídeo final, mitjançant la funció "im2frame".

Per acabar es finalitza el bucle "for" i es tanca la variable que conté el vídeo amb la funció "close".

Taula 48: Conversió de les imatges en fotogrames de vídeo mitjançant la funció "im2frame".

```
for t= 1:length(pngFilesS)  
    Frame=imread(strcat (ImagesFolder, filesep, pngFilesS(t).name));  
    writeVideo (writerObj, im2frame (Frame));  
end  
close (writerObj);
```

3.4 Estructura de l'algorisme final

Un dels aspectes secundaris però no menys importants, és l'estructura de programació que suporta el codi, definint les variables que no tenen res a veure amb la detecció d'angiiodisplàsies, però que són necessàries pel bon funcionament del codi. L'algorisme es pot separar en diversos blocs encarregats d'ordenar, definir o prioritzar les variables que actuaran a posteriori.

3.4.1 Gestió dels fitxers

La gestió dels fitxers que contenen les imatges extretes del codi per part de l'algorisme, és una funcionalitat que s'ha prioritzat perquè l'objectiu de l'estructura de l'algorisme no és només suportar el codi de detecció, sinó facilitar la interacció amb l'usuari final, automatitzant el màxim tots els processos i simplificant la relació entre l'usuari i l'algorisme. Per això s'ha determinat dissenyar una gestió d'arxius versàtil que no generi problemes, sinó que aportes solucions.

Una de les solucions dissenyades per simplificar les accions que ha de desenvolupar l'usuari davant l'algorisme, és que aquest només hagi de definir el directori on es tenen els estudis guardats, mitjançant una interfície visual que simplifica el procés, per tal d'executar completament el codi. Aquest fet comporta que no sigui necessari cap coneixement previ de programació, ni de com està estructurat el codi, per part de l'usuari per poder utilitzar-lo.

Per fer-ho s'utilitza la funció "uigetdir" al principi de l'algorisme, juntament amb les funcions cànon que ja s'han explicat amb anterioritat per tal de determinar el camí on es tenen els estudis guardats, tal i com es mostra en la taula 49.

Taula 49: Inici de l'algorisme final, amb la definició de la variable que contindrà el camí on es tenen guardats els estudis i el seu llistat.

```
close all
clear all
clc

tic;
addpath /Volumes/HUGE/CAPSULAS/dipum_toolbox_2.0.2/

Results_folder=uigetdir;

folders=dir([Results_folder]);

load def.mat
```

En la taula 49 també es pot observar la inicialització de les variables que ens serviran per a la classificació estadística, amb la funció "load" agrupades en l'arxiu "def.mat", la definició de camins (path) per poder utilitzar funcions emmagatzemades en altres carpetes i l'ús de la funció

"dir", per tal de llistar totes les carpetes que estant contingudes dins el directori seleccionat, per tal de determinar el nombre d'estudis que s'han d'analitzar.

Una altra de les capacitats integrades en l'algorisme per la gestió de fitxers és, la possibilitat d'analitzar diverses carpetes sense necessitat d'interacció amb l'usuari. Aquesta funció permet no haver d'estar pendent d'interaccionar amb l'algorisme per tal d'executar-lo per diversos estudis a la vegada.

La implementació d'aquesta funcionalitat, requereix la introducció en el codi d'un bucle que serveixi com a guia perquè l'algorisme vagi itinerant de carpeta. El vector que conté el llistat de carpetes es pot observar a la taula 49, mentre que la inicialització del bucle es pot observar a la taula 50.

Taula 50: *Inici de l'algorisme final, amb la definició de la variable que contindrà el camí on es tenen guardats els estudis i el seu llistat.*

```
for kfolder = 4:length(folders)
```

3.4.2 Gestió de les imatges

De la mateixa manera que és important la gestió dels fitxers que contenen els estudis que s'han d'analitzar, també és important com es tracten les imatges que seran analitzades. Aquest és un tema important perquè la gestió d'aquestes pot suposar la reducció del temps de computació o el fracàs absolut en l'anàlisi d'un estudi.

Els tres punts clau que s'han treballat en aquest aspecte són l'ordre correcte de càrrega de les imatges a la zona de variables del MATLAB, l'eliminació d'imatges repetides i el guardat de les imatges detectades.

El primer punt és necessari per al bon desenvolupament de la detecció d'imatges repetides, per mantenir l'ordre en les imatges que són identificades com portadores de lesió i s'han de guardar a la carpeta de resultats i per la bona creació del vídeo resultant.

Aquesta necessitat d'ordenar les imatges és degut a la configuració predeterminada que té el MATLAB quan llegeix les imatges. El programa no té en consideració el segon valor numèric per ordenar, en altres paraules, el MATLAB ordenaria la següent llista: 1,2,10,50,4,5,60 d'aquesta manera:1,10,2,4,5,50,60. Aquest fet comporta que no existeixi una correlació en el temps si es deixa que el MATLAB llegeixi les imatges de la manera que té predeterminada.

Per evitar aquest fet i obligar al MATLAB a llegir les imatges en ordre correlatiu ascendent, és necessari aplicar una funció que ordeni el vector que conté les imatges correctament. El MATLAB disposa d'una funció interna anomenada "sort", que és capaç d'ordenar les imatges donada una variable d'entrada que especifiqui en quin camp ordenar-les (data, nom,etc) i en quin ordre (ascendent, descendent). Pel nostre cas s'ha decidit utilitzar la data de creació com a variable d'entrada per ordenar, ja que així assegurem

que les imatges estant enumerades correlativament. La data de creació de la imatge és una dada intrínseca, que s'obté en el llistat de les imatges que estant dins de la carpeta que s'analitza.

Taula 51: Secció de codi que ordena les imatges que s'analitzaran i crea la carpeta on es guardaran les deteccions.

```
cd([Results_folder,filesep,folders(kfolder).name])
filesindirs = dir(['*.png']);
S=[filesindirs(:).datenum];
[S,S]=sort(S);
filesindir=filesindirs(S);
name=strcat(folders(kfolder).name, '_RESULTS');
mkdir(Results_folder,name);
```

En la taula 51 es pot observar a banda del codi utilitzat per ordenar les imatges, com s'accedeix a la primera carpeta que s'analitza amb la funció "cd" i com es defineix el nom i es crea la carpeta de resultats, que és on es guardaran les imatges detectades. El nom es defineix mitjançant la funció "strcat" que concatena dos variables "strings" on aprofitem el nom que se li dóna a la carpeta contenidora de les imatges i si afegim l'apèndix "_RESULTS". Pel que fa a la creació de la carpeta, s'utilitza una funció interna del MATLAB, anomenada "mkdir" que crea una nova carpeta amb el nom que es desitja, al directori que se li assigna com a variable d'entrada.

Pel que fa al segon punt, la càpsula endoscòpica com ja s'ha explicat en la introducció, té una ràtio de fotogrames per segon en la captura d'imatges, aquest fet comporta que hi apareguin intervals on la càpsula no efectua cap moviment i s'obtingui una sèrie d'imatges correlatives idèntiques. Aquestes si s'analitzessin, comportaria un gran augment de temps de computació així com de deteccions de tot tipus sense repercutir en un benefici pel diagnòstic.

Per determinar si la imatge que es vol analitzar és igual a la següent, s'ha dissenyat un bucle amb el qual es carrega la imatge immediatament posterior per comprovar si són iguals, mitjançant un "if".

Taula 52: Bucle per eliminar les imatges repetides.

```
for kfile =1:length(filesindir)

I = imread([filesindir(kfile).name]);
kfile2=kfile+1;

if kfile2<=length(filesindir)
Iempty = imread([filesindir(kfile2).name]);
else
Iempty=zeros(576,576,3);
end

if I==Iempty
disp([Results_folder,filesep,filesindir(kfile).name]);
continue
else
```

En la taula 52 es pot observar el procediment pel qual es determina l'anàlisi d'una imatge o no, depenen de si té còpies seves per davant. La secció de codi es troba dins un bucle "for", que té la funció de guia per tal que l'algorisme llegeixi totes les imatges de l'estudi que s'analitza. El primer "if" que es pot observar, s'utilitza com a restricció per no sobrepassar el nombre d'imatges incloses en el vector "filesindir" i provocar un error de lectura, si la variable "kfile" supera la longitud del vector "filesindir", en comptes d'assignar a la variable Iempty la imatge següent a la qual s'analitza, si associa una matriu formada per zeros. En el següent "if" que es pot trobar es realitza el filtratge de les imatges que són iguals, comparant la imatge que s'analitza amb la següent, si el resultat de la comparació és verdader, es mostra en pantalla la imatge que es descarta i es torna al principi del bucle "for" amb la funció "continue" per l'anàlisi de la següent imatge. Si el resultat de la comparació és fals, el processament de la imatge continua.

Per acabar l'últim punt que s'ha tingut en compte en aquest apartat és com es guarden les imatges detectades i la representació d'aquestes. La idea principal era guardar la imatge original sense processar, però es va descartar aquesta idea, pel fet que guardant la imatge original no es podia detectar si realment l'algorisme havia detectat la lesió o si es tractava d'un fals positiu que l'atzar havia fet que es trobés en una imatge on apareixia una lesió. Per això es va dissenyar un processat final que modifiqués mínimament les propietats de la imatge original, perquè remarques les zones detectades pel codi i ajudes a detectar visualment amb més facilitat les regions d'interès.

Taula 53: *Processat per realçar les zones detectades.*

```
if false_positive<55

sqw=strel('disk',1);
fin=imerode((L2==q),sqw);
mask=((L2==q)-fin);
mask=im2bw(mask,graythresh(mask));
R(mask)=255;
checkpoint3=true;

end
```

En la taula 53 es pot observar el codi emprat per realçar les deteccions. Aquesta part del codi es troba al final de l'escala de restriccions, la idea és aplicar un filtre d'erosió d'un píxel de gruix, a la màscara que fa referència a la regió detectada i restar-la de si mateixa, deixant com a producte final una mascara dels límits de la regió. És necessari realitzar una binarització, ja que el resultat de la resta no dona una màscara completament binària. Després de binaritzar s'assignen tots els píxels que es trobin en la posició relativa a la mascara en la imatge original una intensitat de 255, això fa que es defineixi una zona vermella en la imatge original que encercla la regió detectada per l'algorisme, com es pot observar en la figura 54.



Figura 54: Imatge resultant de l'aplicació de l'algorisme de detecció.

Un problema que també es soluciona és la multidetecció de regions en una mateixa imatge, per aquesta raó es pot observar al final de la taula 53 un indicador anomenat "checkpoint3", que es torna verdader sempre que una regió passa tots els filtres. Aquest indicador s'utilitza per validar que alguna regió ha passat tots els filtres i es pot guardar la imatge resultant.

Taula 54: Secció de codi per guardar les imatges.

```
if q==length(L2props) & checkpoint3==true

filename=strcat(Results_folder,filesep,name,filesep,sprintf('detc_img_
%d.png', kfile));

imwrite(R, filename,'png') ;
disp(filesindir(kfile).name)

end
```

Com es pot observar a la taula 54, els requisits per guardar una imatge són que totes les regions hagin estat analitzades i que alguna hagi passat tots els filtres (checkpoint=verdader). Un cop es compleixen els requisits, es

defineix el nom de la imatge que es guardarà i es guarda. En la definició del nom s'especifica el directori on es vol guardar la imatge, a més del nom propi vinculat a aquesta. Per fer-ho s'utilitza una altra vegada la funció "strcat" que uneix les variables tipus "string" que es desitja. Per reescriure i guardar la imatge que s'analitza es fa servir la funció "imwrite". Finalment com a complement visual per l'usuari, es mostra a la finestra de comandes del MATLAB el nom de la imatge que s'ha guardat.

De seccions de codi per guardar imatges n'hi ha dos, una per la detecció de lesions clares i una altra per la detecció de lesions fosques. El funcionament d'ambdues és el mateix (taula 53 i 54), i el posicionament d'aquestes és just en finalitzar l'escala de restriccions, abans d'acabar el bucle "for" que itera sobre les diferents regions detectades.

3.4.3 Integració algorisme de detecció de lesions clares i algorisme de detecció de lesions fosques.

En la integració dels dos algorismes en un únic codi, es buscava la coordinació entre les deteccions realitzades per cada codi, evitant les duplicitats en les deteccions.

La integració dels dos codis suposa decidir un ordre en el qual un dels dos mètodes de detecció s'executa abans que l'altre, d'aquesta manera s'ha acceptat el risc que en executar el primer algorisme es determini un fals positiu en una imatge amb lesió que el segon algorisme de detecció sí que hagués detectat. Aquest factor de risc s'accepta perquè no suposaria un perjudici pel pacient, ja que en produir-se el cas citat, la imatge estaria englobada en el còmput de detectades, l'únic efecte negatiu seria el no saber si realment s'hagués detectat en l'algorisme posterior, o no. Per reduir aquest risc s'ha determinat posar primer l'algorisme de detecció de lesions clares perquè és més restrictiu en la detecció de regions que l'algorisme de detecció de lesions fosques. Tot i així s'ha dissenyat una solució per saber si el segon algorisme hagués detectat la lesió, ja que si es planteja aquest escenari, és necessari saber si s'hagués detectat la lesió o hagués esdevingut un fals negatiu, fet que influiria en la sensibilitat de l'algorisme. Primer s'identifica les imatges que detecta cada secció de codi amb noms diferents, re-definint la variable que determina el nom d'aquestes i es re-analitzen les imatges detectades pel primer codi on apareix la hipòtesi plantejada, amb l'algorisme anterior a la integració, que només conté el segon codi. En el cas dels estudis analitzats per aquest treball, no s'ha obtingut cap imatge que reproduís el cas d'error que s'ha plantejat, per això no s'ha utilitzat aquest mètode resolutiu.

L'eventualitat que determina que una imatge passi a ser analitzada per el segon algorisme és que l'indicador "checkpoint" definit al principi de l'algorisme definitiu com a "false", segueixi en aquest estat. El canvi d'estat de "false" a "true", significaria que s'ha detectat alguna regió sospitosa en el primer codi i es procediria a l'anàlisi de la següent imatge, sense passar pel segon codi.

Taula 55: Principi de la secció que detecta lesions clares, on es defineix l'indicador "checkpoint" com a "false".

```
else
checkpoint=false;
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);
```

Taula 56: Interval de l'algorisme final, on es visualitza el final del codi de detecció de lesions clares i principi del codi de lesions fosques.

```
end
if q==length(L2props) & checkpoint2==true

filename=strcat(Results_folder,filesep,name,filesep,sprintf('detc_img_
%d.png', kfile));

imwrite(R, filename,'png') ;
disp(filesindir(kfile).name)
checkpoint=true;

end
end
if checkpoint == false
Ired=I(:,:,1);
Igreen=I(:,:,2);
```

En la figura 56 es pot observar com, si apareix una detecció en la secció del codi final que detecta les lesions clares i es guarda la imatge, també es redefineix l'indicador "checkpoint" que passa a ser "true". Aquest fet, juntament amb una restricció feta amb un "if", que determina que la imatge que s'analitza només pot continuar el procés si l'indicador "checkpoint" segueix en "false", són les línies de codi que eviten una superposició en les deteccions de les imatges i integren els dos codis de detecció de lesions.

CAPÍTOL 4: RESULTATS

El capítol de resultats, fa referència als valors numèrics claus pel desenvolupament del treball i per l'assoliment dels objectius, tot i així es podria considerar la metodologia desenvolupada per la detecció de lesions provocades per angiодisplàsies com un resultat, ja que és una solució totalment nova per la resolució d'un problema.

4.1 Factors de discriminació

Un dels resultats que s'han obtingut durant el desenvolupament del projecte, ha estat la determinació de valors característics de les lesions, que permeten discriminar-les respecte altres elements, essent aquest un pas bàsic per la detecció de les angiодisplàsies.

A conseqüència de no existir estudis descriptius de les diferents propietats característiques de les lesions provocades per angiодisplàsies, es consideren resultats la descoberta dels elements que s'han estudiat per la implementació de filtres restrictius, ja que poden servir per estudis posteriors sobre aquestes lesions.

Els elements que s'han descobert, que representen més regularment les lesions produïdes per angiодisplàsies, són els patrons estadístics que demostren tenir, les intensitats dels píxels dels diferents canals de color que representen la lesió en una imatge, la diferència entre els nivells d'intensitat que tenen els píxels dels diferents canals i el coeficient entre els eixos de les lesions.

Pel fet que, per motius de metodologia es separen les lesions en dos grups, els resultats de les propietats que defineixen les angiодisplàsies, s'han obtingut per a cadascun dels grups, lesions clares i lesions fosques.

Els valors que es recullen en aquest apartat, pel fet d'haver-se obtingut d'una població de 20 mostres de lesions, són el resultat d'un càlcul estadístic que representa el rang de valors que descriurien la majoria de lesions de cada tipus. En realitzar els càlculs s'ha obtingut valors que no s'han considerat representatius i s'han simbolitzat a les taules amb el caràcter "-".

Taula 57: Valors estadístics característics de les lesions clares.

Rang de valors característics lesions clares	Canal vermell		Canal verd		Canal blau	
	Màxim	Mínim	Màxim	Mínim	Màxim	Mínim
Màxim	252.47	-	144.64	-	88.68	-
Mínim	-	94.98	-	28.23	-	-
Mitjana	203.23	101.53	102.24	36.19	49.49	17.27
Desviació Estàndard	19.74	3.60	20.74	2.05	16.78	1.68
Entropia	6.21	4.02	6.30	4.02	6.14	3.72

Taula 58: Valors estadístics característics de les lesions fosques.

Rang de valors característics lesions fosques	Canal vermell		Canal verd		Canal blau	
	Màxim	Mínim	Màxim	Mínim	Màxim	Mínim
Màxim	193.52	-	89.2	-	40.49	-
Mínim	-	22.01	-	0.10	-	-
Mitjana	160.02	37.04	60.05	10.09	20.81	-
Desviació Estàndard	16.71	2.40	13.23	1.56	6.70	1.95
Entropia	5.77	3.54	5.48	3.26	4.41	2.65

Taula 59: Valors de la diferència entre intensitats dels diferents canals per les lesions fosques.

Diferències entre canals lesions fosques	Canal vermell	Canal verd	Canal blau
Canal vermell	-	>24.37	>48.53
Canal verd	-	-	>18.68
Canal blau	-	-	-

Taula 60: Valors de la diferència entre intensitats dels diferents canals per les lesions clares.

Diferències entre canals lesions clares	Canal vermell	Canal verd	Canal blau
Canal vermell	-	>60.43	>84.53
Canal verd	-	-	>21.78
Canal blau	-	-	-

Taula 61: Valors del coeficient entre els eixos de les lesions.

Coeficient entre eixos de la lesió	
Lesions clares	<2.042
Lesions fosques	<2.65

4.2 Comparació algorisme desenvolupat amb software comercial

L'estudi comparatiu entre el software comercial i el desenvolupat s'ha fet mitjançant l'anàlisi de 6 estudis, no tots ells presenten la malaltia. S'ha utilitzat el software subministrat per la companyia Given Imaging, que segons els especialistes és el que millors resultats dona en el diagnòstic.

El procés de comparació i obtenció dels resultats ha estat, analitzar cadascun dels 6 estudis amb cada algorisme i validar els resultats d'aquests amb l'ajuda del Dr. Marco Antonio.

Els valors que s'han de tenir en compte a l'hora de comparar són: El nombre de deteccions fetes per l'algorisme amb diagnòstic validat com a positiu, nombre de deteccions fetes per l'algorisme amb diagnòstic validat com a negatiu i nombre de lesions amb diagnòstic validat com a positiu que no ha detectat l'algorisme.

A part d'aquests valors i tenint en compte que, com a objectiu del treball també es buscava la reducció del temps de diagnòstic, s'ha comprovat quina disminució del temps del vídeo a analitzar suposa si s'utilitzà el software desenvolupat en aquest projecte. Aquest valor no es pot utilitzar per comparar amb el software comercial perquè, a diferència del software desenvolupat en aquest projecte, aquest no dona com a producte final un vídeo, sinó que mostra els fotogrames que detecta com a lesió.

El mètode que s'ha utilitzat per a la comparació entre els softwares és el càlcul de l'increment dels factors que es consideren clau amb l'equació següent:

$$\frac{\text{Factor algorisme desenvolupat} - \text{Factor algorisme Given Imaging}}{\text{Factor algorisme Given Imaging}} \cdot 100 \quad (2)$$

Taula 62: Resultat de l'anàlisi amb el software de Given Imaging.

Factors	Temps del vídeo estudiat (25 fps)	Nombre de fotogrames	Deteccions totals	Deteccions positives	Deteccions negatives	Falsos negatius
Estudi						
1	00:43:56	65924	16	2	14	5
2	00:10:59	16484	15	0	15	0
3	00:38:41	58034	18	0	18	2
4	00:32:28	48709	10	0	10	11
5	00:18:42	28069	28	3	25	6
6	00:33:59	50999	12	5	7	55

Taula 63: Resultat de l'anàlisi amb el software desenvolupat en el projecte.

Factors	Temps del vídeo obtingut (4 fps)	Nombre de fotogrames	Deteccions totals	Deteccions positives	Deteccions negatives	Falsos negatius
Estudi						
1	00:07:11	65924	1726	7	1719	0
2	00:03:24	16484	816	0	816	0
3	00:05:11	58034	1244	0	1244	2
4	00:01:33	48709	375	10	365	1
5	00:05:05	28.069	1223	9	1214	0
6	00:10:57	50999	2629	13	2616	47

Taula 64: Comparació de resultats.

Estudi	Δ Temps del vídeo	Δ Deteccions totals	Δ Deteccions positives	Δ Deteccions negatives	Δ Falsos negatius
1	-83.65%	+10687.5%	+250%	+12178.57%	-100%
2	-69.04%	+5340%	0.0%	+5340%	0%
3	-86.60%	+6811.11%	0.0%	+6811.11%	0%
4	-95.22%	+3650%	+1000%	+3550%	-90.91%
5	-72.82%	+4267.85%	+200%	+4756%	-100%
6	-67.78%	+21808.33%	+160%	+37271.42%	-14.54%

CAPÍTOL 5: CONCLUSIONS

Analitzant els resultats obtinguts, es pot determinar que s'han assolit tots els objectius plantejats a l'inici del projecte.

S'ha aconseguit el desenvolupament d'un algorisme capaç de detectar lesions provocades per angiодisplàsies en l'intestí prim, utilitzant imatges extretes de l'estudi realitzat per una càpsula endoscòpica.

En els estudis que presentaven lesions, s'ha augmentat un 355% de mitjana la detecció de lesions, i s'ha reduït un 61,09% l'omissió de lesions patològiques, respecte al software comercial més eficaç.

En tots els estudis analitzats s'ha aconseguit reduir la duració del vídeo d'estudi un 79.18% de mitjana.

S'han descobert factors característics clau per la discriminació de les lesions respecte altres elements presents en imatges d'endoscòpia i s'ha determinat el rang de valors que correspon a les lesions provocades per angiодisplàsies.

S'ha desenvolupat una metodologia de treball pel disseny d'algorismes, capaços de discriminar i detectar angiодisplàsies en l'intestí prim.

No obstant els bons resultats, també hi ha diversos factors que es podrien millorar en el cas d'una revisió del projecte, per tal d'augmentar la qualitat d'aquest.

Seria útil una disminució de les deteccions negatives pel cas de l'algorisme desenvolupat, tot i que aquest va lligat a l'augment de deteccions positives, és necessari intentar reduir el nombre de deteccions, ja que aquest fet implicaria una disminució encara més gran de la duració del vídeo final.

Com a validació, seria necessària una revisió dels llindars establerts, utilitzant com a mostra per l'estudi una població més gran d'imatges que presentin lesió. Amb això, es pot verificar que els llindars estant ben establerts i afinar el valor d'aquests.

Per acabar, es podria implementar una millora en la versatilitat a l'hora de binaritzar, per tal de no haver d'utilitzar dos models per la detecció de lesions, com s'utilitza en aquest projecte.

En conclusió, la valoració final del projecte és positiva, en primer lloc per haver assolit l'objectiu principal, malgrat la falta d'informació i estudis en aquest tema. En segon lloc, per haver aconseguit tots els objectius específics, tot i els marges de millora que encara es poden implementar. I en tercer lloc, pel desenvolupament satisfactori d'un algorisme, ara per ara pioner, que pot significar un augment en la qualitat de vida de les persones i el començament de l'establiment de bases, bibliografia i estudis pel desenvolupament d'aquest tipus d'algorismes, que previsiblement tindran un paper important en el sistema de diagnòstic mèdic d'un futur molt proper.

CAPÍTOL 6:

BIBLIOGRAFIA

6.1 Referències bibliogràfiques

González Suárez, Begoña, Josep Maria Dedeu Cuscó, Sara Galter Copa, y Alfredo Mata Bilbao. «Cápsula endoscópica: documento de posicionamiento de la Sociedad Catala de Digestología.» *Gastroenterología y Hepatología*, 2011: 573-583.

Molina Infante, J., y J M Mateos Rodríguez. «Angiodisplasias.» *Revista española de enfermedades digestivas* (Arán Ediciones, S.L.), 2010: 667-667.

Yano, Tomonori, y otros. «Endoscopic classification of vascular lesions of the small intestine.» *Gastrointestinal Endoscopy*, 2008: 169-172.

6.2 Bibliografia de consulta

American society for gastrointestinal endoscopy. *www.asge.org*. Agost de 2014. <http://www.asge.org/press/press.aspx?id=8140> (último acceso: 6 de Març de 2015).

C Gonzalez, Rafael, Richard E Woods, y Steven L Eddins. *Digital Image Processing using MATLAB*. Natick, Massachusetts: Gatesmark, 2009.

C Pedrosa, Marcos. «uptodate.» <http://www.uptodate.com/>. Editado por Lawrence S Friedman. 13 de Febrer de 2015. <http://www.uptodate.com/contents/angiodysplasia-of-the-gastrointestinal-tract> (último acceso: 6 de Març de 2015).

Cedron-Cheng, Hugo. «Cápsula Endoscópica del Intestino Delgado.» *Revista de gastroenterología del Perú*, 2010: 341-349.

Chen-yi, Li, Zhang Bing-ling, Chen Chun-xiao, y Li You-ming. «OMOM capsule endoscopy in diagnosis of small bowel disease.» *Journal of Zhejiang University SCIENCE B*, 2008: 857-862.

G Adler, Douglas, y Christopher J Gostout. «Wireless Capsule Endoscopy.» *Hospital Physician*, 2003: 14-22.

Given Imaging. <http://www.givenimaging.com/>. 2015. <http://www.givenimaging.com/en-int/Innovative-Solutions/Capsule-Endoscopy/Pillcam-SB/PillCam-SB-3/Pages/default.aspx> (último acceso: 21 de Abril de 2015).

—. «neovitalis.» <http://www.neovitalis.dk/>. 2015. <http://www.neovitalis.dk/uploads/PillCam%20SB%203%20FAQs.pdf> (último acceso: 21 de Abril de 2015).

Gunjan, Deepak, Vishal Sharma, Surinder S Rana, y Deepak K Bhasin. «Small bowel bleeding: a comprehensive review.» *Gastroenterology Report*, 2014: 262-275.

IntroMedic. <http://www.intromedic.com/>. 2015. http://www.intromedic.com/eng/sub_products_2.html (último acceso: 23 de Abril de 2015).

JINSHAN SCIENCE & TECHNOLOGY. *jinshangroup*. 2015. <http://english.jinshangroup.com/index.php/Index/product/id/3> (último acceso: 23 de Abril de 2015).

lasermed S.A. <http://www.lasermed.ch/>. 2013. <http://www.lasermed.ch/fr/pillcam-sb3.html> (último acceso: 21 de Abril de 2015).

Luján Sanchis, Marisol. <http://capsula-endoscopica.com/>. 2012. <http://capsula-endoscopica.com/la-capsula-endoscopica/capsula-patency> (último acceso: 10 de Març de 2015).

MathWorks. <http://es.mathworks.com/>. Abril de 2015. <http://es.mathworks.com/help/MATLAB/ref/videowriter.writevideo.html> (último acceso: 21 de Abril de 2015).

Munítiz Ruiz, V, y otros. «Angiodisplasia múltiple de intestino delgado. Un reto diagnóstico y terapéutico.» *Gastroenterología y Hepatología*, 2004: 311-313.

National Institute for Health and Clinical Excellence. «Wireless capsule endoscopy for investigation of the small bowel.» *NICE interventional procedure guidance 101*, 2004: 1-9.

Olympus. *Endocapsule 10 system*. Product Brochure, Olympus, 2015.


Yuce, M.R. «Easy-to-Swallow Wireless Telemetry.» *Microwave Magazine, IEEE*, 2012: 90-101.



Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Pressupost

A faded background image of a large, classical-style building with a central tower and many windows.

“Sistema automàtic de detecció de lesions en imatges endoscòpiques”

TFG presentat per optar al títol de GRAU en
ENGINYERIA BIOMÈDICA
per **Ferran Noya Urgell**

Barcelona, 09 de Juny de 2015

Director: Raúl Benítez Iglesias
Departament ESAII
Universitat Politècnica de Catalunya (UPC)

ÍNDEX PRESSUPOST

CAPÍTOL 7: Pressupost.....	107
7.1 Desglossament de costos.....	109
7.1.1 Hores d'enginyeria	109
7.1.2 Estació de treball	109
7.1.3 Softwares utilitzats.....	109
7.2 Pressupost Total.....	110

CAPÍTOL 7: PRESSUPOST

7.1 Desglossament de costos

7.1.1 Hores d'enginyeria

Les hores s'han dividit en les diverses taques que s'han anat realitzant durant el desenvolupament del projecte.

Estudi previ: 45 €/h

- Cerca d'informació sobre angiодisplàsies: 40 h
- Cerca d'informació sobre càpsules endoscòpiques: 40 h
- Cerca d'informació sobre tècniques actuals de diagnòstic: 70 h
- Estudi per la determinació de factors característics per la discriminació de les angiодisplàsies: 80 h
- Valoració de diverses tècniques per determinar les més adients per emprar: 100 h

Implementació: 45 €/h

- Realització del preprocessat i obtenció de les imatges a partir del vídeo: 35 h
- Implementació dels processos de binarització: 60h
- Segmentació de les regions d'interès: 30 h
- Tècniques de filtratge, reconeixement i caracterització: 70 h
- Programa principal: 50h

Elaboració de la documentació: 35 €/h

- Redacció memòria del projecte: 50 h
- Elaboració dels resultats i les conclusions: 65 h
- Citació de referències i bibliografia: 25 h

TOTAL pressupost hores d'enginyeria: 30.775€

7.1.2 Estació de treball

Per l'elaboració del projecte s'ha necessitat un ordinador, memòries digitals externes per el transport de dades i material d'oficina.

El preu total d'aquest apartat és de: 2.500€

7.1.3 Softwares utilitzats

El software MATLAB per realitzar el processat de les imatges i Microsoft Office per la redacció de la memòria.

Preu TOTAL softwares utilitzats: 3.137 €

7.2 Pressupost Total

En aquest apartat s'exposà el pressupost final incloent tots els apartats anteriors.

Taula 65: Pressupost final del projecte.

Hores d'enginyeria	Tasques	Hores	Preu	TOTAL €
Estudi Previ	Cerca d'informació sobre angiодisplàsies	40	45	1.800 €
	Cerca d'informació sobre càpsules endoscòpiques	40	45	1.800 €
	Cerca d'informació sobre tècniques actuals de diagnòstic	70	45	3.150 €
	Estudi per la determinació de factors característics per la discriminació de les angiодisplàsies	80	45	3.600€
	Valoració de diverses tècniques per determinar les més adients per emprar	100	45	4.500€
Implementació	Realització del preprocessat i obtenció de les imatges a partir del vídeo	35	45	1.575 €
	Implementació dels processos de binarització	60	45	2.700 €
	Segmentació de les regions d'interès	30	45	1.350 €
	Tècniques de filtratge, reconeixement i caracterització	70	45	3.150 €
	Programa principal	50	45	2.250 €
Elaboració de la documentació	Redacció memòria del projecte	50	35	1.750 €
	Elaboració dels resultats i les conclusions	65	35	2.275 €
	Citació de referències i bibliografia	25	35	875 €
	TOTAL			30.775 €
Estació de treball				
	Estació de treball			2.500 €
	TOTAL			2.500 €


Softwares utilitzats		
MATLAB	Llicències de MATLAB i Toolboxes	3.000 €
Microsoft Office	Llicència de Microsoft Office	137 €
	TOTAL	3.137 €
PRESSUPOST FINAL		36.412 €



Escola Universitària d'Enginyeria
Tècnica Industrial de Barcelona
Consorci Escola Industrial de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Annex

A faded, light-colored illustration of a large, classical-style building with a central tower and many windows, serving as a background for the title.

“Sistema automàtic de detecció de lesions en imatges endoscòpiques”

TFG presentat per optar al títol de GRAU en
ENGINYERIA BIOMÈDICA
per **Ferran Noya Urgell**

Barcelona, 09 de Juny de 2015

Director: Raúl Benítez Iglesias
Departament ESAII
Universitat Politècnica de Catalunya (UPC)

ÍNDEX ANNEX

CAPÍTOL 8: Annex	117
8.1 Algorisme principal	119
8.2 Algorisme per extraure fotogrames	126
8.3 Algorisme detector de lesions fosques	127
8.4 Algorisme detector de lesions clares.....	130
8.5 Algorisme semiautomàtic detector de llindars.....	133
8.6 Algorisme automàtic detecció i càlcul de llindars lesions fosques.	134
8.7 Algorisme automàtic detecció i càlcul de llindars lesions clares ...	139
8.8 Algorisme d'entrenament manual pel classificador estadístic	144
8.9 Algorisme d'entrenament per lesions fosques pel classificador estadístic	146
8.10 Algorisme d'entrenament per lesions clares pel classificador estadístic	149

CAPÍTOL 8:

ANNEX

8.1 Algorisme principal

```

close all
clear all
clc

tic;
addpath F:\Capsulas\dipum_toolbox_2.0.2\

Results_folder=uigetdir;

folders=dir([Results_folder]);

load def.mat

for kfolder = 3:length(folders)
    cd([Results_folder,filesep,folders(kfolder).name])
    filesindirs = dir(['*.png']);
    S=[filesindirs(:).datenum];
    [S,S]=sort(S);
    filesindir=filesindirs(S);
    name=strcat(folders(kfolder).name, '_RESULTS');
    mkdir(Results_folder,name);

    for kfile =1:length(filesindir)

        I = imread([filesindir(kfile).name]);
        kfile2=kfile+1;

        if kfile2<=length(filesindir)
            Iempty = imread([filesindir(kfile2).name]);
        else
            Iempty=zeros(576,576,3);
        end

        if I==Iempty
            disp([Results_folder,filesep,filesindir(kfile).name]);
            continue

        else
            checkpoint=false;
            Ired=I(:,:,1);
            Igreen=I(:,:,2);
            Iblue=I(:,:,3);
            IR=rgb2gray(I);

            %%%%%%%%%%%
            %%%%%%%%%%%

            Q=Ired-Igreen;
            Iadpt=adapthisteq(Q);

            se=strel('disk',5);

            Ie=imerode(Iadpt,se);

```

```

Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,122/255);

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ibin), hy, 'replicate');
Ix = imfilter(double(Ibin), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

% Do the WATERSHED
L = watershed(gradmag);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

for etiqueta = AreasSim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');

```



```
gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);

L2props=regionprops(L2, 'Area');
R=I;
checkpoint2=false;
for q=1:length(L2props)
    fp1=[];

    vec=Ired(L2==q);
    edg=1:1:255;
    h=histc(vec,edg);

    A=max(vec);
    B=std2(vec);
    C=mean2(vec);
    D=min(vec);

    if A<252.47 & B>3.60 & B<19.74 & C>101.53 & C<203.23 & D>94.98

        Q_R=entropy(Ired(L2==q));
        Q_G=entropy(Igreen(L2==q));
        Q_B=entropy(Iblue(L2==q));

        if Q_R<6.21 & Q_R>4.02 & Q_G>4.02 & Q_G<6.3 & Q_B>3.72 &
Q_B<6.14

            vecg=Igreen(L2==q);
            Ag=max(vecg);
            Bg=std2(vecg);
            Cg=mean2(vecg);
            Dg=min(vecg);

            if Ag<144.64 & Bg>2.05 & Bg<20.74 & Cg>36.19 & Cg<102.24
& Dg>28.23

                vecb=Iblue(L2==q);
                Ab=max(vecb);
                Bb=std2(vecb);
                Cb=mean2(vecb);
                Db=min(vecb);
```

```

        if Ab<88.68 & Bb<16.78 & Bb>1.68 & Cb>17.27 &
Cb<49.49
            dif_r_g=(A-Ag);
            dif_r_b=(A-Ab);
            dif_g_b=(Ag-Ab);

            if dif_r_g>60.43 & dif_r_b>84.53 & dif_g_b>21.78
                majoraxis=regionprops(L2==q, 'MajorAxisLength');
                minoraxis=regionprops(L2==q, 'MinorAxisLength');

                majoraxisL=majoraxis.MajorAxisLength;
                minoraxiL=minoraxis.MinorAxisLength;
                mamincoef=majoraxisL/minoraxiL;

                if mamincoef<2.042

                    Mask2Eval= imstack2vectors(I, (L2==q));

                    dT1=bayesgauss(Mask2Eval, CA, MA);
                    for i =18:length(CA)
                        fp1 = [fp1 (100*numel(find(dT1 == i))/length(dT1))];
                    end
                    false_positive=sum(fp1);

                    if false_positive<55

                        sqw=strel('disk',1);
                        fin=imerode((L2==q),sqw);
                        mask=((L2==q)-fin);
                        mask=im2bw(mask,graythresh(mask));
                        R(mask)=255;

                        checkpoint2=true;

                    end
                end
            end
        end

        end
        if q==length(L2props) & checkpoint2==true
filename=strcat(Results_folder,filesep,name,filesep,sprintf('detc_img_
%d.png', kfile));
            imwrite(R, filename, 'png') ;
            disp(filesindir(kfile).name)
            checkpoint=true;
        end
    end
    if checkpoint == false

```

```
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);

Iadpt=adapthisteq(Igreen);

se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,graythresh(I2)/1.42);

si=strel('disk',3);
I4=imclose(Ibin,si);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%segmentacion

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(I4), hy, 'replicate');
Ix = imfilter(double(I4), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

% Do the WATERSHED
L = watershed(gradmag);

%figure; imagesc(L);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

Areasim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

for etiqueta = Areasim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');
gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);

L2props=regionprops(L2, 'Area');
checkpoint3=false;
R=I;
for q=1:length(L2props)
    fp2=[];

    vec=Ired(L2==q);
    edg=1:1:255;
    h=histc(vec,edg);

    A=max(vec);
    B=std2(vec);
    C=mean2(vec);
    D=min(vec);

    if A<193.52 & B<16.71 & B>2.40 & C>37.04 & C<160.02 & D>22.01

        Q_R=entropy(Ired(L2==q));
        Q_G=entropy(Igreen(L2==q));
        Q_B=entropy(Iblue(L2==q));
    end
end

```

```

    if Q_R<5.77 & Q_R>3.54 & Q_G>3.26 & Q_G<5.48 & Q_B>2.65 &
Q_B<4.41

        vecg=Igreen(L2==q);
        Ag=max(vecg);
        Bg=std2(vecg);
        Cg=mean2(vecg);
        Dg=min(vecg);

        if Ag<89.2 & Bg>1.56 & Bg<13.23 & Cg>10.09 & Cg<60.05 &
Dg>0.10

            vecb=Iblue(L2==q);
            Ab=max(vecb);
            Bb=std2(vecb);
            Cb=mean2(vecb);
            Db=min(vecb);

            if Ab<40.49 & Bb>1.95 & Bb<6.70 & Cb<20.81
                dif_r_g=(A-Ag);
                dif_r_b=(A-Ab);
                dif_g_b=(Ag-Ab);

                if dif_r_g>24.37 & dif_r_b>48.53 & dif_g_b>18.68

majoraxis=regionprops(L2==q, 'MajorAxisLength');
minoraxis=regionprops(L2==q, 'MinorAxisLength');

                majoraxisL=majoraxis.MajorAxisLength;
                minoraxiL=minoraxis.MinorAxisLength;
                mamincoef=majoraxisL/minoraxiL;
                if mamincoef<2.65

                    Mask2Eval= imstack2vectors(I, (L2==q));

                    dT1=bayesgauss(Mask2Eval, CA, MA);
                    for i =18:length(CA)
                    fp2 =[fp2 (100*numel(find(dT1 == i))/length(dT1))];

                    end
                    false_positive=sum(fp2);
                    if false_positive<55
                        sqw=strel('disk',1);
                        fin=imerode((L2==q),sqw);
                        mask=((L2==q)-fin);
                        mask=im2bw(mask,graythresh(mask));
                        R(mask)=255;
                        checkpoint3=true;

                    end
                end
            end
        end
end
end
end

```

```

        end

        end

        end
    end
    if q==length(L2props) & checkpoint3==true
filename=strcat(Results_folder,filesep,name,filesep,sprintf('detc_img_
%d.png', kfile));
                                imwrite(R, filename,'png') ;
                                disp(filesindir(kfile).name)

                                end

        end
    end

end
end

ImagesFolder=strcat(Results_folder,filesep,name);
pngFiles = dir([ImagesFolder,filesep,'*.png']);
S = [pngFiles(:).datenum];
[S,S] = sort(S);
pngFilesS = pngFiles(S);
VideoFile=strcat(ImagesFolder,filesep,'Final_Video');
writerObj = VideoWriter(VideoFile);
fps= 3;
writerObj.FrameRate = fps;
open(writerObj);
for t= 1:length(pngFilesS)
    Frame=imread(strcat(ImagesFolder,filesep,pngFilesS(t).name));
    writeVideo(writerObj,im2frame(Frame));
end
close(writerObj);
cd ..
end
toc;

```

8.2 Algorisme per extraure fotogrames

```

close all;
clear all;
clc;

tic;

folder = uigetdir;
folders=dir([folder,filesep,'*.mpg']);
cd(folder);
for kfolder = 1:length(folders)

obj = VideoReader(folders(kfolder).name);

```

```
nFotogrames = obj.NumberOfFotogrames;
name=obj.name;
name = strrep(name, '.mpg', '_IMAGES');

    mkdir(folder,name);
    cd([folder,filesep,name]);
% Read one frame at a time.
for k = 1 : nFotogrames
    disp(['frame = ',num2str(k)]);
    I = read(obj, k);

    filename = sprintf('IMG_%d.png', k) ;
    imwrite(I,filename);
end
cd ..
end;

toc;
```

8.3 Algorisme detector de lesions fosques

```
close all
clear all
clc

I=imread('1.png');
figure;imshow(I)
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);

Iadpt=adapthisteq(Igreen);

se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,graythresh(I2)/1.42);

figure;imshow(Ibin);

si=strel('disk',3);
I4=imclose(Ibin,si);
```

```

figure;imshow(I4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%segmentaci?n

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(I4), hy, 'replicate');
Ix = imfilter(double(I4), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag,[]), title('Gradient magnitude (gradmag)');

% Do the WATERSHED
L = watershed(gradmag);

figure;imagesc(L);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

for etiqueta = AreasSim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');

```



```

gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);
figure; imagesc(L2)

L2props=regionprops(L2, 'Area');

for q=1:length(L2props)

    vec=Ired(L2==q);
    edg=1:1:255;
    h=histc(vec,edg);

    A=max(vec);
    B=std2(vec);
    C=mean2(vec);
    D=min(vec);

    if A<193.52 & B<16.71 & B>2.40 & C>37.04 & C<160.02 & D>22.01

        Q_R=entropy(Ired(L2==q));
        Q_G=entropy(Igreen(L2==q));
        Q_B=entropy(Iblue(L2==q));

        if Q_R<5.77 & Q_R>3.54 & Q_G>3.26 & Q_G<5.48 & Q_B>2.65 &
Q_B<4.41

            vecg=Igreen(L2==q);
            Ag=max(vecg);
            Bg=std2(vecg);
            Cg=mean2(vecg);
            Dg=min(vecg);

            if Ag<89.2 & Bg>1.56 & Bg<13.23 & Cg>10.09 & Cg<60.05 &
Dg>0.10

                vecb=Iblue(L2==q);
                Ab=max(vecb);
                Bb=std2(vecb);
                Cb=mean2(vecb);
                Db=min(vecb);

                if Ab<40.49 & Bb>1.95 & Bb<6.70 & Cb<20.81
                    dif_r_g=(A-Ag);
                    dif_r_b=(A-Ab);
                    dif_g_b=(Ag-Ab);

                    if dif_r_g>24.37 & dif_r_b>48.53 & dif_g_b>18.68

                        majoraxis=regionprops(L2==q, 'MajorAxisLength');

```

```
        minoraxis=regionprops(L2==q, 'MinorAxisLength');

        majoraxisL=majoraxis.MajorAxisLength;
        minoraxiL=minoraxis.MinorAxisLength;
        mamincoef=majoraxisL/minoraxiL;
        if mamincoef<2.98
            figure;bar(edg,h);
            q

        end
    end
end

end

end

end
```

8.4 Algorisme detector de lesions clares

```
clear all
close all
clc

I=imread('IMG_645.png');
figure;imshow(I)
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Q=Ired-Igreen;
Iadpt=adapthisteq(Q);

se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,122/255);

figure;imshow(Ibin);

%segmentaci?n
```

```

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ibin), hy, 'replicate');
Ix = imfilter(double(Ibin), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag,[]), title('Gradient magnitude (gradmag)');

% Do the WATERSHED
L = watershed(gradmag);

figure;imagesc(L);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

for etiqueta = AreasSim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');
gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);
figure;imagesc(L2)

L2props=regionprops(L2, 'Area');

for q=1:length(L2props)

```

```

vec=Ired(L2==q);
edg=1:1:255;
h=histc(vec,edg);

A=max(vec);
B=std2(vec);
C=mean2(vec);
D=min(vec);

if A<252.47 & B>3.60 & B<19.74 & C>101.53 & C<203.23 & D>94.98

    Q_R=entropy(Ired(L2==q));
    Q_G=entropy(Igreen(L2==q));
    Q_B=entropy(Iblue(L2==q));

    if Q_R<6.21 & Q_R>4.02 & Q_G>4.02 & Q_G<6.3 & Q_B>3.72 &
Q_B<6.14

        vecg=Igreen(L2==q);
        Ag=max(vecg);
        Bg=std2(vecg);
        Cg=mean2(vecg);
        Dg=min(vecg);

        if Ag<144.64 & Bg>2.05 & Bg<20.74 & Cg>36.19 & Cg<102.24
& Dg>28.23

            vecb=Iblue(L2==q);
            Ab=max(vecb);
            Bb=std2(vecb);
            Cb=mean2(vecb);
            Db=min(vecb);

            if Ab<88.68 & Bb<16.78 & Bb>1.68 & Cb>17.27 &
Cb<49.49

                dif_r_g=(A-Ag);
                dif_r_b=(A-Ab);
                dif_g_b=(Ag-Ab);

                if dif_r_g>60.43 & dif_r_b>84.53 & dif_g_b>21.78

                    majoraxis=regionprops(L2==q, 'MajorAxisLength');
                    minoraxis=regionprops(L2==q, 'MinorAxisLength');

                    majoraxisL=majoraxis.MajorAxisLength;
                    minoraxiL=minoraxis.MinorAxisLength;

```

```
        mamincoef=majoraxisL/minoraxiL;
        if mamincoef<2.042
            figure;bar(edg,h);
            q
        end
    end
end
end
end
end
end
end
```

8.5 Algorisme semiautomàtic detector de llindars

```
close all;
clc;

q=6;

vec=Ired(L2==q);
edg=1:1:255;
h=histc(vec,edg);

A=max(vec);
B=std2(vec);
C=mean2(vec);
D=min(vec);

Q_R=entropy(Ired(L2==q));
Q_G=entropy(Igreen(L2==q));
Q_B=entropy(Iblue(L2==q));

vecg=Igreen(L2==q);
Ag=max(vecg);
Bg=std2(vecg);
Cg=mean2(vecg);
Dg=min(vecg);

vecb=Iblue(L2==q);
Ab=max(vecb);
Bb=std2(vecb);
Cb=mean2(vecb);
Db=min(vecb);

dif_r_g=(A-Ag);
dif_r_b=(A-Ab);
dif_g_b=(Ag-Ab);

majoraxis=regionprops(L2==q,'MajorAxisLength');
minoraxis=regionprops(L2==q,'MinorAxisLength');
```

```
majoraxisL=majoraxis.MajorAxisLength;  
minoraxiL=minoraxis.MinorAxisLength;  
mamincoef=majoraxisL/minoraxiL;
```

8.6 Algorisme automàtic detecció i càlcul de l'indar lesions fosques

```
close all  
clear all  
clc  
  
folder = uigetdir;%busquem el nom del directori  
  
filesindir = dir([folder,filesep,'*.png']);%seleccio de les imatges  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%thresholding calc  
max_red=[];  
std2_red=[];  
mean2_red=[];  
min_red=[];  
  
max_blue=[];  
std2_blue=[];  
mean2_blue=[];  
min_blue=[];  
  
max_green=[];  
std2_green=[];  
mean2_green=[];  
min_green=[];  
  
entr_red=[];  
entr_blue=[];  
entr_green=[];  
  
dif_r_bV=[];  
dif_r_gV=[];  
dif_g_bV=[];  
  
f1=[];  
r=[];  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
for kfile =1:length(filesindir)  
I = imread([folder,filesep,filesindir(kfile).name]);  
figure;imshow(I)  
Ired=I(:,:,1);  
Igreen=I(:,:,2);  
Iblue=I(:,:,3);  
IR=rgb2gray(I);
```

```

Iadpt=adapthisteq(Igreen);

se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,graythresh(I2)/1.42);

figure;imshow(Ibin);

si=strel('disk',3);
I4=imclose(Ibin,si);

figure;imshow(I4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%segmentaci?n

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(I4), hy, 'replicate');
Ix = imfilter(double(I4), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag,[]), title('Gradient magnitude (gradmag)');

% Do the WATERSHED
L = watershed(gradmag);

figure;imagesc(L);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

% Properties of each region
Lprops = regionprops(L,'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

```

```

    AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

    E = L;

    for etiqueta = AreasSim
        E(L == etiqueta) = 0;
    end

    hy2 = fspecial('sobel');
    hx2 = hy2';
    Iy2 = imfilter(double(E), hy2, 'replicate');
    Ix2 = imfilter(double(E), hx2, 'replicate');
    gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

    L2=watershed(gradmag2);
    figure;imagesc(L2)

    L2props=regionprops(L2, 'Area');

    for q=1:length(L2props)

        vec=Ired(L2==q);
        edg=1:1:255;
        h=histc(vec,edg);

        A=max(vec);
        B=std2(vec);
        C=mean2(vec);
        D=min(vec);

        if A<174 & B>5.92 & B<15.5 & C>65.2 & C<148.7 & D>51

            Q_R=entropy(Ired(L2==q));
            Q_G=entropy(Igreen(L2==q));
            Q_B=entropy(Iblue(L2==q));

            if Q_R<5.6 & Q_R>3.8 & Q_G>3.5 & Q_G<5.3 & Q_B>2.9 & Q_B<4.2

                vecg=Igreen(L2==q);
                Ag=max(vecg);
                Bg=std2(vecg);
                Cg=mean2(vecg);
                Dg=min(vecg);

                if Ag<83 & Bg>4.4 & Bg<12.1 & Cg>22.2 & Cg<51.6 Dg>10

                    vecb=Iblue(L2==q);
                    Ab=max(vecb);
                    Bb=std2(vecb);
                    Cb=mean2(vecb);
                end
            end
        end
    end

```



```

        Db=min(vecb);

        if Ab<36 & Bb>2.6 & Bb<6.2 & Cb>4.2 & Cb<18.9
            dif_r_g=(A-Ag);
            dif_r_b=(A-Ab);
            dif_g_b=(Ag-Ab);

            if dif_r_g>42 & dif_r_b>66 & dif_g_b>23

                figure;bar(edg,h);
                q
                f1=[f1 q];

                max_red=[max_red A];
                std2_red=[std2_red B];
                mean2_red=[mean2_red C];
                min_red=[min_red D];

                max_blue=[max_blue Ab];
                std2_blue=[std2_blue Bb];
                mean2_blue=[mean2_blue Cb];
                min_blue=[min_blue Db];

                max_green=[max_green Ag];
                std2_green=[std2_green Bg];
                mean2_green=[mean2_green Cg];
                min_green=[min_green Dg];

                entr_red=[entr_red Q_R];
                entr_blue=[entr_blue Q_B];
                entr_green=[entr_green Q_G];

                dif_r_bV=[dif_r_bV dif_r_b];
                dif_r_gV=[dif_r_gV dif_r_g];
                dif_g_bV=[dif_g_bV dif_g_b];
                majoraxis=regionprops(L2==q, 'MajorAxisLength');
                minoraxis=regionprops(L2==q, 'MinorAxisLength');

                majoraxisL=majoraxis.MajorAxisLength;
                minoraxiL=minoraxis.MinorAxisLength;
                mamincoef=majoraxisL/minoraxiL;

                r=[r mamincoef]
            end
        end
    end
end

end
end

end

r=r((2:(length(r)-1)))

max_red=max_red(2:(length(max_red)-1));
std2_red=std2_red(2:(length(std2_red)-1));
mean2_red=mean2_red(2:(length(mean2_red)-1));

```

```

min_red=min_red(2:(length(min_red)-1));

max_blue=max_blue(2:(length(max_blue)-1));
std2_blue=std2_blue(2:(length(std2_blue)-1));
mean2_blue=mean2_blue(2:(length(mean2_blue)-1));
min_blue=min_blue(2:(length(min_blue)-1));

max_green=max_green(2:(length(max_green)-1));
std2_green=std2_green(2:(length(std2_green)-1));
mean2_green=mean2_green(2:(length(mean2_green)-1));
min_green=min_green(2:(length(min_green)-1));

entr_red=entr_red(2:(length(entr_red)-1));
entr_blue=entr_blue(2:(length(entr_blue)-1));
entr_green=entr_green(2:(length(entr_green)-1));

dif_r_bV=dif_r_bV(2:(length(dif_r_bV)-1));
dif_r_gV=dif_r_gV(2:(length(dif_r_gV)-1));
dif_g_bV=dif_g_bV(2:(length(dif_g_bV)-1));

mamincoef_llin=mean2(r)+2*std2(r)

max_red_llin=mean2(max_red)+2*std2(max_red);
std2_red_llin=mean2(std2_red)+2*std2(std2_red);
mean2_red_llin=mean2(mean2_red)+2*std2(mean2_red);
min2_red_llin=mean2(min_red)+2*std2(min_red);

max_red_llin2=mean2(max_red)-2*std2(max_red);
std2_red_llin2=mean2(std2_red)-2*std2(std2_red);
mean2_red_llin2=mean2(mean2_red)-2*std2(mean2_red);
min2_red_llin2=mean2(min_red)-2*std2(min_red);

max_blue_llin=mean2(max_blue)+2*std2(max_blue);
std2_blue_llin=mean2(std2_blue)+2*std2(std2_blue);
mean2_blue_llin=mean2(mean2_blue)+2*std2(mean2_blue);
min2_blue_llin=mean2(min_blue)+2*std2(min_blue);

max_blue_llin2=mean2(max_blue)-2*std2(max_blue);
std2_blue_llin2=mean2(std2_blue)-2*std2(std2_blue);
mean2_blue_llin2=mean2(mean2_blue)-
2*std2(mean2_blue);
min2_blue_llin2=mean2(min_blue)-2*std2(min_blue);

max_green_llin=mean2(max_green)+2*std2(max_green);
std2_green_llin=mean2(std2_green)+2*std2(std2_green);

mean2_green_llin=mean2(mean2_green)+2*std2(mean2_green);
min2_green_llin=mean2(min_green)+2*std2(min_green);

max_green_llin2=mean2(max_green)-2*std2(max_green);
std2_green_llin2=mean2(std2_green)-
2*std2(std2_green);
mean2_green_llin2=mean2(mean2_green)-
2*std2(mean2_green);
min2_green_llin2=mean2(min_green)-2*std2(min_green);

```

```
entr_red_llin=mean2(entr_red)+2*std2(entr_red);
entr_blue_llin=mean2(entr_blue)+2*std2(entr_blue);
entr_green_llin=mean2(entr_green)+2*std2(entr_green);

entr_red_llin2=mean2(entr_red)-2*std2(entr_red);
entr_blue_llin2=mean2(entr_blue)-2*std2(entr_blue);
entr_green_llin2=mean2(entr_green)-
2*std2(entr_green);

dif_r_bV_llin=mean2(dif_r_bV)+2*std2(dif_r_bV);
dif_r_gV_llin=mean2(dif_r_gV)+2*std2(dif_r_gV);
dif_g_bV_llin=mean2(dif_g_bV)+2*std2(dif_g_bV);

dif_r_bV_llin2=mean2(dif_r_bV)-2*std2(dif_r_bV);
dif_r_gV_llin2=mean2(dif_r_gV)-2*std2(dif_r_gV);
dif_g_bV_llin2=mean2(dif_g_bV)-2*std2(dif_g_bV);
```

8.7 Algorisme automàtic detecció i càlcul de llinars lesions clares

```
close all
clear all
clc

folder = uigetdir;%busquem el nom del directori

filesindir = dir([folder,filesep,'*.png']);%seleccio de les imatges
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%thresholding calc
max_red=[];
std2_red=[];
mean2_red=[];
min_red=[];

max_blue=[];
std2_blue=[];
mean2_blue=[];
min_blue=[];

max_green=[];
std2_green=[];
mean2_green=[];
min_green=[];

entr_red=[];
entr_blue=[];
entr_green=[];

dif_r_bV=[];
dif_r_gV=[];
dif_g_bV=[];
```

```

f1=[];
r=[];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for kfile =1:length(filesindir)
I = imread([folder,filesep,filesindir(kfile).name]);

figure;imshow(I)
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Q=Ired-Igreen;
Iadpt=adapthisteq(Q);

se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,122/255);

figure;imshow(Ibin);

%segmentaci?n

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ibin), hy, 'replicate');
Ix = imfilter(double(Ibin), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag,[]), title('Gradient magnitude (gradmag)');

% Do the WATERSHED
L = watershed(gradmag);

figure;imagesc(L);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

for etiqueta = AreasSim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');
gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);
figure;imagesc(L2)

L2props=regionprops(L2, 'Area');

for q=1:length(L2props)

    vec=Ired(L2==q);
    edg=1:1:255;
    h=histc(vec,edg);

    A=max(vec);
    B=std2(vec);
    C=mean2(vec);
    D=min(vec);

    if A<225 & B>5.6 & B<18.6 & C>112.5 & C<181.4 & D>99.9

        Q_R=entropy(Ired(L2==q));
        Q_G=entropy(Igreen(L2==q));
        Q_B=entropy(Iblue(L2==q));
```

```

if Q_R<5.9 & Q_R>3.99 & Q_G>4.3 & Q_G<6 & Q_B>3.9 & Q_B<5.8

vecg=Igreen(L2==q);
Ag=max(vecg);
Bg=std2(vecg);
Cg=mean2(vecg);
Dg=min(vecg);

if Ag<128 & Bg>5.8 & Bg<20.8 & Cg>41.2 & Cg<94.2 & Dg>29

    vecb=Iblue(L2==q);
    Ab=max(vecb);
    Bb=std2(vecb);
    Cb=mean2(vecb);
    Db=min(vecb);

    if Ab<80 & Bb<16.6 & Bb>4.6 & Cb>16.6 & Cb<42.3
        dif_r_g=(A-Ag);
        dif_r_b=(A-Ab);
        dif_g_b=(Ag-Ab);

    if dif_r_g>67 & dif_r_b>98 & dif_g_b>25

figure;bar(edg,h);
q

f1=[f1 q];

max_red=[max_red A];
std2_red=[std2_red B];
mean2_red=[mean2_red C];
min_red=[min_red D];

max_blue=[max_blue Ab];
std2_blue=[std2_blue Bb];
mean2_blue=[mean2_blue Cb];
min_blue=[min_blue Db];

max_green=[max_green Ag];
std2_green=[std2_green Bg];
mean2_green=[mean2_green Cg];
min_green=[min_green Dg];

entr_red=[entr_red Q_R];
entr_blue=[entr_blue Q_B];
entr_green=[entr_green Q_G];

dif_r_bV=[dif_r_bV dif_r_b];
dif_r_gV=[dif_r_gV dif_r_g];
dif_g_bV=[dif_g_bV dif_g_b];

```

```

majoraxis=regionprops(L2==q, 'MajorAxisLength');
minoraxis=regionprops(L2==q, 'MinorAxisLength');

majoraxisL=majoraxis.MajorAxisLength;
minoraxisL=minoraxis.MinorAxisLength;
mamincoef=majoraxisL/minoraxisL;

r=[r mamincoef]

end
end

end

end

end

end

end

mamincoef_llin=mean2(r)+2*std2(r);

max_red_llin=mean2(max_red)+2*std2(max_red);
std2_red_llin=mean2(std2_red)+2*std2(std2_red);
mean2_red_llin=mean2(mean2_red)+2*std2(mean2_red);
min2_red_llin=mean2(min_red)+2*std2(min_red);

max_red_llin2=mean2(max_red)-2*std2(max_red);
std2_red_llin2=mean2(std2_red)-2*std2(std2_red);
mean2_red_llin2=mean2(mean2_red)-2*std2(mean2_red);
min2_red_llin2=mean2(min_red)-2*std2(min_red);

max_blue_llin=mean2(max_blue)+2*std2(max_blue);
std2_blue_llin=mean2(std2_blue)+2*std2(std2_blue);
mean2_blue_llin=mean2(mean2_blue)+2*std2(mean2_blue);
min2_blue_llin=mean2(min_blue)+2*std2(min_blue);

max_blue_llin2=mean2(max_blue)-2*std2(max_blue);
std2_blue_llin2=mean2(std2_blue)-2*std2(std2_blue);
mean2_blue_llin2=mean2(mean2_blue)-
2*std2(mean2_blue);
min2_blue_llin2=mean2(min_blue)-2*std2(min_blue);

max_green_llin=mean2(max_green)+2*std2(max_green);
std2_green_llin=mean2(std2_green)+2*std2(std2_green);
mean2_green_llin=mean2(mean2_green)+2*std2(mean2_green);
min2_green_llin=mean2(min_green)+2*std2(min_green);

max_green_llin2=mean2(max_green)-2*std2(max_green);
std2_green_llin2=mean2(std2_green)-
2*std2(std2_green);
mean2_green_llin2=mean2(mean2_green)-
2*std2(mean2_green);
min2_green_llin2=mean2(min_green)-2*std2(min_green);

```

```

entr_red_llin=mean2(entr_red)+2*std2(entr_red);
entr_blue_llin=mean2(entr_blue)+2*std2(entr_blue);
entr_green_llin=mean2(entr_green)+2*std2(entr_green);

entr_red_llin2=mean2(entr_red)-2*std2(entr_red);
entr_blue_llin2=mean2(entr_blue)-2*std2(entr_blue);
entr_green_llin2=mean2(entr_green)-
2*std2(entr_green);

dif_r_bV_llin=mean2(dif_r_bV)+2*std2(dif_r_bV);
dif_r_gV_llin=mean2(dif_r_gV)+2*std2(dif_r_gV);
dif_g_bV_llin=mean2(dif_g_bV)+2*std2(dif_g_bV);

dif_r_bV_llin2=mean2(dif_r_bV)-2*std2(dif_r_bV);
dif_r_gV_llin2=mean2(dif_r_gV)-2*std2(dif_r_gV);
dif_g_bV_llin2=mean2(dif_g_bV)-2*std2(dif_g_bV);

```

8.8 Algorisme d'entrenament manual pel classificador estadístic

```

close all
clear all
clc

%create a data base of positive masks

I1=imread('1.png');
I2=imread('2.png');
I3=imread('3.png');
I4=imread('4.png');
I5=imread('5.png');
I6=imread('6.png');
I7=imread('7.png');
I8=imread('IMG_255.png');
I9=imread('IMG_610.png');
I10=imread('IMG_630.png');
I11=imread('IMG_631.png');
I12=imread('IMG_640.png');
I13=imread('IMG_645.png');
I14=imread('IMG_650.png');
I15=imread('IMG_655.png');
I16=imread('IMG_660.png');
I17=imread('IMG_665.png');

%first seventeen masks are more usual types of lesions

falsepositivetype1=roipoly(I1);
falsepositivetype2=roipoly(I2);
falsepositivetype3=roipoly(I3);
falsepositivetype4=roipoly(I4);
falsepositivetype5=roipoly(I5);

```



```
falsepositivetype6=roipoly(I6);
falsepositivetype7=roipoly(I7);
falsepositivetype8=roipoly(I8);
falsepositivetype9=roipoly(I9);
falsepositivetype10=roipoly(I10);
falsepositivetype11=roipoly(I11);
falsepositivetype12=roipoly(I12);
falsepositivetype13=roipoly(I13);
falsepositivetype14=roipoly(I14);
falsepositivetype15=roipoly(I15);
falsepositivetype16=roipoly(I16);
falsepositivetype17=roipoly(I17);

filename='lesion_types.mat';

save(filename, 'falsepositivetype1', 'falsepositivetype2', 'falsepositive
type3', 'falsepositivetype4', 'falsepositivetype5', 'falsepositivetype6',
'falsepositivetype7', 'falsepositivetype8', 'falsepositivetype9', 'falsep
ositivetype10', 'falsepositivetype11', 'falsepositivetype12', 'falseposit
ivetype13', 'falsepositivetype14', 'falsepositivetype15', 'falsepositivet
ype16', 'falsepositivetype17');

x1=imstack2vectors(I1, falsepositivetype1);
x2=imstack2vectors(I2, falsepositivetype2);
x3=imstack2vectors(I3, falsepositivetype3);
x4=imstack2vectors(I4, falsepositivetype4);
x5=imstack2vectors(I5, falsepositivetype5);
x6=imstack2vectors(I6, falsepositivetype6);
x7=imstack2vectors(I7, falsepositivetype7);
x8=imstack2vectors(I8, falsepositivetype8);
x9=imstack2vectors(I9, falsepositivetype9);
x10=imstack2vectors(I10, falsepositivetype10);
x11=imstack2vectors(I11, falsepositivetype11);
x12=imstack2vectors(I12, falsepositivetype12);
x13=imstack2vectors(I13, falsepositivetype13);
x14=imstack2vectors(I14, falsepositivetype14);
x15=imstack2vectors(I15, falsepositivetype15);
x16=imstack2vectors(I16, falsepositivetype16);
x17=imstack2vectors(I17, falsepositivetype17);

[C1, m1]=covmatrix(x1);
[C2, m2]=covmatrix(x2);
[C3, m3]=covmatrix(x3);
[C4, m4]=covmatrix(x4);
[C5, m5]=covmatrix(x5);
[C6, m6]=covmatrix(x6);
[C7, m7]=covmatrix(x7);
[C8, m8]=covmatrix(x8);
[C9, m9]=covmatrix(x9);
[C10, m10]=covmatrix(x10);
[C11, m11]=covmatrix(x11);
[C12, m12]=covmatrix(x12);
[C13, m13]=covmatrix(x13);
[C14, m14]=covmatrix(x14);
[C15, m15]=covmatrix(x15);
[C16, m16]=covmatrix(x16);
[C17, m17]=covmatrix(x17);
```

```
CA
=cat(3,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,C17);
MA
=cat(2,m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17);

filename2='CA_MA.mat'
save(filename2,'CA','MA')
```

8.9 Algorisme d'entrenament per lesions fosques pel classificador estadístic

```
close all
clear all
clc
tic;

folder = uigetdir;%busquem el nom del directori

filesindir = dir([folder,filesep,'*.png']);%seleccio de les imatges
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

CAA=[];
MAA=[];
f1=[];
f2=[];

for kfile =1:length(filesindir)

I = imread([folder,filesep,filesindir(kfile).name]);
%I=imread('IMG_255.png');
figure;imshow(I)
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);

Iadpt=adapthisteq(Igreen);

se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,graythresh(I2)/1.42);
```

```

figure;imshow(Ibin);

si=strel('disk',3);
I4=imclose(Ibin,si);

figure;imshow(I4);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

%segmentaci?n

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(I4), hy, 'replicate');
Ix = imfilter(double(I4), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag,[]), title('Gradient magnitude (gradmag)');

% Do the WATERSHED
L = watershed(gradmag);

figure;imagesc(L);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

```

```

for etiqueta = AreasSim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');
gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);
figure;imagesc(L2)

L2props=regionprops(L2, 'Area');

for q=1:length(L2props)

    vec=Ired(L2==q);
    edg=1:1:255;
    h=histc(vec,edg);

    A=max(vec);
    B=std2(vec);
    C=mean2(vec);
    D=min(vec);

    if A<174 & B>5.92 & B<15.5 & C>65.2 & C<148.7 & D>51

        Q_R=entropy(Ired(L2==q));
        Q_G=entropy(Igreen(L2==q));
        Q_B=entropy(Iblue(L2==q));

        if Q_R<5.6 & Q_R>3.8 & Q_G>3.5 & Q_G<5.3 & Q_B>2.9 & Q_B<4.2

            vecg=Igreen(L2==q);
            Ag=max(vecg);
            Bg=std2(vecg);
            Cg=mean2(vecg);
            Dg=min(vecg);

            if Ag<83 & Bg>4.4 & Bg<12.1 & Cg>22.2 & Cg<51.6 Dg>10

                vecb=Iblue(L2==q);
                Ab=max(vecb);
                Bb=std2(vecb);
                Cb=mean2(vecb);
                Db=min(vecb);

                if Ab<36 & Bb>2.6 & Bb<6.2 & Cb>4.2 & Cb<18.9
                    dif_r_g=(A-Ag);
                    dif_r_b=(A-Ab);
                    dif_g_b=(Ag-Ab);

                    if dif_r_g>42 & dif_r_b>66 & dif_g_b>23

```

```
majoraxis=regionprops (L2==q, 'MajorAxisLength');
minoraxis=regionprops (L2==q, 'MinorAxisLength');

        majoraxisL=majoraxis.MajorAxisLength;
        minoraxisL=minoraxis.MinorAxisLength;
        maminccoef=majoraxisL/minoraxisL;

        if maminccoef<2.65

                x1=imstack2vectors (I, (L2==q));
                [C1 m]=covmatrix(x1);
                CAA=[CAA C1];
                MAA=[MAA m];
                q

        end

        end

        end

        end

        end

        end

        end

        end

CAA=reshape (CAA,3,3, (length (CAA) /3));
        save ('NewCA_MA_2.mat', 'CAA', 'MAA');
        toc;
```

8.10 Algorisme d'entrenament per lesions clares pel classificador estadístic

```
close all
clear all
clc
tic;

folder = uigetdir;%busquem el nom del directori
```

```

filesindir = dir([folder,filesep,'*.png']);%seleccio de les imatges
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

CAA=[];
MAA=[];

for kfile =1:length(filesindir)

I = imread([folder,filesep,filesindir(kfile).name]);
%I=imread('IMG_669.png');
figure;imshow(I)
Ired=I(:,:,1);
Igreen=I(:,:,2);
Iblue=I(:,:,3);
IR=rgb2gray(I);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Q=Ired-Igreen;
Iadpt=adapthisteq(Q);
se=strel('disk',5);

Ie=imerode(Iadpt,se);
Iobr=imreconstruct(Ie,Iadpt);

I2 = medfilt2(Iobr,[15 15]);

Ibin=im2bw(I2,122/255);
figure;imshow(Ibin);

%segmentaci?n

% Use the gradient magnitude as the segmentation function
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Ibin), hy, 'replicate');
Ix = imfilter(double(Ibin), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);

figure;imshow(gradmag,[]), title('Gradient magnitude (gradmag)');

% Do the WATERSHED
L = watershed(gradmag);

figure;imagesc(L);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Properties of each region
Lprops = regionprops(L, 'Area');

% Filter the regions - size

maxlimit=10000;
minlimit=4;

AreasSim = find([Lprops.Area] < minlimit |
[Lprops.Area]>maxlimit);

E = L;

for etiqueta = AreasSim
    E(L == etiqueta) = 0;
end

hy2 = fspecial('sobel');
hx2 = hy2';
Iy2 = imfilter(double(E), hy2, 'replicate');
Ix2 = imfilter(double(E), hx2, 'replicate');
gradmag2 = sqrt(Ix2.^2 + Iy2.^2);

L2=watershed(gradmag2);
figure;imagesc(L2)

L2props=regionprops(L2, 'Area');

for q=1:length(L2props)

    vec=Ired(L2==q);
    edg=1:1:255;
    h=histc(vec,edg);

    A=max(vec);
    B=std2(vec);
    C=mean2(vec);
    D=min(vec);

    if A<252.47 & B>3.60 & B<19.74 & C>101.53 & C<203.23 & D>94.98

        Q_R=entropy(Ired(L2==q));
        Q_G=entropy(Igreen(L2==q));
        Q_B=entropy(Iblue(L2==q));
    end
end

```

```

        if Q_R<6.21 & Q_R>4.02 & Q_G>4.02 & Q_G<6.3 & Q_B>3.72 &
Q_B<6.14

        vecg=Igreen(L2==q);
        Ag=max(vecg);
        Bg=std2(vecg);
        Cg=mean2(vecg);
        Dg=min(vecg);

        if Ag<144.64 & Bg>2.05 & Bg<20.74 & Cg>36.19 & Cg<102.24
& Dg>28.23

            vecb=Iblue(L2==q);
            Ab=max(vecb);
            Bb=std2(vecb);
            Cb=mean2(vecb);
            Db=min(vecb);

            if Ab<88.68 & Bb<16.78 & Bb>1.68 & Cb>17.27 &
Cb<49.49

                dif_r_g=(A-Ag);
                dif_r_b=(A-Ab);
                dif_g_b=(Ag-Ab);

                if dif_r_g>60.43 & dif_r_b>84.53 & dif_g_b>21.78

                    majoraxis=regionprops(L2==q, 'MajorAxisLength');
                    minoraxis=regionprops(L2==q, 'MinorAxisLength');

                    majoraxisL=majoraxis.MajorAxisLength;
                    minoraxisL=minoraxis.MinorAxisLength;
                    mamincoef=majoraxisL/minoraxisL;
                    if mamincoef<2.042

                        x1=imstack2vectors(I, (L2==q));
                        [C1 m]=covmatrix(x1);
                        CAA=[CAA C1];
                        MAA=[MAA m];
                    end
                end
            end
        end
    end
end
end

```



```
end  
end  
end
```

```
CAAA=reshape(CAA,3,3,(length(CAA)/3));  
save('NewCA_MA_3.mat','CAAA','MAA');  
toc;
```