

# Detecció automàtica de capes retinals en imatges OCT

**Jordi Rosés Castellsaguer**  
Grau d'Enginyeria Informàtica  
Àrea d'Intel·ligència Artificial

Director (Consultor): **Dr. Joan M. Nuñez Do Ríó**  
Professor responsable de l'assignatura: **Dr. Carles Ventura Royo**

12 de juny de 2019



Aquesta obra està subjecta a una llicència de Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>“Detecció automàtica de capes retinals en imatges OCT”</i>
<b>Nom de l'autor:</b>	<i>Jordi Rosés Castellsaguer</i>
<b>Nom del consultor/a:</b>	<i>Dr. Joan M. Nuñez Do Rio</i>
<b>Nom del PRA:</b>	<i>Dr, Carles Ventura Royo</i>
<b>Data de lliurament (mm/aaaa):</b>	<i>06/2019</i>
<b>Titulació o programa:</b>	<i>Grau d'Enginyeria Informàtica</i>
<b>Àrea del Treball Final:</b>	<i>Intel·ligència Artificial</i>
<b>Idioma del treball:</b>	<i>Català</i>
<b>Paraules clau</b>	<i>Imatges OCT Segmentació d'imatges Xarxes neuronals de convolució (CNN)</i>

### **Resum del Treball:**

La tomografia de coherència òptica (OCT) és una tecnologia no invasiva que permet obtenir imatges dels teixits oculars. Donada la seva gran utilitat, és necessari el desenvolupament d'eines de suport al diagnòstic que ajudin a processar automàticament la imatge per tal de facilitar-ne l'anàlisi i fer més evidents les característiques de la retina.

En aquest context, es proposa fer un estudi comparatiu entre tres procediments de detecció automàtica de capes retinals en imatges OCT, i en els tres casos transformant un problema de segmentació d'imatges en un altre de classificació de píxels:

- 1) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals de primer i segon ordre com entrades a un model de classificació tipus Random Forest.
- 2) Detecció de fronteres, també amb dues classes, i fent servir subimatges com entrades a un model de xarxa neuronal de convolució (CNN).
- 3) Classificació dels píxels de la imatge entre múltiples regions (“fons d'imatge”, “interior globus ocular”, “fronteres entre capes”, “capa retinal 1”, “capa retinal 2”, etc), i fent servir subimatges com entrades a un model de xarxa neuronal de convolució (CNN).

Veurem que s'obtenen millors resultats plantejant l'exercici com un problema de dues classes (amb valors AUC per sobre de 0,95), encara que la opció multi-classe, amb resultats no tan bons de classificació, ens permet estalviar procediments posteriors de processat dels resultats.

**Abstract:**

Optical Coherence Tomography (OCT) is a non-invasive technology that allows images of ocular tissues to be obtained. Given its usefulness, it is necessary to develop diagnostic support tools that help automatically process the image in order to facilitate its analysis and to make retina features more evident.

In this context, our proposal is to make a comparative study between three automatic processes for detection of retinal layers in OCT images, all of them transforming an image segmentation problem into a pixel classification one:

- 1) Border detection with two classes (“no border” vs “border”) and using local first and second order statistics as entries to a Random Forest type classification model.
- 2) Border detection, also with two classes, and using sub-images as inputs to a Convolutional Neural Network (CNN) model.
- 3) Classification of the pixels in the image between multiple regions (“image background”, “inner eyeball”, “boundaries between layers”, “retinal layer 1”, “retinal layer 2”, etc.), and using sub-images such as inputs to a Convolutional Neural Network (CNN) model.

We will see that better results are obtained by considering the exercise as a two-class problem (with AUC values above 0.95). Even though the multi-class option does not yield as good classification results, it eliminates the necessity of any subsequent result processing.

# Índex

<b>1. Introducció</b> .....	<b>1</b>
1.1 Context i justificació del treball .....	1
1.2 Objectius del treball .....	4
1.3 Enfocament i mètodes proposats .....	4
1.4 Planificació del treball .....	7
1.5 Breu sumari de productes obtinguts .....	9
1.6 Breu descripció dels capítols de la memòria .....	10
<b>2. Dades</b> .....	<b>11</b>
2.1 Origen de les dades .....	11
2.2 Format de les dades .....	12
<b>3. Models de classificació</b> .....	<b>13</b>
3.1 Introducció .....	13
3.2 Random Forest .....	15
3.3 Xarxes Neuronals de Convolució (CNN) .....	16
<b>4. Mètodes</b> .....	<b>18</b>
4.1 Introducció .....	18
4.2 Tractament i processat de les dades .....	20
Reducció del soroll .....	21
Normalització dels nivells de gris .....	22
Descomposició de les imatges en subimatges més petites .....	22
Extracció de característiques locals .....	24
“Targets” .....	25
Estructura de dades .....	27
4.3 Mostreig .....	28
Balanceig de classes .....	28
Entrenament / Validació / Test .....	28
4.4 Definició del model .....	29
4.5 Validació del model .....	31

Validació en fase d'entrenament .....	31
Validació en fase de test .....	31
Corbes ROC .....	33
4.6 Post-processat dels resultats del model .....	33
4.7 Particularitats de cada mètode .....	35
<b>5. Experiments i resultats .....</b>	<b>36</b>
5.1 Introducció .....	36
5.2 Mètode 1:	
Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals com entrada .....	36
5.3 Mètode 2:	
Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir subimatges com entrada .....	43
5.4 Mètode 3:	
Classificació de píxels de la imatge entre múltiples regions, fent servir subimatges com entrada .....	51
5.5 Discussió .....	54
<b>6. Conclusions .....</b>	<b>58</b>
<b>7. Bibliografia .....</b>	<b>60</b>

## Llista de figures

Figura 01.	Capas de la retina etiquetades [2-06] .....	2
Figura 02.	Organització temporal de les tasques .....	9
Figura 03.	Fotograma original etiquetat (6 capas) .....	12
Figura 04.	Imatge OCT original .....	12
Figura 05.	Fase de processat de les dades .....	19
Figura 06.	Fase de mostreig, definició de models i post-processat .....	19
Figura 07.	Fase d'entrenament i validació de models .....	20
Figura 08.	Descomposició de la imatge en subimatges més petites (procediment "sliding window") .....	22
Figura 09.	Imatge OCT amb regió de 25x25 píxels .....	23
Figura 10.	Subimatges de 9x9, 15x15 i 25x25 píxels .....	24
Figura 11.	Imatge d'etiquetes ("targets groundtruth") processades amb dues classes ("píxels no frontera" vs "píxels frontera") .....	26
Figura 12.	Etiquetes multi-classe ("groundtruth") .....	27
Figura 13.	Secció vertical (columna de píxels) d'una imatge de probabilitats .....	34
Taula 14.	Optimització d'hyper-paràmetres amb models Random Forest .....	37
Figura 15.	Corba ROC per a subimatges de mida 35x35 .....	38
Figura 16.	Imatge OCT original i imatge d'etiquetes amb les fronteres originals ("groundtruth") entre regions .....	39
Figura 17.	Imatge amb les probabilitats de cada píxel de ser una frontera entre capas .....	39
Figura 18.	Imatges de probabilitats binaritzades en funció de diferents llindars .....	40
Figura 19.	Secció vertical (columna de píxels) de la imatge de probabilitats amb els màxims locals en vermell .....	41
Figura 20.	Fronteres entre capas retinals obtingudes per l'addició dels màxims locals de cada secció vertical .....	42
Figura 21.	Fronteres entre capas retinals obtingudes per l'addició d'únicament 6 dels màxims locals de cada secció vertical .....	42
Taula 22.	Valors de validació per a xarxa CNN en fase d'optimització de paràmetres .....	43
Figura 23.	Arquitectura de xarxa CNN .....	44
Taula 24.	Valors de validació per a xarxa CNN en fase d'entrenament .....	44
Taula 25.	Valors de validació per a xarxa CNN en fase de test .....	45
Figura 26.	Evolució del F1-Score en funció de la mida de la subimatge .....	45
Figura 27.	Corba ROC per a subimatges de mida 35x35 .....	46
Figura 28.	Imatge OCT original i imatge d'etiquetes amb les fronteres originals ("groundtruth") entre regions .....	46
Figura 29.	Imatge amb les probabilitats de cada píxel de ser una frontera entre capas .....	47
Figura 30.	Imatges de probabilitats binaritzades en funció de diferents llindars .....	48

Figura 31.	Secció vertical (columna de píxels) de la imatge de Probabilitats amb els màxims locals en vermell .....	49
Figura 32.	Fronteres entre capes retinals obtingudes per l'addició dels màxims locals de cada secció vertical .....	50
Figura 33.	Fronteres entre capes retinals obtingudes per l'addició d'únicament 6 dels màxims locals de cada secció vertical .....	50
Taula 34.	Valors de validació per a xarxa CNN amb un bloc "conv-pool" .....	52
Taula 35.	Valors de validació per a xarxa CNN amb doble capa "conv"...	52
Figura 36.	Arquitectura de xarxa CNN .....	53
Figura 37.	Imatge OCT original i imatge d'etiquetes amb les regions originals ("groundtruth") .....	53
Figura 38.	Imatge amb la classificació de cada píxel .....	54
Figura 39.	Exemples de subimatge de 25x25 píxels extrems de la imatge i etiquetades com a negatius i positius .....	55
Figura 40.	Histogrames d'intensitats de gris dels exemples de subimatges etiquetades com a negatius o positius .....	56



# 1. Introducció

## 1.1. Context i justificació del treball

La Tomografia de Coherència Òptica (“Optical Coherence Tomography” o OCT) és una tecnologia que, mitjançant l'ús de la llum, permet obtenir imatges de les capes sub-superficials de materials translúcids o opacs a una resolució equivalent a la d'un microscopi de baix consum [2-02].

Al llarg de les darreres dècades, aquesta tecnologia ha rebut molt reconeixement en la comunitat mèdica, i en concret, en Oftalmologia, pel fet de ser una tecnologia no invasiva amb la que obtenir imatges dels teixits oculars, i a més, amb una resolució molt més alta (millor que 10  $\mu\text{m}$ ) que altres modalitats d'imatge com la ressonància magnètica o els ultrasons [2-02].

També en Oftalmologia, aquest tipus d'imatges permeten analitzar la retina i el segment anterior de l'ull, així com avaluar l'organització cel·lular, la integritat del fotorceptor i el gruix axonal en el glaucoma, la degeneració macular, l'edema macular diabètic, l'esclerosi múltiple i altres malalties oculars o patologies sistèmiques que presentin signes oculars [2-01, 2-02, 2-03].

En general, els principals avantatges de la OCT són:

- S'obtenen imatges de superfície en viu a una resolució quasi microscòpica
- S'obtenen imatges directes i instantànies de la morfologia dels teixits
- No hi ha preparació de la mostra ni del subjecte
- No s'utilitzen radiacions ionitzants

Quant a les imatges generades, les A-scan s'obtenen mitjançant exploracions de profunditat axial, i proporcionen informació relacionada amb la longitud de l'ull, mentre que la tomografia de tall transversal (B-scan) pot obtenir-se mitjançant la combinació lateral d'exploracions.

En la imatge OCT de la figura 01 [2-06] podem observar una secció de la retina amb les diferents capes que la conformen etiquetades:

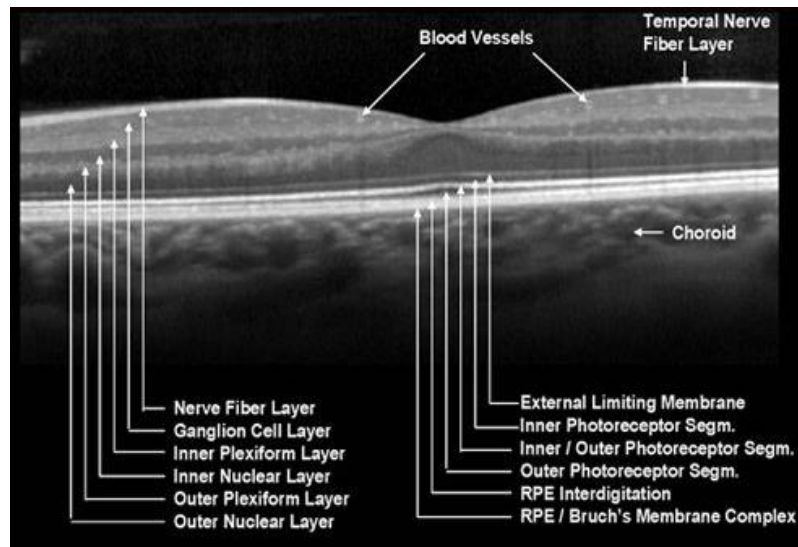


Figura 01. Capes de la retina etiquetades [2-06]

Des de la seva presentació l'any 1991 [1-30], la OCT ha anat creixent en popularitat fins a estar avui totalment consolidada com un estàndard en Oftalmologia per la seva capacitat d'informar a temps real sobre l'estructura ocular, el diagnòstic de malalties, l'avaluació de la seva progressió, per a avaluar la resposta de teràpies, així com per a ajudar a entendre la patogènia de malalties i crear noves teràpies. En trobem aplicacions tan en clíniques especialitzades, com en recerca i fabricació [1-12].

No obstant això, l'anàlisi i la segmentació manual d'aquest tipus d'imatges presenta, entre altres, els següents inconvenients:

- Costosa (temporal)
- Massa subjectiva
- Gran variabilitat

Per tant, amb l'ànim d'incrementar el benefici que comporta l'ús d'aquesta tecnologia, creiem que seria de molta utilitat desenvolupar una eina que en millori la usabilitat, i per tant, proposem dissenyar un sistema capaç de segmentar aquest tipus d'imatges de forma automàtica.

En visió artificial, la segmentació d'imatges és el procés mitjançant el qual es divideix el contingut de la imatge digital en diferents fragments o conjunts de píxels en funció d'un criteri determinat, normalment amb la intenció d'obtenir una major claredat de la informació que hi apareix.

Apart d'aquest factor, creiem que, en concret, segmentar les imatges OCT comportaria una sèrie d'avantatges molt atractius:

- Facilitaria molt la feina de l'oftalmòleg que podria analitzar les seccions de la retina més ràpid (cost temporal) i de forma molt més eficient.
- Permetria assolir uns nivells d'estandardització elevats i, per tant, superar la subjectivitat inherent de l'anàlisi per humans.
- Permetria reduir costos relacionats amb l'anàlisi de les imatges
- Amb major precisió, l'oftalmòleg pot fer un millor diagnòstic, i per tant, pot anticipar-se a l'aparició de possibles patologies i potser evitar lesions.

A més, la incorporació de procediments d'Intel·ligència Artificial al processat d'imatges d'aquest tipus permetria efectivament en una primera fase visualitzar de forma nítida les diferents regions de la imatge, però també establiria un bon punt de partida per al futur anàlisi d'altres aspectes de la imatge com, per exemple, detecció i reconeixement de possibles patologies.

Per tant, en aquest treball, sense ànims de pretendre obtenir un sistema que arribi a substituir les qualitats de l'expert, ens proposem dissenyar una eina que l'ajudi a segmentar o a detectar-ne ràpidament les diferents capes retinals.

En aquesta direcció, revisem la bibliografia relacionada amb sistemes que hagin adreçat el mateix objectiu i en trobem múltiples referències molt interessants:

Kafieh et al. [1-13] proposen un mètode de segmentació de capes retinals basat en l'ús de mapes de difusió, que utilitza mesures estadístiques per a caracteritzar la textura local de la distribució d'intensitat de la imatge. Aquest procediment mostra bons resultats fins i tot en condicions de poc contrast en les que resulta més difícil trobar la frontera entre capes. El mètode es va provar amb 23 conjunts de dades de dos grups de pacients (amb glaucoma i sense). També Cabrera et al. [1-16] proposen un mecanisme per a segmentar les capes retinals utilitzant mesures de textura estadística. Chiu et al. [1-17] presenten un treball per a la segmentació automàtica de capes retinals utilitzant teoria de grafs i programació dinàmica. Els resultats exposats mostren que aconseguen segmentar amb exactitud vuit capes en ulls normals d'adult amb un grau d'èxit comparable al d'un expert.

De la mateixa manera, per a la segmentació d'imatges OCT amb seccions retinals, Guha Roy et al. [1-08] proposen una arquitectura profunda totalment de convolució (ReLayNet), per a la segmentació de capes de retina i masses de fluids en

exploracions OCT de l'ull. ReLayNet recorre una primera seqüència de blocs de convolució ("encoders" o codificadors) amb els que aprèn una jerarquia de característiques contextuais, i tot seguit, passa per una segona seqüència de blocs de convolució ("decoders" o descodificadors) amb els que s'obté la segmentació semàntica de la imatge.

## 1.2. Objectius del treball

En el context del que acabem d'exposar, l'objectiu principal que ens proposem assolir amb aquest treball és el de dissenyar un procediment basat en Intel·ligència Artificial capaç de detectar de forma automàtica múltiples capes de la retina en imatges OCT, amb uns resultats suficientment bons i amb uns temps d'execució suficientment petits.

Com a sub-objectius necessaris per tal d'aconseguir concloure amb èxit l'empresa que ens proposem, plantegem els següents punts:

- Revisió de literatura sobre segmentació d'imatges, i en concret, d'imatges mèdiques.
- Revisió de literatura sobre models basats en xarxes de convolució (CNN).
- Tractament i processat de les dades.
- Implementació d'un "pipeline" d'entrenament, validació i test dels models de classificació escollits.
- Disseny i optimització de models de classificació, i en especial, de xarxes neuronals amb arquitectures profundes, que funcionin correctament amb les dades que hem processat prèviament.

Per sobre dels objectius eminentment pràctics que acabem d'exposar, des d'un punt de vista personal, ens proposem endegar aquest estudi amb la intenció última de comprendre com funcionen les xarxes neuronals profundes.

## 1.3. Enfocament i mètodes proposats

Una de les primeres decisions que prenem en arrencar aquest treball és la de substituir un problema de segmentació d'imatges per un altre problema de classificació de píxels.

Per tant, quan parlem aquí de segmentació d'imatges ho fem referint-nos a la implementació de procediments d'aprenentatge supervisat que

assignin a cada un dels píxels de la imatge una probabilitat de pertànyer a  $n$  classes conegudes amb anterioritat.

Sense intenció de fer una taxonomia exhaustiva d'alternatives ni tampoc d'entrar aquí en el detall de models específics, exposem a continuació algunes disjuntives interessants a tenir en compte abans d'arrencar les feines:

a) Classificació de subimatges o d'imatges senceres:

Una de les estratègies més conegudes a l'hora de classificar píxels d'una imatge rep el nom de "Sliding-Window". Aquesta consisteix en dividir la imatge en sub-imatges més petites, normalment quadrades i de mida senar, centrades en els píxels que volem classificar. Si desitgem utilitzar models tradicionals de classificació, obtenim a partir de cada fragment d'imatge una sèrie de característiques (p. ex. estadístics de primer i segon ordre) de la funció d'intensitat corresponent, i utilitzem aquests com a base de dades d'entrada al sistema. Si per contra volem treballar amb arquitectures neuronals profundes, les entrades del sistema esdevenen les subimatges, i mitjançant diferents nivells d'anàlisi (les capes de la xarxa) aprenem les característiques necessàries per a realitzar la classificació.

Actualment, aquests plantejaments segueixen sent molt útils, però disposem també de models que permeten classificar imatges senceres sense necessitat de fragmentar-les en subimatges més petites. Aquest és el cas de les Fully Convolutional Networks (FCN) [1-08, 1-10], un tipus de xarxes basades en dues fases: "downsampling" i "upsampling". En aquest cas l'entrada del sistema és la imatge completa, sense fragmentar, i la sortida esdevé la mateixa imatge segmentada. Les FCN s'estructuren seqüencialment al voltant d'un primer recorregut amb el que es captura el context i, amb un segon camí expansiu simètric respecte el primer que permet una localització precisa.

b) Problema amb dues classes o amb més de dos:

En funció de com s'han processat les etiquetes de cada píxel en la imatge, la classificació d'aquests pot plantejar-se com un problema de dues classes o multi-classe.

En el primer cas, disposem únicament de 2 classes de píxels: els píxels corresponents a regions de la imatge, i els píxels corresponents a fronteres entre regions.

En el problema multi-classe, cada regió de la imatge ve etiquetada com a una regió independent. En primer lloc, tenim igualment la

classe dels píxels que coincideixen amb fronteres entre regions, i després tenim tantes etiquetes com regions hi ha en la imatge.

c) Amb extracció de característiques o sense:

Abans de l'enorme desenvolupament que hem vist la darrera dècada dels models basats en xarxes profundes (Deep Learning), era comú, un cop fragmentada la imatge en subimatges més petites, extreure una nova base de dades amb mesures estadístiques de primer i segon ordre calculades a partir de la distribució de la funció d'intensitat (en nivells de gris o color) en cada una de les subimatges. Aquesta és una tasca de processat amb un cost gegantí de temps i computació que feia moltes d'aquestes estratègies inviables.

Per una banda, la possibilitat d'executar feines d'aquest tipus amb grans costos computacionals en el núvol (p. ex. AWS o Google Cloud Platform) permet resoldre part del problema.

Per una altra, el desenvolupament d'algorismes basats en xarxes neuronals profundes de convolució (CNN) permet alimentar els models de classificació directament amb les imatges (o subimatges) disponibles, de manera que ja no és necessària la fase d'extracció de característiques. A més, apart de l'avantatge que per si sol ja representa estalviar fases com aquesta, més endavant comentarem la millora en resultats d'aquests procediments.

En funció de quina esdeingui l'estratègia escollida, aquesta determinarà clarament el tipus de processat de les dades i el tipus de model seleccionat, i en definitiva, tot l'algorisme d'aprenentatge, i per tant, cal tenir clar des d'un punt de vista d'estratègia, què es vol fer.

A mode de comparativa, i basant-nos en el que acabem d'exposar, proposem resoldre el problema de la classificació de píxels mitjançant 3 estratègies diferents:

- 1) Detecció de fronteres amb dues classes ("no frontera" vs "frontera") i fent servir estadístics locals de primer i segon ordre com entrades al model
- 2) Detecció de fronteres amb dues classes ("no frontera" vs "frontera") i fent servir subimatges com entrades al model
- 3) Classificació dels píxels de la imatge entre múltiples regions ("fons d'imatge", "interior globus ocular", "fronteres entre capes",

“capa retinal 1”, “capa retinal 2”, etc), i fent servir subimatges com entrades al model

En els tres casos es tracta de classificar cada un dels píxels de la imatge per separat, i no tota la imatge de cop com a única entrada al model, de manera que caldrà aplicar el model de classificació que correspongui a cada una de les regions que envolten el píxel (subimatges).

#### **1.4. Planificació del treball**

Les principals tasques en que s'estructura aquest treball es poden agrupar de la següent manera:

- a) Introducció:
  - Revisió de bibliografia relacionada amb l'anàlisi d'imatges OCT
  - Revisió de bibliografia relacionada amb segmentació automàtica d'imatges mitjançant classificació de píxels
  - Revisió de bibliografia relacionada amb models de classificació, i en concret, amb models CNN
- b) Pre-processament de les imatges:
  - Importació d'imatges i vídeo
  - Estandardització de les mides de les imatges
  - Extracció d'etiquetes a partir de vídeo
  - Identificació de les etiquetes corresponents a cada capa retinal
  - Emmagatzematge de les imatges en format .csv (.txt)
- c) Implementació del codi per a l'entrenament i la validació de models de classificació:
  - Càrrega d'imatges pre-processades
  - Processament per a ML
  - Separació dels conjunts per a entrenament, validació i test
  - Definició de l'arquitectura dels models
  - Entrenament de models
  - Validació de models (mesures de "performance")
  - En cas de mals resultats, repetir tasques a partir de la definició de l'arquitectura
- d) Redacció del document
  - Redactat amb antecedents sobre anàlisi d'imatges OCT
  - Redactat amb antecedents sobre segmentació automàtica d'imatges

- Redactat sobre les característiques de la base de dades, i el seu processat
- Redactat sobre els models usats
- Redactat sobre la "performance" obtinguda (presentació de resultats)
- Redactat de conclusions
- Preparació d'índexs, figures, formats específics, etc.

Els recursos necessaris per a poder dur les tasques anteriors a bon port es poden dividir bàsicament en 2 tipus de programari:

1) Programari per al processament de les imatges:

Les imatges es troben originàriament en diferents mides i en formats .avi (vídeo) i .tiff, de manera que és necessari pre-processar-les. En primer lloc, cal escriure el codi que permeti llegir les imatges en qualsevol dels seus formats, modificar les seves mides, extreure les imatges etiquetades del format vídeo, etc.

Per a poder resoldre aquesta fase de forma ràpida utilitzarem MATLAB. Aquest llenguatge ens permet importar les imatges (i els vídeos), treballar-les com a matrius i emmagatzemar-les com a .png i .csv (o .txt) de forma molt senzilla sense haver d'escriure gaires línies de codi.

2) Programari per a la implementació del codi per a l'entrenament i la validació de models de classificació:

Un cop emmagatzemades totes les imatges (juntament amb les matrius que contenen les etiquetes), cal escriure tots els codis necessaris per a l'entrenament i la validació de la xarxa. Aquesta part l'executarem amb llenguatge Python, i utilitzarem principalment les llibreries Numpy i Pandas per a la manipulació general de les dades, i Keras per a la definició de l'arquitectura de la xarxa i per al seu entrenament i validació.

Apart de les eines que ens proporcionin aquestes llibreries per a validar la xarxa, utilitzarem també altres mesures de qualitat per a mesurar la "performance" de tot el sistema (p. ex. F1-Score i altres) definides per nosaltres mateixos.

No es preveu la utilització de cap procediment basat en BigData ni d'execució d'entrenament de xarxes en el núvol, per tant, en principi, totes les execucions tindran lloc en entorns locals.



Per a l'organització temporal de les feines es proposa el diagrama de la figura 02.

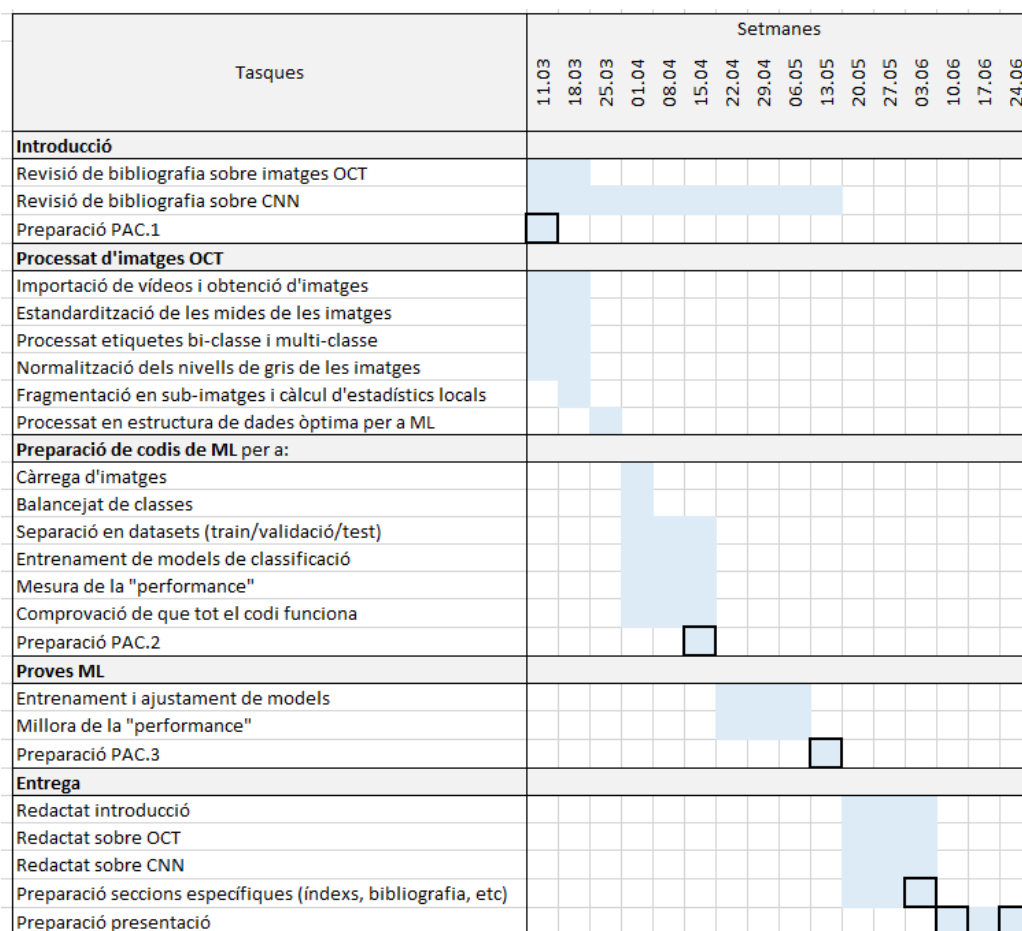


Figura 02. Organització temporal de les tasques

### 1.5. Breu sumari de productes obtinguts

En aquest treball ens proposem fer recerca sobre diferents procediments per a la detecció de capes retinals en imatges OCT, per tant, els productes obtinguts i mostrats al final del treball consisteixen eminentment en descripcions sobre arquitectures específiques per a diferents models de classificació, així com taules de resultats.

A més, el resultat final que ens ha de permetre realment validar la feina feta és l'exposició d'imatges OCT amb les capes retinals correctament identificades fent ús dels models entrenats.

## **1.6. Breu descripció dels altres capítols de la memòria**

En el segon capítol hi exposem de forma breu l'origen de les dades, explicant com, quan i en quines condicions es van obtenir, per a quin tipus de pacients, i entre altres, en quin format les trobem.

En el capítol tercer hi expliquem les estratègies que hem considerat en termes de models de classificació, quins models hem decidit utilitzar, el perquè, i quines són les seves característiques més rellevants.

En el capítol 4 entrem ja en detall i expliquem amb profunditat, per a cada una de les 3 estratègies de segmentació adoptades, tots els passos que constitueixen els seus "pipelines" de processat de dades, entrenament, validació i optimització de models, així com els mecanismes de post-processat adoptats.

En el capítol 5 hi exposem, per a cada una de les 3 estratègies de segmentació plantejades, els resultats obtinguts. Tanquem el capítol amb un darrer apartat de discussió en el que fem una revisió crítica dels productes presentats.

Finalment, adjuntem un capítol amb les conclusions.

## 2. Dades

### 2.1. Origen de les dades

Les imatges SD-OCT ("Spectral Domain - Optical Coherence Tomography") que hem utilitzat en aquest treball han estat seleccionades d'un registre electrònic de pacients amb degeneració macular relacionada amb l'edat entre els anys 2009-2014 del Moorfields Eye Hospital de Londres. Els pacients són tant homes com dones entre 49 i 92 anys. Per tal de formar part de l'estudi, aquests havien d'estar diagnosticats clínicament de degeneració macular deguda a la mitjana edat. Els pacients seleccionats, per tant, formaven part d'un tractament per AMD avançada en un ull, i tenien un historial de tres anys de AMD intermèdia.

Les imatges van ser recaptades com a imatges SD OCT Topcon (Topcon Medical Systems, Oakland, NJ) i van ser seleccionades en funció de la seva bona resolució de cara a la segmentació. Els subjectes amb patologies maculars diferents de la presència de druses van ser exclosos de l'estudi. Tots els pacients amb cubs maculars van ser escanejats amb 512 A-scans i 128 B-scans (nominalment, quadrats de 6x6 mm centrats en la fovea). Es va utilitzar un dispositiu 3D OCT-2000, amb una resolució axial d'imatge de 6µm i una velocitat de captació de 50.000 escanejats axials per segon.

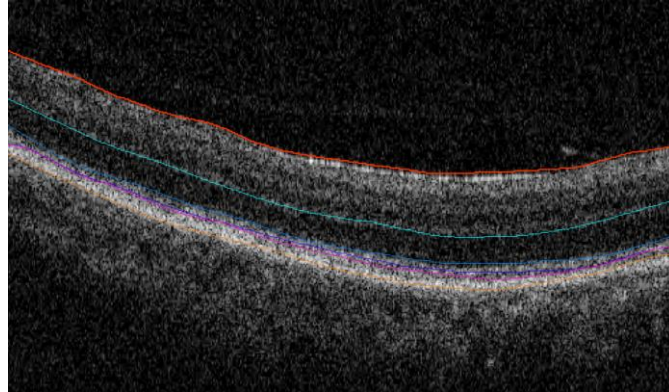
Cada un dels 128 fotogrames (B-scans) ha estat segmentat inicialment de forma automàtica amb un programa de Iowa Reference Algorithms (Retinal Image Analysis Lab, Iowa Institute for Biomedical Imaging, Iowa City, IA) [1-19, 1-20]. Posteriorment, tots els errors de segmentació en cada un dels fotogrames han estat corregits manualment.

Les fronteres etiquetades amb la segmentació són 6 (fig.03), i es corresponen amb la membrana limitadora interna (ILM), el límit entre la capa plexiforme exterior (OPL) i la capa nuclear externa (ONL), el límit entre el mioide i l'el·lipsoide dels segments interns, el límit entre el segment interior (IS) i el segment exterior d'unió (OSJ) i la vora interior i exterior de l'epiteli pigmentari de la retina (RPE).

## 2.2. Format de les dades

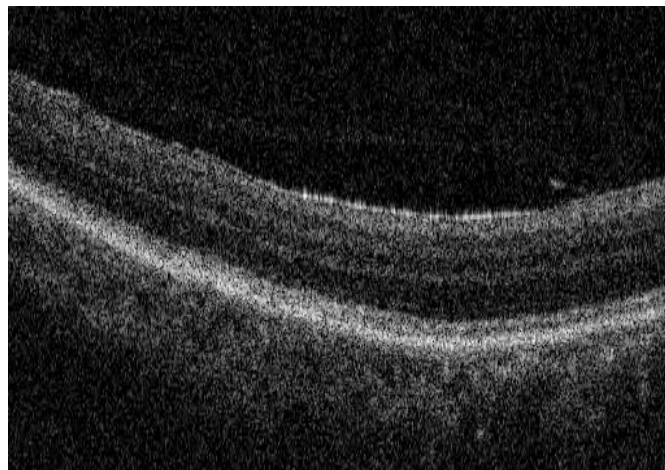
La base de dades amb la que treballem s'obté amb dos tipus diferents de fitxers.

Per una banda, disposem de 12 fitxers de vídeo en format .avi, cada un d'ells construïts a partir de la seqüència de 128 fotogrames. Aquests fotogrames són cada una de les imatges OCT amb què treballem, i contenen les 6 etiquetes "groundtruth" que hem comentat (fig.03).



*Figura 03. Fotograma original etiquetat (6 capes)*

Per altra banda, disposem de 12 paquets d'imatges en format .tiff, cada un dels quals conté 128 imatges diferents. Aquestes imatges són les mateixes que podem trobar en el vídeo, però no venen etiquetades (fig.04), per tant són les imatges OCT que segmentarem.



*Figura 04. Imatge OCT original*

## 3. Models de classificació

### 3.1. Introducció

Prenent com a punt de partida les disjuntives exposades en el primer capítol, aprofundim en aquest ja una mica més en els detalls de les estratègies adoptades de classificació.

Tal com hem comentat en apartats anteriors, ens proposem fer una comparativa entre 3 estratègies de classificació:

- 1) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals de primer i segon ordre com entrades al model
- 2) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir subimatges com entrades al model
- 3) Classificació dels píxels de la imatge entre múltiples regions (“fons d’imatge”, “interior globus ocular”, “fronteres entre capes”, “capa retinal 1”, “capa retinal 2”, etc), i fent servir subimatges com entrades al model

Per tant, una de les primeres decisions que prenem a l'hora d'iniciar el projecte és la de transformar un problema tradicional de segmentació d'imatges en un altre problema de classificació de píxels.

En aquest sentit, diversos investigadors han aplicat ja la mateixa estratègia fent ús de models de classificació de diferents tipus. En concret, pel que fa a l'aplicació de models tipus Random Forest trobem múltiples referències:

Per una banda, Schroff et al. [1-21] proposen un procediment per a classificar píxels amb models Random Forest. La contribució d'aquest article és triple. Per una banda, mostren com classificadors K-Nearest Neighbors poden mapejar-se en arquitectures Random Forest. En segon lloc, demostren que la capacitat discriminatòria d'aquests classificadors millora incorporant el context espacial. Finalment, demostren que la capacitat de Random Forest de combinar múltiples funcions condueix a un augment del rendiment quan s'utilitzen

simultàniament variables més típiques de visió (p.e. colors, bancs de filtres i funcions HOG).

També Knauer et al. [1-22] proposen un nou enfocament per a la segmentació d'imatges d'alta qualitat basada en un espai de característiques ric i d'alta dimensionalitat, i utilitzen Random Forest per solucionar diversos problemes de segmentació binaris i multi-classe. És una propietat intrínseca de l'enfocament basat en arbres de decisió, que qualsevol decisió es basa només en un petit subconjunt de característiques d'entrada. Per tant, l'ús d'arbres els permet fer una extracció de característiques seqüencial, i mostren que l'enfocament permet una segmentació ràpida i de gran qualitat. A més, un avantatge d'aquest enfocament és que es pot utilitzar fàcilment en entorns de paral·lelització.

Lefkovits et al. [1-23] utilitzen un algoritme de Random Forest per tractar conjunts de dades que per la seva mida esdevenen intractables, i l'utilitzen per a reduir-ne les dimensions i per a segmentar imatges de tumors cerebrals.

Altres investigadors, en canvi, han optat per utilitzar models basats en Xarxes Neuronals de Convulsió (CNN) amb el mateix propòsit:

Per exemple, Giusti et al. [1-24] proposen un sistema per a segmentar imatges amb xarxes profundes de convulsió evitant costos alts de computació accelerant el procés diversos ordres de magnitud fent ús de programació dinàmica.

També Gouk et al. [1-25] proposen un procediment per a classificar píxels en imatges, fent èmfasi també ells en la necessitat d'accelerar el processos.

Tschopp et al. [1-26], conscients de la quantitat de càlculs redundants que es realitzen amb el procediment de "sliding-window", proposen una extensió de la llibreria Caffe per mirar d'augmentar el rendiment de sistemes de segmentació amb arquitectures CNN.

A més, davant la disjuntiva que plantejàvem de treballar amb la imatge sencera ("Fully Convolutional Networks") o amb extracció i anàlisi de subimatges ("sliding window"), també hem indicat que els tres plantejaments escollits girarien entorn a la segona opció.

El procediment de finestres corredisses (en anglès "sliding window") ha estat un dels més usats les darreres dècades per a l'extracció i processat d'informació a partir d'imatges, i la bibliografia que pot trobar-se al respecte és molt extensa:

Fang et al. [1-06] i Alonso-Caneiro et al. [1-03] generen per a cada imatge OCT un mapa amb les probabilitats d'ocurrència de cada capa retinal. Per aconseguir-ho, fragmenten les imatges OCT en subimatges més petites i les utilitzen com a entrades en una Xarxa Neuronal de Convolució (CNN). La sortida d'aquesta xarxa entrenada són, d'aquesta manera, les probabilitats d'ocurrència de cada capa retinal associades a cada píxel concret de la imatge. A partir d'aquest mapa de probabilitats, s'obté la segmentació discreta final de cada píxel mitjançant un mètode de cerca gràfica.

Glumov et al. [1-28] proposen el procediment de "sliding window" per a processar imatges i detectar objectes, i a més, ho fan de forma efectiva en termes de complexitat computacional i de qualitat de detecció. Presenten les principals etapes de la transformació de dades: processament preliminar d'imatges, càlcul recursiu de funcions locals, generació del camp de valors per a la funció discriminant i localització d'objectes.

Helwan et al. [1-29] dissenyen també un sistema d'aprenentatge automàtic basat en finestres corredisses, aquest cas per al reconeixement i detecció de ventricles esquerres en imatges cardíques MR.

De les 3 estratègies que ens interessa analitzar en aquesta comparativa, i recolzant-nos en la bibliografia revisada, decidim utilitzar un model de classificació tipus Random Forest en el primer plantejament, idoni per a entrades de tipus estadístic, mentre que per a les estratègies 2 i 3 ens centrem en un model basat en xarxes neuronals profundes de convolució idoni per a entrades en forma d'imatge.

A continuació, expliquem les principals característiques de cada un d'aquests models.

### **3.2. Random Forest**

En aquest treball, el model que utilitzem per a variables de tipus estadístic és Random Forest, de la família dels mètodes "ensemble".

Random Forest [1-18] és un tipus de models de la família dels mètodes "ensemble" per a classificació (i regressió) basat en l'agregació de múltiples arbres de decisió, que es combinen per a fer una predicció, donant com a sortida global la classe de la moda (vot de la majoria) de les classes entregades pels arbres individuals.

### 3.3. Xarxes neuronals de convolució (CNN)

Les xarxes neuronals de convolució (CNN) [1-01, 1-09, 2-04, 2-05] són un tipus de xarxes neuronals profundes molt utilitzades en classificació de dades espacials (p. ex. visió artificial i sèries temporals), que representen una evolució de les xarxes multicapa (MLP).

Les CNN exploten especialment bé les característiques espacials de les dades mitjançant la connexió local entre neurones de capes adjacents, és a dir, les sortides d'un subconjunt de 'n' neurones en la capa oculta 'm' esdevenen les entrades de cada neurona específica en la capa oculta següent 'm+1'. Aquest mecanisme rep el nom de "sparse connectivity", i permet codificar comportaments no lineals fàcilment simplement apilant subconjunts de neurones d'aquesta manera.

En una CNN, les capes s'especialitzen en quatre tipus diferenciats d'accions [2-13]:

- 1) "Convolution"
- 2) "Pooling"
- 3) "Flattening"
- 4) "Full Connection"

I en funció de les accions anteriors i de la forma com s'executin en cada cas, podem construir una CNN mitjançant la seqüenciació dels següents operadors (o capes ocultes):

**Operador de convolució:** genera un "kernel" o màscara de convolució i l'aplica a les múltiples dimensions de les entrades rebudes amb l'objectiu d'obtenir una determinada sortida.

**Max Pooling:** aquest operador divideix la imatge d'entrada en un conjunt de finestres no solapades, i per a cada una d'elles, en calcula el valor màxim. En resum, el que fa és aplicar una operació de sub-mostreig no lineal, que permet reduir el volum de computació en capes posteriors mitjançant l'eliminació de valor no màxims, i aporta una forma d'invariància a la translació. Amb aquest operador cada finestra queda substituïda per un únic valor.

**Average Pooling:** aquest operador divideix la imatge d'entrada en un conjunt de finestres no solapades, i per a cada una d'elles, en calcula el valor mig. Amb aquest operador cada finestra queda substituïda per un únic valor [2-10].

**"Flatten":** és un operador que converteix totes les estructures bidimensionals resultants en un sol vector. Aquest pas és fonamental de cara a poder passar de les capes destinades a operacions de convolució/"pooling" a les capes amb neurones connectades [2-13].



**“Locally-Connected Layer”**: és similar a l’operador de convolució, però en aquest cas, a cada finestra o subregió de l’espai se li aplica un filtre (o conjunt de pesos) diferent [2-11].

**“Dense Layer” / “Fully-Connected Layer”**: és una capa neuronal tradicional, en la que cada neurona d’una capa oculta rep entrades de totes les neurones de la capa anterior, i per tant, ambdues capes es troben “densament” connectades [2-12].

**“Drop Out”**: és un operador que permet superar el problema tradicional de “over-fitting”. Ho fa descartant (assignant valor 0) de forma aleatòria un percentatge de les neurones durant el període d’entrenament. Se li assigna un valor entre 0 i 1, que correspon al percentatge de neurones a desconnectar [2-12].

Totes les arquitectures CNN que hem utilitzat en aquest treball, incloses totes les que hem anat descartant durant la fase d’optimització, han estat implementades utilitzant la llibreria KERAS de Python. Així mateix, en totes hem utilitzat l’optimitzador “Adam” [1-27] proporcionat per la mateixa llibreria, amb tots els seus valors per defecte:

```
Learning Rate = 0.001  
beta_1 = 0.9  
beta_2 = 0.999  
epsilon = None  
decay = 0.0  
AMSGrad = False
```

Donat que els nostres vectors de “targets” prenen la forma de valors enters (no llistat de vectors amb valors binaris), i que les nostres classes són mútuament excloents, com a funció objectiu hem utilitzat en totes les proves l’opció de “Sparse Categorical Crossentropy”, també proporcionada per la mateixa llibreria.

## 4. Mètodes

### 4.1. Introducció

Tal com hem comentat en apartats anteriors, ens proposem fer una comparativa entre 3 estratègies de classificació:

- 1) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals de primer i segon ordre com entrades al model
- 2) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir subimatges com entrades al model
- 3) Classificació dels píxels de la imatge entre múltiples regions (“fons d’imatge”, “interior globus ocular”, “fronteres entre capes”, “capa retinal 1”, “capa retinal 2”, etc), i fent servir subimatges com entrades al model

En aquest capítol expliquem els tres procediments amb detall, així com la seqüència de feines en que s’estructura cada estratègia. Algunes d’aquestes tasques són comunes als tres casos, i d’altres són específiques de només alguna d’elles.

El processat de les imatges i tots els passos que executem des del material original fins a disposar d’una base de dades derivada en format Python llesta per a alimentar els diferents models (mostreig no inclòs) es poden sintetitzar en el diagrama de la figura 05.

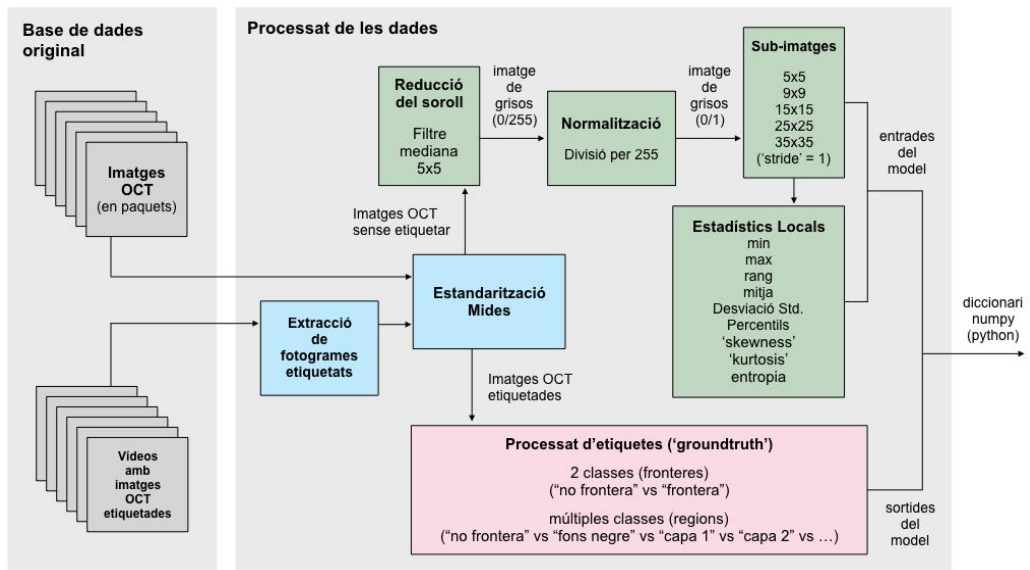


Figura 05. Fase de processat de les dades

Un cop disposem de les dades processades i organitzades en forma de llistats de diccionaris de Python (llibreria “numpy”), organitzem diferents subgrups de dades que gestionarem al llarg del procés en funció del que necessitem, i ja podem passar a l’entrenament de models com mostra la figura 06.

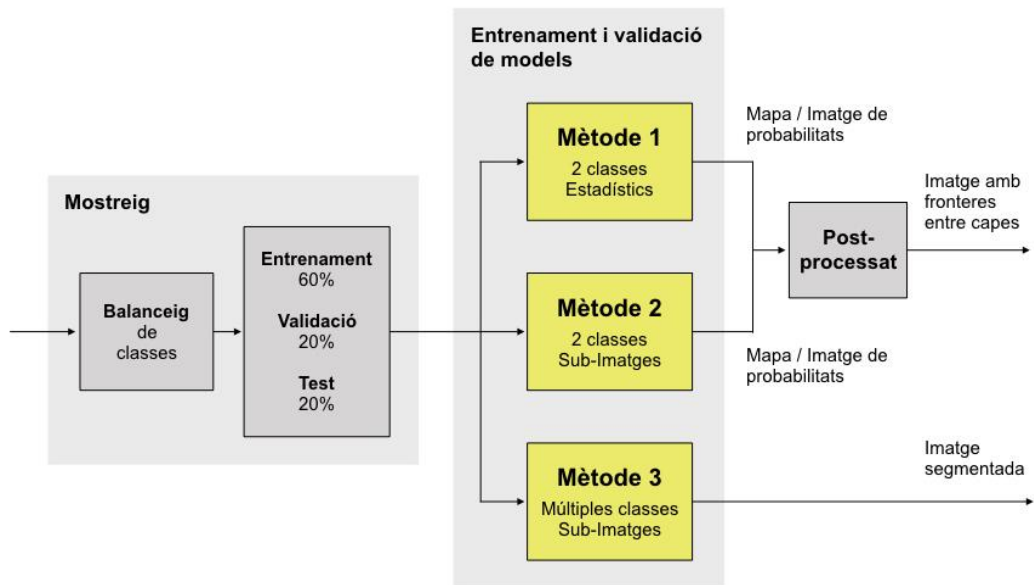


Figura 06. Fases de mostreig, definició de models i post-processat

Podem veure que en dos dels mètodes que volem treballar farà falta disposar d’un darrer bloc de post-processat de resultats.

La fase d'entrenament i validació de models pot sintetitzar-se amb una mica més de detall en el diagrama de la figura 07.

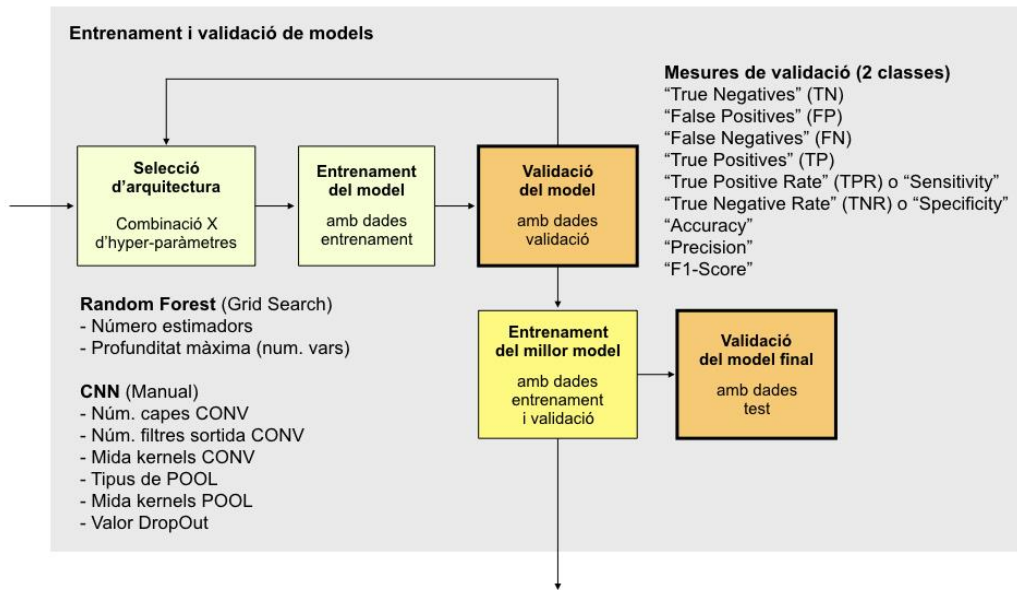


Figura 07. Fase d'entrenament i validació de models

A continuació aprofundim en cada una d'aquestes fases i les expliquem amb més detall.

## 4.2. Tractament i processat de les dades

Com ja hem comentat en el capítol 2, disposem de 12 vídeos, cada un d'ells conformat a partir de 128 fotogrames o imatges prèviament etiquetades per un expert.

Les fronteres entre capes retinals que han estat reconegudes i etiquetades corresponentment per l'expert són 6.

El processat dels vídeos consisteix en els següents passos:

- Importació dels vídeos i extracció de fotogrames, i importació de cada una de les corresponents imatges sense etiquetar, per al tractament per separat de cada parell d'imatges (procés iteratiu).
- Estandardització de les mides de la imatge original i la imatge d'etiquetes perquè totes tinguin la mateixa dimensió.
- Pas de les imatges a nivells de gris (les originals venen en color, per les etiquetes).
- Binarització de la imatge, de manera que s'assigna valor 1 als píxels de les etiquetes originals, i valor 0 a la resta.

- f) Generació d'una tercera imatge d'etiquetes multi-classe a partir de les etiquetes de dues classes.
- g) Guardat de tres fitxers: la imatge original en nivells de gris, i les imatges amb les etiquetes (per a dues i més classes).

Donat que aquestes feines són relativament senzilles i no requereixen de procediments gens sofisticats, per a dur-les a terme utilitzem MATLAB.

El principal motiu de l'elecció d'aquest programari és la senzillesa de les seves estructures de dades, que permet manipular imatges com a matrius (sense diferenciar entre files i columnes) i aplicar sobre elles operacions (transposició, etc.) de forma molt ràpida i sense necessitat d'importar o configurar cap tipus de llibreria.

Un cop desmuntats els vídeos i pre-processats els seus fotogrames, disposem de 1.536 conjunts de 3 imatges amb:

- a) La imatge original en nivells de gris (fig.04)
- b) La imatge d'etiquetes per a dues classes (fronteres vs capes retinals) (fig.11)
- c) La imatge d'etiquetes multi-classe (regions de la imatge) (fig.12)

## **Reducció del soroll**

En Visió per Computador, és comú començar amb una primera fase de processat de les imatges per a la reducció del soroll i l'ajustament del contrast.

La reducció del soroll és important perquè un sòl píxel blanc o negre enmig d'una subimatge actua com un valor extrem que segurament afectarà el procés d'extracció de característiques i alterarà els valors finals d'algunes variables (per exemple, la intensitat mitjana de la subimatge).

L'ajust de contrast és important per a ressaltar determinats aspectes de la imatge.

Les tècniques de processat més habituals per a la reducció del soroll i l'ajustament del contrast són:

- Filtrat amb mediana
- Equalització de l'histograma
- Descomposició Wavelet
- Normalització de contrast local
- Operadors locals i globals

En aquest treball, un cop importades les imatges OCT, apliquem a la imatge mitjançant una operació de convolució, un filtre de mediana amb una finestra de mida 5x5 píxels Això ens permet suavitzar la imatge i eliminar valors d'intensitat extrems.

### Normalització dels nivells de gris

Sabent que el valor màxim d'intensitat en una imatge de nivells de gris és 255, dividim per aquest els valors discrets d'intensitat de cada píxel, de manera que ens queden tots normalitzats amb valors continus entre 0 i 1.

### Descomposició de les imatges en subimatges més petites

Com que hem decidit transformar un problema de segmentació d'una imatge en un altre de classificació píxel a píxel, cal descompondre cada imatge OCT en subimatges més petites (fig.08).

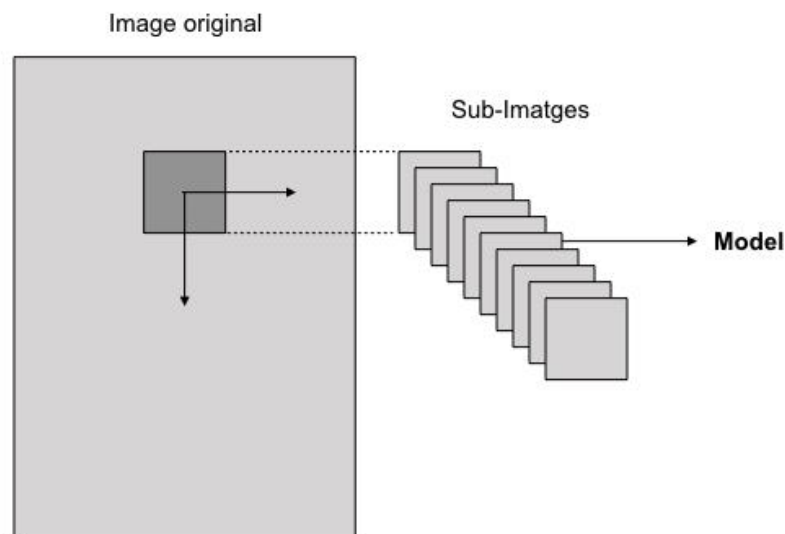


Figura 08. Descomposició de la imatge en subimatges més petites (procediment "sliding window")

Decidim que aquestes subimatges siguin quadrades (tot i que podrien no ser-ho), i per tal d'associar cada subimatge amb un píxel concret, les seves dimensions són senars, i així, les coordenades del seu píxel central esdevenen les coordenades de referència per a cada subimatge.

D'entrada no sabem quina és la mida de subimatge que millors resultats donarà, per tant, obtenim inicialment subimatges de diferents mides amb

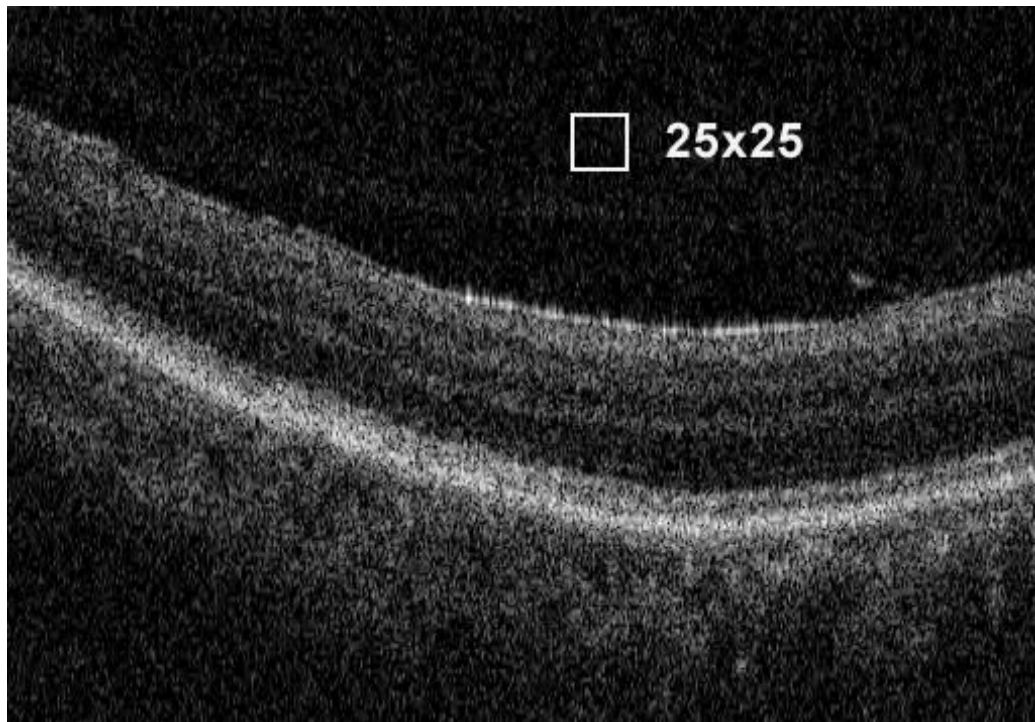
la intenció que aquest esdevingui també un paràmetre a discutir en els experiments.

Les mides de subimatge escollides són:

- 5 x 5 píxels
- 9 x 9 píxels
- 15 x 15 píxels
- 25 x 25 píxels
- 35 x 35 píxels

El motiu pel qual no es consideren finestres més grans és que, donat que algunes capes retinals són molt fines, suposem que amb subimatges de mida per sobre de 35x35 píxels poden coincidir més de dues capes retinals en una mateixa subimatge.

Per tal de fer-nos una idea de la superfície que representen les subimatges, en la figura 09 podem observar la relació de mida que existiria entre una imatge OCT sencera i una finestra de mida 25x25 píxels, i per tant, podem a simple vista comprendre de quina manera finestres grans poden generar-nos problemes en la classificació per coincidència entre capes.



*Figura 09. Imatge OCT amb regió de 25x25 píxels*

Quant al solapament entre subimatges, es decideix establir una distància ("stride") d'un píxel per al desplaçament entre subimatges consecutives,

cosa que implica que hi hagi molts píxels de solapament entre subimatges.

En definitiva, a partir de cada imatge OCT generem 5 conjunts de dades nous amb subimatges de diferents mides.

En la figura 10 podem observar 3 exemples de subimatges de mides 9x9, 15x15 i 25x25, i totes tres centrades en el mateix píxel.

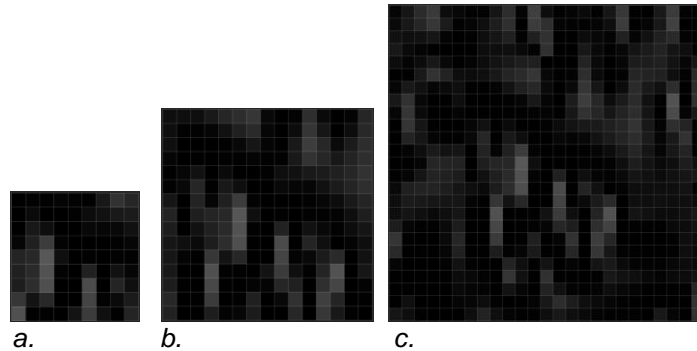


Figura 10. Sub-imatges de (a) 9x9, (b) 15x15 i (c) 25x25 píxels

### **Extracció de característiques locals**

En els models de xarxes de convolució (CNN), les entrades al model són les mateixes subimatges que acabem d'emmagatzemar, però en els models Random Forest no és així, i per tant, cal sotmetre les subimatges a un procés addicional d'extracció o d'enginyeria de característiques que ens permeti construir una nova base de dades derivada que és la que utilitzarem en aquest tipus de models.

En Visió per Computador, la fase d'extracció de característiques és el procés mitjançant el qual s'obtenen una sèrie de variables derivades de cada una de les subimatges, normalment estadístics de primer i segon grau, que caracteritzen la distribució de la funció d'intensitat en aquesta regió de la imatge. Un cop calculats els diferents valors, aquests queden associats a cada píxel concret sobre el que se centra cada finestra, de manera que el resultat d'aquesta operació és una base de dades amb variables estadístiques on cada fila es correspon amb unes coordenades específiques de l'espai de la imatge.

Els estadístics locals de primer i segon ordre que hem calculat en aquest treball per a cada subimatge són:

- Intensitat mínima
- Intensitat màxima
- Rang de valors d'intensitat (nivells de gris)



- Intensitat mitja
- Desviació estàndard de la intensitat
- Percentils d'intensitat 1, 5, 10, 25, 50, 75, 90, 95, 99
- "Skewness"
- "Kurtosis"
- Entropia

Per a cada una d'aquestes variables podem esperar un determinat comportament i una porció específica d'informació que podem utilitzar de forma eficient per a calcular millor els diferents píxels de la imatge.

### **"Targets"**

En sistemes d'aprenentatge automàtic supervisat, és necessari disposar d'un llistat d'etiquetes associades a cada una de les instàncies de la base de dades que permetin entrenar el model, de manera que aquest aprengui mitjançant l'algorisme corresponent quina relació matemàtica existeix entre les entrades del model i les etiquetes proporcionades ("targets"). Un cop entrenat el model, és aquesta lògica apresada la que li permetrà en el futur fer classificacions a partir de noves dades.

Per tant, un cop processades les entrades dels diferents models (les subimatges i els descriptors estadístics locals) cal decidir quina és la millor estratègia alhora de generar etiquetes.

En la base de dades processada, cada observació és una finestra o subimatge de píxels, per tant, el valor d'etiqueta que associem a cada subimatge és el mateix que correspongui al seu píxel central. Però l'etiqueta ("target") que assignem a cada píxel dependrà de l'estratègia global de classificació que haguem previst, per tant, és molt important sotmetre aquesta qüestió a debat abans de començar.

La base de dades original únicament proporciona etiquetes ("targets groundtruth") pels píxels de la imatge que corresponen a fronteres entre regions (fig.03).

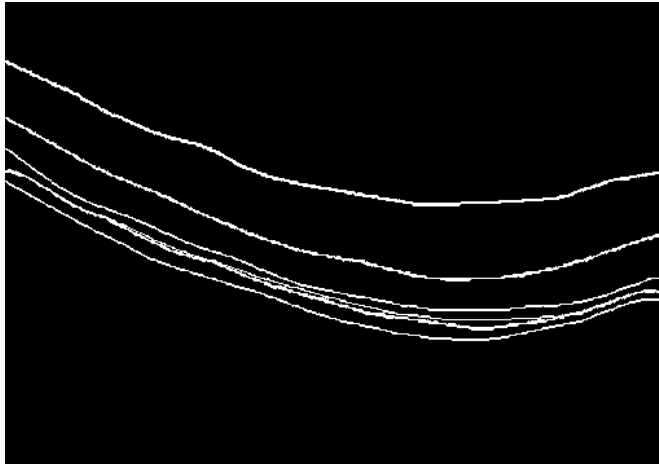
La resta de píxels pertanyen a regions no frontereres de la imatge (p. ex. fons negre o gruix de les capes retinals) i no tenen etiquetes.

Amb aquest punt de partida, tenim diferents opcions.

- Podem definir un problema en el que l'objectiu sigui detectar fronteres.

En aquest cas, processem les imatges d'etiquetes (fig.11) de manera que gairebé tots els píxels siguin negres (classe 0), i

únicament els píxels fronterers tinguin valor (classe 1). Això seria un plantejament de classificació amb només dues classes (“no frontera” vs “frontera”).



*Figura 11. Imatge d'etiquetes (“targets groundtruth”) processades amb dues classes (“píxels no frontera” vs “píxels frontera”)*

Amb dues classes el problema de classificació és més senzill perquè es redueix a calcular la probabilitat de cada píxel d'esdevenir una frontera entre capes retinals (siguin quines siguin aquestes), i per tant, es pot considerar un problema de detecció binària.

Si per contra, volem que el model sigui capaç de reconèixer fronteres entre capes retinals concretes, podem assignar un valor diferent d'etiqueta a cada límit entre parells de capes. D'aquesta manera, no tindríem només un problema de classificació de dues classes (“no frontera” vs “frontera”) sinó un problema amb múltiples classes (“no frontera” vs “frontera 1” vs “frontera 2” vs...).

- O podem definir un problema en el que l'objectiu sigui detectar regions.

En aquest cas, cal assignar una etiqueta diferent a cada regió de la imatge que vulguem identificar (fig.12):

- Píxels del fons de la imatge OCT, que bàsicament estarà format per píxels negres (amb valors d'intensitat propers a zero) i el trobarem en la part superior de la imatge.
- Píxels de la regió interior del globus ocular, caracteritzada per un degradat de nivells de gris en la part inferior de la imatge.
- Píxels de cada una de les capes retinals, conformant el seu gruix.

- Píxels fronterers. Decidim deixar-los sense assignar a cap de les regions, i per tant, cal considerar una classe addicional en la classificació.

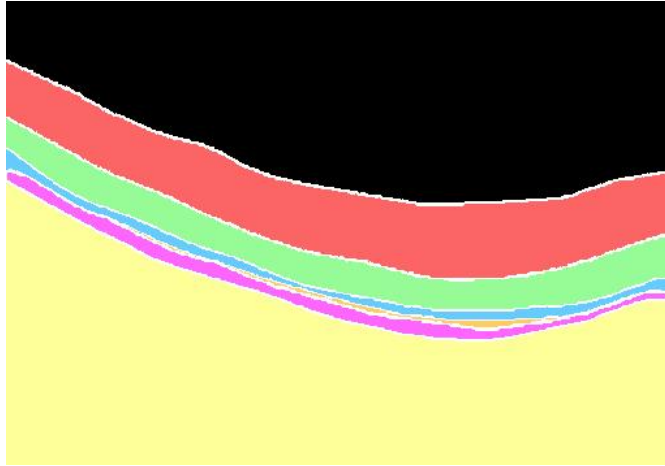


Figura 12. Etiquetes multi-classe originals ("groundtruth")

Per no limitar-nos, decidim processar les etiquetes generant diferents llistats de "targets": uns amb només dues classes de fronteres, altres amb múltiples classes de fronteres, i altres amb etiquetatge de regions, de manera que en funció de la metodologia utilitzada sigui possible agafar-ne unes o altres.

Com hem explicat anteriorment, en aquest treball, ens proposem comparar tres procediments diferents, però atenent únicament a la gestió de les etiquetes ("targets"), considerarem:

- Problema de detecció de fronteres de 2 tipus diferents: "no frontera" vs "frontera" (procediments 1 i 2)
- Problema d'identificació dels píxels propis de cada regió (procediment 3)

### **Estructura de dades**

Un cop desmuntades i processades totes les imatges i obtingudes les finestres, així com tots els estadístics i els diferents paquets d'etiquetes ("targets"), s'emmagatzema tot junt en una única base de dades (una per a cada mida de finestra), que pren la forma d'un llistat de diccionaris (numpy), i que és la que utilitzem en tot el que resta de treball.

Per tant, aquesta informació emmagatzemada en una mateixa estructura de dades, constitueix un llistat d'observacions o instàncies. Cada observació fa referència a un píxel concret de la imatge, i associat a aquest píxel tenim les següents variables:

- Coordenades del píxel (fila i columna)
- Valor original de nivells de gris
- 5 subimatges, o matrius (una per a cada mida) amb valors normalitzats de nivells de gris
- 5 llistats de mesures estadístiques, un per a cada mida de finestra
- Un valor binari (0 o 1) d'etiqueta ("target") per a dues classes
- Un valor enter d'etiqueta per a múltiples classes

A partir d'aquí, en funció de si estem utilitzant un model o un altre, o de quina estratègia estiguem considerant, seleccionarem unes entrades i unes etiquetes concretes en funció del que ens convingui.

#### **4.3. Mostreig**

Un cop disposem de les entrades dels diferents models (les subimatges i els descriptors estadístics locals) i les sortides (llistats de "targets"), organitzem diferents conjunts de dades en funció a diversos objectius.

#### **Balanceig de classes**

Si treballem amb dues classes ("no frontera" vs "frontera"), la classe majoritària és la de "negatiu" ("píxels no frontera"), és a dir, de píxels que no són una frontera entre capes, i aquesta excedeix molt en mida la classe de "positiu" ("píxels frontera"). Aquest fet pot dificultar més endavant l'aprenentatge del model, per tant, donat que tenim suficients subimatges, decidim descartar una part dels píxels "negatiu" per tal que el nombre d'observacions de cada classe estigui més igualat.

#### **Entrenament / Validació / Test**

Dividim aleatòriament la base de dades processada en diferents subconjunts que administrarem al llarg de la definició del model en funció del que necessitem:

- Entrenament:

Utilitzem el 60% de les observacions per a entrenar els diferents models, i en cap moment utilitzem aquestes instàncies per a validar o testejar, ni per a cap altra finalitat.

- Validació:

Reservem un 20% de les observacions per a validar els models en les fases d'entrenament i optimització d'hyper-paràmetres.

Un cop hem escollit una arquitectura concreta fruit de la fase d'optimització, tornem a entrenar el model final escollit, però ara utilitzant també en l'entrenament les dades de validació, és a dir, el model final s'entrena amb el 80% de les observacions.

- Test:

Reservem el 20% de les observacions per, un cop hem acabat el procés d'entrenament, tornar a validar el model final fent servir un conjunt de dades que no hagi intervingut durant l'entrenament.

Tot i la aleatorietat de la confecció de les mostres, ens assegurem de seguir tenint classes balancejades, tant en els paquets d'entrenament com de validació. En el conjunt de dades de test ja no és important tenir les classes balancejades.

A més, com que tenim intenció de validar visualment els resultats sobre una imatge concreta, deixem una imatge OCT de banda, i no la fem servir en cap moment més que per a l'examen final.

#### **4.4. Definició del model**

Dels 3 procediments que volem comparar en aquest treball, com ja hem dit, el primer utilitza entrades de tipus estadístic, mentre els altres dos utilitza com a entrades les mateixes subimatges. Per tant, en el primer cas utilitzem un model Random Forest, i en els altres dos implementem una xarxa neuronal de convolució (CNN).

Un cop hem implementat tot el "pipeline" d'entrenament per a cada tipus de model, amb la finalitat d'identificar l'arquitectura idònia per a cada cas, tan per als models Random Forest com per a les xarxes neuronals de convolució (CNN) es proven diferents valors d'hyper-paràmetres (diferents en cada cas), i se seleccionen les configuracions amb millors resultats.

En el cas de models Random Forest, s'estudien diferents configuracions en funció dels següents hyper-paràmetres:

- Número d'estimadors
- Profunditat màxima dels arbres (número màxim de variables)

Aquest anàlisi s'executa d'acord amb un procediment de "Grid Search", amb el que s'entrena el model diversos cops per a cada combinació d'hyper-paràmetres i s'emmagatzemen els resultats. Finalment, es calcula la mitja dels resultats corresponents a cada configuració i es comparen. El model seleccionat com a final és el que obté valors més alts en les diferents mesures de validació.

Amb les xarxes de convolució, es proven també múltiples valors de paràmetres, tots ells per a arquitectures amb un o dos blocs multicapa de "convolució+pooling", 10 epochs i una mida de "batch" de 128.

Donades les característiques de les arquitectures CNN, no és possible aplicar aquí un procediment "Grid Search" (al menys de la manera com l'apliquem amb altres models) perquè determinades combinacions de valors no són compatibles. Per exemple, si en un bloc "conv-pool" hem usat una determinada mida de filtre o "kernel", en el següent bloc no totes les mides de filtre o "kernel" seran compatibles amb el primer, i això fa que no es pugui resoldre la optimització amb un procés simplement iteratiu.

La conseqüència del que acabem d'exposar és que la optimització de xarxes CNN esdevé un procés iteratiu manual, treballant l'arquitectura CNN amb l'únic recurs d'anar fent proves i aprenent dels errors, i per tant, convertint-se en un procediment molt artesanal.

De tota manera, els valors que hem provat per a generar configuracions diferents de CNN són els següents:

- Número de capes de convolució = [1,2]
- Número de filtres en la sortida de la convolució = [32,64,128,256]
- Mides de kernels per a la convolució = [1,3,5,7]
- Tipus de "pooling" = ['max','average']
- Mides de kernels per al "pooling" = [1,3,5,7]
- Valor de "DropOut" = [ 0.15 , 0.2 , 0.25 , 0.3 , 0.4 , 0.5 ]

El criteri que utilitzem per a finalitzar l'entrenament de xarxes CNN és el de fixar el nombre d'epochs. Arribat a aquest valor, l'entrenament s'atura. Com que en funció de l'arquitectura CNN el temps d'entrenament pot oscil·lar entre 30 minuts i 10 hores, hem gestionat el nombre d'epochs en funció del temps requerit d'entrenament, no comparant en cap cas resultats d'entrenament amb diferents valors d'epochs.

Aquestes arquitectures es construeixen i s'entrenen amb l'ajut de les llibreries de Python: Tensorflow, Keras i Sklearn.

## 4.5. Validació del model

Per a poder mesurar la correctesa de la solució adoptada, cal definir una sèrie de mesures capaces de quantificar de forma objectiva l'èxit del procediment.

Aquestes mesures es calculen després d'haver executat el model sobre una base de dades que no hagin intervingut en l'entrenament del mateix.

De tota manera, durant l'entrenament del model, disposem de mesures de validació després de cada iteració. Aquestes ens serveixen com a mesures orientatives sobre com està evolucionant el procediment, però en cas que les prenem com a mesures de validació final.

Per tant, quant a mesures de validació del model, parlarem de dues fases de validació:

- 1) Mesures de validació durant l'entrenament del model
- 2) Mesures de validació final o de test, sobre un conjunt de dades independent

### Validació en fase d'entrenament

Per tal de poder avaluar l'evolució de l'entrenament, l'algorisme que utilitzem ens permet obtenir per a cada iteració les següents mesures de validació durant la fase d'entrenament:

**“Loss”**: mesura de la no correctesa del model. Ens dona una percepció sobre quant properes són les prediccions del model respecte els valors "groundtruth". Valors petits corresponen a millors models, per tant, ens interessa minimitzar-la.

**“Accuracy”**: proporció de subimatges correctament classificades en la classe que els toca.

Per a nosaltres es tracta de mesures estrictament orientatives donat que no controlem la partició que realitza l'algorisme d'observacions per a entrenament, validació i test durant l'entrenament.

### Validació en fase de test

Per a mesurar la correctesa de la solució adoptada es prenen les següents mesures de validació:

**“True Negatives” (TN):** Nombre de subimatges de la classe 0 (no corresponents a una frontera entre capes retinals) correctament classificades com a classe 0.

**“False Positives” (FP):** Nombre de subimatges de la classe 0 (no corresponents a una frontera entre capes retinals) incorrectament classificades com a classe 1.

**“False Negatives” (FN):** Nombre de subimatges de la classe 1 (corresponents a una frontera entre capes retinals) incorrectament classificades com a classe 0.

**“True Positives” (TP):** Nombre de subimatges de la classe 1 (corresponents a una frontera entre capes retinals) correctament classificades com a classe 1.

**“True Positive Rate” (TPR) o “Sensitivity”:** proporció de subimatges de la classe 1 (corresponents a una frontera entre capes retinals) correctament classificades com a tals.

**“True Negative Rate” (TNR) o “Specificity”:** proporció de subimatges de la classe 0 (no corresponents a una frontera entre capes retinals) correctament classificades com a tals.

**“Accuracy”:** proporció de subimatges correctament classificades en la classe que els toca.

$$ACC = ( TP + TN ) / ( TP + TN + FP + FN )$$

El problema d'aquesta mesura és que, en cas de classes no balancejades (és a dir, conjunts de dades en que es disposa de moltes més mostres d'una classe que de l'altra) por portar a engany, perquè pot mostrar valors alts i només haver classificat totes les observacions en la classe majoritària.

**“Precision”:** De les subimatges de la classe 1 (corresponents a una frontera entre capes retinals) (correctament o no) classificades com a classe 1, proporció d'aquestes correctament classificades.

$$Pr = TP / ( TP + FP )$$

**“F1-Score”:** Proporció de la correctesa de la classificació en ambdues classes.

$$F1 = 2TP / ( 2TP + FP + FN )$$

Aquesta mesura és la que utilitzem com a referència donat que és la més restrictiva i que valida la classificació de les dues classes.



Aquestes mesures de validació en fase de test es calculen sobre un conjunt de subimatges no utilitzades durant l'entrenament:

## **Corbes ROC**

La corba ROC ("Receiver Operating Characteristic Curve") és un recurs visual que il·lustra l'habilitat discriminadora d'un determinat classificador per a dues classes en funció de les probabilitats obtingudes per a cada classe i la seva binarització mitjançant múltiples llindars de decisió.

Amb altres paraules, per a unes determinades entrades del model, aquest entrega com a sortida les probabilitats de que cada observació pertanyi a una classe o a l'altra. Per tal d'obtenir la classe a la que finalment assignem cada instància, s'aplica un determinat llindar de binarització. Si en comptes d'aplicar un sol llindar, re-calem l'assignació de classe de forma iterativa per a un conjunt de llindars diferents distribuïts homogèniament, podem generar un grafisme en forma de corba (ROC) que relaciona el ratio de positius ben classificats ("True Positive Rate" o TPR) amb el ratio de classificació de negatius ("False Positive Rate" o FPR), i per tant, ens dona una molt bona mesura del bon comportament del model. El TPR també es coneix com a "sensitivity", "recall" o simplement probabilitat de detecció (de fronteres entre capes retinals en aquest cas), mentre que el FPR també es coneix com a probabilitat de falsa alarma i pot calcular-se com a "1 - specificity".

A partir de la corba ROC, també s'acostuma a calcular l'àrea sota la corba ("Area under the curve" o AUC), donat que permet quantificar numèricament la correctesa del model i sintetitzar-la en un sol valor entre 0 i 1, sent 0 el pitjor comportament, i 1 el millor.

### **4.6. Post-processat del resultat del model**

En els dos primers mètodes que volem comparar, la sortida del model és la probabilitat associada a un determinat píxel d'esdevenir frontera entre regions de la imatge, per tant, per a cada imatge OCT sotmesa al model, obtenim un mapa de probabilitats.

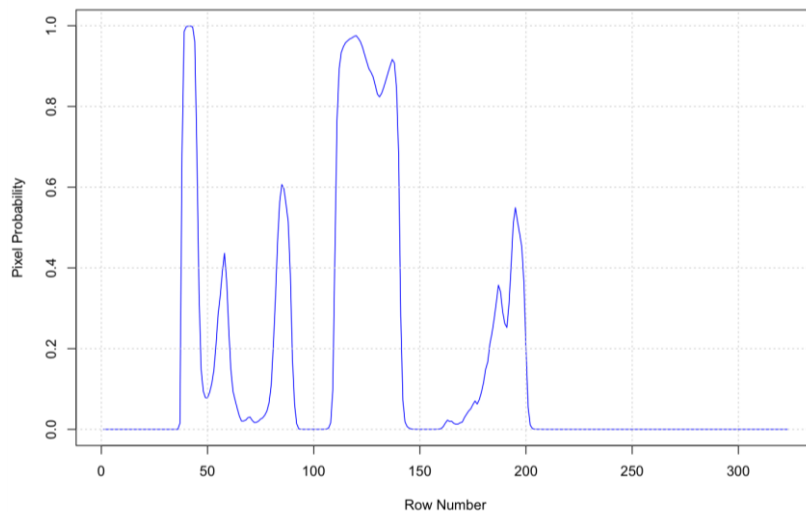
A partir d'aquesta imatge de probabilitats, hem de trobar un mecanisme de post-processat que ens permeti obtenir unes fronteres ben definides (amb valor unívoc o binari, sense massa gruix, sense discontinuïtats, etc.).

Provem diferents opcions:

Opció 1: Es proposa establir un llindar per a les probabilitats amb l'objectiu de binaritzar la imatge. D'aquesta manera obtenim

una imatge amb píxels de valor 0 o 1. Els píxels amb valor 1 corresponen a fronteres entre capes, mentre que els de valor 0 constitueixen segments de la imatge, o amb altres paraules, a capes de la retina.

Opció 2: A partir de la imatge de probabilitats, la descomponem en cada una de les seves seccions verticals, cada una de les columnes de píxels, i la analitzem. En la figura 13 ensenyem una d'aquestes seccions, amb la probabilitat d'esdevenir frontera en l'eix vertical, i el número de fila (de píxels de la imatge) en l'eix horitzontal.



*Figura 13. Secció vertical (columna de píxels) d'una imatge de probabilitats*

Per a cada una de les corbes de probabilitats, obtenim els seus màxims locals, i en seleccionem els més rellevants, esperant que coincideixin amb els punts de separació entre capes retinals.

Una manera de seleccionar un subconjunt de màxims locals és filtrar-los mitjançant un llindar.

Una altra manera d'obtenir el subconjunt de màxims locals és seleccionar, per a cada secció vertical, els 6 màxims que tinguin valor de probabilitat més alt.

Amb el tercer mètode estudiat no apliquem cap operació de post-processat. En aquest cas, el resultat del model és directament la classe (o regió de la imatge) a la que s'assigna cada píxel.

#### 4.7. Particularitats de cada mètode

Un cops explicades totes les fases del “pipeline” de treball, des del processat inicial de les dades fins la obtenció de tres models (un per a cada estratègia comparada), analitzem breument, a mode de resum, quines són les particularitats que aproximen o diferencien aquestes estratègies.

Similituds:

- Els mecanismes de processat (suavitzat de nivells de gris, etc.) de les imatges senceres abans de fragmentar-les són els mateixos en els tres mètodes.
- En els tres casos utilitzem procediments basats en “sliding window”, per tant, hem de processar les subfinestres en tots. En la primera estratègia, a més, a partir de les subimatges hem de calcular-ne els descriptors estadístics.

Diferències:

- En la primera estratègia fem servir estadístics locals com entrades del model, mentre que en les altres utilitzem directament les subimatges. Aquesta decisió condiona clarament el tipus de model a utilitzar. Per als estadístics fem servir models Random Forest mentre que en els altres dos casos usem xarxes de convolució.
- Tot i que en les estratègies 2 i 3 fem servir el mateix tipus de models CNN, el fet que en una treballem amb dues classes i en l’altre en tinguem més pot provocar que les arquitectures òptimes resultants siguin diferents. És de preveure que la tercera solució requereixi una arquitectura CNN una mica més complexa.
- En els mètodes 1 i 2 necessitem operacions de post-processat perquè, com que el resultat del model són mapes de probabilitats, serà necessari refinar-los fins obtenir unes fronteres ben definides.

## 5. Experiments i resultats

### 5.1. Introducció

Tal com hem anat recordant en cada capítol, ens proposem fer una comparativa entre 3 estratègies diferents:

- 1) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals de primer i segon ordre com entrades al model
- 2) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir subimatges com entrades al model
- 3) Classificació dels píxels de la imatge entre múltiples regions (“fons d’imatge”, “interior globus ocular”, “fronteres entre capes”, “capa retinal 1”, “capa retinal 2”, etc), i fent servir subimatges com entrades al model

A continuació mostrem els resultats obtinguts per a cada una d’aquestes estratègies.

### 5.2. Mètode 1:

**Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals com entrada**

A partir de la base de dades d’estadístics locals obtinguts tal com hem explicat anteriorment, entrenem i validem diferents configuracions d’un model Random Forest.

La taula 14 mostra els 5 millors resultats obtinguts amb diferents mides de finestra i diferents arquitectures per a models Random Forest:

Taula 14. Optimització d'hyper-paràmetres amb models Random Forest

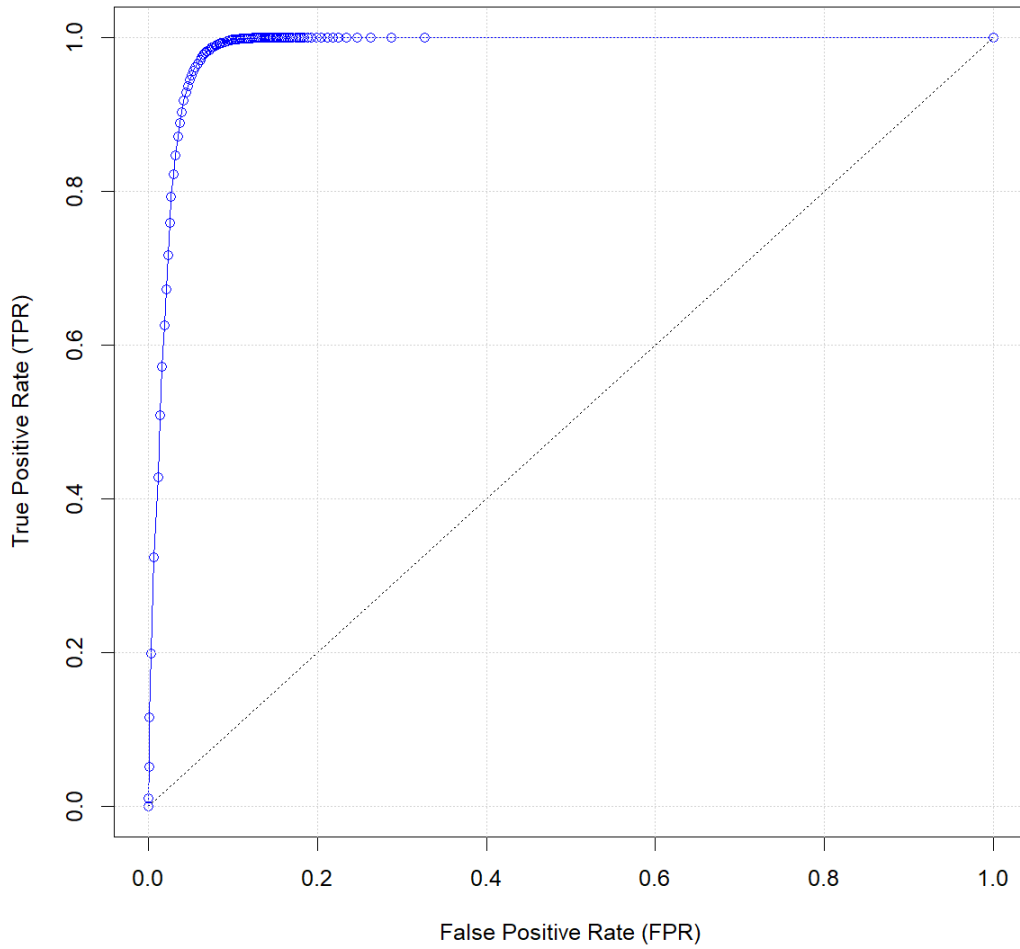
Mides subimatge	Hyper-Paràmetres		Validació				
	Número d'estimadors	Profunditat màxima	TPR	TNR	Accuracy	Precision	F-Score
9x9	<b>500</b>	<b>19</b>	<b>0,9847</b>	<b>0,8712</b>	<b>0,9278</b>	<b>0,8841</b>	<b>0,9316</b>
	200	19	0,9847	0,8702	0,9274	0,8833	0,9312
	100	19	0,9833	0,8694	0,9263	0,8825	0,9302
	50	19	0,9829	0,8685	0,9256	0,8818	0,9296
	25	19	0,9800	0,8700	0,9249	0,8826	0,9288
15x15	<b>200</b>	<b>19</b>	<b>0,9903</b>	<b>0,8868</b>	<b>0,9386</b>	<b>0,8976</b>	<b>0,9417</b>
	500	19	0,9904	0,8862	0,9384	0,8971	0,9414
	100	19	0,9905	0,8853	0,9380	0,8964	0,9411
	50	19	0,9888	0,8852	0,9371	0,8962	0,9402
	25	19	0,9883	0,8855	0,9369	0,8963	0,9401
25x25	<b>200</b>	<b>19</b>	<b>0,9946</b>	<b>0,8876</b>	<b>0,9413</b>	<b>0,8992</b>	<b>0,9445</b>
	100	19	0,9944	0,8877	0,9413	0,8992	0,9444
	500	19	0,9947	0,8873	0,9412	0,8989	0,9444
	50	19	0,9937	0,8876	0,9408	0,8990	0,9440
	25	19	0,9917	0,8873	0,9397	0,8987	0,9429
35x35	<b>100</b>	<b>19</b>	<b>0,9957</b>	<b>0,8905</b>	<b>0,9427</b>	<b>0,8996</b>	<b>0,9452</b>
	500	19	0,9958	0,8901	0,9426	0,8993	0,9451
	50	19	0,9951	0,8909	0,9426	0,8999	0,9451
	200	19	0,9956	0,8901	0,9425	0,8993	0,9450
	25	19	0,9945	0,8879	0,9408	0,8974	0,9434

Analitzant els resultats obtinguts, el millor F-Score (mesura de validació més restrictiva) s'aconsegueix amb un model Random Forest amb 100 estimadors i profunditat màxima 19, i una mida de subimatge de 35x35.

Entrenem aquest model i el validem amb les dades de test, no utilitzades fins ara.

La figura 15 mostra la corba ROC per als resultats obtinguts pel model en fase de test.

### Classification Performance / ROC Curve



*Figura 15. Corba ROC per a subimatges de mida 35x35*

En aquest cas, obtenim un valor  $AUC = 0,9823$ .

A mode de test final, executem el model sobre tots els píxels d'una imatge que havíem deixat sense utilitzar (fig.16), i n'obtenim el mapa de probabilitats (fig.17).

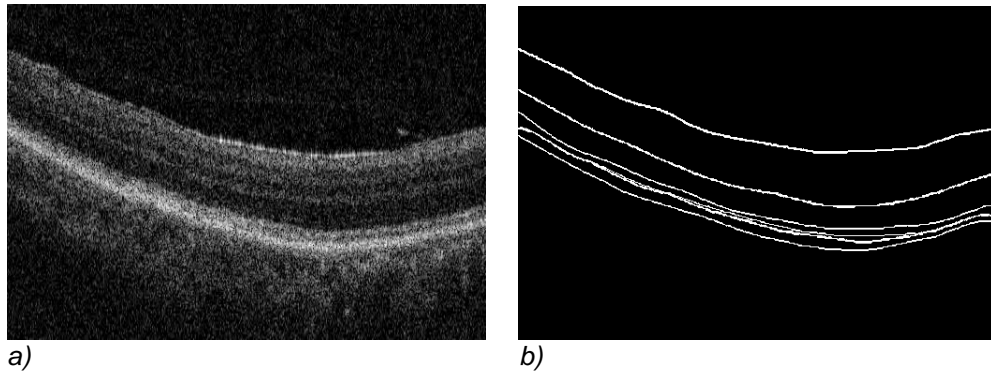


Figura 16. (a) Imatge OCT original i  
(b) imatge d'etiquetes amb les fronteres originals ("groundtruth") entre regions

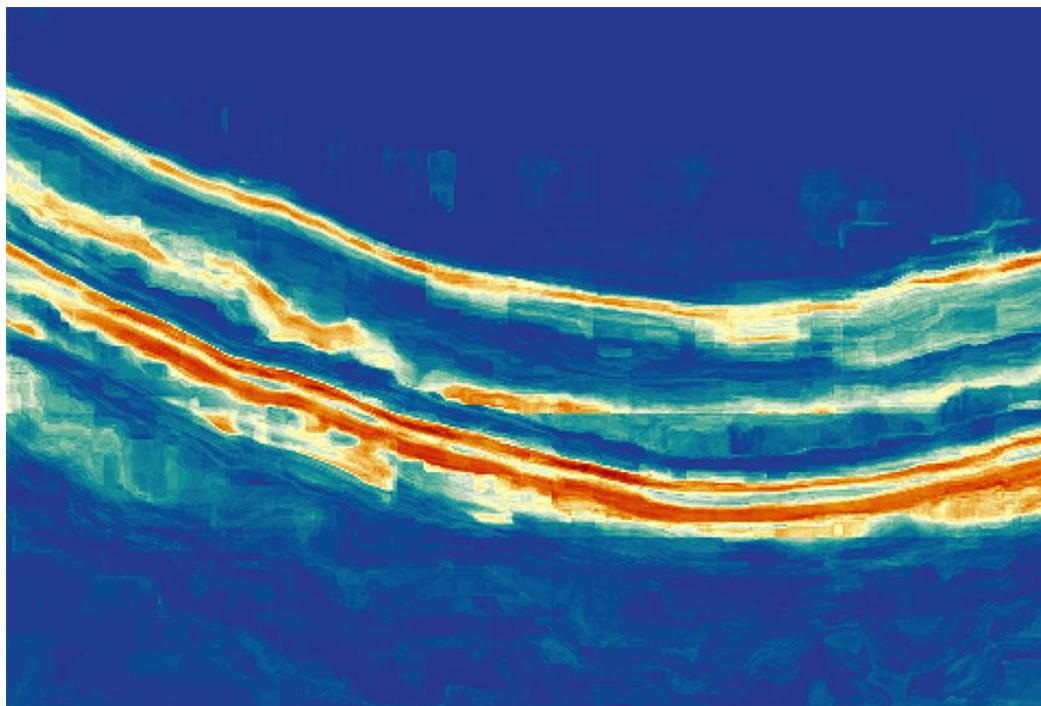
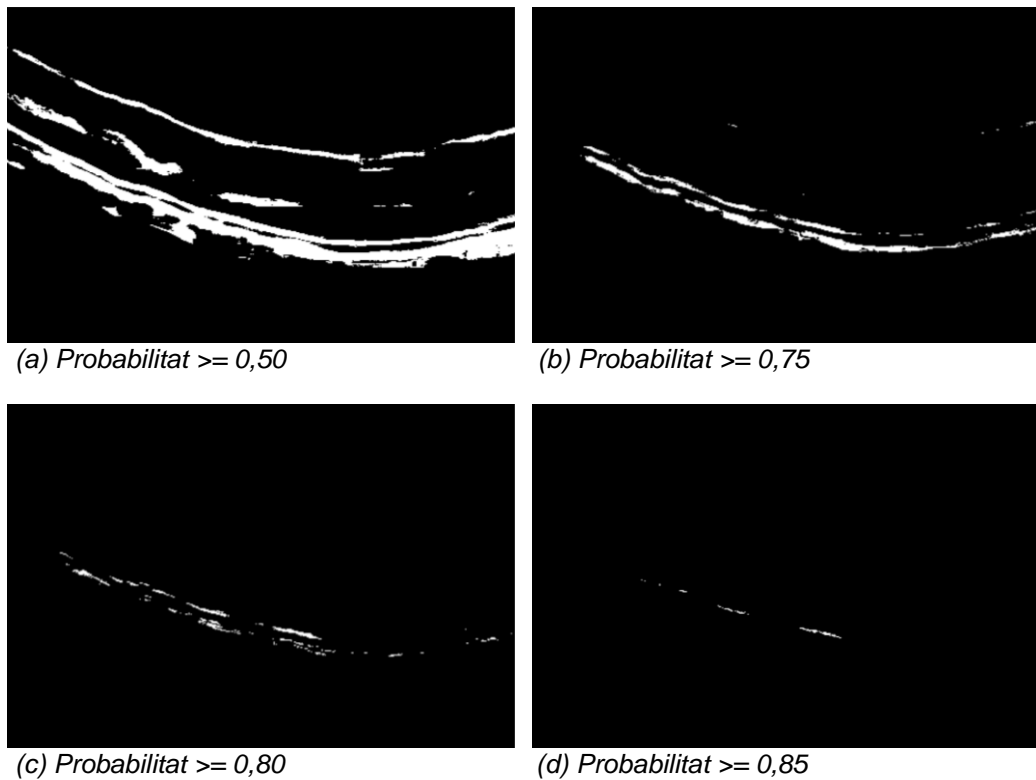


Figura 17. Imatge amb les probabilitats de cada píxel de ser una frontera entre capes

En aquest mapa de probabilitats (fig.17), la probabilitat de ser frontera amb valor 0 correspon al color blau, mentre que el valor 1 correspon al color vermell. Els píxels de color groc o taronja representarien probabilitats amb valors continus entre 0 i 1.

Per a obtenir unes fronteres ben definides (sense massa gruix, sense discontinuïtats, etc.), provem les diferents opcions que hem explicat:

Opció 1: Estudiem diferents llindars per a les probabilitats amb l'objectiu de binaritzar la imatge. Les imatges de la figura 18 mostren un mateix mapa de probabilitats binaritzat amb diferents valors de llindar.



*Figura 18. Imatges de probabilitats binaritzades en funció de diferents llindars*

Veiem que amb llindars relativament baixos (0,50-0,85) gairebé perdem la detecció, cosa que indica que els valors de probabilitat obtinguts per a les fronteres entre capes no són gaire propers a 1.

Opció 2: Obtenim ara els màxims locals de cada una de les seccions verticals del mapa de probabilitats i els analitzem (fig.19).



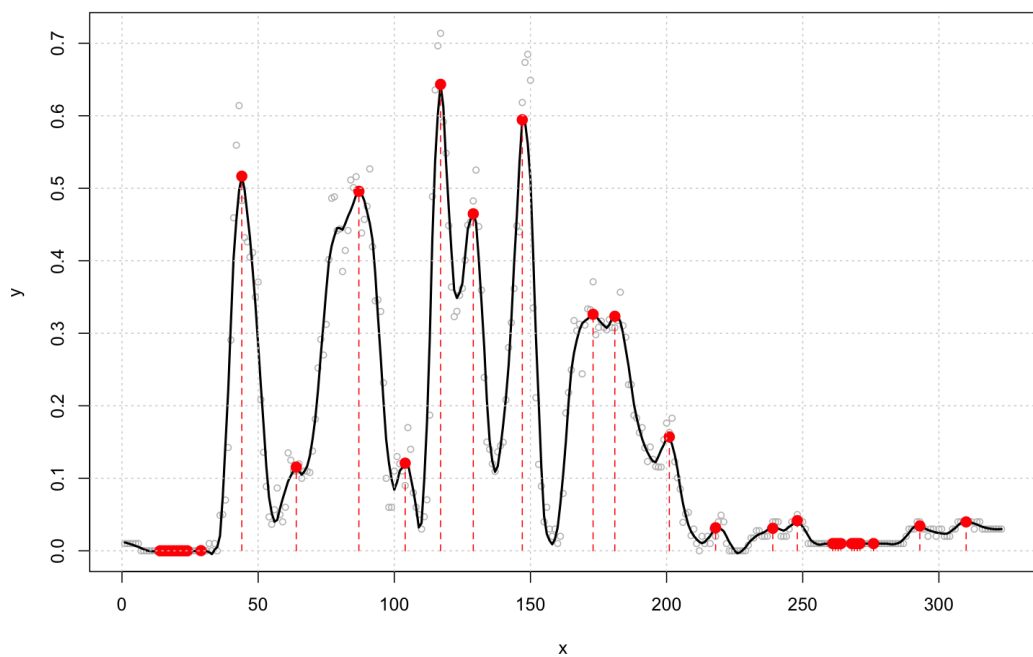
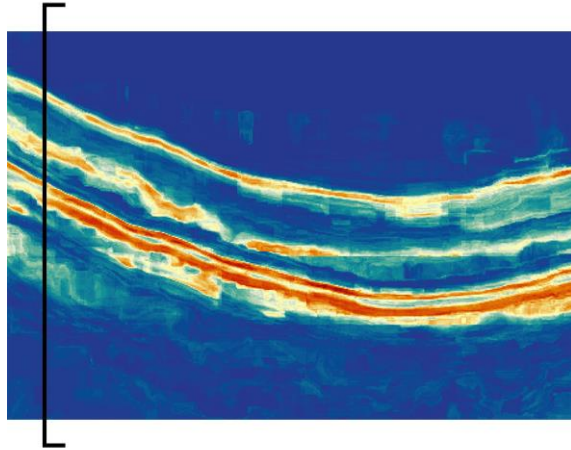
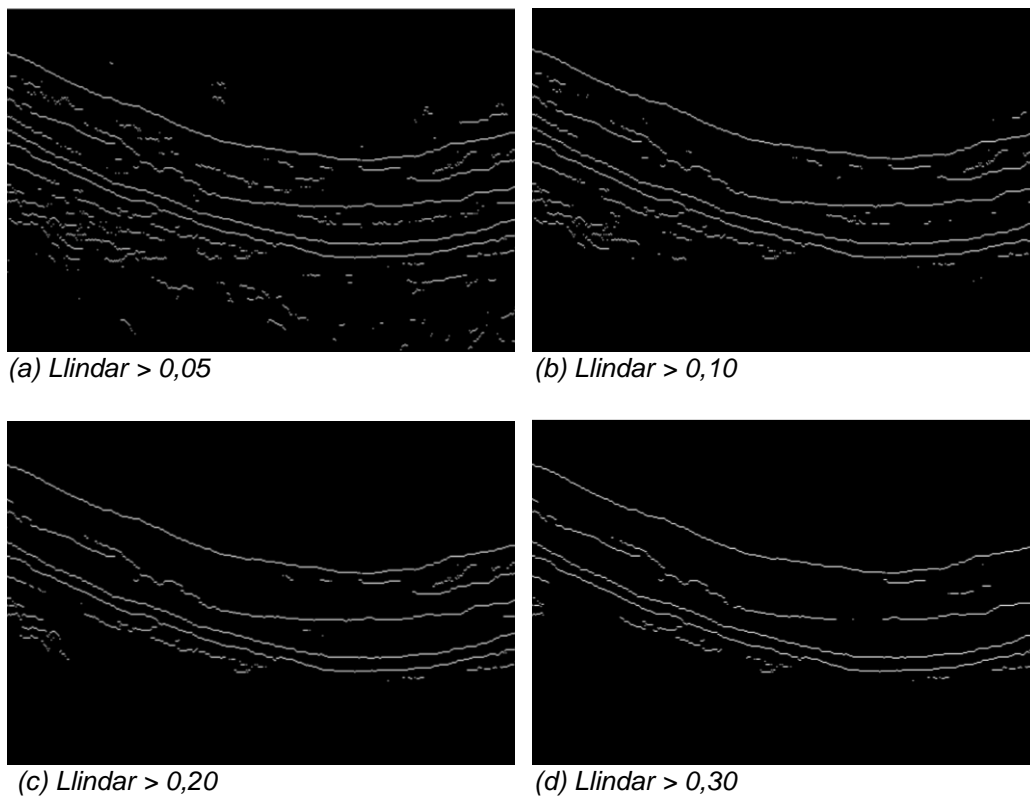


Figura 19. Secció vertical (columna de píxels) de la imatge de probabilitats amb els màxims locals en vermell (gràfic inferior)

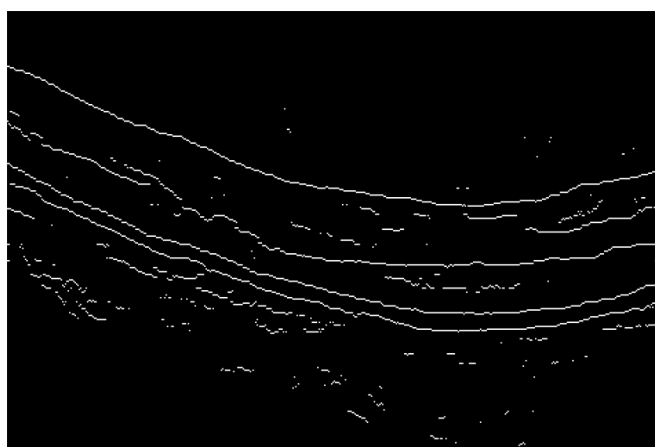
A continuació seleccionem els màxims locals més rellevants, i reconstruïm la imatge únicament amb les corbes conformades pels punts seleccionats.

Les figura 20 mostra les imatges reconstruïdes per a diferents valors de llindar:



*Figura 20. Fronteres entre capes retinals obtingudes per l'addició dels màxims locals de cada secció vertical*

Una altra manera d'obtenir el subconjunt de màxims locals és seleccionar, per a cada secció vertical, els 6 màxims que tinguin valor de probabilitat més alt, i d'aquesta manera obtenim la imatge de la figura 21.



*Figura 21. Fronteres entre capes retinals obtingudes per l'addició d'únicament 6 dels màxims locals de cada secció vertical*

### 5.3. Mètode 2: Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir subimatges com entrada

Entrenem múltiples combinacions d’hyper-paràmetres per a una arquitectura de xarxa neuronal de convolució (CNN), i en la taula 22 mostrem els millors resultats obtinguts amb diferents paràmetres:

*Taula 22. Valors de validació per a xarxa CNN en fase d’optimització de paràmetres*

Mida de subimatge	Mida del kernel	Número de filtres	Mida pooling	Tipus de pooling	Drop Out	Precisió (entrenament)
<b>35</b>	<b>1</b>	<b>28</b>	<b>1</b>	<b>max</b>	<b>0,15</b>	<b>0,93216</b>
35	1	28	1	max	0,3	0,92979
35	1	28	1	max	0,25	0,9296
35	1	28	1	max	0,2	0,92649
35	1	28	3	max	0,15	0,92558
35	1	28	1	max	0,4	0,92339
35	1	28	3	max	0,2	0,92339

Atenent a l’estructura de capes i funcionalitats exposades en l’apartat anterior, l’arquitectura CNN escollida es construeix mitjançant la seqüenciació de les següents capes (fig.23):

- a) Capa de convolució 2D (“Conv2D”)
  - Número de filtres de sortida = 28
  - Mides de la finestra de convolució (o “kernel”) = (1,1)
- b) Capa “MaxPooling2D”
  - Mides de la finestra de “pooling” = (1,1)
- c) Flatten (flattening the 2D arrays for fully Connected layers)
- d) Dense (128, amb funció d’activació “Relu”)
- e) Dropout (0,15)
- f) Dense ( num de classes=2, amb funció d’activació “softmax”)
- g) Compile (amb optimitzador “adam”, loss=sparse categorical crossentropy, metrics=”accuracy”)

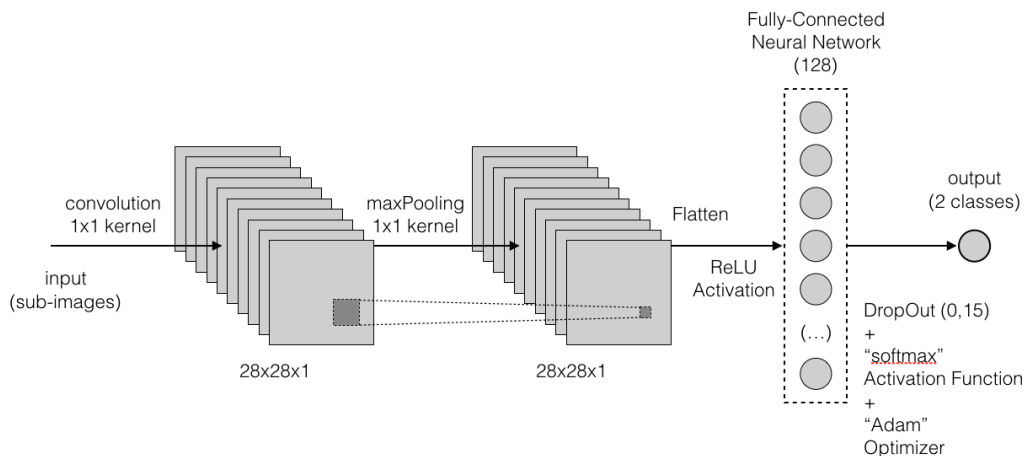


Figura 23. Arquitectura de xarxa CNN

S'executa el codi de Python que entrena la xarxa amb “epochs” = 10 i “batch size” = 128.

En validació en fase d'entrenament (mostra per a validació) obtenim els resultats de la taula 24:

Taula 24. Valors de validació per a xarxa CNN en fase d'entrenament

	Entrenament		Validació		Test
Mida Subimatge	Loss	Accuracy	Loss	Accuracy	Accuracy
5x5	0,4331	0,7895	0,4292	0,7906	0,7055
9x9	0,3668	0,8379	0,3620	0,8407	0,7369
15x15	0,3152	0,8714	0,3201	0,8663	0,7492
25x25	0,2401	0,9116	0,2373	0,9124	0,8540
35x35	0,1093	0,9439	0,1954	0,9337	<b>0,8648</b>

Observem que a mida que la subimatge creix els resultats obtinguts en fase d'entrenament són millors.

De tota manera, aquestes mesures no contemplen la correcció de la classificació en relació a cada una de les classes, sinó només la classificació global d'observacions, independentment de la classe a la que pertanyin.

En fase de test, per a cada una de les mides de subimatge utilitzades obtenim les mesures finals de validació de la taula 25.

Taula 25. Valors de validació per a xarxa CNN en fase de test

Mida Subimatge	TN	FP	FN	TP
5x5	96.705	22.798	49.001	75.321
9x9	93.790	24.863	38.793	84.574
15x15	103.361	13.994	45.989	75.916
25x25	50.781	7.210	9.839	48.962
35x35	30.015	4.241	5.036	29.347

Mida Subimatge	TPR	TNR	Precision	FScore
5x5	0,605	0,809	0,767	0,677
9x9	0,685	0,790	0,772	0,726
15x15	0,622	0,880	0,844	0,716
25x25	0,832	0,875	0,871	0,851
35x35	0,853	0,876	0,873	<b>0,863</b>

Igual com passava en la validació en fase d'entrenament, observem a la figura 26 que obtenim millors resultats amb finestres més grans de subimatge.

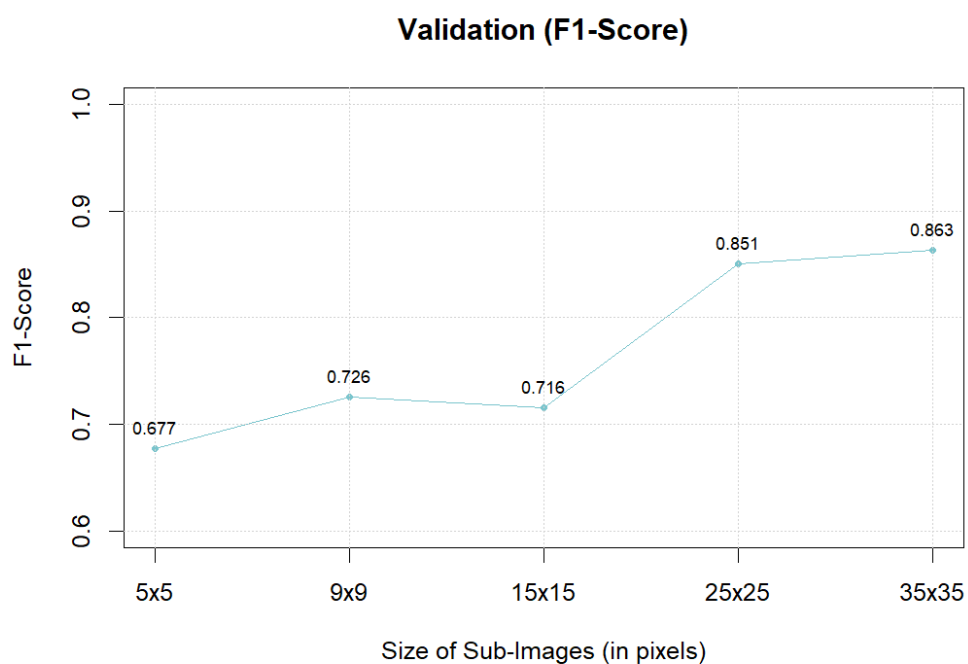


Figura 26. Evolució del F1-Score en funció de la mida de la subimatge

En la figura 27 mostrem la corba ROC per als resultats obtinguts pel model en fase de test. En aquest cas, obtenim un valor AUC = 0,9587.

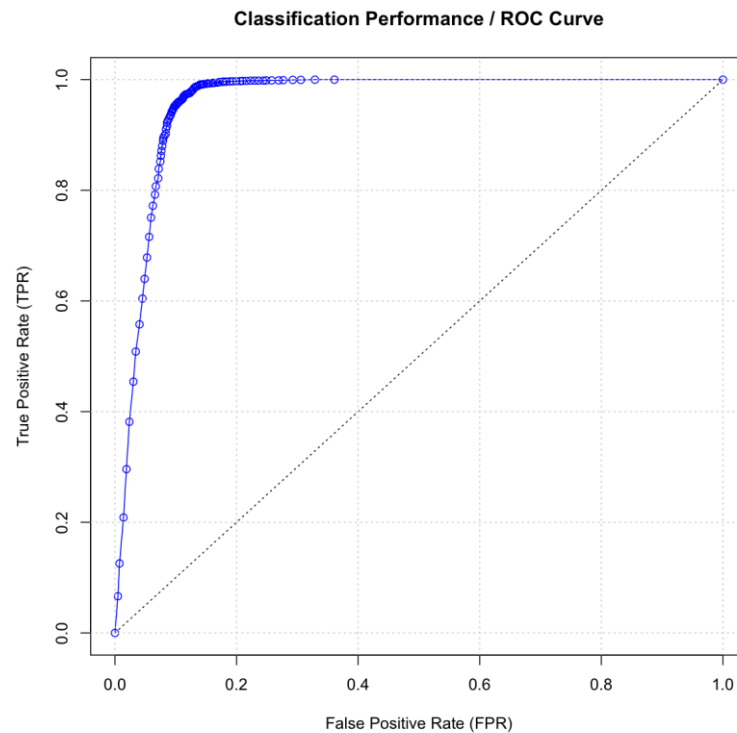


Figura 27. Corba ROC per a subimatges de mida 35x35

A continuació fem un mapeig de les probabilitats obtingudes per a una imatge concreta (fig. 28) sobre els seus píxels i n'obtenim el mapa de probabilitats de la figura 29.

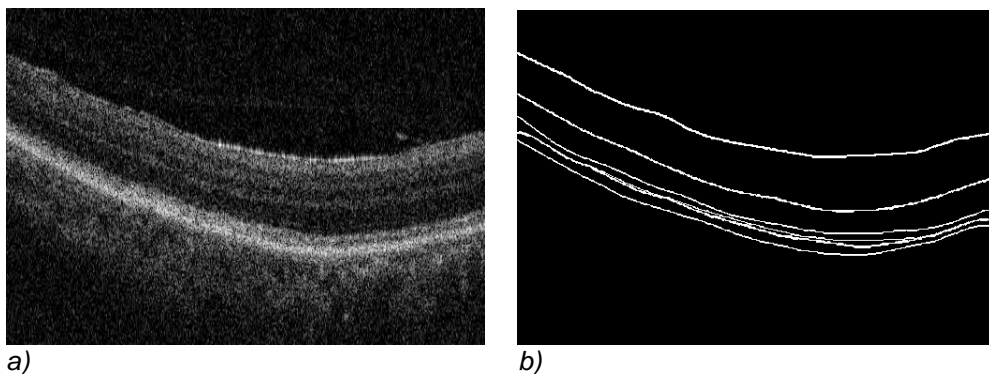
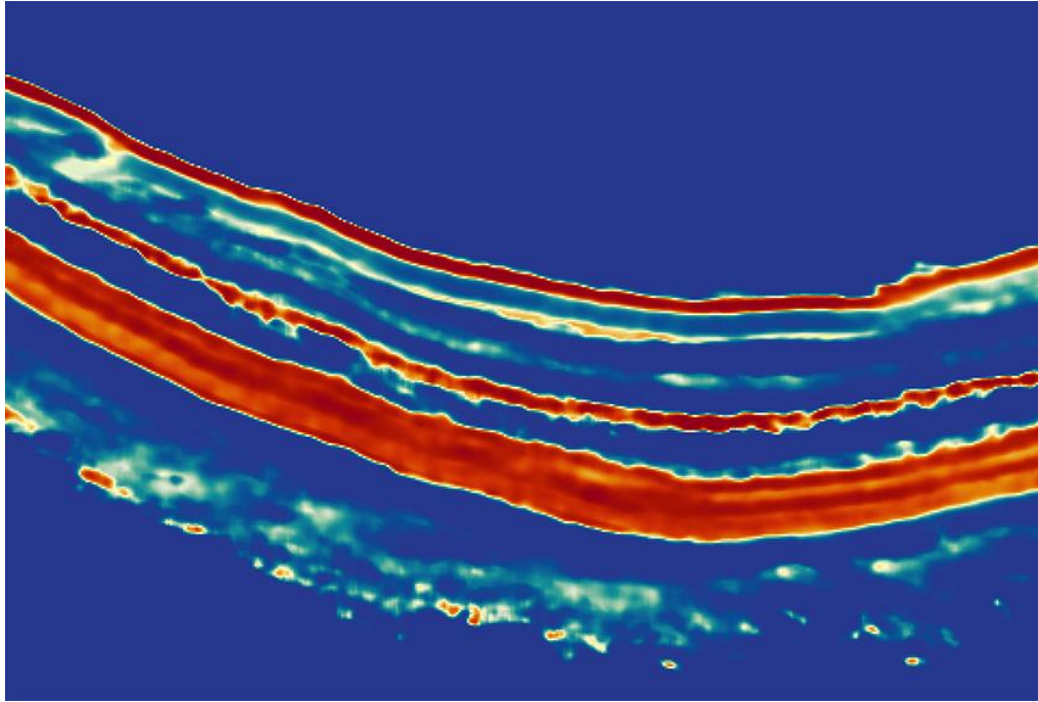


Figura 28. (a) Imatge OCT original i (b) imatge d'etiquetes amb les fronteres originals ("groundtruth") entre regions



*Figura 29. Imatge amb les probabilitats de cada píxel de ser una frontera entre capes*

A continuació, com en el mètode anterior, per a obtenir unes fronteres ben definides provem les diferents opcions que hem explicat:

Opció 1: En la figura 30 mostrem imatges resultants de binaritzar el mapa de probabilitats amb diferents llindars.

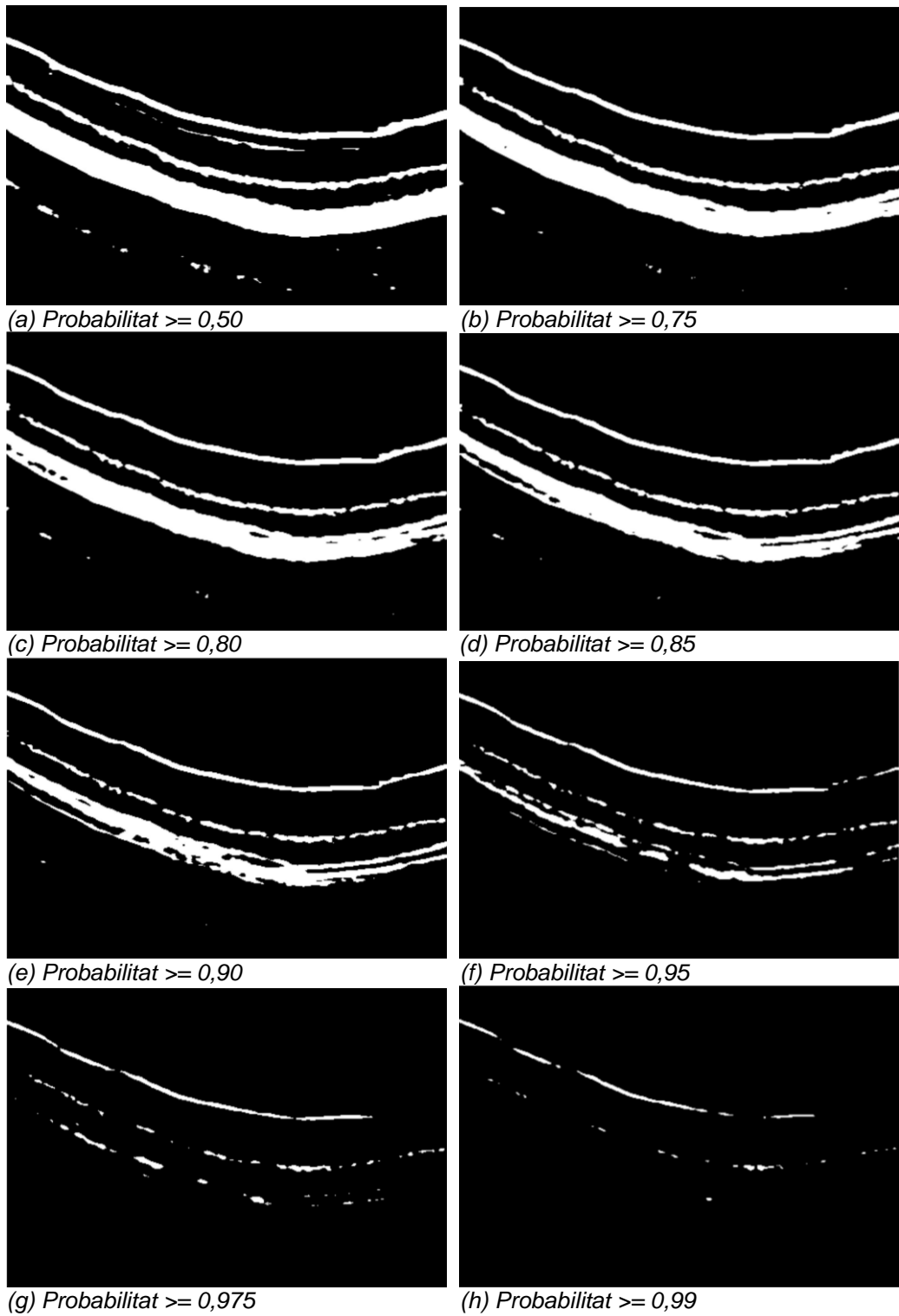
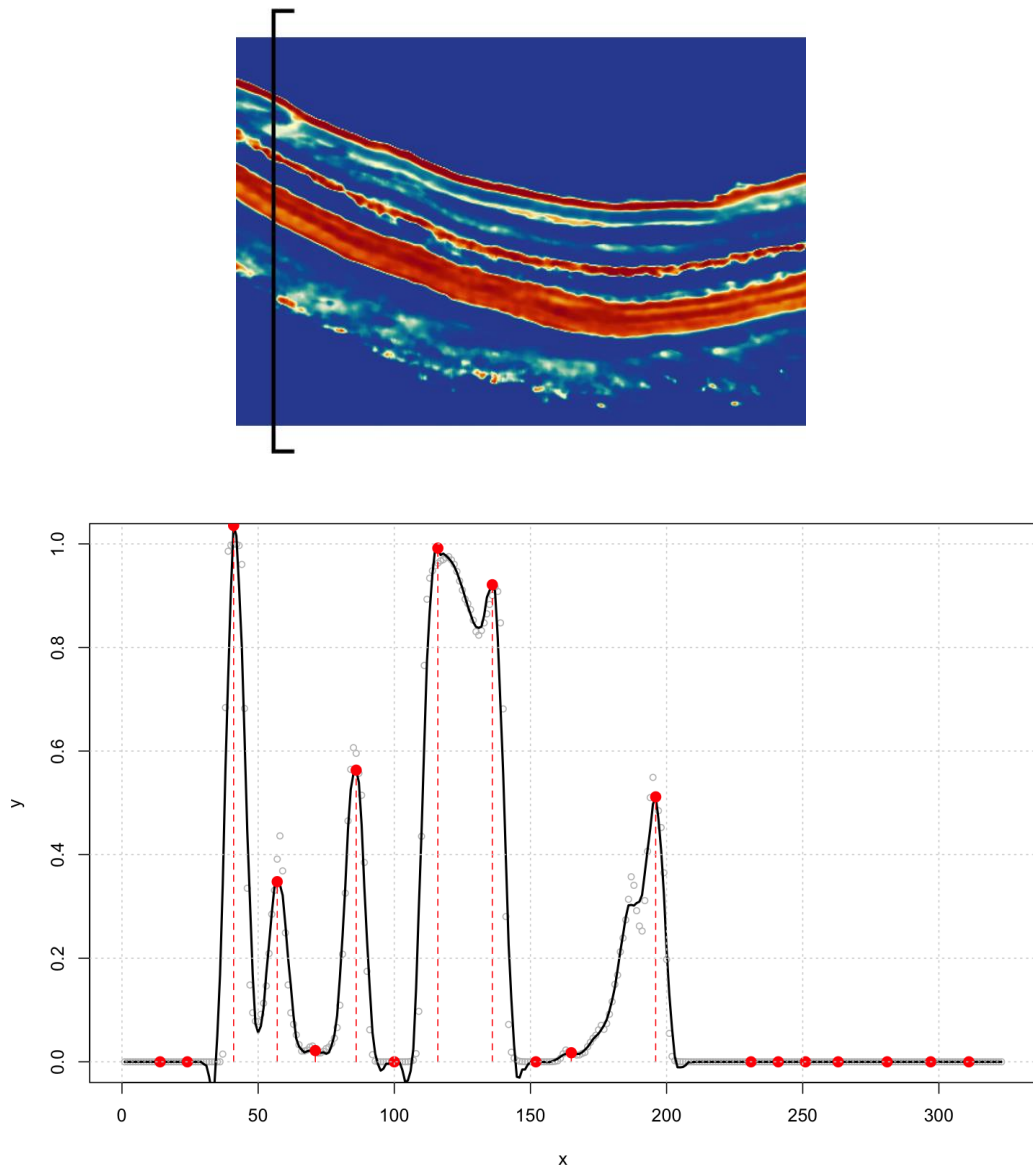


Figura 30. Imatges de probabilitats binaritzades en funció de diferents llindars

Veiem que ara perdem la detecció a partir de valors de llindar més alts que abans (0,95), cosa que indica que els valors de probabilitat obtinguts amb el mètode 1 són més alts i per tant, més ben detectats.

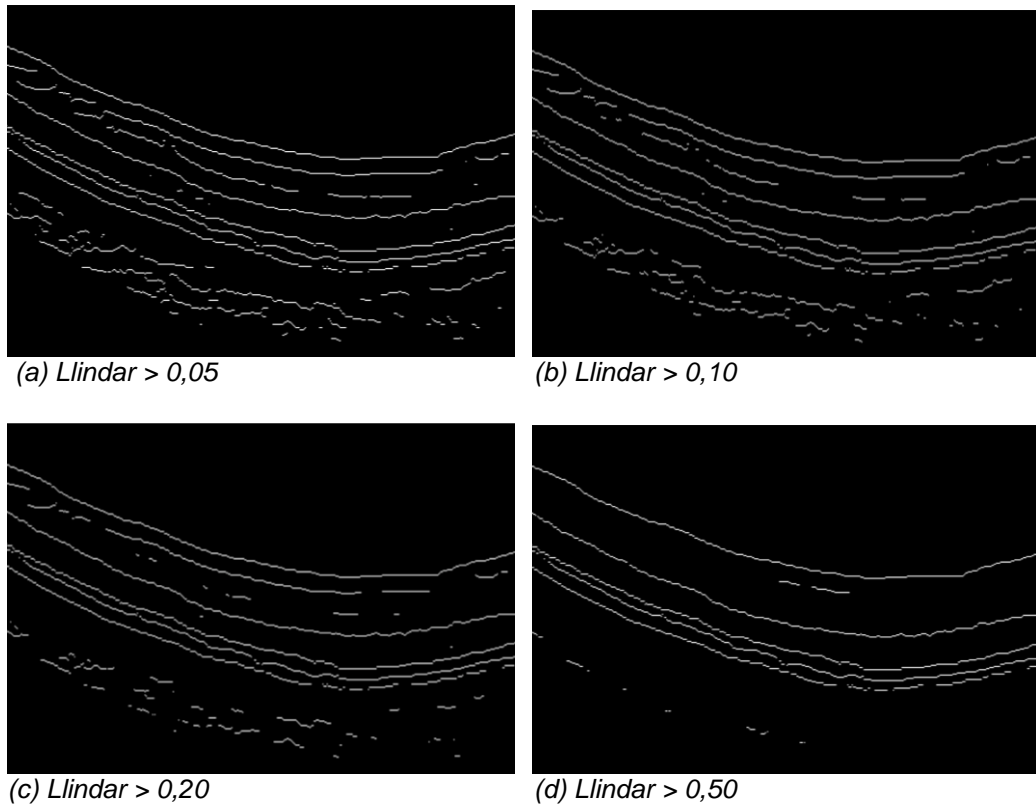


Opció 2: A partir de la imatge de probabilitats, la descomponem en cada una de les seves seccions verticals (columnes de píxels) i obtenim els seus màxims locals (fig.31).



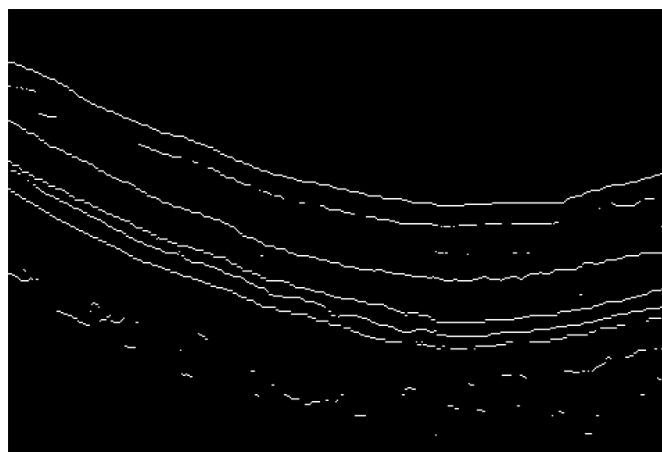
*Figura 31. Secció vertical (columna de píxels) de la imatge de probabilitats amb els màxims locals en vermell (gràfic inferior)*

Igual com hem fet amb el mètode 1, seleccionem els màxims locals i reconstruïm la imatge únicament amb les corbes conformades pels màxims locals seleccionats. En la figura 32 mostrem imatges reconstruïdes per a diferents valors de llindar.



*Figura 32. Fronteres entre capes retinals obtingudes per l'addició dels màxims locals de cada secció vertical*

Mitjançant la selecció dels 6 màxims locals que tinguin valor de probabilitat més alt, obtenim la imatge de la figura 33:



*Figura 33. Fronteres entre capes retinals obtingudes per l'addició d'únicament 6 dels màxims locals de cada secció vertical*

#### 5.4. Mètode 3: Classificació dels píxels de la imatge entre múltiples regions, fent servir sub-imatges com entrada

En aquest cas, disposem d'un total de 8 classes diferents:

- una classe correspon als píxels del fons negre de la imatge
- una classe correspon als píxels de totes les fronteres entre capes
- 6 classes corresponen a les 6 capes retinals

En primer lloc, provem diferents configuracions de xarxes CNN amb un únic bloc "conv-pool", i iterem amb un número baix d'epochs (3) a través dels següents valors:

- Mides de finestra = [ 5x5 , 9x9 , 15x15 , 25x25 , 35x35 ]
- Número de blocs "2DConv-Pool" = 1
- Tipus de "pooling" = ['max', 'average']
- Número de filtres (Conv) = [ 32 , 64 , 128 ]
- Mides de kernel (Conv) = [ 1 , 3 , 5 ]
- Mides de kernel ("pool") = [ 1 , 3 , 5 ]
- Valors de DropOut = [ 0.15 , 0.20 , 0.25 ]

De totes les iteracions, l'arquitectura de model CNN amb un sol bloc "conv-pool" que millors resultats ens ha donat ha estat la constituïda per les següents capes:

- a) Capa de convolució 2D ("Conv2D")  
Número de filtres de sortida = 28  
Mides de la finestra de convolució (o "kernel") = (1,1)
- b) Capa "MaxPooling2D"  
Mides de la finestra de "pooling" = (1,1)
- c) Flatten (flattening the 2D arrays for fully Connected layers)
- d) Dense (128, amb funció d'activació "Relu")
- e) Dropout (0,2)
- f) Dense ( num de classes=8, amb funció d'activació "softmax")
- g) Compile (amb optimitzador "adam", loss=sparse categorical crossentropy, metrics="accuracy")

Amb:

Número d'epochs = 10  
Mida de Batch = 128

Tractant-se d'un problema multi-classe, no disposem de positius ni negatius, i per tant, no podem presentar els resultats amb el mateix format que en els 2 mètodes anteriors.

En la taula 34 mostrem els millors resultats que hem obtingut per diferents mides de finestra i amb arquitectures CNN amb un sòl bloc “conv-pool”.

*Taula 34. Valors de validació per a xarxa CNN amb un bloc “conv-pool”*

Mida Subimatge	Entrenament		Validació		Test
	Loss	Accuracy	Loss	Accuracy	Accuracy
5x5	1,7927	0,2565	1,8200	0,2492	0,3400
9x9	1,5772	0,3586	1,5869	0,3548	0,3776
15x15	1,2930	0,4764	1,3235	0,4680	0,3919
25x25	0,9721	0,5913	1,1404	0,5399	0,4280
35x35	0,7344	0,6889	1,1365	0,5755	0,4777

Donat que, des del punt de vista de les mesures de validació els resultats obtinguts amb un sòl bloc “conv-pool” no són prou bons, seguim provant diferents arquitectures CNN, però treballant ara amb més d’un bloc de convolució en la seva estructura, i després de només 5 epochs i mida de batch 128, obtenim els resultats de la taula 35.

*Taula 35. Valors de validació per a xarxa CNN amb doble capa “conv”*

Mida Subimatge	Entrenament		Validació		Test
	Loss	Accuracy	Loss	Accuracy	Accuracy
35x35	0,6122	0,7321	0,9328	0,6340	0,5313

Veiem que els resultats, tot i ser igualment valors molt baixos, milloren una mica.

L’estructura del model CNN que hem entrenat anteriorment i que per limitacions del calendari adoptem com a “definitiva” presenta la següent configuració (fig.36):

- a) Capa de convolució 2D (“Conv2D”)
  - Número de filtres de sortida = 64
  - Mides de la finestra de convolució (o “kernel”) = (3,3)
- b) Capa de convolució 2D (“Conv2D”)
  - Número de filtres de sortida = 128
  - Mides de la finestra de convolució (o “kernel”) = (3,3)

- c) Capa “MaxPooling2D”  
Mides de la finestra de “pooling” = (2,2)
- d) Dropout (0,25)
- e) Flatten (flattening the 2D arrays for fully Connected layers)
- f) Dense (128, amb funció d’activació “Relu”)
- g) Dropout (0,2)
- h) Dense ( num de classes=8, amb funció d’activació “softmax”)
- i) Compile (amb optimitzador “adam”, loss=sparse categorical crossentropy, metrics=”accuracy”)

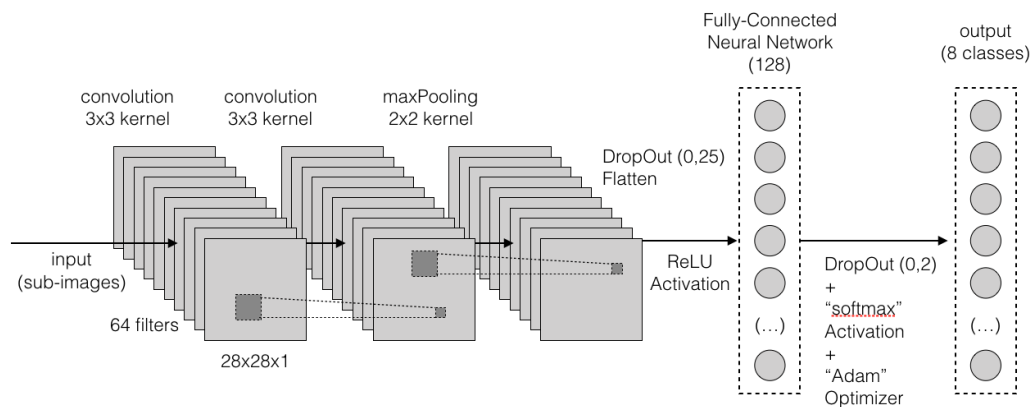


Figura 36. Arquitectura de xarxa CNN

Un cop definit el model final, l’executem sobre tots els píxels d’una imatge (fig.37).

Tractant-se ara d’un problema multi-classe, ja no presentem un mapa de probabilitats, sinó una imatge amb la classificació obtinguda, identificant cada una de les 8 classes.

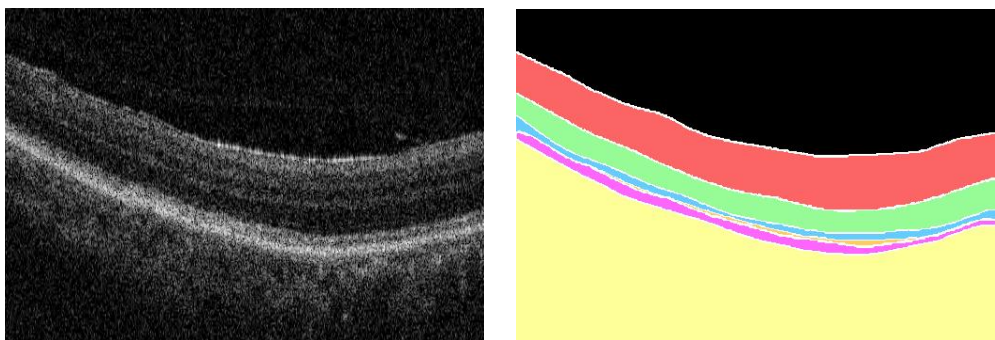
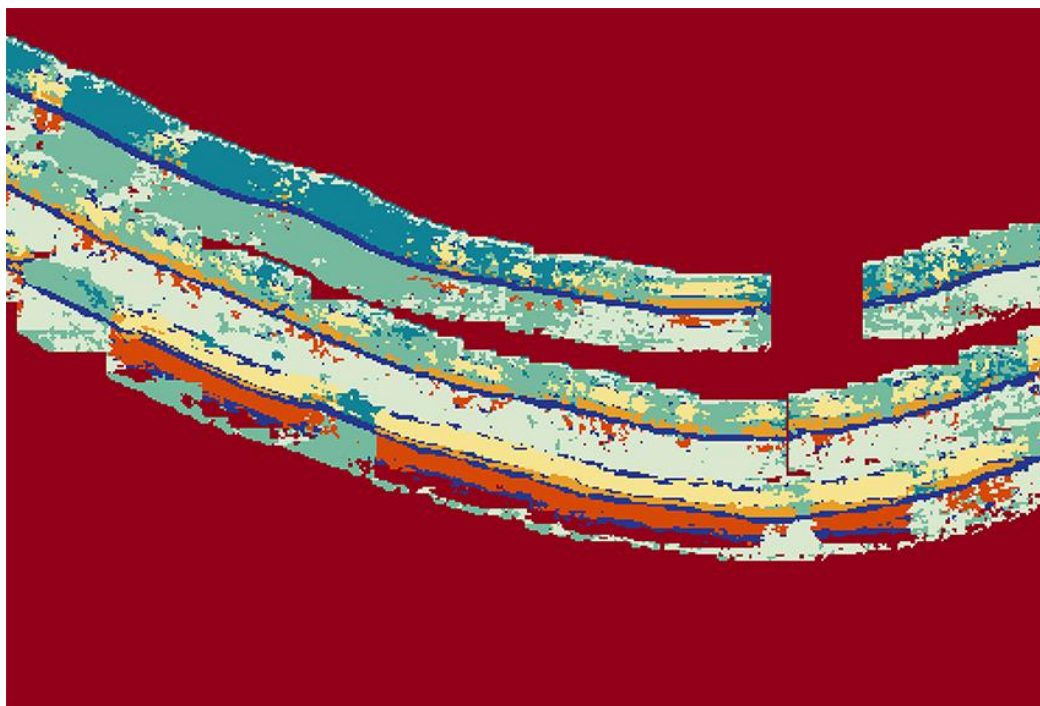


Figura 37. (a) Imatge OCT original i  
(b) imatge d’etiquetes amb les regions originals (“groundtruth”)



*Figura 38. Imatge amb la classificació de cada píxel*

La imatge mostrada (Fig.38) correspon a la classificació de tots els píxels d'una imatge amb l'arquitectura final del model CNN especificada anteriorment, i executada per a subimatges de 35x35 píxels. Les mides de la imatge final és més petita per l'efecte del "padding".

## 5.5. Discussió

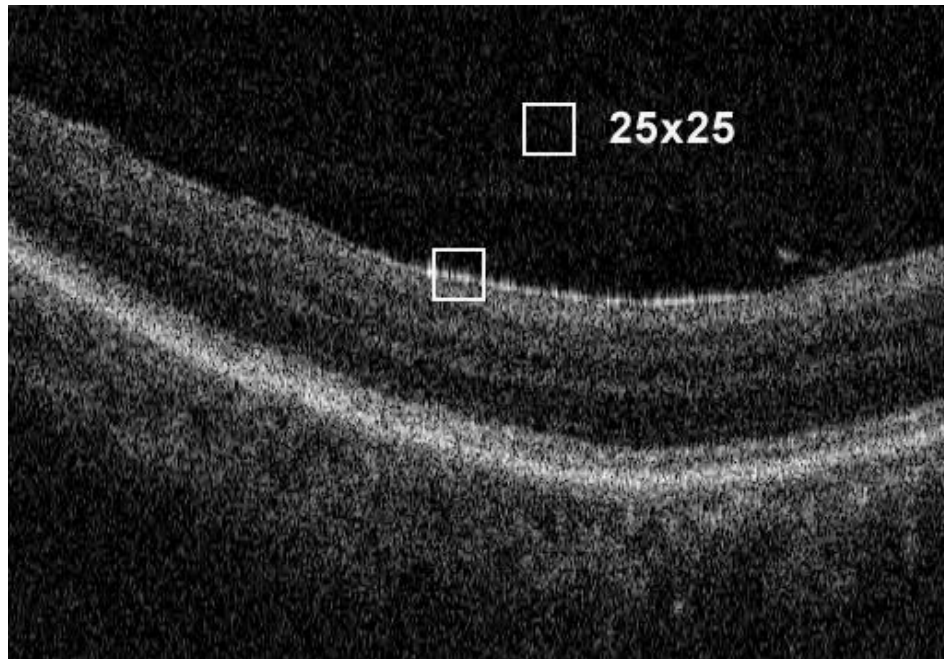
Observant els resultats del mètode 1, veiem que, en el processat dels màxims locals de cada secció, tant amb el filtrat de màxims amb llindars (amb valor 0,20) (fig.20) com amb la selecció dels 6 màxims locals més rellevants (fig.21), obtenim relativament bé 4 de les 6 fronteres entre capes retinals. Podem observar:

- 4 fronteres (de 6) gairebé contínues d'esquerra a dreta de la imatge.
- Tot de línies discontinües, algunes corresponents a altres fronteres, i d'altres senzillament mal detectades (falsos positius). Si ara apliquéssim un mecanisme de neteja (p. ex. descartar totes les línies amb mida per sota d'un llindar), segurament les perdríem totes, i ens quedarien només les 4 fronteres bones.

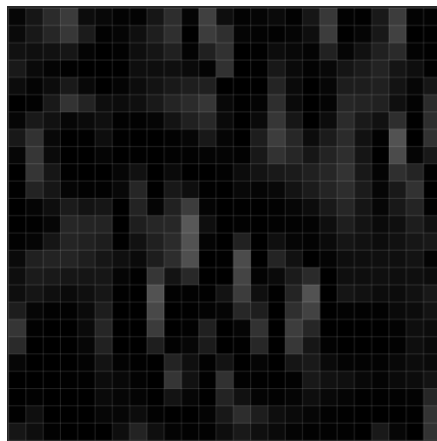
No és estrany que els resultats de classificar dues classes amb entrades de tipus estadístic (mètode 1) siguin bons, ja que, si observem detingudament en la figura 39 el tipus de subimatge que no contenen

fronteres (fig.39b) i les que sí (fig.39c), veiem clarament que les primeres contenen distribucions d'intensitat relativament homogènies, mentre que les segones presenten la discontinuïtat pròpia del salt entre capes.

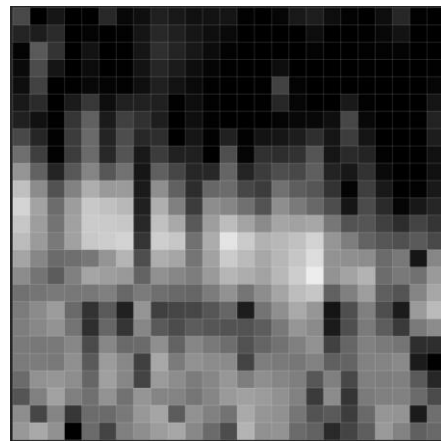
Aquesta característica és molt fàcilment explicable amb mesures estadístiques com el rang de nivells de gris o la variabilitat.



a.



b.



c.

*Figura 39. Exemples de subimatges de 25x25 píxels  
extretes de la imatge (a) i  
etiquetades com a (b) negatiu o (c) positiu*

Des d'un punt de vista de descriptors estadístics de primer i segon ordre, les finestres etiquetades com a negatiu presenten una distribució (fig.40) de valors d'intensitat relativament homogenis, mentre que les finestres que inclouen múltiples capes retinals presenten distribucions més heterogènies, i en el millor dels casos, bimodals.

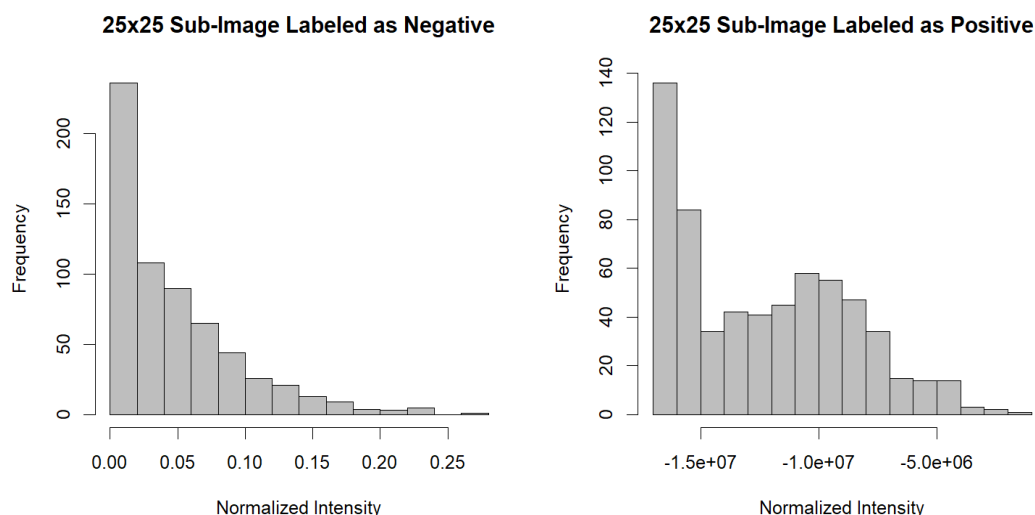


Figura 40. Histogrames d'intensitats de gris dels exemples de subimatges etiquetades com a (figura 39.b) negatius o (figura 39.c) positius

Pel que fa al mètode 2, comparant les imatges de probabilitats obtingudes amb els mètodes 1 (fig.17) i 2 (fig.29), podem dir que en el segon cas els valors de probabilitat són més alts, generant una imatge menys borrosa i amb molt més contrast entre classes. Per tant, parlant estrictament de les probabilitats obtingudes amb els diferents models, podem afirmar que són més acurats en aquest segon cas.

Quant al filtrat de màxims locals, amb un llindar de 0,50, veiem en la figura 32 que obtenim un resultat prou bo, detectant 5 fronteres gairebé completes en tota la seva longitud, i sense gaires falsos positius.

En aquest segon mètode obtenim:

- 5 fronteres (de 6, una més que amb el mètode 1) gairebé contínues d'esquerra a dreta de la imatge.
- Tot de línies discontinües, algunes corresponents a altres fronteres, i d'altres senzillament mal detectades (falsos positius). Si ara apliquéssim un mecanisme de neteja (p. ex. descartar totes les línies amb mida per sota d'un llindar), segurament les perdríem totes, i ens quedarien només les 4 fronteres bones.

En aquests dos mètodes (dues classes), observem que, des d'un punt de vista de correctesa dels models, atenent únicament a un conjunt de mesures de validació, els resultats obtinguts són bons.

Dit això, tot i aquests bons resultats des d'un punt de vista quantitatiu, en ambdós casos és necessari aplicar diversos mecanismes de refinament.



Els principals inconvenients que trobem són:

- El model entrega les fronteres en termes de probabilitat (d'existència de frontera), i per tant, cal binaritzar aquests valors (assignar-los de forma inequívoca a una classe o a una altra).
- Un cop binaritzades les fronteres el gruix d'aquestes no és un sol píxel, i per tant, és necessari afinar-les. En alguns casos, aquesta tasca no és fàcil perquè el gruix de les línies fa que múltiples fronteres apareguin solapades.
- Algunes de les fronteres trobades presenten discontinuïtats amb trams no detectats, de manera que cal reconstruir aquests trams que s'han perdut.

De tota manera, un cop superats aquests inconvenients mitjançant diferents feines de post-processat de les sortides del model, s'acaben obtenint resultats prou bons.

Observant els resultats del mètode 3, tot i que els números de les mesures de validació no són tan bons com en els altres dos casos, en la imatge final (fig.38) sí veiem que molts píxels han estat ben segmentats

Veiem que les regions del fons negre de la imatge i de la part interior del globus ocular han estat assignades a la mateixa classe (color vermell), i també que molts píxels de capes retinals també hi han estat assignades. És possible que es degui a una qüestió de mal balancejat entre classes, perquè aquestes dues regions tenen una mida molt superior a la de les capes retinals. En el futur caldria revisar aquesta fase.

A més, apart d'aquesta qüestió, entenem que allargant la fase d'optimització seria possible treballar una arquitectura CNN més complexa, i aconseguiríem una millor segmentació.

## 6. Conclusions

En aquest treball ens proposàvem detectar capes retinals en imatges OCT, i hem decidit estructurar-ho en forma de comparativa entre 3 procediments diferenciats:

- 1) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir estadístics locals de primer i segon ordre com entrades al model
- 2) Detecció de fronteres amb dues classes (“no frontera” vs “frontera”) i fent servir subimatges com entrades al model
- 3) Classificació dels píxels de la imatge entre múltiples regions (“fons d’imatge”, “interior globus ocular”, “fronteres entre capes”, “capa retinal 1”, “capa retinal 2”, etc), i fent servir subimatges com entrades al model

Hem començat processant la base de dades d’imatges OCT i hem entrenat i validat diferents models de classificació en funció del que corresponia en cada un dels tres casos.

En el primer cas hem utilitzat un model tipus Random Forest, en el segon hem entrenat una xarxa neuronal de convolució (CNN) amb un sòl bloc “conv-pool”, i en el darrer cas, hem fet ús d’una xarxa també CNN però amb múltiples blocs de convolució.

En els dos primers casos, en els que els models detecten les fronteres entre capes retinals enlloc de les pròpies capes, hem obtingut resultats de classificació molt bons ( $AUC=0,9823$  i  $AUC=0,9587$  respectivament). Com a principal inconvenient tenim que ambdós procediments requereixen de procediments addicionals que permetin segmentar la imatge a partir de les sortides dels models. Tot i així, les imatges finals que hem mostrat (figs. 20c i 32d) arriben a resultats prou bons: amb el mètode 1 podem detectar relativament bé 4 (de les 6) capes retinals, mentre que amb el mètode 2 en podem localitzar 5 (de 6).

En el tercer procediment en canvi, per una banda els resultats obtinguts no són els esperats en termes de mesures de validació del model, però tractant-se d'una estratègia de segmentació directa, presenta l'avantatge que no faria falta aplicar recursos de post-processat com en els mètodes 1 i 2.

Siguin quins siguin els resultats obtinguts en aquest treball, sens dubte, el principal objectiu que ens proposàvem ha estat assolit sobradament: el de fer recerca amb xarxes neuronals profundes i ser capaços, no només de programar-les i fer-les funcionar correctament amb una base de dades concreta, sinó també de comprendre-les i educar una determinada intuïció sobre què fa falta modificar o re-dissenyar per a millorar-ne els resultats.

Quant a la previsió temporal del treball, encara que al llarg del curs els treballs han anat transcorrent amb fluïdesa i hem anat assolint les diferents fites que ens havíem proposat en la planificació, les darreres setmanes hem hagut d'alterar lleugerament els nostres plans pel fet que les xarxes amb sortides multi-classe no acabaven de funcionar com preteníem.

Pel que fa a la planificació de futures feines, tot i haver assolit amb èxit els objectius que ens havíem proposat, atenent a les mesures de validació dels models usats, els resultats presentats per a la classificació multi-classe són inferiors al que esperàvem, o sigui que en el futur seria necessari revisar l'arquitectura de xarxa neuronal multi-bloc que hem utilitzat per tal d'intentar millorar-ne el comportament.

## 7. Bibliografia

### Articles:

- [1-01] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6.
- [1-02] M. Ali and B. Hadj, "Segmentation of oct skin images by classification of speckle statistical parameters," *2010 IEEE International Conference on Image Processing*, Hong Kong, 2010, pp. 613-616.
- [1-03] D. Alonso-Caneiro, S. Read, J. Hamwood, S. Vincent, M. Collins, "Use of convolutional neural networks for the automatic segmentation of total retinal and choroidal thickness in OCT images", 1<sup>st</sup> Conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam, The Netherlands
- [1-04] P. M. Arabi, N. Krishna, N. V. Deepa, V. Ashwini and H. M. Prathibha, "A comparison of OCT and retinal fundus images for age-related Macular degeneration," *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Delhi, 2017, pp. 1-5.
- [1-05] G. C. Y. Chan, S. A. A. Shah, T. B. Tang, C. -. Lu, H. Muller and F. Meriaudeau, "Deep Features and Data Reduction for Classification of SD-OCT Images: Application to Diabetic Macular Edema," *2018 International Conference on Intelligent and Advanced System (ICIAS)*, Kuala Lumpur, 2018, pp. 1-4.
- [1-06] L. Fang, D. Cunefare, C. Wang, R. Guymer, S. Li, S. Farsiu, "Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search", *Biomedical Optics Express* 2732, Vol. 8, No. 5, 1 May 2017
- [1-07] K. Gopinath, J. Sivaswamy and T. Mansoori, "Automatic glaucoma assessment from angio-OCT images," *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, Prague, 2016, pp. 193-196.

- [1-08] A. Guha Roy, S. Conjeti, S. Phani Krishna Karri, D. Sheet, A. Katouzian, C. Wachinger, N. Navab, "ReLayNet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks", *Biomedical Optics Express* 3627, Vol. 8, No. 8, 1 August 2017
- [1-09] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, Beijing, 2017, pp. 721-724.
- [1-10] O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Computer Vision and Pattern Recognition, 2015*.
- [1-11] J. Sugmk, S. Kiattisin and A. Leelasantitham, "Automated classification between age-related macular degeneration and Diabetic macular edema in OCT image using image segmentation", *The 7th 2014 Biomedical Engineering International Conference*, Fukuoka, 2014, pp. 1-4.
- [1-12] J. Fujimoto, E. Swanson. "The Development, Commercialization, and Impact of Optical Coherence Tomography". *Investigative Ophtalmology & Visual Science (IOVS)*, Special Issue, Vol.57, No.9
- [1-13] R. Kafieh, H. Rabbani, M.D. Abramoff, M. Sonka. "Intra-retinal layer segmentation of 3D optical coherence tomography using coarse grained diffusion map". *Med Image Anal.* 2013 December; 17(8): . doi:10.1016/j.media.2013.05.006.
- [1-14] G. Lemaître, M. Rastgoo, J. Massich, C.Y. Cheung, T.Y. Wong, E. Lamoureux, D. Milea, F. Mériaudeau, D. Sidibé. "Classification of SD-OCT Volumes Using Local Binary Patterns: Experimental Validation for DME Detection" Hindawi Publishing Corporation, *Journal of Ophthalmology*, Volume 2016, Article ID 3298606, 14 pages
- [1-15] M. Awais, H. Muller, F. Meriaudeau. "Classification of SD-OCT images using Deep learning approach"
- [1-16] D. Cabrera, H.M. Salinas, C.A. Puliafito. "Automated detection of retinal layer structures on optical coherence tomography images". 12 December 2005 / Vol. 13, No. 25 / *OPTICS EXPRESS* 10200
- [1-17] S.J. Chiu, X.T. Li, P. Nicholas, C.A. Toth, J.A. Izatt, S. Farsiu. "Automatic segmentation of seven retinal layers in SDOCT"

images congruent with expert manual segmentation". 30 August 2010 / Vol. 18, No. 18 / OPTICS EXPRESS 19413

- [1-18] L. Breiman. "Random Forests", 2001
- [1-19] Abramoff, Michael D., Mona K. Garvin, and Milan Sonka. "Retinal imaging and image analysis." *IEEE reviews in biomedical engineering* 3 (2010): 169-208.
- [1-20] Antony, Bhavna, et al. "Automated 3-D method for the correction of axial artifacts in spectral-domain optical coherence tomography images." *Biomedical optics express* 2.8 (2011): 2403-2416.
- [1-21] F. Schroff, A. Criminisi, A. Zisserman. "Object Class Segmentation using Random Forests". Proceedings of the British Machine Vision Conference 2008, Leeds, September 2008
- [1-22] U. Knauer, U. Seiffert. "Fast image segmentation based on boosted random forests, integral images, and features on demand". 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)
- [1-23] L. Lefkovits, S. Lefkovits, S. Emerich, M. Vaida. "Random Forest Feature Selection Approach for Image Segmentation". Conference ICMV 2016, Nice
- [1-24] A. Giusti, D.C. Cirezan, J. Masci, L.M. Gambardella, J. Schmidhuber. "Fast Image Scanning with Deep Max-Pooling Convolutional Neural Networks". Technical Report No. IDSIA-01-13, February 8, 2013
- [1-25] H. Gouk, A.M. Blake. "Fast Sliding Window Classification with Convolutional Neural Networks". IVCNZ '14 Proceedings of the 29th International Conference on Image and Vision Computing New Zealand, Pages 114-118. November 2014
- [1-26] F. Tschopp, J.N.P. Martel, S.C. Turaga, M. Cook, J. Funke. "Efficient Convolutional Neural Networks for Pixelwise Classification on Heterogeneous Hardware Systems". 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)
- [1-27] D.P. Kingma, J. Lei Ba. "Adam: A Method for Stochastic Optimization". 3rd International Conference for Learning Representations, San Diego, 2015
- [1-28] N.I. Glumov, E.I. Kolomiyetz, V.V. Sergeev. "Detection of objects on the image using a sliding window mode". Optics &

Laser Technology, Volume 27, Issue 4, August 1995, Pages 241-249

- [1-29] A. Helwan, D.U. Ozsahin. "Sliding Window Based Machine Learning System for the Left Ventricle Localization in MR Cardiac Images". Applied Computational Intelligence and Soft Computing, Volume 2017, Article ID 3048181
- [1-30] D. Huang, E.A. Swanson, C.P. Lin, J.S. Schuman, SW.G. tinson, W. Chang, M.R. Hee, T. Flotte, K. Gregory, C.A. Puliafito, et al. "Optical coherence tomography". Science. 1991

**Webs:**

- [2-01] [www.thorlabs.com](http://www.thorlabs.com)  
(2019-04)
- [2-02] [https://en.wikipedia.org/wiki/Optical\\_coherence\\_tomography](https://en.wikipedia.org/wiki/Optical_coherence_tomography)  
(2019-04)
- [2-03] <https://newgradoptometry.com/complete-list-ocular-diseases-optical-coherence-tomography-oct/>  
(2019-04)
- [2-04] <https://towardsdatascience.com/how-to-teach-a-computer-to-see-with-convolutional-neural-networks-96c120827cd1>  
(2019-04)
- [2-05] <https://developer.nvidia.com/discover/convolutional-neural-network>  
(2019-04)
- [2-06] <https://www.pinterest.com.au/pin/500040364871634798/>  
(2019-04)
- [2-07] <https://webeye.ophth.uiowa.edu/eyeforum/tutorials/retinal-detachment-med-students/index.htm>  
(2019-04)
- [2-08] <https://www.mathworks.com/matlabcentral/fileexchange/43518-graph-based-segmentation-of-retinal-layers-in-oct-images>  
(2019-04)
- [2-09] <http://www.northerneyesurgeons.com.au/oct-scanning>  
(2019-04)
- [2-10] <https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.averagepooling2dlayer.html>  
(2019-04)
- [2-11] <https://keras.io/layers/local/>  
(2019-04)
- [2-12] <https://www.quora.com/In-Keras-what-is-a-dense-and-a-dropout-layer>  
(2019-04)
- [2-13] <https://www.quora.com/What-is-the-meaning-of-flattening-step-in-a-convolutional-neural-network>  
(2019-04)