



UNIVERSITAT DE
BARCELONA

Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Comparació de xarxes de segmentació semàntica d'aplicacions dermatològiques en l'entorn de l'ISIC Challenge.

Albert Morales Llobet

Director: Mr. Simone Balocco
Realitzat a: Departament de
Matemàtiques i Informàtica
Barcelona, 18 de gener de 2020

Agraïments

M'agradaria agrair profundament a Mr. Balocco Simone per la seva ajuda durant tot el treball. M'ha guiat amb gran saviesa responent a tots els dubtes que li presentava, sense dir-me exactament el que havia de fer en tot moment, m'ha donat pistes per quins camins havia de seguir i suport incondicional en tota la recerca. D'aquesta manera sense ell no hagués sigut possible aconseguir aquests resultats.

Agrair a la Facultat de Matemàtiques i Informàtica totes les facilitats en forma de material o d'infraestructures que m'han ofert durant el llarg de la carrera.

Finalment, remarcar que sense el companys i els professors amb el suport i l'aprenentatge transmès no hagués sigut possible aquest procés.

Abstract

Cancer affects millions of people every single year around the world. Making it the most common type of cancer, skin cancer owns the record of being responsible to end the life of thousands of human beings. Therefore finding a way to detect it and to classify it is a major issue. At the same time, we cannot forget about the importance of prevention and research about new ways to cure it.

Recently, there's been a significant improvement in the field of neural networks, more specific in the design and research about different kinds of networks that help to classify pixels from an image.

This project will analyze multiple sorts of models of neural networks such as U-net or DeepLab v3+, to further get an estimation of its performance regarding semantic segmentation using dermatologic images as a training set. Finally, we will emulate ISIC Challenge 2017 to see how well our networks do.

Yielding an accuracy of almost 94% and a F-Score around 86% DeepLab v3+ model has claimed the best result of all of them. As a result, further research in semantic segmentation's field has been encouraged so melanoma's detection process can be catalyzed. That way survival rate will directly be increased.

Abstracto

El cáncer afecta a millones de personas cada año en todo el mundo. Siendo el más común, el cáncer de piel es el responsable de terminar con la vida de miles de personas. Por este motivo encontrar una manera de poder detectarlo y clasificarlo es crucial. Al mismo tiempo no se tiene que olvidar la importancia de la prevención y la innovación de nuevas curas.

En estos últimos años se ha avanzado a pasos de gigante en el ámbito de las redes neuronales, específicamente en el diseño y la investigación sobre diferentes tipos de redes que ayudan a clasificar píxeles de una imagen.

En el proyecto se analizarán distintos modelos de redes como la U-net o DeepLab v3+ entre otras, para tener una estimación del rendimiento de estas redes en relación a la segmentación de imágenes dermatológicas. Finalmente se reproducirá un simulacro de participación en el ISIC Challenge del 2017.

Con unos resultados significativos de exactitud de casi un 94% y una F-Score de cerca del 86% el modelo DeepLab v3+ ha liderado los experimentos realizados. Motivando de esta manera una futura investigación de estas tecnologías para segmentar imágenes y de esta manera estar un paso más cerca de acelerar la

detección de lunares. Aumentando de manera directa la expectativa de supervivencia del paciente.

Abstracte

El càncer afecta a milions de persones cada any arreu del món. Sent el més comú el càncer de pell, és el responsable d'acabar amb la vida de milers de persones. Per aquest motiu trobar una manera de poder detectar-lo i classificar-lo és crucial. Tanmateix sense oblidar la importància de la prevenció i la innovació de noves cures.

Aquests últims anys s'ha avançat a passos de gegant en l'àmbit de les xarxes neuronals, més en específic en el disseny i recerca sobre diferents tipus de xarxes que ajuden a classificar píxels d'una imatge.

Al projecte s'analitzaran diferents models de xarxes com la U-net o DeepLab v3+ entre altres, per tenir una estimació del rendiment d'aquestes en referència a la segmentació d'imatges dermatològiques. S'acabarà reproduint un simulacre de participació al ISIC Challenge del 2017.

Amb uns resultats significatius d'exactitud de casi un 94% i una F-Score de quasi el 86% el model DeepLab v3+ ha liderat els experiments duts a terme. Motivant així una futura recerca d'aquestes tecnologies per segmentar imatges i fer un pas més a la detecció prematura d'aquesta malaltia. Augmentant de manera directa la expectativa de supervivència del pacient.

Índex

1	Introducció	9
1.1	Objectius	10
2	Càncer	11
2.1	Càncer de pell	11
2.1.1	No-melanoma	12
2.1.2	Melanoma	12
3	Estat de l'art	15
4	Dataset	17
5	Deep Learning	21
5.1	Història	21
5.2	Neurona	21
5.3	Estructura	23
5.3.1	Funcions d'activació	23
5.4	Aprenentatge	27
5.4.1	Cost function	27
5.4.2	Gradient descent	27
5.4.3	Backpropagation	28
5.5	Convolutional Neural Networks (CNN)	28
5.5.1	Convolutional layer	30
5.5.2	Pooling	31
5.5.3	Batch normalization	32
5.5.4	Fully connected layer	32
5.5.5	Dropout layer	32
5.5.6	SoftMax layer	33
5.6	Transfer Learning	33
6	Metodologia	34
6.1	Hardware	34
6.2	Software	34
6.3	Data Augmentation en MATLAB	35
6.4	Networks	35

6.4.1	Simple CNN.....	36
6.4.2	U-net.....	36
6.4.3	DeepLab v3 amb resnet18	38
7	Experiments.....	41
7.1	Mètrica d'avaluació de la xarxa.....	41
7.2	Mètode d'avaluació de la xarxa	43
7.3	Primers experiments	44
7.4	U-net.....	46
7.5	DeepLab v3+ amb ResNet-18	47
7.6	U-Net vs DeepLab-ResNet-18.....	48
8	Resultats.....	49
9	Conclusions i treball futur.....	57
9.1	Conclusions	57
9.2	Treball futur.....	57
10	Bibliografia	59

Índex de figures

Figura 1. Comparació dels diferents tipus de càncer, en nou casos i morts al 2019	12
Figura 2. Tant per cent de supervivència en cinc anys depenent de la fase de detecció del càncer	13
Figura 3. Tant per cent dels casos de supervivència en relació a la fase de detecció	13
Figura 4. Estructura de la xarxa de Yuan Y.	15
Figura 5. Exemples del diferents objectes a les imatges.....	18
Figura 6. Exemple d'una imatge amb la seva corresponent màscara.....	19
Figura 7. Histogrames de la mida de les imatges.....	20
Figura 8. Exemple visual d'una neurona	21
Figura 9. Esquema d'una neurona d'una ANN	22
Figura 10. Esquema d'una estructura de ANN	23
Figura 11. Gràfic d'una funció binària a l'esquerra. Fórmula d'una funció binària a la dreta (2)	24
Figura 12. Gràfic d'una funció d'activació lineal a l'esquerra. Fórmula d'una funció d'activació lineal a la dreta (3).....	24
Figura 13. Gràfic d'una funció sigmoid a l'esquerra. Fórmula d'una funció sigmoid a la dreta (4)	25
Figura 14. Gràfic d'una funció TanH a l'esquerra. Fórmula d'una funció TanH a la dreta (5).....	25
Figura 15. Gràfic d'una funció ReLU a l'esquerra. Fórmula d'una funció ReLU a la dreta (6).....	26
Figura 16. Gràfic d'una funció Leaky ReLU a l'esquerra. Fórmula d'una funció Leaky ReLU a la dreta (7).....	26
Figura 17. Gràfic d'una funció paramètrica ReLU a l'esquerra. Fórmula d'una funció paramètrica ReLU a la dreta (8)	26
Figura 18. Exemples de funcions de pèrdua amb diferents tipus de valors d'aprenentatge.....	27
Figura 19. Exemple de CNN.....	29
Figura 20. Operació de convolució a una imatge $M \times N \times 3$ i un kernel de $3 \times 3 \times 3$	30
Figura 21. Exemples d'Atrous convolutions amb rates: 1, 2 i 3. D'esquerra a dreta	31
Figura 22. Exemple del resultat d'aplicar max pooling(superior) i average pooling(inferior)	32
Figura 23. Exemple visual de com MATLAB implementa data augmentation	35
Figura 24. Exemple d'estructura U-net	37
Figura 25. Exemple d'estructura encorder-decoder	38
Figura 26. Estructura d'una ResNet-18 a l'esquerra i d'una ResNet-50 a la dreta.....	39
Figura 27. Gràfic que representa l'accuracy de les xarxes en relació al temps de predicció[31]	40

Figura 28. Atrous Spatial Pyramid Pooling(ASPP)	40
Figura 29. Com es distribueixen les dades de sortida per analitzar-se'n el seu resultat	41
Figura 30. Esquema visual de com es distribueixen els sets i com es calcula l'error	43
Figura 31. Taula de com es reparteix el dataset a cada iteració a l'esquerra i exemple visual a la dreta	44
Figura 32. A la part superior de la imatge es veu una funció d' <i>accuracy</i> en relació a cada iteració. A la part inferior es pot veure la funció de pèrdua	49
Figura 33. A l'esquerra, resultats de 25 imatges amb el contorn del ground truth(vermell) i l'obtingut per la U-net utilitzada. A la dreta el resultat resultant del model DeepLab v3+	53
Figura 34. Resultats del ISIC Challenge 2017	54
Figura 35. Resultats generats per la DeepLab v3+. A la línia superior hi ha mals resultats, a la línia del mig hi ha molt bons resultats i a la línia inferior hi ha resultats mediocres.....	55
Figura 36. Resultats generats per la U-net amb la mateixa distribució de la Figura 35	56

Índex de taules

Taula 1. Taula de la distribució de les imatges de cada base de dades.	17
Taula 2. La distribució d'imatges presentada pel Challenge	19
Taula 3. Taula de la distribució dels dataset per cada iteració amb una k=4 i dos tests.	45
Taula 4. Taula de la distribució del dataset per cada iteració amb k=5	46
Taula 5. Taula dels resultats	49
Taula 6. Taula dels resultats de la U-net al modificar el nombre d'iteracions.....	50
Taula 7. Taula dels resultats de la U-net al modificar el learning rate.....	50
Taula 8. Taula dels resultats de la U-net al modificar els tipus de transformacions	50
Taula 9. Taula dels resultats de la U-net al modificar la profunditat.....	51
Taula 10. Taula dels resultats al augmentar les iteracions amb i sense data augmentation	51
Taula 11. Resultats de la xarxa DeepLab v3+ modificant el nombre d'èpoques/iteracions	52
Taula 12. Resultats despres de canviar el learning rate inicial	52
Taula 13. Taula de resultats si s'aplica una learning rate variat	52
Taula 14. Taula dels paràmetres d'entrada de la U-net i la DeepLab v3+	53

1 Introducció

En aquests últims anys la ISIC(International Skin Imaging Collaboration) ha creat un projecte anomenat Melanoma Project el qual pretén reduir la mortalitat per causes dels melanomes i les biòpsies innecessàries. El repte que proposen conjuntament amb el ràpid creixement en l'àmbit de les xarxes neuronals, han catapultat els avenços en aquest camp de la medicina.

Una vegada s'entén la importància de detectar aquest tipus de malaltia a temps sorgeix la resposta immediata de crear una tecnologia que faciliti i agilitzi aquesta detecció.

Aquesta tecnologia per tant es podria aplicar com a aplicació pel públic en general, com a software utilitzat pels metges a consultes a distància o com a eina recurrent a la qual el metge pot consultar en qualsevol moment per complementar la seva decisió.

Paral·lelament a aquest projecte i promogut per la ISIC s'ha de continuar treballant en diferents àmbits que tot i no tenir relació directa en el desenvolupament de la tecnologia sí que afecten a una futura implementació d'aquesta. La privacitat, la qualitat de la imatge i la interoperabilitat són els aspectes que a dia d'avui no compleixen encara els paràmetres mínims per tal de que s'ajustin als estàndards requerits(tecnològics, de tècnica i de terminologia).

És important diferenciar quins passos existeixen a la hora d'identificar i classificar un càncer. Existeixen tres tipus diferents d'aplicacions, el primer es basa en la detecció de la lesió en si, en altres paraules, localitzar diferents lesions en una imatge d'una part del cos humà.

La segona tracta de segmentar una imatge d'una lesió per poder extreure diferents característiques que aportaran informació sobre la morfologia de la piga. Seguint el ABCDE del melanoma, farà referència a la asimetria(A), al contorn(B) i al diàmetre(D).

Finalment la tercera aplicació és la que classificarà la lesió en una de les següents categories: melanoma, carcinoma de cèl·lules basals(BCC), carcinoma de cèl·lules escatoses (SCC) o una lesió diferent no relacionada amb un càncer de pell.

En aquest projecte es treballaran les aplicacions referents a la segona categoria, la segmentació d'una lesió.

1.1 Objectius

En primer lloc l'objectiu del projecte és obtenir una xarxa neuronal que sigui capaç de segmentar una imatge dermatològica explorant els diferents tipus de xarxes i endinsant-nos en l'estat de l'art en aquest àmbit.

Tanmateix es comprovaran els resultats utilitzant la metodologia presentada per el ISIC Challenge 2017, el que permetrà posicionar la xarxa obtinguda en relació a la dels altres participants.

2 Càncer

Essent la segona causa de mortalitat als Estats Units, el càncer és una de les majors preocupacions a les societats desenvolupades[1]11. No tan sols acaba aquí el problema, estudis recents adverteixen de que el ràpid desenvolupament global provocarà una reducció de la mortalitat relativa a malalties infeccioses a països menys desenvolupats, pel que convertirà el càncer en la major causa de mortalitat a nivell global[2].

Coneixem el càncer com el conjunt de malalties que inclouen el creixement anormal de les cèl·lules amb la capacitat d'envair o reproduir-se a altres part del cos. La capacitat d'expandir-se a altres parts es coneix com a metàstasi, aquesta acostuma a succeir a les últimes etapes del càncer i probablement serà el causant de la mort del pacient.

Tot i que és difícil comprovar que ha causat el càncer, hi ha diferents factors els quals s'ha demostrat que modifiquen el nivell de risc per contraure la malaltia. El tabac és responsable del 25% al 30% de les morts per càncer fins a un 87% a les morts per càncer de pulmó; l'alcohol; d'un 30% a un 35% de les morts per càncer tenien algun tipus de relació amb la dieta de l'individu; el pes; la contaminació; l'exposició a elements radioactius; virus; entre molts altres[3].

Els càncers es poden classificar en quatre categories[4]:

- **Carcinomes:** S'origina a l'epiteli. (Càncer de pell, càncer de pròstata, ...)
- **Sarcomes:** S'origina al teixit conjuntiu, com cartílags, músculs, etc.
- **Leucèmia:** S'origina a la sang. (LAL, LLC, LAM, LMC)
- **Limfomes:** S'origina al teixit limfoide. (Limfoma No-Hodgkin i malaltia de Hodgkin)

2.1 Càncer de pell

Com el seu nom indica els càncers de pell són aquells que s'originen a l'epidermis. El principal risc per contreure aquest tipus de càncer està molt relacionat amb l'exposició a la llum ultraviolada entre altres factors de risc. El càncer de pell és el més comú als Estats Units, es calcula que un de cada cinc persones desenvoluparan un al llarg de la seva vida[5].

Les dades dels càncers de pell en general no solen ser acurades en tant que la gran majoria d'aquests són no-melanomes, els del tipus més benigne, i acostumen a no ser reportats de manera tant estricta com els melanomes per exemple. Aquests segons si que estan ben datats[6].

Els càncers de pell es classifiquen en dos categories: melanomes, i no-melanomes.

2.1.1 No-melanoma

Existeixen dos tipus de càncer de pell no-melanoma, carcinoma de cèl·lules basals(BCC) i carcinoma de cèl·lules escatoses (SCC)[7].

Els BCC solen aparèixer a les parts exposades al sol com el nas(cap en general) i espatlles, són de textura suau i sobresurten de la pell. És el tipus de càncer de pell més comú i el que menys risc suposa pel pacient. Normalment es tracta amb cirurgia tot i que en casos estranys pot arribar a quimioteràpia si està molt estès.

Els SCC com el BCC, sol ser causat per l'exposició al sol. Comença com un petit bony i a mesura que es va fent gran apareix necrosis al centre. Aquest tipus de càncer si que tendeix a estendre's.

2.1.2 Melanoma

Els càncers de tipus melanoma són els més agressius de la família dels càncers de pell però a la vegada el menys comú dels dos. S'estima que aquest 2019 es detectaran casi 100.000 nou casos als Estats Units, això representa un 5.5% del total dels casos de càncer detectats durant aquest any, convertint el càncer de melanoma com el cinquè més freqüent com es pot veure a la Figura 1[8].

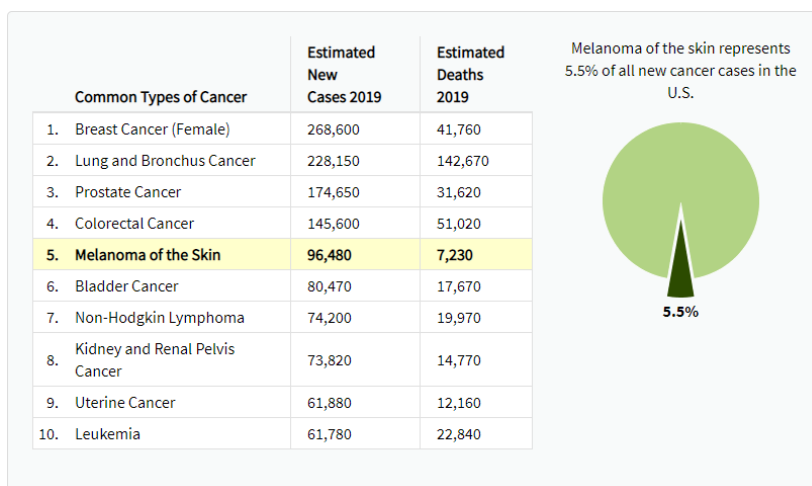


Figura 1. Comparació dels diferents tipus de càncer, en nou casos i morts al 2019

Els melanomes es generen a les pigues, és quan els melanòcits comencen a créixer en nombre de forma ràpida. Aquest tipus de càncer no té cap limitació pel que fa a les parts del cos on pot aparèixer, en els homes els llocs més freqüents són al cap i el pit i en les dones als braços i les cames.

Els homes de 65 anys de pell clara són els més propensos a contreure aquest càncer, altres factors de risc que engloben els càncers de pell en general també s'apliquen com serien: exposició al sol, tenir un sistema immunitari dèbil, historial familiar, estar submís a determinats tipus de radiacions, dissolvents i altres tipus de materials, entre altres. En aquest en particular el fet de tenir una quantitat de pigues elevada també contribueix en un major factor de risc.

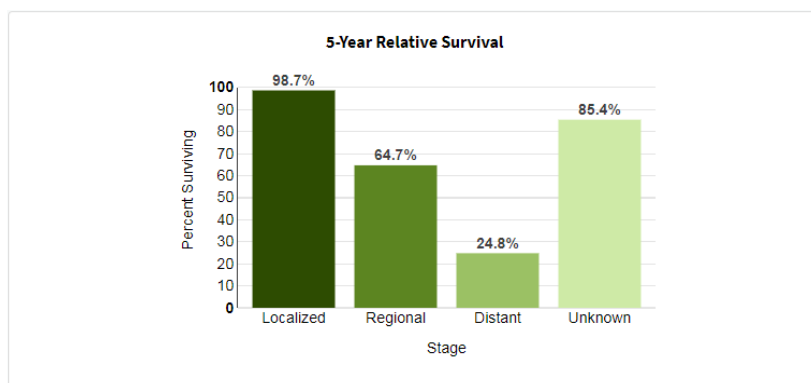


Figura 2. Tant per cent de supervivència en cinc anys dependent de la fase de detecció del càncer

Com es pot observar a la Figura 1 tot i ser el cinquè en casos detectats té un nombre de morts més petit en comparació als altres. Això és degut, entre altres coses, a que el càncer es produeix en zones externes del cos (En rares ocasions es poden produir als ulls, als intestins o a la boca). També l'índex de supervivència durant els següent cinc anys és molt elevat per tots aquells càncers que han sigut detectats i tractats en la primera etapa, accentuant la relació entre esperança de vida i fase de detecció del càncer – Figura 2 i Figura 3-.

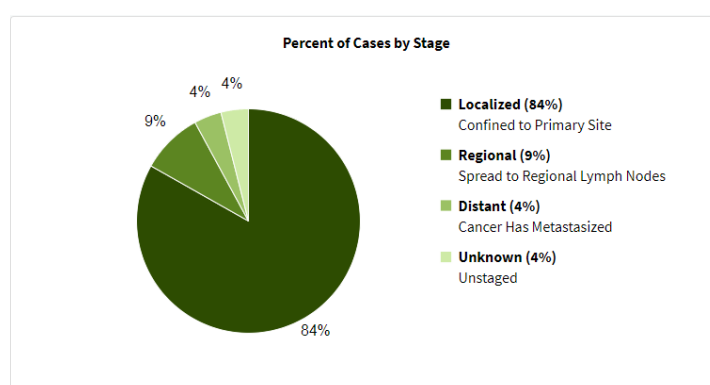


Figura 3. Tant per cent dels casos de supervivència en relació a la fase de detecció

A nivell mèdic hi ha tres maneres de diagnosticar el pacient: examinant el cos de manera general complementant-ho amb l'historial mèdic del pacient; fent un examen de la pell; i la més intrusiva, una biòpsia[6].

A nivell personal és molt important detectar qualsevol canvi a la pell, analitzar-la i visitar a un doctor, catalitzant així el diagnòstic. Hi ha uns signes anomenats "ABCDE" del melanoma que fan entendre quins són els factors clau per detectar aquesta malaltia, en conseqüència, qualsevol individu pot fer-ho de forma regular.

"ABCDE":

- **A - (Asymmetrical):** Com d'iguals són les dues meitats de la piga.
- **B - (Border):** Com de definit esta el contorn.
- **C - (Color):** Com d'homogeni és el color.
- **D - (Diameter):** La mida de la piga. 6mm és la mida alarmant, si bé els melanomes poden ser més petits.
- **C - (Evolving):** Si la piga canvia.

El fet de poder treballar amb una imatge treta del mòbil, obtinguda per l'usuari, obra la porta a un munt de possibilitats. Crear aplicacions que accelerin o complementin la detecció d'aquests càncers atreu a diferents camps de la ciència. Especialment en l'àmbit de la informàtica, l'aprenentatge automàtic combinat amb el processament d'imatges.

3 Estat de l'art

Va ser Binder M. et al, un dels precursors del diagnòs de melanomes utilitzant una xarxa neuronal, quan el 1994 va aconseguir entrenar una xarxa satisfactòriament[9]. Tot i els bons resultats, l'estudi mancava d'una quantitat representativa d'imatges tant de positius com negatius.

En l'àmbit de la segmentació semàntica hi ha un canvi significatiu abans i després de les Convolutional Neural Networks(CNN). Previ a les xarxes els mètodes més utilitzats eren clàssics de l'aprenentatge automàtic. Fer *threshold* d'histogrames, *clustering*, segmentació morfològica, *fuzzy logic*, etc.

Va ser amb la revolució de les CNN quan, Collobert R i Pinheiro O.[10], fan una adaptació en el camp del processament d'imatges, de la feina de Jordan del 1986 sobre les Recurrent Neural Networks(RNN), per classificar píxels.

A partir d'aleshores múltiples experiments van dur-se a terme fins que les xarxes es van anar desenvolupant fins al dia d'avui.

Moral F. al 2017 entrenant un model de FCN-MatConvNet va aconseguir uns resultats de 0.64 ± 0.15 *Jaccard index* i un *Sørensen-Dice* de 0.77 ± 0.12 per la segmentació de 49 imatges amb una base de dades de 4400 imatges[11]. Al fer l'extrapolació, és a dir, al testejar contra imatges d'una altra base de dades, els resultats van ser de *Jaccard index* = 0.49 ± 0.16 i *Sørensen-Dice* = 0.64 ± 0.14 . Cal notar que va fer més tipus d'experiments amb diferents resultats com es pot consultar al seu treball, tot i que els que ens fixem són els més rellevants pel procediment del nostre experiment.

Yuan Y. implementant una CDNN amb l'estructura de la Figura 4 va aconseguir 0.765 *Jaccard index* i un *Sørensen-Dice* de 0.849 a l'ISIC Challenge del 2017 [12].

Conv	Filter size	No. of features	Deconv	Filter size	No. of features
conv-1-1	3×3	16	deconv-1	3×3	256
conv-1-2	3×3	32	ups-1	2×2	256
pool-1	2×2	32	deconv-2-1	3×3	256
conv-2-1	3×3	64	deconv-2-2	3×3	128
conv-2-2	3×3	64	ups-2	2×2	128
pool-2	2×2	64	deconv-3-1	4×4	128
conv-3-1	3×3	128	deconv-3-2	3×3	128
conv-3-2	4×4	128	ups-3	2×2	128
pool-3	2×2	128	deconv-4-1	3×3	64
conv-4-1	3×3	256	deconv-4-2	3×3	32
conv-4-2	3×3	256	ups-4	2×2	32
pool-4	2×2	256	deconv-5-1	3×3	16
conv-5	3×3	512	output	3×3	1

Figura 4. Estructura de la xarxa de Yuan Y.

Berseth M., en segon lloc amb un *Jaccard index* de 0.762 va utilitzar una xarxa U-net de profunditat tres[13].

Al 2018, el guanyador Qian C. Et al, van utilitzar una xarxa pròpia inspirada en DeepLab i PSPNet amb un *Jaccard index* de 0.802[14]. Extreuen les característiques amb una ResNet101 i les passa a diferents blocs de *pooling* i dilatació.

El segon lloc, Du H. i Seok J. Y. Et al, amb un *Jaccard index* de 0.799, format per un model DeepLab que utilitza característiques ja entrenades per PASCAL VOC 2012[15].

És difícil veure en detall els resultats d'aquests treballs ja que en documents públics presentats no expliquen qualificativament els resultats obtinguts, es basen més en una aproximació numèrica.

4 Dataset

En qualsevol àmbit relacionat amb l'anàlisi d'imatges el primer pas indispensable és trobar o crear una col·lecció d'imatges que reuneixi uns determinats criteris. Els criteris més importants són la quantitat i la qualitat de les imatges. Per quantitat s'entén com el número d'imatges que conté la col·lecció i la qualitat es refereix a com variades són les imatges, que tinguin les màscares apropiades i la qualitat de la mateixa.

Una de les bases de dades més grans relacionades amb lesions de pell és l'ISIC Archive. Aquest arxiu és una col·lecció de múltiples bases de dades - Taula 1 - que actualment està formada per 23906 imatges(Novembre 2019)[16][21], les imatges han sigut capturades des de diferents dispositius clínics de diferents centres i asseguren una bona qualitat a la vegada que respecten la privacitat del pacient.

Nom DB	JID Editorial Images	Dermos-copedia	HAM10000	MSK-1	MSK-2	MSK-3	MSK-4	MSK-5	SONIC	UDA-1	UDA-2
Nº imatges	100	5	10015	1100	1535	225	947	111	9251	557	60

Taula 1. Taula de la distribució de les imatges de cada base de dades.

Tot i aquesta quantitat d'imatges no tot son avantatges, les fotografies presenten tot tipus d'objectes – Figura 5 - que poden dificultar la tasca:

- Marques de retolador. (a)
- Vora negra a causa de la projecció de la lent. (b)
- Gel. (c)
- Adhesius de colors.(d)
- Regle per mesurar, ja sigui virtual o físic.(e)
- Pels.(f)

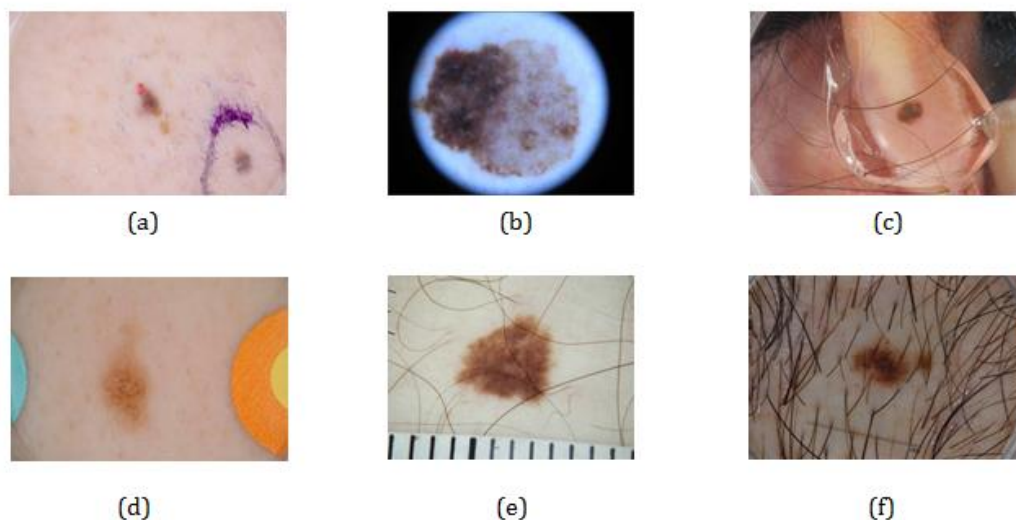


Figura 5. Exemples del diferents objectes a les imatges

A la vegada no totes les imatges de l'arxiu tenen un diagnòstic, el que fa una tasca molt difícil el fet d'haver de seleccionar les imatges una per una depenent si tenen una segmentació disponible.

L'ISIC Challenge va començar el 2016, aquell any l'organització partia el challenge en tres parts: segmentació, extracció de característiques i classificació de la lesió. Per la segmentació es presentaven 900 imatges dermatològiques, amb les màscares corresponents, per l'entrenament i 379 pel test[17][21].

El 2017 el format era el mateix que el de l'any anterior: segmentació; extracció de característiques; i classificació[18][21].

L'any següent, el 2018, el format continua igual als dos últims anys amb les tres mateixes fases. Les dades disponibles només augmenten en 594, pel que fa a la segmentació des de l'any passat[19][21].

El 2019 el format del challenge canvia i passa de tenir tres fases a tenir-ne una, la última dels anys anteriors. Tracta de classificar les 25331 imatges, casi tot l'arxiu complet, en nou categories[20][21].

Els primers experiments es van començar a fer amb la base de dades del 2017 i al trobar la del 2018 no hi havia un increment tant significatiu com per canviar-la completament. Per això el dataset triat és la recopilació d'imatges disponibles per l'ISIC Challenge 2017[18][21]. I ara bé no es descarta fer un dels últims experiments amb les dades del 2018.

El dataset consta de 2750 imatges on totes contenen metadata adjunta on s'especifica informació com: l'edat de la persona; la localització de la lesió; el

diagnòstic; la segmentació(Figura 6); etc. Així, en el nostre cas podrem fer ús de la combinació de la imatge i la seva segmentació per entrenar la xarxa.

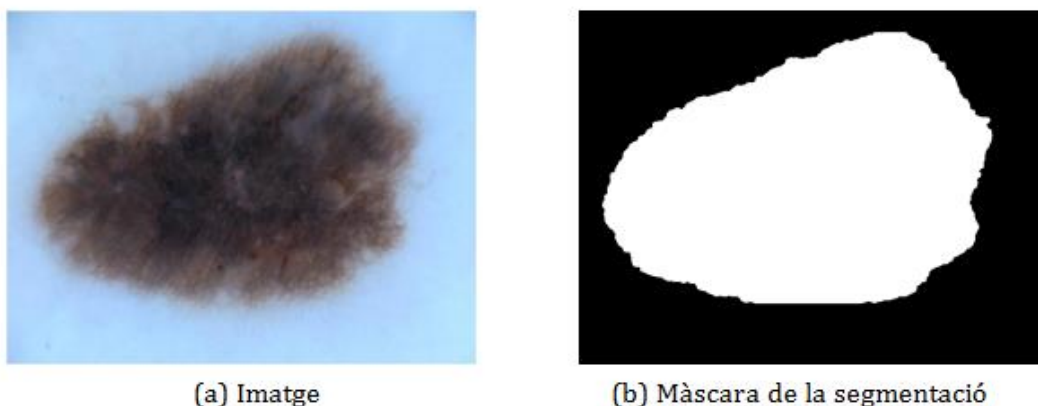


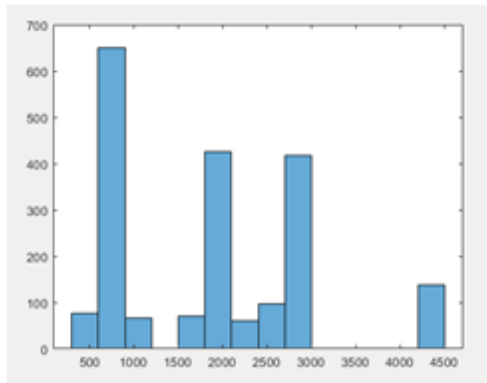
Figura 6. Exemple d'una imatge amb la seva corresponent màscara

Més endavant explicarem els experiments duts a terme durant aquest projecte a la vegada de com hem utilitzat les dades ja que tot i aquesta és la distribució que se'ns presenta – Taula 2 - hem modificat aquesta per analitzar la nostra xarxa més detingudament.

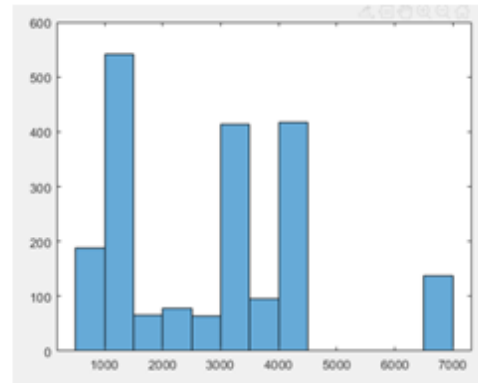
	Training	Validation	Test	Total
Nº imatges	2000	150	600	2750
%	73%	5%	22%	100%

Taula 2. La distribució d'imatges presentada pel Challenge

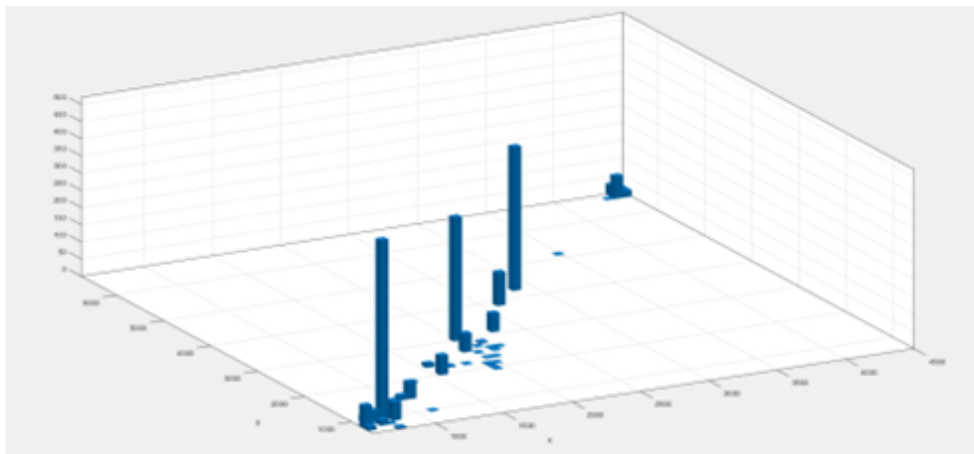
És important saber com estem modificant les imatges del nostre dataset, ja que per raons que explicarem més endavant, modificarem la seva mida a 64x64x3 píxels. L'eix de les X oscil·la entre 500p i 3000p – Figura 7(a) - tot i que hi ha algunes mostres que poden arribar als 4500 píxels, l'eix de les Y – Figura 7(b) - va aproximadament de 600p fins 4500p amb puntes de 7000 píxels.



(a) Mida en l'eix de les x



(b) Mida en l'eix de les y



(c) Mida en l'eix de les x i les y

Figura 7. Histogrames de la mida de les imatges

Com es pot veure en – Figura 7(c) - les imatges tenen aproximadament el mateix *ratio* X-Y, és a dir, que quan fem la transformació no respectarem aquest *ratio* i estarem fent les imatges de x7 a x45 vegades més petites en l'eix de les X i x9 a x70 en l'eix de les Y (sense tenir en compte el grup més extrem de les imatges més grans on aquesta reducció pot arribar fins a x100).

5 Deep Learning

5.1 Història

“A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY” és el primer document conegut relacionat amb les xarxes neuronals[22]. Datat del 1943 i escrit per Warren S. McCulloch i Walter H. Pitts, els converteix en els pares del que es coneix com a perceptró, element fonamental d'una xarxa neuronal. L'escrit fa un estudi sobre el comportament físic de les neurones i, a partir d'aquest, poder quantificar i simular-ho en un circuit elèctric.

Al 1959 Bernard Widrow i Marcian Hoff van desenvolupar dos models el ADALINE i el MADALINE, sense entrar en cap tipus de detall, aquests dos models van tenir un paper clau en el futur de les xarxes neuronals. MADALINE va ser la primera xarxa amb una aplicació real, per un problema real[23].

Els avanços d'aquesta tecnologia no van parar des de llavors i avui en dia aquestes xarxes són presents a una infinitats d'àmbits quotidians.

5.2 Neurona

També anomenat perceptró, és l'element que confecciona una capa en la nostra xarxa, cada capa serà un conjunt de n neurones. Per entendre una neurona en una ANN(Artificial neural network) és important saber la composició i el funcionament d'una neurona del cos humà en tant que el que pretenen aquestes són emular-les.

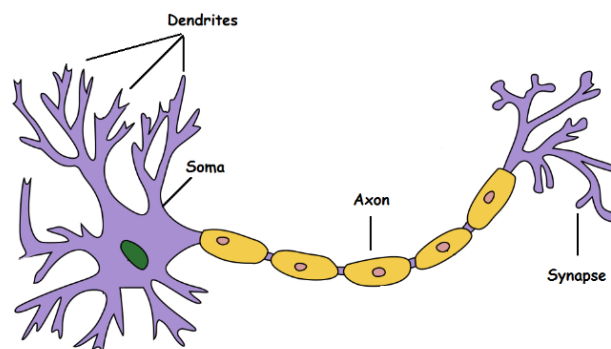


Figura 8. Exemple visual d'una neurona

Les neurones del cervell, veure Figura 8, es poden desglossar en les següents parts:

- **Dendrites:** Són els receptors de les senyals enviades per les altres neurones.
- **Soma:** Processa la informació rebuda pels dendrites i la lliura a l'axó.
- **Axó:** És per on circula l'impuls nerviós generat pel soma fins la sinapsi, per ser transmesa.
- **Sinapsi:** Transmet l'impuls nerviós a la següent neurona. És el responsable de connectar les neurones.

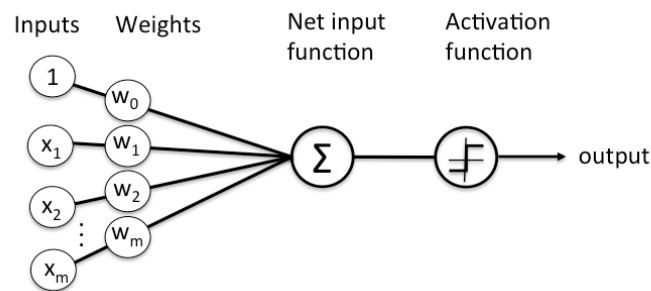


Figura 9. Esquema d'una neurona d'una ANN

Una neurona ANN, com es pot veure a la Figura 9, està formada per:

- **Inputs:** És el paràmetre d'entrada que l'usuari dona al sistema neuronal per que faci la predicció o els resultats de les neurones de la capa anterior.
- **Pesos:** És la importància de la connexió entre neurones en forma de valor.
- **Funció:** Computa els valors d'entrada per aconseguir un de sol. Acostuma a ser una suma i compte amb l'addició d'un *bias*.
- **Funció d'activació:** Aplica una transformació al resultat de la funció per tal de determinar si la neurona serà activada o no.
- **Output:** Valor que rebran les neurones a continuació d'aquestes directament connectades.

Si anomenem els inputs x , els pesos w i el *bias* b on x i w seran vectors de mateixes dimensions. L'output de la neurona serà la classificació, utilitzant la funció d'activació escaiguda, del producte escalar de x i w sumant-hi b .

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \quad (1)$$

5.3 Estructura

Per definir una ANN una de les propietats que ha de complir és de com a mínim tenir tres capes de neurones: una d'entrada; un capa intermèdia; i una de sortida. Com a norma sempre hi haurà una d'entrada i una de sortida, en canvi, pot haver-hi tantes d'intermèdies com sigui necessari o es desitgi, altrament anomenades capes ocultes (*Hidden layers* en anglès).

Aquestes capes ocultes són les que donen complexitat a la xarxa, depenent de quantes n'hi hagin serà capaç d'aprendre més característiques, veure Figura 10.

Com el resultat de cada capa es calcula a partir d'una capa anterior (suma ponderada), és a dir, d'unes característiques, es pot dir que a mesura que avances de capes les característiques que reconeixerà seran més complexes. D'aquí la importància de la profunditat.

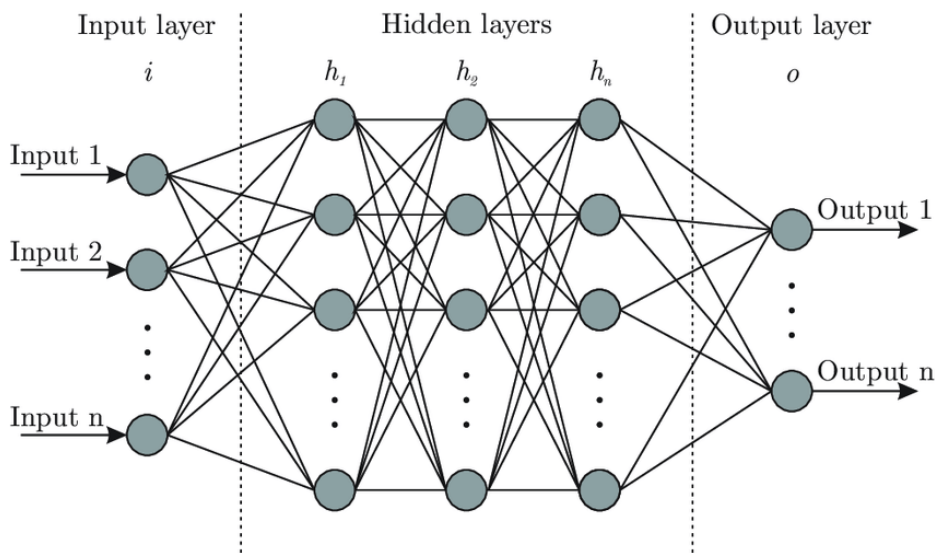


Figura 10. Esquema d'una estructura de ANN

5.3.1 Funcions d'activació

Les funcions d'activació determinen si una neurona s'activarà, no s'activarà o amb la intensitat que ho farà, depenent del tipus de funció. En definitiva juguen un paper clau al resultat i rendiment de la xarxa.

5.3.1.1 Binary

Fent la funció de *switch*, binari, 0 o 1, activarà la neurona o no depenent del *threshold* que s'escaigui. Activar la neurona s'entén com enviar la senyal que ha calculat la funció de la neurona a les següents sense modificar el valor.

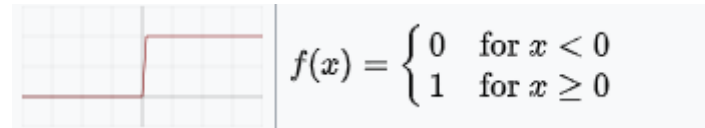


Figura 11. Gràfic d'una funció binària a l'esquerra. Fórmula d'una funció binària a la dreta (2)

Com només té dos resultats limita molt les classes d'output pel que la converteix en una funció simple incapaç de classificar en múltiples categories.

5.3.1.2 Linear Activation

A diferència de l'anterior la funció linear té múltiples valors de sortida, aquest valor és el resultat de la multiplicació del valor de la suma de pesos i bias amb el número de la funció. Pel que converteix la senyal de sortida en un valor proporcional al d'entrada.



Figura 12. Gràfic d'una funció d'activació linear a l'esquerra. Fórmula d'una funció d'activació linear a la dreta (3)

Les funcions linears estan subjectes a dos grans problemes. Utilitzant aquesta funció la derivada és un valor constant ($f'(x)=1$) pel que farà impossible utilitzar el mètode de *backpropagation* per entrenar el model. També implica que el resultat de la suma de totes les capes sigui una funció linear, ja que el resultat de la combinació linear continua sent linear. No deixa de ser un model de regressió linear.

5.3.1.3 Sigmoid

La funció *sigmoid* o també coneguda com a *logistic function*, entra a la categoria de funcions no linears, discrimina valors propers a zero mentre interpreta com a iguals valors extrems.

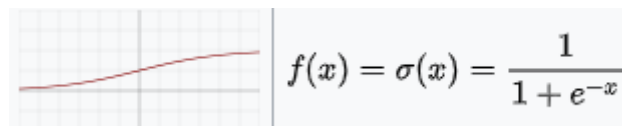


Figura 13. Gràfic d'una funció sigmoide a l'esquerra. Fórmula d'una funció sigmoide a la dreta (4)

Un dels punts claus és que normalitzarà el valor de sortida de cada neurona entre 0 i 1. Aquesta varietat en el gradient de la funció farà que a la que el valor d'entrada sigui una mica gran (en valor absolut) ja vulgui classificar aquell valor com a 0 o 1. De resultes que les prediccions poden ser binàries si no hi ha termes intermedis donant-li així, una propietat de fil de doble fulla, on per una banda tenim la claredat dels resultats però pot excedir en la homogeneïtzació dels mateixos. Mencionar també que el cost de computació és elevat.

5.3.1.4 TanH

Diminutiu de tangent hiperbòlica comparteix la majoria de les propietats de la funció sigmoide. La gran diferència es troba en que aquesta està centrada al zero, és a dir els valors van de 1 a -1 mentre que a la sigmoide van de 1 a 0.

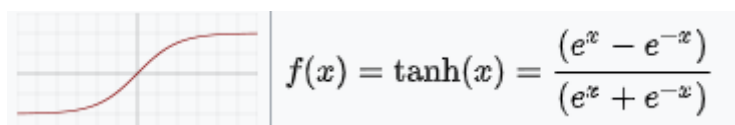


Figura 14. Gràfic d'una funció TanH a l'esquerra. Fórmula d'una funció TanH a la dreta (5)

Els valors de sortida tindran una distribució semblant a la funció sigmoide, on valors grans (valor absolut) seran igualment classificats com a 1 o -1 i els valors centrals (al voltant de 0) tindran senyals de sortida bastant diferents.

Com la sigmoide pot pecar de poca classificació pels valors elevats i a nivell computacional és cara.

5.3.1.5 ReLU

ReLU és una altra funció d'activació, serveix per referir-se a Rectified linear unit, unitat linear rectificada en català. Si bé el comú denominador en les funcions no linears ha sigut l'alt cost de computació la *ReLU* és una excepció en tant que la tendència d'una xarxa amb funcions *ReLU* és a convergir molt ràpid.

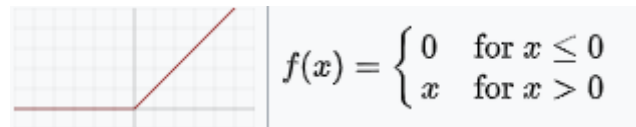


Figura 15. Gràfic d'una funció ReLU a l'esquerra. Fórmula d'una funció ReLU a la dreta (6)

Entenent la importància de l'algorisme de *backpropagation* alhora de la influència de les derivades al càlcul d'aquest, és molt rellevant entendre la derivada d'una funció *ReLU*. La derivada d'una *ReLU* és una funció amb sortida binària.

El problema de la *Dying ReLU* és quan el valor d'entrada s'aproxima a zero o és negatiu. El valor de la derivada al ser zero no permet l'execució de l'algorisme de *backpropagation*, dit d'una altra manera, la xarxa no pot aprendre.

5.3.1.6 Leaky ReLU

L'objectiu de la Leaky ReLU és solucionar el problema de la Dying ReLU.

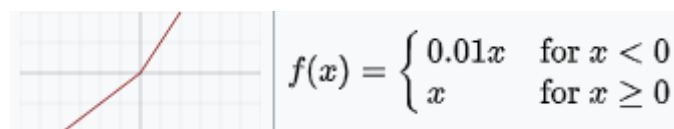


Figura 16. Gràfic d'una funció Leaky ReLU a l'esquerra. Fórmula d'una funció Leaky ReLU a la dreta (7)

Al tenir un lleuger pendent a l'eix negatiu de les x, evita que la derivada de la funció per aquells valors sigui zero. Aquesta lleugera modificació permet que l'algorisme de *backpropagation* es pugui aplicar però continua sense poder aprendre valors negatius de forma consistent.

5.3.1.7 Parametric ReLU

Les *ReLU* paramètriques volen introduir els valors negatius com a paràmetre d'aprenentatge. Sent α , de la figura X, el factor a optimitzar en l'aprenentatge.

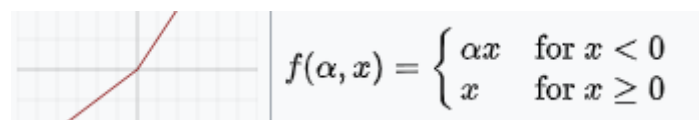


Figura 17. Gràfic d'una funció paramètrica ReLU a l'esquerra. Fórmula d'una funció paramètrica ReLU a la dreta (8)

Tot i permetre l'aprenentatge d'aquests valors els resultats poden presentar una alta variància en els resultats depenent del problema que estiguin resolvent.

5.4 Aprenentatge

5.4.1 Cost function

A l'hora d'aprendre el primer pas és entendre quin error hem comés, per així, rectificar-lo. La funció de cost és el responsable de quantificar aquest error per tal de classificar com de correctes són les prediccions del model.

Una manera de donar un valor a l'error de les prediccions és el *mean squared error*(MSE).

El MSE calcula la mitjana dels errors al quadrat. Sempre serà un número positiu i penalitza els erros com més grans siguin, d'aquí que faci el quadrat de l'error.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (9)$$

On Y és el valor real, \hat{Y} la predicció de la xarxa i n el nombre de prediccions.

5.4.2 Gradient descent

L'algorisme de descens de gradient té com a objectiu trobar el mínim d'una funció donada. En el cas de les xarxes neuronals, vol trobar el valor de la funció de cost on el model prediu amb menys error per, d'aquesta manera, poder considerar-lo com el més òptim.

És un algorisme recursiu, on a cada iteració el que fa és: calcular el pendent de la funció en el punt actual, derivant; calcular la distància del pas que ha de fer en la direcció on el pendent decreix; fer el pas; mirar si el valor absolut del pas és més petit que el *threshold*, si és així acaba, sinó continua iterant.

El learning rate és la variable que controla com de gran és el pas, aquest valor té molta transcendència en com de ràpid entrenarà o si trobarà el punt mínim.

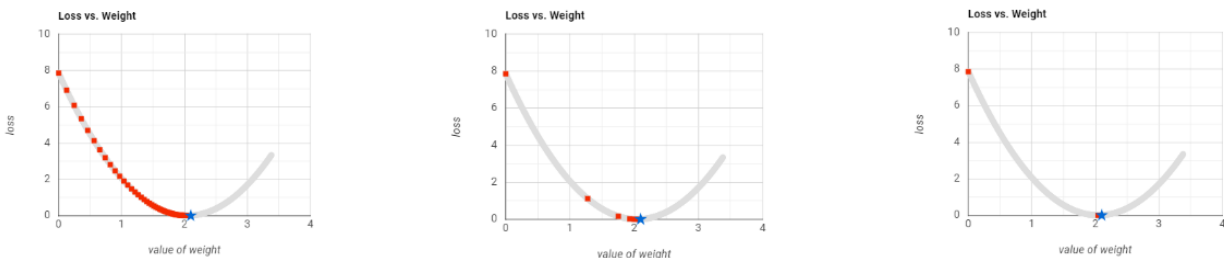


Figura 18. Exemples de funcions de pèrdua amb diferents tipus de valors d'aprenentatge

Com es veu a la imatge (a) de la Figura 18 el valor és molt petit i necessita moltes iteracions per arribar al valor mínim, en canvi en la imatge (c) de la Figura 18 arriba

en un sol pas. Al cap i a la fi és un intercanvi d'eficiència i risc, si es vol utilitzar *learning rates* grans es pot arribar a punts més allunyats de la funció però la possibilitat de passar-se del mínim és molt alta. Si per altra banda, s'utilitza *learning rates* petits és molt possible que trobi el mínim però trigarà molt en trobar-lo, i si hi ha limitacions d'iteracions o de temps, no arribarà mai.

5.4.3 Backpropagation

Si entenem el resultat d'una xarxa neuronal com el càlcul d'unes dades d'entrada amb uns pesos w i uns *bias* b podem arribar a la conclusió que si modifiquem aquests valors canviarem el valor de sortida de la xarxa[24].

L'objectiu de *backpropagation* és calcular les derivades parcials $\partial C/\partial w$ i $\partial C/\partial b$ de la funció de cost C respecte w i b , en altres paraules, el gradient descendent de l'error per canviar els pesos i els *bias*.

No s'entrarà al nivell matemàtic dels càlculs i les funcions que s'utilitzen per propagar l'error. El concepte que sí que és fonamental d'entendre és que l'error d'una neurona donada propagarà l'error a la capa anterior cap aquelles neurones que li hagin proporcionat la major informació. Assumint que aquelles neurones que tenen més influència al càlcul d'aquella seran les causants de l'error.

L'algorisme funciona de la següent manera:

- Es dona un input.
- Es calcula la capa de sortida a partir de la d'entrada, és a dir, es fa *forward propagation*.
- Es calcula l'error entre la predicció i el resultat esperat.
- Es propaga l'error de la capa de sortida a la d'entrada.
- Es calcula el gradient i es canvien els pesos i els *bias*.

5.5 Convolutional Neural Networks (CNN)

Fins aquest moment a la hora de parlar de xarxes neuronals ho hem fet des del punt de vista del nivell més senzill d'aquestes, les *Feedforward neural network*(FNN). Si s'ha entès com funcionen, es pot observar un problema si, per exemple, volem que el nostre input sigui una imatge- Figura 19 -.

L'entrada d'una FNN és d'una dimensió, les imatges per contra, es representen com una matriu a nivell computacional, és a dir dues dimensions com a mínim. Depenent de quin tipus d'espai de color tingui pot tenir múltiples dimensions, RGB acostuma a ser la més comú i aquest té tres dimensions.

La diferència de dimensions de l'entrada presenta instantàniament tres grans qüestions: les xarxes neuronals no poden analitzar imatges; transformar una imatge de n dimensions a una de una per passar-la com a paràmetre; i modificar el model plantejat per les FNN.

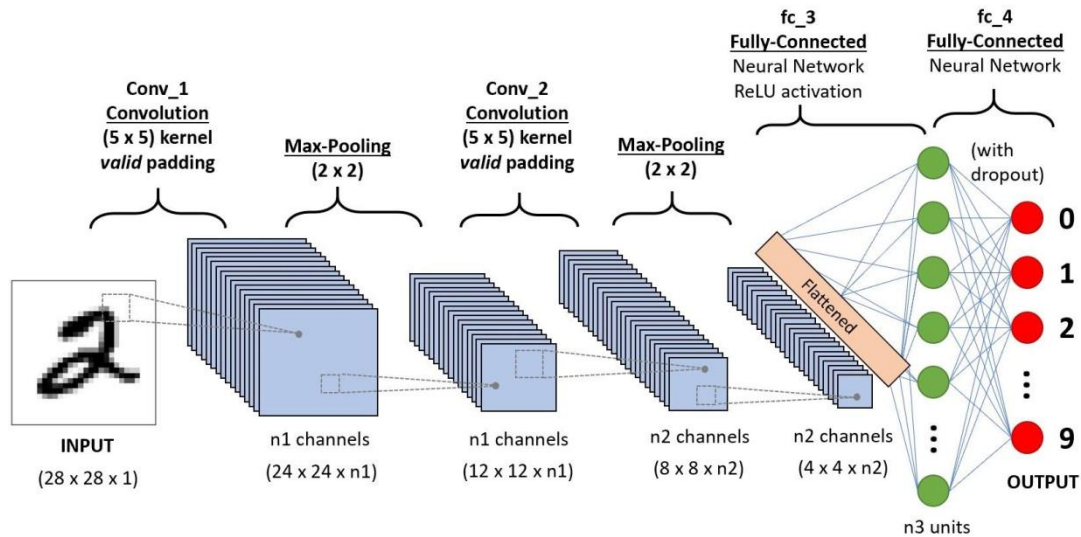


Figura 19. Exemple de CNN

La primera afirmació és completament falsa, les xarxes neuronals no només poden fer-ho sinó que produeixen resultats excepcionals en el camp de l'anàlisi d'imatges.

És llavors reduir la dimensió de la imatge d'entrada una solució? No. Tot i que es pot fer, especialment de dues (imatges en escala de gris per exemple) a una dimensió, el fet de reduir les dimensions trenca moltes relacions de dependència espacial entre píxels.

Amb les dues primeres hipòtesis descartades, és crear un nou model de xarxa neuronal la solució? Sí. S'anomenen *Convolutional neural networks* (CNN) i permeten capturar les dependències espacials i temporals d'una imatge.

Les CNN estan formades per una entrada i una sortida, com qualsevol xarxa neuronal i dos gran blocs, el primer que extreu les característiques de la imatge i el segon que classifica les característiques.

Per extreure característiques la xarxa contindrà diferents combinacions de *pooling layers*, *batch normalization layers*, les capes de funció d'activació i capes de convolució. Aquestes combinacions són les que determinen les diferents tipus de CNN del mercat actualment.

La classificació generalment es duu a terme amb una *Fully Connected layer* i una *softMax layer*.

5.5.1 Convolutional layer

L'operació de convolució és una operació matemàtica que s'aplica en el cap del processament d'imatges, és l'acció de multiplicar dues matrius de diferent mida per generar un de tercer de la mateixa dimensió[26].

En el processament d'imatges és té com a paràmetres una imatge i un filtre, com a resultat serà una matriu n_{out} que dependrà de la mida del filtre k , del *padding* que s'apliqui p , del *stride* s i de la mida de la imatge n_{in} .

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1 \tag{10}$$

En la convolució el filtre s'itera en els dos eixos de la imatge en passos de la dimensió del *stride*.

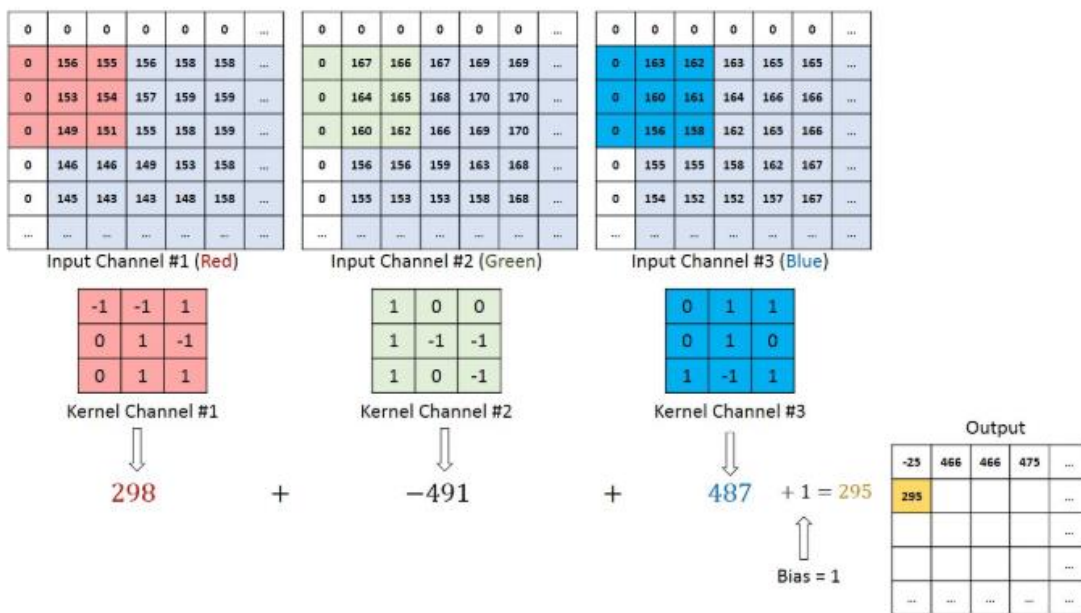


Figura 20. Operació de convolució a una imatge $M \times N \times 3$ i un kernel de $3 \times 3 \times 3$

El filtre conegut normalment com a *kernel*, és una matriu de dimensions imparells, acostuma a ser de 3, 5 o 7 i és el responsable de guardar la informació de la característica. Cada *kernel* reconeixerà una característica en particular, com la direcció del gradient, el color, les formes, etc.

Com es pot veure a la Figura 20 quan la imatge té múltiples canals de colors cada *kernel* s'iterarà al canal que li pertorqui.

El *padding* marca com tractarà els contorns de la matriu la convolució.

Valid padding Vs *Same padding*:

- **Valid:** El filtre només s'iterarà a filtres on tot el *kernel* estigui dins de la imatge pel que la matriu de sortida es reduirà.
- **Same:** La mida de la matriu de sortida serà de la mateixa mida que la d'entrada si *stride* es 1.

La *atrous convolution* és una variació d'una convolució normal basada en espaiar els valors del *kernel*. El paràmetre que determinarà la distància entre els elements que el filtre computarà serà el factor de dilatació. Veure Figura 21[26].

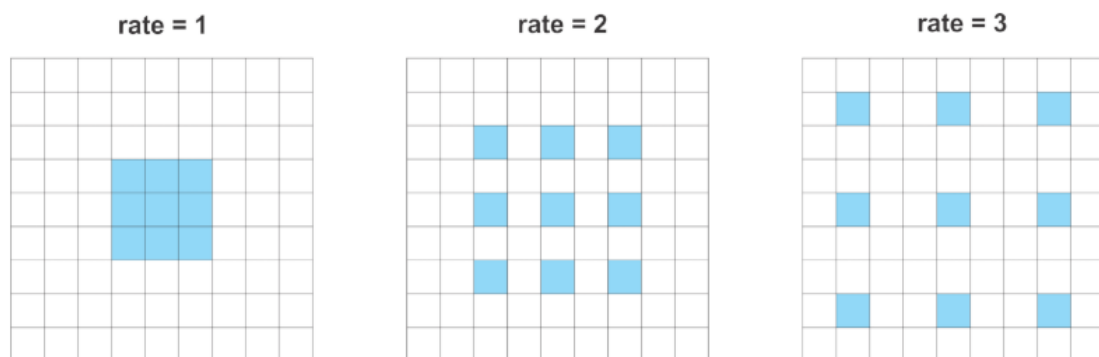


Figura 21. Exemples d'Atrous convolutions amb rates: 1, 2 i 3. D'esquerra a dreta

5.5.2 Pooling

Pooling és l'algorisme que donat unes dimensions, redueix la matriu donada per principalment, reduir el cost computacional de la xarxa i donar importància a les característiques dominants[26].

Si diem que apliquem un *pooling* de 2x2, com a la figura X, reduïrem la matriu que li apliquem a la meitat. Hi ha dues maneres de fer el càlcul del valor resultant:

- *Max Pooling*: Agafa com a valor resultat del filtre, el valor màxim.
- *Average Pooling*: Fa una mitjana de tots els valors que engloba el filtre.

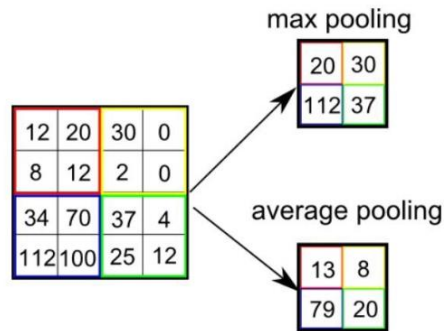


Figura 22. Exemple del resultat d'aplicar max pooling(superior) i average pooling(inferior)

5.5.3 Batch normalization

Normalitza els valors d'entrada d'una capa, per reduir la covariància. Ho fa extraient la mitjana de l'últim *batch* i el divideix per la desviació estàndard[25].

El *batch normalization* és clau per accelerar el *learning rate* de la xarxa, ens permet utilitzar un valor més elevat ja que controla que els valors d'activació no siguin molt grans(valor absolut).

També limita el sobre entrenament perquè afegeix soroll a cada capa oculta, similar a les *drop-out layers*.

5.5.4 Fully connected layer

Al fer-se normalment al final de l'estructura d'una xarxa, la matriu de característiques serà d'una dimensió, el que fa que aplicar unes connexions de neurona-neurona amb *ratio* 1:1 no sigui tant costós. A sobre permet aprendre noves funcions no linears de característiques d'alt nivell en aquell estat de la xarxa.

5.5.5 Dropout layer

Les capes de drop-out volen prevenir l'efecte del sobre entrenament, es basa en deixar fora momentàniament unitats de la xarxa. Aquesta acció de ser descartat per aquella iteració es basa en probabilitats, per a casi la majoria de dissenys de xarxes 0.5 es considera un valor òptim[27].

Un dels efectes secundaris de deixar neurones d'una capa fora és que les altres tindran més responsabilitat i generaran subxarxes que al cap i a la fi afegiran robustesa al model al final de l'entrenament.

5.5.6 SoftMax layer

S'utilitza com a capa final de la xarxa per classificar. Calcula la probabilitat que el resultat pertanyi a una classe en un problema amb múltiples classes.

Com a paràmetres li entra un vector de nombres reals i els normalitza en una distribució de probabilitats proporcional als exponents dels valors d'entrada.

5.6 Transfer Learning

De la mateixa manera que els humans utilitzen experiències o coneixements passats per aprendre'n de nous, les xarxes neuronals es poden dissenyar a partir d'aquest mateix concepte.

Hi ha diferents maneres de realitzar el *transfer learning*, en aquest apartat es parlarà sobre el *transfer learning per deep learning* i més exactament en l'àmbit de la segmentació d'imatges.

L'objectiu principal d'aquesta manera de dissenyar un model és suplementar informació(característiques i pesos) de les característiques de baix nivell a una xarxa que realitza una tasca específica.

Aquesta informació s'integra introduint al model de la xarxa que realitza aquesta tasca específica la xarxa ja entrenada com a entrada de la mateixa, és a dir, es substitueixen les primeres capes de la xarxa ja dissenyada per una de pre-entrenada.

La xarxa ja entrenada sol haver estat entrenada prèviament amb una base de dades significativament gran, especialment si es compara amb la tasca en la que la volem integrar. Complementant i aportant més informació, d'aquesta manera ajuda a sobreposar-se al fet de que la base de dades que s'estigui utilitzant sigui limitada, entre altres propietats que aporta aquesta pràctica. És important saber quan i quan no és una bona pràctica, no sempre és beneficiosa en termes de resultats.

6 Metodologia

6.1 Hardware

L'equip utilitzat està antiquat per la tasca que volem realitzar, tot i que evidentment és possible duu a terme els experiments que presentarem, no és ni molt menys recomanable. Limita altament els nostres paràmetres d'entrada i, últimament, les expectacions i conclusions finals.

- Processador: AMD FX-8350
- Gràfica: NVIDIA GeForce GTX 960 4GB
- Memòria RAM: 16GB DDR4
- Hard Drive: WDC WD10EZEX-00ZF5A0 (1TB, SATA 3)

6.2 Software

A nivell de software no presenta tantes limitacions com el hardware utilitzat, si bé el sistema operatiu emprat és antic ni molt menys serà un factor determinant a l'obtenció de resultats. Tots els altres productes són els últims accessibles al públic en el moment d'aquest projecte.

També és important remarcar que els controladors de la gràfica han variat durant l'execució del projecte i com a consens s'ha escollit triar l'última versió que s'ha utilitzat just abans d'entregar el projecte.

- OS: Windows 7 64-bits
- IDE: MATLAB R2019b(Acadèmic use)
- GPU Drivers: 441.41
- NVIDIA CUDA 10.2

Add-ons:

- Deep Learning Toolbox Model for ResNet-18 Network – 19.2.0
- Parallel Computing Toolbox – 2.1
- Image Processing Toolbox – 11.0
- Computer Vision Toolbox – 9.1
- Deep Learning Toolbox - 13.0
- Statistics and Machine Learning Toolbox - 11.6
- Simulink – 10.0
- Database Toolbox – 9.2

6.3 Data Augmentation en MATLAB

A diferència de tècniques anteriors per augmentar les dades d'un dataset en compte de crear manualment dades prèvies a l'entrenament i passar-les com a paràmetre, la funció *augmentedImageDatastore* de MATLAB ens deixa crear un objecte que es passa per paràmetre a l'entrenament[28].

L'objecte és un conjunt de transformacions geomètriques com: translació; rotació; escalat; reflexió; i *shear*. Com es pot veure a la Figura 23, durant l'execució a cada iteració es generarà una combinació aleatòria de les possibles transformacions i s'aplica als les imatges d'aquell *minibatch*.

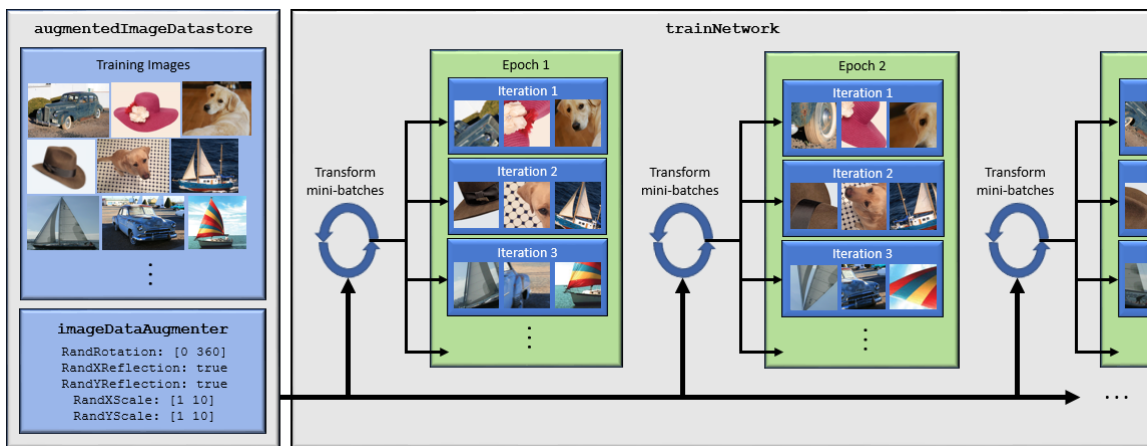


Figura 23. Exemple visual de com MATLAB implementa data augmentation

Aquest procés és molt beneficiós perquè augmenta el dataset relativament al nombre d'èpoques de l'entrenament, és a dir, no genera imatges innecessàriament i no omplen espai a memòria.

6.4 Networks

Per entendre el perquè de la utilització d'aquestes tres xarxes, és important entendre el procés d'aprenentatge durant el treball. La primera xarxa forma part del primer tast personal relacional amb les xarxes neuronal, sent així la primera xarxa que vaig dissenyar.

A partir d'aquest punt em vaig anar endinsant i aprofundint en el tema mentre aprenia els paràmetres de les xarxes amb els que es podien jugar, trobant d'aquesta manera dissenys de xarxes per a la segmentació d'imatges més avançades pel que fa a l'estat de l'art.

La xarxa U-net va ser la que pel moment va semblar tenir millor resultats, especialment en l'àmbit que estava treballant. Com vaig fer amb la primera xarxa, mentre feia el *fine tuning* i aprenia sobre la xarxa continuava fent recerca sobre l'estat de l'art fins que vaig ser introduït al *transfer learning*.

Aquest últim fet va resultar en la tercera i última xarxa que vaig treballar sent la més complexa i difícil d'entendre però alhora produïa els millors resultats fins el moment i va resultar coincidir amb l'últim estat de l'art pel que fa la segmentació de melanomes.

6.4.1 Simple CNN

La primera network utilitzada és molt senzilla. Com a paràmetres d'entrada té una imatge de dimensions 128x128x3, 32 filtres per capa de mida 3[29].

Consta de tres combinacions de capes de convolució, amb ReLU com a capa de funció d'activació i una capa de *batch normalization*. El padding és el mateix per totes, utilitzarem same padding. En canvi, el factor de dilatació per fer una convolució de tipus *atrous*: 1 a la primera; 2 a la segona; i 4 a la tercera.

L'últim bloc de la xarxa és una combinació de capes per classificar els píxels. Una capa de convolució amb un filtre de 1, u capa de tipus softMax i una de classificació.

6.4.2 U-net

L'estructura d'una U-net com el seu nom indica, té forma de u, veure Figura 24. La branca de l'esquerra conté els *encoders*, a la dreta conté els *decoders* i a la zona inferior hi ha un pont entre ambdues.

Els *encoders* són els encarregats d'extreure les característiques de la imatge, per fer-ho cada *encoder* està format per dues capes de convolució amb *padding = same*, ReLU com a funció d'activació i filtres de mida 3x3 amb el nombre de característiques que pertoqui a cada nivell, el doble que al nivell anterior. Aquestes fan créixer la imatge en el tercer eix mentre la redueixen en els dos eixos de les dimensions.

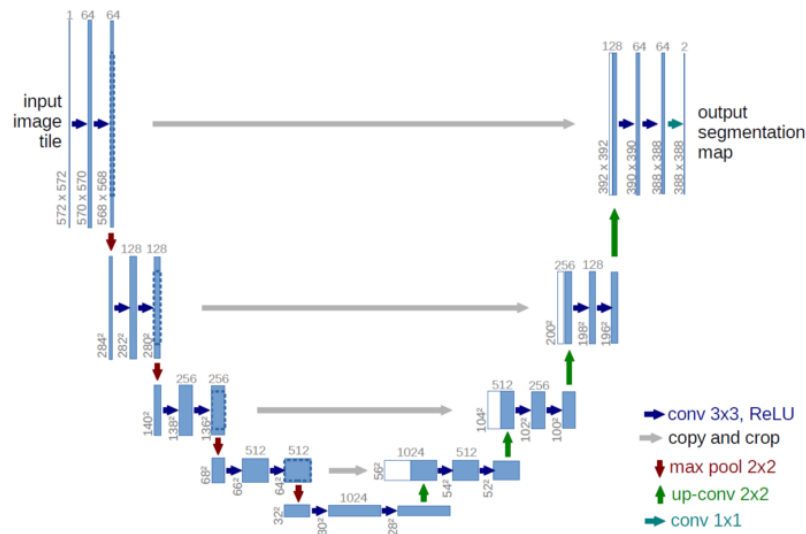


Figura 24. Exemple d'estructura U-net

L'altre part que forma els *encoders* són les capes de *max pooling* com expliquem en la secció anterior, aquesta funció serveix per reduir la imatge així el filtre en el següent set de convolucions abraçarà el doble d'informació(en el cas d'un *max pooling 2x2*) dels píxels més dominants de la imatge. Com a contrapartida tot i augmentar el nivell d'extracció de les característiques al mateix temps es perd informació espacial. Per això és necessiten els *decoders*.

El pont té dues convolucions formant dues *fully connecter layer*(FCL) per trobar més característiques.

Els *decoders* tenen la funció de reconstruir les dimensions de la imatge d'entrada per recuperar les dependències espacials. Com als *encoders* els *decoders* tenen dos capes de convolució amb filtres 3x3 amb *same padding*, tants com siguin necessaris a cada nivell amb funció d'activació de tipus ReLU.

A diferència dels *encoders* amb les capes de *max pooling*, els *decoders* tenen una funció per incrementar la mida de la imatge. No entrarem en detall a explicar aquest procés però direm que es fa utilitzant *transposed convolutions*.

Cada *decoder* i cada *encoder* de cada nivell estan connectats de forma directa, saltant-se connexions, concatenen la sortida de l'*encoder* del nivell x amb l'entrada del *decoder* d'aquell nivell.

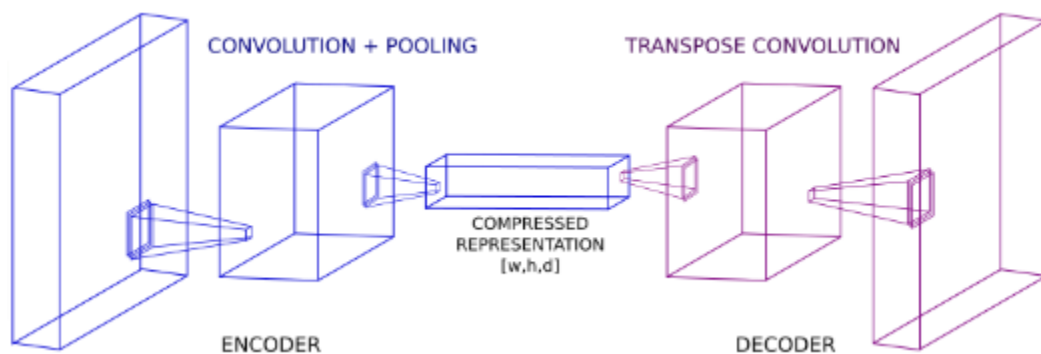


Figura 25. Exemple d'estructura encorder-decoder

En definitiva es pot dir que una U-net és una *fully convolutional network (FCN)* ja que només conté *convolutional layers* i com el seu creador Olaf Ronneberger et al. descriuen - <<In this paper, we build upon a more elegant architecture, the so-called “fully convolutional network”>>[30]. Aconsegueix uns resultats remarcables en l'àmbit de segmentació semàntica en la biomedicina.

6.4.3 DeepLab v3 amb resnet18

DeepLab utilitza una xarxa prèviament entrenada per extreure les característiques de la imatge. En el nostre cas, ResNet-18.

ResNet-18 és un model de xarxa que es centra en fer una concatenació del resultat no lineal de la convolució amb l'entrada d'abans d'haver fet aquestes convolucions. Entre cada convolució i abans de cada funció d'activació s'aplica *batch normalization*. Les funcions d'activació són de tipus ReLU.

L'entrada de la xarxa, és a dir la imatge, sempre ha de tenir com a mínim 244x244 dimensions ja que la primera capa sempre és una convolució d'un filtre 7x7 amb 64 característiques i *stride* = 2. La segona capa sempre és un 3x3 *max pool* amb *stride* = 2. En la versió de 18 capes hi ha:

- Dues capes de convolució de 64 característiques cada una.
- Dues capes de convolució de 128 característiques cada una.
- Dues capes de convolució de 256 característiques cada una.
- Dues capes de convolució de 512 característiques cada una.

El final està format per una combinació de tres capes, una *average pool*, una capa completament connectada amb la capa de funció *softmax* per la classificació.

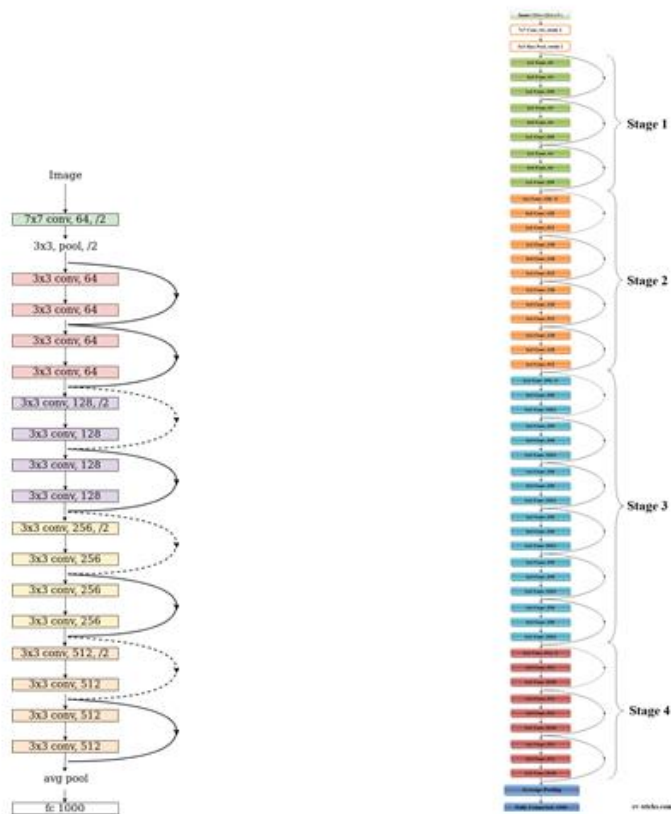


Figura 26. Estructura d'una ResNet-18 a l'esquerra i d'una ResNet-50 a la dreta

ResNet té diferents dimensions de xarxes, aquesta dimensió ve donada per la quantitat de capes que té. Van des de 18- Figura 26 imatge de l'esquerra- fins a 152. Si be l'increment de les capes redueix l'error de la xarxa, també augmenta el cost computacional.

Fet que, partint de que hi ha limitacions de hardware, ha promogut que la xarxa escollida sigui la de 18 capes. Sobretot si es té en compte la relació entre temps estimat de predicció i *accuracy*, veure Figura 27.

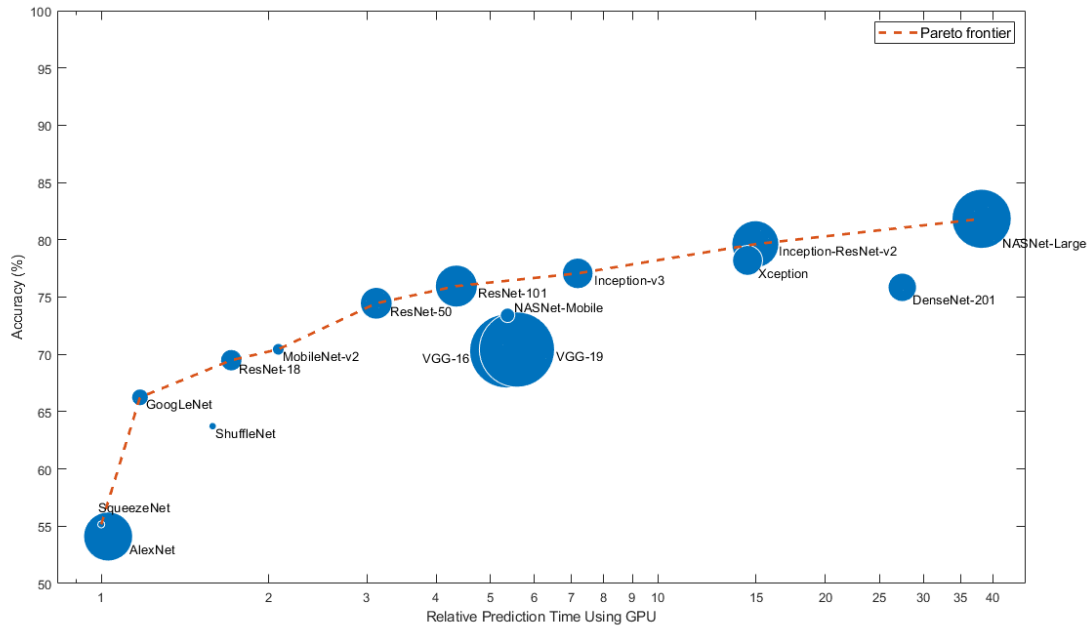


Figura 27. Gràfic que representa l'accuracy de les xarxes en relació al temps de predicció[31]

DeepLab v3+ és un model format per un *encoder* i un *decoder* per augmentar la precisió a la segmentació. A part d'aquest fet és igual que la DeepLab v3 estàndard, basada en les capes de convolucions amb funció ReLU i *batch normalization* on l'últim bloc és una *Atrous Spatial Pyramid Pooling*(ASPP)[32].

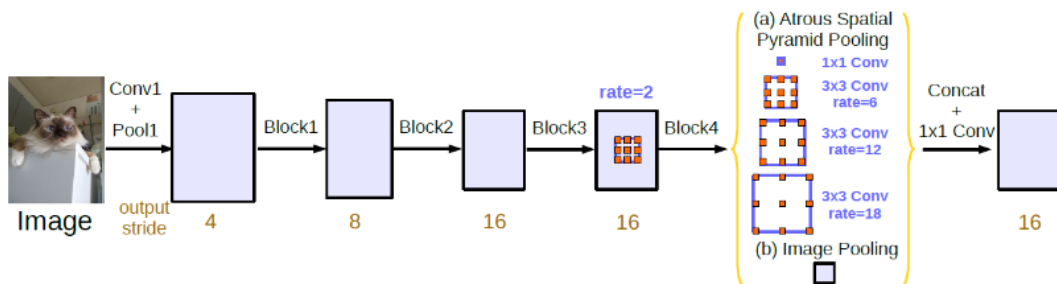


Figura 28. Atrous Spatial Pyramid Pooling(ASPP)

Les ASPP, Figura 28, són múltiples *atrous layers* executades en paral·lel a partir d'una *Multi Grid*, per exemple, *Multi Grid* = 1,2,4 com a la primer xarxa de totes. El resultat és la concatenació d'aquestes múltiples convolucions passades per un filtre 1x1[33].

A nivell abstracte, per poder entendre el model que hem creat, a l'utilitzar els dos models mencionats anteriorment. Tenim un *encoder-decoder* on l'*encoder* és la xarxa ja entrenada, *ResNet* i la ASPP que és el que pretenem entrenar, finalment acaba amb un *decoder* i un bloc de classificació.

7 Experiments

7.1 Mètrica d'avaluació de la xarxa

Com quan s'entrena a l'avaluar una xarxa els paràmetres que s'utilitzen de test han de tenir les dades reals(*ground truth*) també. A l'entrenar aquest *ground truth* serveix per calcular l'error per així modificar els pesos i *bias*, en altres paraules, aprendre.

Per avaluar la xarxa una tècnica estesa és crear una matriu on a un eix té les classes predites pel model i a l'altre les classes reals. Els valors de la Figura 29 són el número de vegades que aquella combinació ha existit. Aquesta matriu s'anomena matriu de confusió.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figura 29. Com es distribueixen les dades de sortida per analitzar-se'n el seu resultat

Si diem que tenim dos tipus de classes, com en el nostre cas, la matriu tindrà dimensions 2x2. Sempre serà quadrada, aquesta relació ve donada a causa del número de classes i de la lògica de la mateixa, si una classe existeix sempre podrà ser predita.

Donada A com a primera classe i B com a segona classe:

- **True positive(TP):** Els elements A predits com a classe A. Predicció correcta.
- **True negative(TN):** Els elements B predits com a classe B. Predicció correcta.
- **False positive(FP):** Els elements B predits com a classe A. Predicció incorrecta.
- **False negative(FN):** Els elements A predits com a classe B. Predicció incorrecta.

Amb aquests paràmetres es poden calcular índexs que descriuen millor el rendiment del model.

- *Accuracy*: La relació entre les prediccions encertades i el total de respostes. En el nostre cas respon a quants píxels el model ha predit que eren píxels i quants píxels de fons el model ha predit que seria fons. És una de les mètriques més utilitzades.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (11)$$

- *Precision*: La relació entre les prediccions correctes que el model ha donat d'una classe i les que ha predit d'aquella classe en total. Respon a la pregunta de quants d'aquells que el model ha predit com a classe A són en veritat classe A.

$$Precision = \frac{TP}{(TP + FP)} \quad (12)$$

- *Recall*: La relació entre les prediccions correctes que el model ha donat d'una classe i tot el conjunt real d'aquella classe. En altres paraules, de totes les nostres prediccions sobre la classe A, quin percentatge hem encertat.

$$Recall = \frac{TP}{(TP + FN)} \quad (13)$$

- *Specificity*: La relació entre els que el model prediu que no són d'una classe i tots aquells que no corresponen a la classe. Per exemple, els que el nostre model prediu com a fons de la imatge i precisament no ho són.

$$Specificity = \frac{TN}{(FP + TN)} \quad (14)$$

- *F-score*: És la mitjana harmònica entre la *precision* i el *recall*. També conegut com *Sørensen-Dice*, valora com de semblants són ambdós índex i com grans. Per exemple, si tens un *recall* molt elevat però una precisió molt baixa o viceversa, la *F-score* serà pràcticament zero.

$$FScore = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad (15)$$

- *Jaccard index*: Serveix per determinar com d'iguals o diferents dos conjunts de dades són.

$$Jaccard = \frac{TP}{(TP + FP + FN)} \quad (16)$$

7.2 Mètode d'avaluació de la xarxa

El *cross-validation* és una tècnica àmpliament utilitzada per avaluar el rendiment d'un model en el camp de l'aprenentatge automàtic. L'objectiu d'aquesta tècnica es centre en explicar quina resposta dona el model al ser presentat un altra base de dades diferent.

És una tècnica que es recomana utilitzar especialment si no es disposa de sets suficientment grans en tant que s'analitza el model tantes vegades com mostres diferents es generin. Ajudarà a reconèixer tendències com el *overfitting* o el biaix de selecció.

K-fold cross-validation és el terme que s'empra per referir-se a quantes vegades es farà la re-mostra. *K* indica el número de vegades i *fold* significa plec, és a dir, quants plecs es faran.

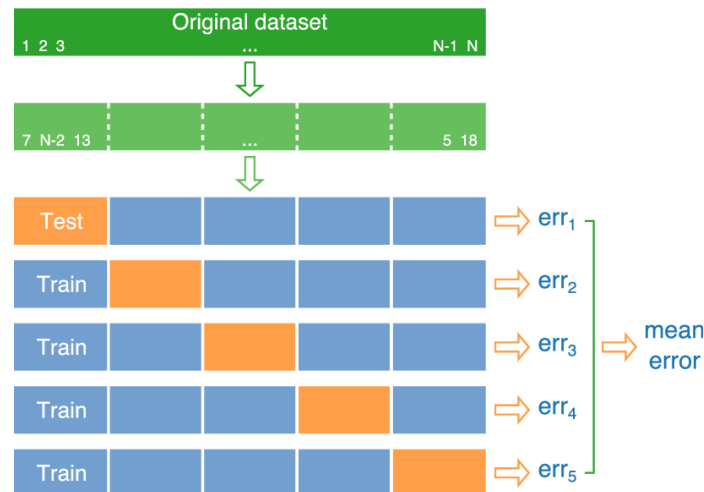


Figura 30. Esquema visual de com es distribueixen els sets i com es calcula l'error

L'algorisme funciona de la següent manera:

- Es reordenen les dades del *dataset* de forma aleatòria.
- Es divideix el *dataset* en *k* parts.
- Per cada part:

- S'escull un conjunt com el de test i la resta de parts $(k-1)$ com a *training*.
- S'entrena el model amb el grup de *training*.
- S'extreuen les mètriques d'anàlisis.
- Es fa la mitjana del conjunt de resultats de cada iteració de l'algorisme

En el nostre cas hem afegit un set de validació, per tant, a l'algorisme modifiquem la part de la iteració. Si diem que a és l'índex de la fracció escollida, 1 és l'inici, k és el final i assumim que és circular ($a+1=1$ quan $a=k$):

- ⋮
- S'escull un conjunt com el de *validation*. És a . (Figura 31 en color groc)
 - S'escull un conjunt de test. És $a + 1$. (Figura 31 en color blau)
 - S'escull un conjunt de *training*. És tot el *dataset* - $(a + (a+1))$. (Figura 31 en color taronja)
- ⋮

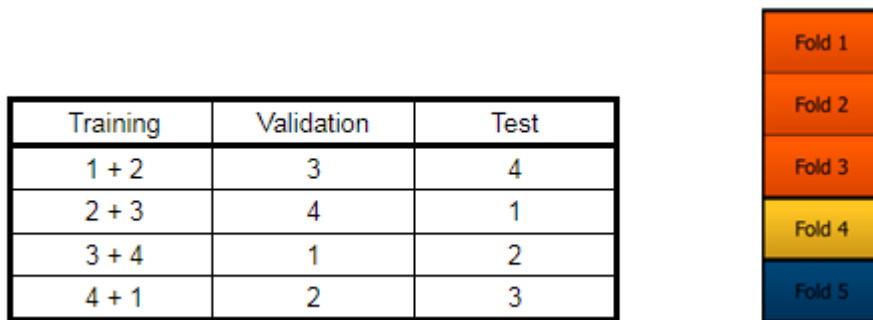


Figura 31. Taula de com es reparteix el dataset a cada iteració a l'esquerra i exemple visual a la dreta

7.3 Primers experiments

Tot i que l'objectiu dels experiments no és explicar de forma temporal quan van ser duts a terme, si que és important notar que hi ha conseqüències que van motivar canvis de tècnica o valors.

Per entrenar aquesta xarxa es van utilitzar dos tipus de k -fold cross-validation, $k = 4$ i $k=1$. Aquest canvi va ser degut al temps de computació del primer experiment. També es feien dos tests diferents.

Recordem el *dataset* disponible pel challenge: 2000 imatges de *training*; 150 per *validation*; i 600 per *testing*. Anomenarem les dades de *training* com a set X , el de *validation* com a set Y i el destinat al *testing* pel challenge, Z .

Les dades es van separar de la següent manera:

Per $k=4$:

- Es creen 4 grups de les dades del set X, en total tindrem 4 grups de 500 imatges.
- Escollim un de la partició anterior com a *validation* -> 500 imatges. 25%
- Escollim un grup de la partició pel test 1 -> 500 imatges. 25%
- Agrupem els dos grups restants que passaran a ser el nostre *training* -> $500 \times 2 = 1000$ imatges. 50%
- Escollim les dades del set Z com les del test 2 -> 600 imatges. Aquest test serà immutable, els quatre primers punts a cada *fold* canviaran però el set de test mai ho farà.

Training	Validation	Test
1 + 2	3	4
2 + 3	4	1
3 + 4	1	2
4 + 1	2	3
		Avg
		Std
1 + 2	3	Test 2
2 + 3	4	
3 + 4	1	
4 + 1	2	
		Avg
		Std

Taula 3. Taula de la distribució dels dataset per cada iteració amb una $k=4$ i dos tests.

En resum, el set serà avaluat com a quatre *k-fold* però afegint-hi un test més. Com es podrà veure als resultats això passa perquè els resultats eren significativament diferents.

Per $k=1$:

- Només es fa un *fold* i és el mateix que el de la primera iteració al *k-fold* per $k=4$.
- Continuàvem complementant el *testing* afegint un altre test per veure si canviava molt al presentar-li dades noves.

Molt important entendre que aquests canvis pràcticament van ser forçats, tenint en compte els coneixements i experiència en la matèria fer un *k-fold* de $k=1$ per entendre els canvis era superior a la pèrdua de consistència als resultats.

Aquest set d'experiments es va centrar en canvis a les opcions d'entrenament i el *data augmentation*.

Les opcions d'entrenament que es van canviar són, entre altres: Max epochs, són el nombre de vegades que una imatge passarà per la xarxa; Learn rate drop factor, el percentatge que es modifica el factor d'aprenentatge; Learn rate drop period, cada quantes èpoques es canvia el factor d'aprenentatge; Initial learn rate, el valor inicial del factor d'aprenentatge; i validation patience, els nombre de vegades que la funció d'error supera o és igual al mínim fins aleshores.

També és pot veure la transició de la primera xarxa a la següent, no els posem a la secció de la u-net perquè els primers experiments amb la u-net van ser fent k-fold = 1.

Relatiu al data augmentation, en aquest apartat és binari, s'aplica augment o no s'aplica augment. L'augment és rotacions de 0 a 360 graus i escalats de 0.5 a 1.5 vegades.

Els experiments van ser modificar el factor inicial d'aprenentatge de 0.001 a 0.00001, veure l'impacte del factor de paciència 4 o *maxepochs*, canviar el màxim d'iteracions 20-25 i aplicar augmentació a les dades.

7.4 U-net

Aquests experiments són els més complexos, més costosos i desenvolupats. Al trobar una xarxa que ja està dissenyada i més que comprovat que té bons resultats en l'àmbit de la segmentació d'imatges. En aquest apartat si que considerem, dintre de les limitacions de hardware i la quantitat de les dades emprades, obtenir uns resultats sòlits i extrapolables.

El mètode utilitzat va ser un *k-fold cross-validation* de $k=5$ sense fer un segon test amb dades externes, perquè es va comprovar a la transició de la xarxa anterior a aquesta que la diferència es reduïa significativament.

Training	Validation	Test
1 + 2 + 3	4	5
2 + 3 + 4	5	1
3 + 4 + 5	1	2
4 + 5 + 1	2	3
5 + 1 + 2	3	4

Taula 4. Taula de la distribució del dataset per cada iteració amb $k=5$

El *dataset* escollit ha sigut el set X de 2000 imatges. És a dir, que a l'aplicar el $k=5$ *fold* tindrem:

- El *training* estarà format per 1200 imatges, un 60% de les dades totals.
- El *validation* estarà format per 400 imatges, 20% de les dades totals.
- El test tindrà la mateixa mida que el set de validació, 400 imatges, 20% de les dades totals.

Com als experiments anteriors aquí ens centrarem amb els canvis a les opcions d'entrenament, el *data augmentation* i, com a novetat, la profunditat de la xarxa.

En l'apartat de l'entrenament ens endinsem en la importància del nombre d'iteracions, altrament entès com el número d'èpoques multiplicat per la mida de la *batch*, de 240 iteracions a 420 iteracions en increments de 60 i un extra de 900 iteracions. També modificarem el *learning rate*, de 0.01- a 0.00005 dividint a la meitat cada experiment.

Pel que fa a *data augmentation* utilitzarem paràmetres més suaus, en el sentit que els tipus de transformacions que aplicarem seran molt lleugeres. Les rotacions aniran de -15 a 15 graus, les translacions de la imatge en píxels de -3 a 3 píxels i si s'aplica reflexió.

La profunditat de la xarxa anirà de dos a cinc.

7.5 DeepLab v3+ amb ResNet-18

La combinació de que el *transfer learning* va ser un dels últims conceptes a aprendre així com la utilització de *DeepLab v3+* fan que fer proves amb la xarxa en si sigui molt més limitades.

Els paràmetres que accepta apart del model que es vol utilitzar és el factor de dilatació per les capes de convolució, tanmateix continua tenint els paràmetres d'entrenament.

En conseqüència els paràmetres que vam modificar van ser el *drop rate*, el factor d'entrenament inicial i el màxim d'èpoques.

Per avaluar el rendiment es va fer simulant el model de puntuació de l'ISIC challenge 2017. El set X com el d'entrenament, 2000 imatges; el set Y com el de validació, 150 imatges; i el set Z de test, 600 imatges. Lluny de donar-nos una bona estimació del rendiment del challenge permet situar-nos a la taula de resultats del mateix.

Els factors que vam canviar van ser:

- *Max epoch*: 10, 5, 3. Amb 30 per provar el nou *drop rate factor* de 0.95.
- *Initial learning rate*: 0.001 i 0.0001.
- *Drop rate factor*: 0.05, 0.1 i 0.95. Alhora amb el pas d'aquest. És a dir 0.95 s'aplicava cada època, mentre que el de 0.05 i 0.1 cada 2 èpoques.

7.6 U-Net vs DeepLab-ResNet-18

Fins al moment la manera que s'intentava optimitzar els resultats era a partir de la *F-score*, però al voler comparar les tres xarxes fent una simulació del challenge es presentava una nova mètrica, el *Jaccard index*.

Aquest últim apartat d'experiments s'afrontava com si se'ns formulés la següent pregunta. Si haguessis de presentar-te a l'ISIC Challenge quina xarxa amb quins paràmetres triaries?

Amb la introducció del nou índex i de la nova manera d'avaluar un model amb el *dataset* donat es van triar els paràmetres que, presumptament, millors resultats havien donat fins el moment i es va entrenar cada model. Com comentarem als resultats hi havia resultats que obrien preguntes o definien tendències i aquesta manera d'entrenar, sense fer *folds*, donava la oportunitat a provar alguns d'aquests.

8 Resultats

Els resultats dels primers experiments no se'ls pot donar tota la credibilitat que es voldria ja que només es fa un *fold*, tot i així si que es poden extreure algunes conclusions.

Nº epoch req.	Training	Validation	Test	Acc	Prec	Recall	Speci	Fscore
13	1 + 2	3	4	0.86576	0.60171	0.76109	0.88886	0.67208
13	2 + 3	4	1	0.86227	0.59163	0.76832	0.883	0.6685
14	3 + 4	1	2	0.86558	0.60037	0.76655	0.88743	0.67336
15	4 + 1	2	3	0.85207	0.58952	0.72726	0.88132	0.65119
			Avg	86.14%	59.58%	75.58%	88.52%	66.63%
			Std	0.64%	0.61%	1.93%	0.36%	1.03%
13	1 + 2	3	Test 2	0.80133	0.5498	0.74807	0.81722	0.6685
13	2 + 3	4		0.79566	0.53928	0.7608	0.80606	0.6311
14	3 + 4	1		0.80515	0.55659	0.74818	0.82215	0.63832
15	4 + 1	2		0.80486	0.55642	0.74392	0.82304	0.63666
			Avg	80.18%	55.05%	75.02%	81.71%	64.36%
			Std	0.44%	0.81%	0.73%	0.78%	1.69%

Taula 5. Taula dels resultats

En el primer i únic experiment els resultats, veure Taula 5, són sorprenents per ser la primera prova tenint en compte la xarxa que es presenta. Com a contrapartida pot presentar un alt sobre entrenament ja que no fem cap mena d'augment a les dades ni hi ha cap *drop-out layer*. Per això aquesta anomalia al final del gràfic d'entrenament de la Figura 32.

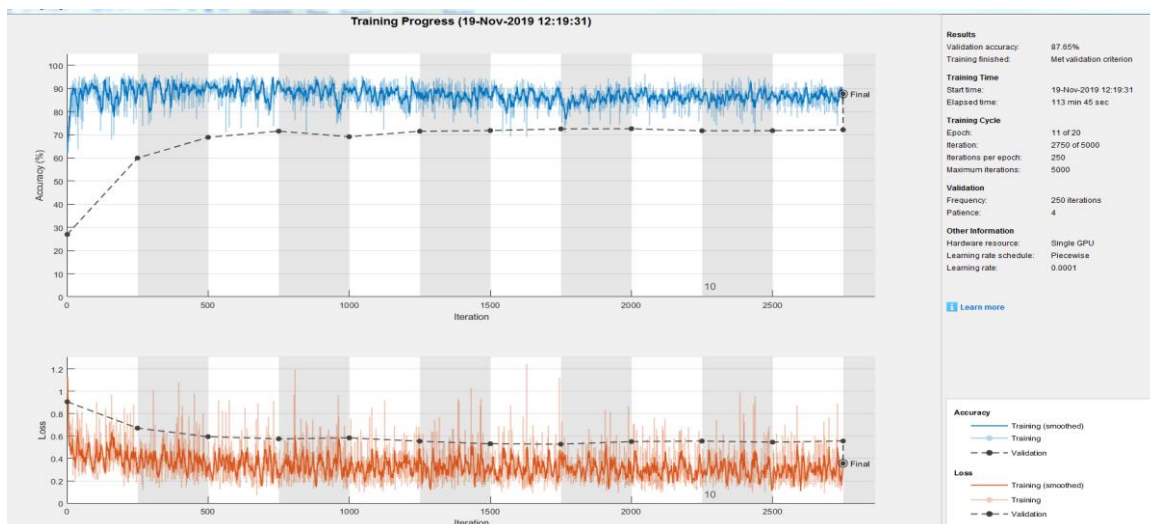


Figura 32. A la part superior de la imatge es veu una funció d'*accuracy* en relació a cada iteració. A la part inferior es pot veure la funció de pèrdua

Els millors resultats que es van aconseguir va ser amb un *learning rate* de 0.001 i *validation patience* de 4. La xarxa tenia un *max epoch* de 20 però va acabar l'entrenament a la 11a. Pel test número 1, el que sortia del set X, va donar 0.89 d'*accuracy*, 0.65 de *precision*, 0.81 de *recall*, 0.91 d'*especificity* i una *F-Score* de 0.72. Al test 2, 0.84 d'*accuracy*, 0.61 de *precision*, 0.77 de *recall*, 0.86 d'*especificity* i *F-Score* de 0.72.

Sembren un molts bon resultats però per donar-li qualsevol tipus de credibilitat faltaria fer un anàlisi més exhaustiu amb més *folders*.

Els resultats de la u-net es poden separar en 4 proves, com es pot veure a la Taula 6, les iteracions juguen un paper molt important en l'aprenentatge. Com més iteracions més *F-Score* obteníem, de 4 a 5 èpoques aconseguíem un augment de 4 punts de la *F-Score* i un descens de 2 punts de desviació estàndard. De 5 a 7 iteracions només sembla reduir-se la desviació de la *F-Score* i augmenta un punt l'*accuracy*. Si s'augmenta a 15 èpoques s'aconsegueix una *F-Score* de 0.72 però augmenta la desviació estàndard i es perd un punt d'*accuracy*.

Epoch	MiniBatch	Ite	Learning Rate	Acc	Prec	Recall	Speci	Fscore
4	20	240	1.00E-03	0.87 ± 0.02	0.67 ± 0.09	0.59 ± 0.13	0.93 ± 0.04	0.62 ± 0.07
5	20	300	1.00E-03	0.88 ± 0.02	0.70 ± 0.08	0.63 ± 0.09	0.94 ± 0.03	0.66 ± 0.04
6	20	360	1.00E-03	0.87 ± 0.02	0.63 ± 0.08	0.69 ± 0.03	0.91 ± 0.03	0.66 ± 0.04
7	20	420	1.00E-03	0.89 ± 0.01	0.76 ± 0.08	0.59 ± 0.08	0.96 ± 0.03	0.66 ± 0.02
15	20	900	1.00E-03	0.88 ± 0.04	0.67 ± 0.12	0.79 ± 0.05	0.90 ± 0.05	0.72 ± 0.05

Taula 6. Taula dels resultats de la U-net al modificar el nombre d'iteracions

Al canviar els *learning rates* es veu clarament un descens del rendiment en tots els aspectes, especialment al només fer 300 iteracions la falta d'aprenentatge és molt evident. Veure Taula 7.

Epoch	MiniBatch	Ite	Learning Rate	Acc	Prec	Recall	Speci	Fscore
5	20	300	1.00E-03	0.88 ± 0.02	0.70 ± 0.08	0.63 ± 0.09	0.94 ± 0.03	0.66 ± 0.04
5	20	300	5.00E-04	0.87 ± 0.01	0.67 ± 0.08	0.63 ± 0.13	0.93 ± 0.04	0.64 ± 0.05
5	20	300	1.00E-04	0.86 ± 0.01	0.61 ± 0.04	0.62 ± 0.07	0.91 ± 0.02	0.61 ± 0.03
5	20	300	5.00E-05	0.84 ± 0.02	0.56 ± 0.04	0.60 ± 0.06	0.89 ± 0.02	0.58 ± 0.03

Taula 7. Taula dels resultats de la U-net al modificar el learning rate

Com es veu en la següent Taula 8 vam canviar els valors del paràmetres d'entrada de l'augment de les dades. A mesura que afegim noves transformacions els resultats redueixen la desviació estàndard de a *F-Score* a canvi d'una mica de pèrdua de les altres mètriques. Notar també que la translació és tant petita que casi no afecta als resultats.

RandRotation	RandTranslation	RandReflection	Epoch	Acc	Prec	Recall	Speci	Fscore
-	-	-	5	0.88 ± 0.01	0.69 ± 0.05	0.66 ± 0.13	0.93 ± 0.02	0.66 ± 0.07
-15 15	-	-	5	0.87 ± 0.02	0.66 ± 0.06	0.66 ± 0.07	0.92 ± 0.03	0.66 ± 0.04
-15 15	-3 3	-	5	0.88 ± 0.02	0.70 ± 0.08	0.63 ± 0.09	0.94 ± 0.03	0.66 ± 0.04
-15 15	-3 3	TRUE	5	0.87 ± 0.01	0.62 ± 0.03	0.67 ± 0.04	0.91 ± 0.01	0.65 ± 0.03

Taula 8. Taula dels resultats de la U-net al modificar els tipus de transformacions

En l'estudi de la profunditat no s'han pogut extreure gaires conclusions altres que fer obvi les limitacions del hardware. Com es pot observar a la Taula 9, els resultats per profunditats de 2, 3 i 4 no varien pràcticament, la diferència abismal ha sigut pel que fa al cost computacional. En profunditats de 2 i 3 el procés era molt més ràpid que per profunditat, amb profunditat 4 l'ordinador ja es notava que li costava però funcionava. Al posar profunditat 5 el procés a diferents punts de l'entrenament donava errors de memòria a la GPU.

Depth	Epoch	MiniBatch	Ite	Learning Rate	Acc	Prec	Recall	Speci	Fscore
2	5	20	300	1.00E-03	0.86 ± 0.01	0.61 ± 0.05	0.71 ± 0.01	0.90 ± 0.02	0.65 ± 0.03
3	5	20	300	1.00E-03	0.87 ± 0.02	0.64 ± 0.07	0.67 ± 0.04	0.91 ± 0.03	0.65 ± 0.03
4	5	20	300	1.00E-03	0.87 ± 0.01	0.62 ± 0.03	0.67 ± 0.04	0.91 ± 0.01	0.65 ± 0.03
5	5	20	300	1.00E-03	-	-	-	-	-

Taula 9. Taula dels resultats de la U-net al modificar la profunditat

Al haver considerat 5 èpoques, 300 iteracions, poc temps d'entrenament es va fer dos més experiments amb 15 èpoques, el resultat va millorar a la vora del 7-10% però va augmentar lleugerament la desviació estàndard.

RandRotation	RandTranslation	RandReflection	Epoch	Ite	Learning Rate	Acc	Prec	Recall	Speci	Fscore
-	-	-	15	900	1.00E-03	0.92 ± 0.01	0.93 ± 0.04	0.63 ± 0.06	0.99 ± 0.01	0.75 ± 0.04
-15 15	-3 3	TRUE	15	900	1.00E-03	0.88 ± 0.04	0.67 ± 0.12	0.79 ± 0.05	0.90 ± 0.05	0.72 ± 0.05

Taula 10. Taula dels resultats al augmentar les iteracions amb i sense *data augmentation*

La diferencia de resultats entre tenir i no tenir *data augmentation* enganya una mica en aquesta Taula 10, la raó és probablement, que 900 iteracions és un punt òptim a parar l'entrenament si no s'aplica cap augment a les dades. Mentre que quan s'augmenten les dades al tenir imatges lleugerament diferents a cada època tot i tenir una *accuracy* alta a la validació, l'entrenament té molta variància, en altres paraules, l'amplitud de la funció de cost es va reduint a mesura que avança l'entrenament. Deixar entrenar més temps el model o augmentar el *learning rate* al model amb *data augmentation* milloraria els resultats.

Els resultats de la xarxa DeepLab amb com a *backbone* ResNet-18 ha sigut la que ha donat millor resultats. Els resultats de baixar les èpoques, com es veu a la Taula 11, cauen a mesura que es redueixen les iteracions.

Epoch	Learning Rate	Acc	Prec	Recall	Speci	Fscore
3	1.00E-03	0.92	0.92	0.71	0.98	0.8
5	1.00E-03	0.92	0.95	0.71	0.99	0.81
10	1.00E-03	0.93	0.95	0.73	0.99	0.82

Taula 11. Resultats de la xarxa DeepLab v3+ modificant el nombre d'èpoques/iteracions

Com havíem vist amb els resultats de la u-net si teníem poques èpoques i baixàvem el factor d'aprenentatge inicial teníem una pèrdua de *F-Score* final. Veure Taula 12.

Epoch	Learning Rate	Acc	Prec	Recall	Speci	Fscore
3	1.00E-03	0.92	0.92	0.71	0.98	0.8
3	1.00E-04	0.9	0.87	0.68	0.97	0.76

Taula 12. Resultats després de canviar el learning rate inicial

A part de veure com influencien les èpoques també vam provar per primera vegada de forma formal, que passava si reduïem el factor d'aprenentatge cada iteració. Els resultats de la Taula 13, mostren com al multiplicar el factor d'aprenentatge per un número més petit cau la *F-Score* final. Però, si incrementem les èpoques llavors teníem guanys mínims.

Epoch	Learning Drop Rate	Learning Rate	Acc	Prec	Recall	Speci	Fscore
5	0.05	1.00E-03	0.92	0.94	0.7	0.99	0.8
5	0.10	1.00E-03	0.92	0.93	0.72	0.99	0.81
30	0.95	1.00E-03	0.93	0.95	0.74	0.99	0.83

Taula 13. Taula de resultats si s'aplica una learning rate variat

També s'observa que si augmentem tres vegades el màxim d'iteracions, veure Taula 11 fila 3 (10 iteracions) i Taula 13 fila 3 (30 iteracions), tampoc augmenta la *F-Score* tant dràsticament. Aquesta comparació dona pas a l'últim set d'experiments en el qual presentem una xarxa com a l'escollida per presentar-nos al challenge.

El *validation patience* el tenim oblidat, però després d'analitzar alguns resultats als primers experiments i a la u-net hem arribat a la conclusió que si l'experiment acabava per aquesta condició sol tenir millors resultats. En conseqüència li donarem una oportunitat.

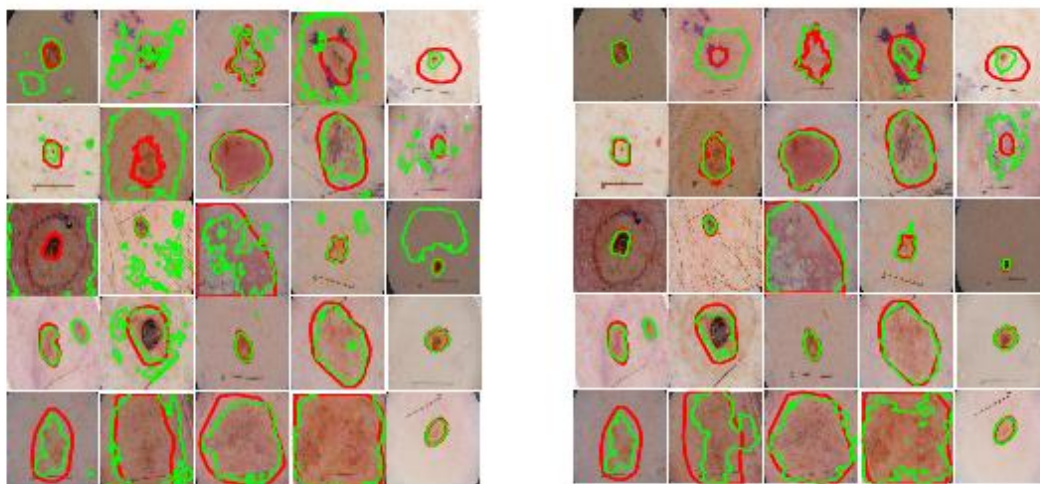


Figura 33. A l'esquerra, resultats de 25 imatges amb el contorn del ground truth(vermell) i l'obtingut per la U-net utilitzada. A la dreta el resultat resultant del model DeepLab v3+

Amb la U-net s'han escollit els següents paràmetres: *Learning rate* = 1.00E-03, *depth encoder* = 4, mida de la imatge 128x128, les mateixes transformacions, *validation patience* = 8, *max epoch* = 25. Va finalitzar a la 19a època.

El resultat amb U-net és el següent:

- *Accuracy*: 0.868
- *Precision*: 0.678
- *Recall*: 0.808
- *Specificity*: 0.886
- *F-Score*: 0.737
- *Jaccard index*: 0.584

	Data Augmentation			Depth	Epoch	Epoch final	MiniBatch	Size	Learning Rate
	RandRotation	RandTranslation	RandReflection						
U-net	-15 15	-3 3	TRUE	4	25	19	8	128x128x3	0.001
DeepLab	-15 15	-3 3	TRUE	-	15	11	8	244x244x3	0.001

Taula 14. Taula dels paràmetres d'entrada de la U-net i la DeepLab v3+

Per la DeepLab s'ha intentat agafar els millors paràmetres possibles. El *learning drop rate* no ha produït cap canvi significatiu i sí que fa augmentar el temps d'aprenentatge, per aquest motiu escollirem no posar-ne. El factor inicial amb millors resultats tant de la u-net com als pocs experiments que hem pogut fer amb la última xarxa ha sigut 1.00E-03, per tant aquest serà l'escollit. L'aprenentatge no

l'hem variat des del principi, només en fases als primers experiments el vam tocar però no l'hem treballat en particular, utilitzarem el mateix set de transformacions. Un canvi que farem és afegir èpoques en comparació a la Taula 11, però ni molt menys com a l'últim resultat de la Taula 13. Va finalitzar a la 11a època.

El resultat amb DeepLab v3+ és el següent:

- *Accuracy*: 0.937
- *Precision*: 0.900
- *Recall*: 0.820
- *Specificity*: 0.972
- *F-Score*: 0.858
- *Jaccard index*: 0.752

Aquest resultat no ens situaria com a primers però no estariem ni molt menys lluny del primer participant, al voltant de la posició número sis. El *Score* de la Figura 34 fa referència al *Jaccard index*.

Rank	User	Title	Organization	Documentation	Date	Score
1	Yading Yuan	Test-15.4.6+finals	Mt.Sinai	[icon]	Thu, 2 Mar 2017, 12:43:58 am	0.765 [icon]
2	Matt Berseth	rc13 - Final Submission	NLP LOGIX / WISEEYEA	[icon]	Wed, 1 Mar 2017, 10:42:36 pm	0.762 [icon]
3	popleyi	MResNet-Seg	USYD - BMIT	[icon]	Thu, 2 Mar 2017, 4:24:23 am	0.760 [icon]
4	Euijoon Ahn	ResNet (Single-Scale with additional training images)	USYD-BMIT	[icon]	Wed, 1 Mar 2017, 5:17:21 am	0.758 [icon]
5	RECOD Titans	3	RECOD Titans	[icon]	Thu, 2 Mar 2017, 3:26:16 am	0.754 [icon]
6	Jeremy Kawahara	final segmentation - on test	Jer	[icon]	Thu, 2 Mar 2017, 5:07:40 am	0.752 [icon]
7	Jahanifar Zamani	Saliency Map Contour Propagation	NedMos - Tarbiat Modares University	[icon]	Wed, 1 Mar 2017, 5:55:51 pm	0.749 [icon]
8	INESC TECNALIA	INESC TECNALIA	INESC TEC Porto / Tecnalia	[icon]	Wed, 1 Mar 2017, 9:41:57 pm	0.735 [icon]
9	Vic Lee	task1_final_Alice	Computer Vision Institute, Shenzhen University	[icon]	Thu, 2 Mar 2017, 2:53:05 am	0.718 [icon]

Figura 34. Resultats del ISIC Challenge 2017

És interessant entendre els resultats de forma visual també. Per classificar-los de forma visual utilitzarem tres categories:

- **Molt bon resultat:** Aquelles imatges on la predicció i el *ground truth* són molt semblants.
- **Resultat mediocre:** Aquelles imatges on la predicció i el *ground truth* tenen una relació bastant semblant.
- **Mal resultat:** Aquelles imatges on clarament hi ha una diferència notable entre el resultat esperat i l'obtingut.

Els resultats visuals s'han extret de les dues últimes xarxes(U-net i DeepLabv3+) en el seu estat final, en els experiments previs només utilitzàvem una mètrica quantitativa per la millora.

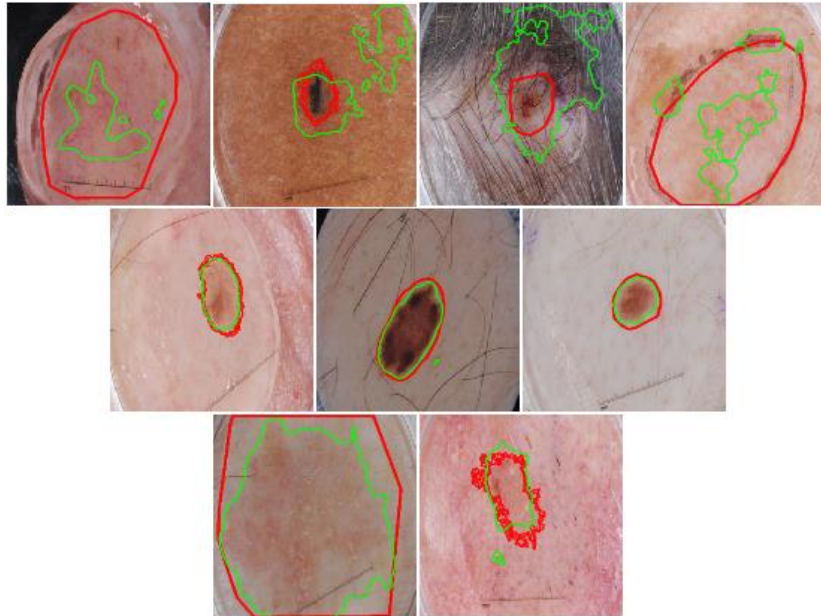


Figura 35. Resultats generats per la DeepLab v3+. A la línia superior hi ha mals resultats, a la línia del mig hi ha molt bons resultats i a la línia inferior hi ha resultats mediocres.

Com es pot veure a la Figura 35 a la fila superior hi ha uns exemples de mals resultats. Aquests mals resultats poden tenir múltiples causes, els objectes com els pels o les pintades semblen tenir a vegades un impacte negatiu al resultat. Determinats tons de pell també influeixen al resultat per això es creu que incrementar la base de dades augmentaria lleugerament els resultats.

A la segona línia de la Figura 35 destaquen els bons resultats. Acostumen a tenir bons resultats aquelles pigues que tenen un color pronunciadament diferent al de la pell i tenen formes més regulars.

Per acabar, a la fila inferior de la Figura 35 es poden observar aquells resultats que es podrien considerar acurats però no perfectes. La diferència sol ser que a vegades el *ground truth* és irregular i la nostra predicció per contra, és molt més regular i sembla discriminar menys. En canvi a vegades passa tot el contrari, on el *ground truth* és molt genèric i la predicció tendeix a mostrar un contorn molt més definit de la piga.

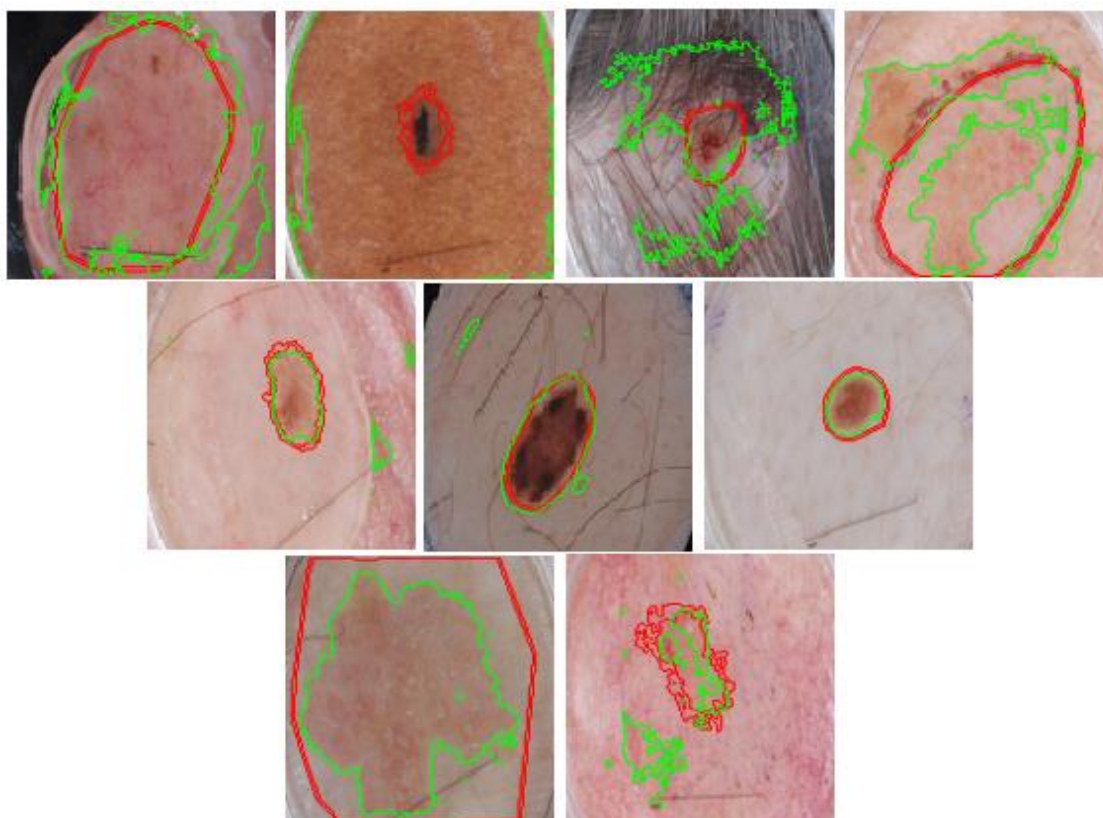


Figura 36. Resultats generats per la U-net amb la mateixa distribució de la Figura 35

Si agafem els mateixos exemples que a la Figura 35 hi ho fem amb la U-net, veure Figura 36, podem apreciar com no canvia molt. A part de les diferències clares entre els resultats de les dues xarxes que comentarem a les conclusions, només es canviaria de categoria l'exemple de la part inferior dreta, que passaria de mediocre a mal resultat. I és consideraria posar la imatge de la cantonada superior esquerra a molt bona.

Pel que fa a les causes de perquè els resultats no semblen diferir molt de la hipòtesis extreta al analitzar la Figura 35.

9 Conclusions i treball futur

9.1 Conclusions

La primera xarxa es va descartar del la comprovació final en l'àmbit del Challenge ja que els primers resultats al analitzar-la no eren prometedors, a la vegada que era molt simple. Va servir com a eina d'aprenentatge personal per introduir-me a les xarxes neuronals.

Referent a la U-net, ha sigut la xarxa que més nombre de resultats ha donat en tant que ha sigut la xarxa en que s'ha dedicat més temps a analitzar. Tot i no ser la que millors resultats ens ha donat als nostres experiments si que és extensament emprada en l'àmbit de segmentació d'imatges amb sòlids resultats. Al no haver aplicat cap tipus de *transfer learning* és probable que aprengui característiques molt més definides pel set de dades d'entrenament i li costi generalitzar. La manca d'aquesta transferència també és probable que beneficiï un nombre de èpoques molt més elevat comparada a la última xarxa utilitzada.

Si comparem a nivell visual alguns exemples de resultats de les dues últimes xarxes, U-net i DeepLab v3+- Figura 33 -es poden extreure les següents conclusions: la U-net acostuma a tenir contorns molt més irregulars amb múltiples zones d'interès i en situacions tendeix a donar resultats extrems, on és massa precís o on determina una zona molt gran com a resultat; DeepLab dona resultats significativament més definits i uniformes, ara bé, prioritza *precision* sobre *recall* i des d'un punt de vista mèdic interessaria més que fos al revés.

El resultat no ens l'hem de prendre com a completament resolutori, pot ser que el model estigui fent *overfitting* per diferents motius com podria ser la falta d'imatges d'entrenament. Al cap i a la fi, 2.000 imatges durant 11 èpoques, s'estima que serien unes 22.000 imatges d'entrenament tot i que la gran majoria serien semblants. 30º de rang de rotació, 6 píxels en cada eix i una reflexió quan algunes imatges són lleugerament simètriques no presenten casos completament nous a la xarxa.

No obstant no és un mal resultat i s'anima a continuar treballant en aquesta línia, ja que la xarxa DeepLab v3+ és relativament nova i té moltes possibilitats.

9.2 Treball futur

Una vegada finalitzats els experiments arribem a conclusions que indueixen possibles millores o aplicacions futures, són les següents:

- Processar prèviament les imatges. Nosaltres al procés previ només modificàvem la mida de les imatges. Es proposa canviar el seu espai de color, de RGB a CIELAB o HSV.
- També és recomana, si el hardware ho permet, incrementar la mida de les imatges. Nosaltres hem utilitzat 128x128 per la U-net i 244x244 per la DeepLab v3+, es recomanaria arribar a uns 512x512 píxels.
- Pel que fa a modificar la mida de les imatges també es proposaria en un futur treball, treballar amb imatges que no siguin quadrades. Per exemple, una mitjana de totes les imatges d'entrenament de la mida de cada eix.
- Incrementar significativament les iteracions d'entrenament. En el nostre treball s'ha variat de 5 a 20 èpoques amb un cas excepcional de 30. Es recomanaria incrementar de manera significativa aquest número i veure quin impacte tindria, especialment en l'entorn de la U-net.
- Com a opcions d'entrenament hem utilitzat l'algorisme SGDM(Stochastic gradient descent with momentum) per optimitzar. Es convida a recercar altres algorismes com RMSProp o Adam(Adaptative moment estimation).
- A resultes del punt anterior, utilitzar un factor d'entrenament variable. En el treball s'ha utilitzat en alguns casos però al modificar-se per èpoques i realitzar-se'n poques no millorava resultats de forma clara. Multiplicar el *learning rate* inicial per un valor de 0.75 a 0.99 cada època durant 100, 200 o 300 èpoques seria un bon punt de partida.
- Utilitzar ResNets amb més capes potencialment augmentaria els resultats de l'experiment. Com seria ResNet-34, ResNet-50 o ResNet-101.
- El següent pas natural, ja fora de la segmentació, seria aplicar els valors extrems d'aquesta segmentació per classificar les lesions.
- Crear una aplicació per dispositius mòbils en la qual es pugui aplicar aquesta tecnologia juntament amb la classificació.

10 Bibliografia

- [1] Heron M. Deaths: Leading causes for 2017. National Vital Statistics Reports; vol 68 no 6. Hyattsville, MD: National Center for Health Statistics. 2019.
- [2] Bray, F.; Jemal, A.; Grey, N.; Ferlay, J.; Forman, D. "Global cancer transitions according to the Human Development Index (2008-2030): a population-based study". The Lancet
- [3] Anand P, Kunnumakkara AB, Sundaram C, Harikumar KB, Tharakan ST, Lai OS, Sung B, Aggarwal BB (September 2008). "Cancer is a preventable disease that requires major lifestyle changes". *Pharmaceutical Research*. 25 (9):2097116. doi:10.1007/s11095-008-9661-9. PMC 2515569. PMID 18626751.
- [4] National Cancer Institute, What Is Cancer? Updated <2/9/2015>. Available at: <https://www.cancer.gov/about-cancer/understanding/what-is-cancer>. Accessed <11/20/2019>.
- [5] Skin Cancer Foundation. Skin Cancer Facts & Statistics. Updated <5/31/2019>. Available at: <https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/>. Accessed <11/20/2019>.
- [6] PDQ® Adult Treatment Editorial Board. PDQ Melanoma Treatment. Bethesda, MD: National Cancer Institute. Updated <11/9/2019>. Available at: <https://www.cancer.gov/types/skin/patient/melanoma-treatment-pdq>. Accessed <12/6/2019>. [PMID: 26389388]
- [7] NHS. Overview: Skin cancer(non-melanoma). Updated <1/6/2019>. Available at: <https://www.nhs.uk/conditions/non-melanoma-skin-cancer/>. Accessed <12/6/2019>.
- [8] SEER National Cancer Institute. Cancer Stat Facts: Melanoma of the Skin. Available at: <https://seer.cancer.gov/statfacts/html/melan.html>. Accessed <12/6/2019>.
- [9] M. Binder, A. Steiner, M. Schwarz, S. Knollmayer, K. Wolff, and H. Pehamberger, "Application of an artificial neural network in epiluminescence microscopy pattern analysis of pigmented skin lesions: A pilot study," *Br. J. Dermatol.*, vol. 130, pp. 460–465, 1994 Codella N, Gutman D, Celebi ME, Helba B, Marchetti MA, Dusza S, Kalloo A, Liopyris K, Mishra N, Kittler H, Halpern A. "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging

- (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)". arXiv:1710.05006[cs.CV] Available: <https://arxiv.org/abs/1710.05006>
- [10] Moral F. Redes completamente convolucionales en la segmentación semántica de lesiones melanocíticas. <6/21/2017>.
- [11] Yuan Y. Automatic skin lesion segmentation with fully convolutional-deconvolutional network. Available at: <https://arxiv.org/pdf/1703.05165.pdf>. <9/28/2017>.
- [12] Berseth M. ISIC 2017 - Skin Lesion Analysis Towards Melanoma Detection. Available at: <https://arxiv.org/ftp/arxiv/papers/1703/1703.00523.pdf>.
- [13] Qian C, Jiang H. and Liu T. ISIC 2018 - Skin Lesion Analysis. Available throughout: <https://challenge2018.isic-archive.com/leaderboards/>. <7/28/2018>.
- [14] Du H. Seok J.Y Et al. Team HolidayBurned at ISIC CHALLENGE 2018. Available throughout: <https://challenge2018.isic-archive.com/leaderboards/>.
- [15] ISIC Archive Gallery. Available at: <https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery>.
- [16] Gutman, David; Codella, Noel C. F.; Celebi, Emre; Helba, Brian; Marchetti, Michael; Mishra, Nabin; Halpern, Allan. "Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)". eprint arXiv:1605.01397. 2016.
- [17] Codella N, Gutman D, Celebi ME, Helba B, Marchetti MA, Dusza S, Kalloo A, Liopyris K, Mishra N, Kittler H, Halpern A. "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)". arXiv: 1710.05006 [cs.CV] Available: <https://arxiv.org/abs/1710.05006>
- [18] MathWorks. Pretrained Deep Neural Networks. Available at: <https://es.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html>

- [19] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, Allan Halpern: “Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)”, 2018; <https://arxiv.org/abs/1902.03368>
- [20] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, Josep Malvehy: “BCN2000: Dermoscopic Lesions in the Wild”, 2019; arXiv:1908.02288.
- [21] Tschandl, P. et al. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data 5:180161 doi: 10.1038/sdata.2018.161 (2018).
- [22] McCulloch, W.S. & Pitts, W. Bulletin of Mathematical Biophysics (1943) 5: 115. <https://doi.org/10.1007/BF02478259>
- [23] Widrow B, Hoff M.E. Adaptive Switching Circuits. <6/30/1960>.
- [24] Nielsen M. Neural Networks and Deep Learning: How the backpropagation algorithm Works. Available at: <http://neuralnetworksanddeeplearning.com/index.html>. Dec 2019.
- [25] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Available at: <https://arxiv.org/pdf/1502.03167v3.pdf>. <3/2/2015>
- [26] Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning. Available at: <https://arxiv.org/pdf/1603.07285.pdf>. <1/11/2018>
- [27] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple Way to Prevent Neural Networks from Overfitting. Available at: <http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>. <6/14>
- [28] MathWorks. Preprocess Images for Deep Learning: Augment Images for Training with Random Geometric Transformations. Available at: <https://es.mathworks.com/help/deeplearning/ug/preprocess-images-for-deep-learning.html>.
- [29] MathWorks. Semantic Segmentation Examples: Create a Semantic Segmentation Network. https://es.mathworks.com/help/vision/ug/semantic-segmentation-examples.html#mw_6ab02754-d2fa-4330-8bea-3eeec77279da

- [30] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham
- [31] Chen, L., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." *Computer Vision — ECCV 2018*, 833-851. Munic, Germany: ECCV, 2018.
- [32] Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. Available at: arXiv preprint arXiv:1706.05587 (2017)