Universitat Oberta de Catalunya (UOC)

Màster Universitari en Enginyeria Informàtica

Àrea: Intel·ligència Artificial

Reconeixement intel·ligent de malalties oculars mitjançant arquitectures d'aprenentatge profund

Autor: Jordi Coll i Corbilla

Director del TFM: Joan M. Nuñez Do Rio

Professor: Carles Ventura Royo

Londres, 31 de desembre del 2019

Índex

Índex de figures 1

Índex de taules 4

1 Abstracte 6

2 Introducció 8

3 Justificació i motivació del TFM 19

4 Objectius del TFM 21

4.1 Objectiu General 21

4.2 Objectiu Específic 23

5 Estructuració del treball 24

5.1 Planificació i execució del treball 24

5.1.1 Relació de tasques 24

5.1.2 Planificació temporal 26

6 El conjunt de dades 27

6.1 El fitxer d'anotacions especials 29

6.2 Imatges de validació 31

7 Metodologia 33

7.1 Preprocessament 33

7.1.1 Anàlisi del dataset i generació del ground truth 34

7.1.2 Tractament de les imatges 36

7.1.3 Generació de la seqüenciació d'elements 41

7.1.4 Augment de dades 42

7.2 Model Inception 51

7.2.1 Descripció del model 51

7.2.2 Procés d'entrenament 55

7.2.3 Procés de validació 55

7.2.4 Experiments 59

7.2.4.1 Experiment inicial 59

7.2.4.2 Experiment amb pesos o parcialitat en cada classe de sortida 63

7.2.4.3 Experiment amb Augment de dades 68

7.2.4.4 Experiment amb aprenentatge transferit 72

7.2.4.5 Fine-Tuning 76

7.3 Model VGG-16 79

7.3.1 Descripció del model 79

7.3.2 Procés d'entrenament 81

7.3.3 Procés de validació 82

7.3.4 Experiments 83

7.3.4.1 Experiment inicial 83

7.3.4.2 Experiment amb aprenentatge transferit i augment de dades 87

7.3.4.3 Fine-Tuning 91

8 Discussió 93

9 Conclusió 98

9.1 Dificultats 99

9.2 Objectius assolits 100

10 Tecnologia 101

10.1 Frameworks utilitzats 101

10.2 Preparació del maquinari 101

11 Annexos 103

11.1 Fitxer d'anotacions de les dades 103

11.2 Fitxer d'explicacions especials 103

11.3 Codi Font 104

11.4 Llistat d'imatges descartades 104

11.5 Llistat d'imatges amb un nou ground truth 104

11.6 Resum del model Inception 105

11.7 Resum del model VGG 105

12 Glossari 106

13 Bibliografia 107

Índex de figures

Figura 1. Retinografia ull esquerre i dret (vist des del front) sense cap anomalia. 9

Figura 2. Càmera retinal (Roletschek, 2019)..... 10

Figura 3. (1) Retinopatia diabètica, (2) Glaucoma, (3) Cataracta, (4) Degeneració de la màcula, (5) Hipertensió, (6) Miopia, (7) Druses, (8) membrana epirretiniana. 11

Figura 4. Classificació de malalties oculars mitjançant CNNs. 16

Figura 5. Representació del diagrama de flux 17

Figura 6. Diagrama de planificació temporal 26

Figura 7. Histograma d'edat dels pacients 28

Figura 8. Classificació de les diferents malalties 28

Figura 9. Exemple de registre oftàlmic del conjunt de dades..... 29

Figura 10. Representació dels grups d'imatges (entrenament i validació) 31

Figura 11. Exemple d'imatges de prova amb diferents formats..... 37

Figura 12. Llista d'imatges tractades..... 38

Figura 13. Algorisme de retall d'imatges..... 39

Figura 14. Efecte de mirall de les imatges 40

Figura 15. Redimensionat d'imatges..... 41

Figura 16. Desglossament de dades..... 42

Figura 17. Exemple de matriu de confusió..... 47

Figura 18. Algorismes d'augment de dades utilitzats 50

Figura 19. Augment de dades: escalat, saturació, retall, il·luminació, contrast i d'altres. 50

Figura 20. Model Inception 52

Figura 21. Capes afegides al model Inception..... 53

Figura 22. Model Inception 54

Figura 23. Mostra de dades d'entrada etiquetades 56

Figura 24. Exemple de sortida del model per a la seva validació visual 57

Figura 25. Exemple de matriu de confusió 58

Figura 26. Accuracy d'entrenament i Accuracy de validació 61

Figura 27. Pèrdua de l'entrenament i pèrdua de la validació 61

Figura 28. Matriu de confusió de l'experiment inicial 63

Figura 29. Execució de l'experiment amb pesos model Inception 65

Figura 30. Accuracy d'entrenament i Accuracy de validació 66

Figura 31. Matriu de confusió 66

Figura 32. Exemple de resultat del classificador 67

Figura 33. Resultat de l'execució del model 69

Figura 34. Accuracy d'entrenament i Accuracy de validació 70

Figura 35. Matriu de confusió 71

Figura 36. Exemple de sortida de les dades de validació 72

Figura 37. Sortida de l'execució amb aprenentatge transferit 74

Figura 38. Accuracy d'entrenament i Accuracy de validació 74

Figura 39. Matriu de confusió 75

Figura 40. Matriu de confusió final 78

Figura 41. Model VGG16 80

Figura 42. Precisió i Precisió de validació 85

Figura 43. Pèrdua i pèrdua de validació 85

Figura 44. Matriu de confusió 86

Figura 45. Execució de l'experiment amb aprenentatge transferit i model VGG16 88

Figura 46. Accuracy d'entrenament i Accuracy de validació 89

Figura 47. Matriu de confusió 90

Figura 48. Matriu de confusió final 92

Figura 49. Predicció de sortida 94

Figura 50. Matrius de confusió (Inception vs. VGG) 95

Figura 51. Resultats ODIR amb model Inception 96

Figura 52. Resum de resultats 98

Índex de taules

Taula 1. Relació de tasques 24

Taula 2. Desequilibri de dades 32

Taula 3. Ground truth per a imatges específiques 35

Taula 4. Desequilibri de dades 43

Taula 5. Càlcul dels pesos per al desequilibri 43

Taula 6. Mètriques utilitzades 44

Taula 7. Nombre d'Imatges a generar per classe minoritària 49

Taula 8. Configuració de l'experiment 59

Taula 9. Resultats bàsics amb la xarxa Inception 60

Taula 10. Configuració de l'experiment 63

Taula 11. Resultats sobre les dades d'entrenament amb pesos amb la xarxa Inception 64

Taula 12. Resultats sobre les dades de validació amb pesos amb la xarxa Inception 64

Taula 13. Resultats sobre les dades de validació amb pesos amb la xarxa Inception 67

Taula 14. Configuració de l'experiment 68

Taula 15. Resultats sobre les dades d'entrenament amb pesos amb la xarxa Inception 68

Taula 16. Resultats sobre les dades de validació amb pesos amb la xarxa Inception 69

Taula 17. Resultats puntuació final 71

Taula 18. Configuració de l'experiment 72

Taula 19. Resultats sobre les dades d'entrenament amb aprenentatge transferit 75

Taula 20. Resultats sobre les dades de validació amb aprenentatge transferit 75

Taula 21. Resultats puntuació final 76

Taula 22. Configuració de l'experiment 76

Taula 23. Resultats dels experiments mitjançant Fine-tuning 77

Taula 24. Configuració de l'experiment 83

Taula 25. Resultats bàsics amb la xarxa VGG-16 84

Taula 26. Configuració de l'experiment 87

Taula 27. Resultats sobre les dades d'entrenament amb pesos amb la xarxa VGG-16 87

Taula 28. Resultats sobre les dades de validació amb pesos amb la xarxa VGG-16 87

Taula 29. Resultats sobre les dades de validació amb pesos amb la xarxa VGG-16 90

Taula 30. Resultats dels experiments mitjançant Fine-tuning 91

Taula 31. Resultat dels models 93

Taula 32. Configuració dels models 94

1 Abstracte

Resum: Les patologies de la retina són la causa més comuna de ceguesa infantil a tot el món. La detecció ràpida i automàtica de malalties és crítica i urgent per a reduir la càrrega de treball dels oftalmòlegs. Els oftalmòlegs diagnostiquen malalties basant-se en el reconeixement de patrons mitjançant la visualització directa o indirecta de l'ull i les seves estructures circumdants. La dependència amb el fons d'ull i la seva anàlisi fa que el camp de l'oftalmologia sigui perfectament adequat per beneficiar-se d'algorismes de deep learning. Cada malaltia té diferents estadis de severitat que es poden deduir verificant l'existència de lesions específiques i cada lesió es caracteritza per certs trets morfològics on diverses lesions de diferents patologies tenen característiques similars. Observem que els pacients poden veure's afectats simultàniament per diverses patologies i en conseqüència, la detecció de malalties oculars presenta una classificació amb múltiples etiquetes amb un principi de resolució complex.

S'estudien dues solucions de deep learning per la detecció automàtica de múltiples malalties oculars. Les solucions que s'han escollit són a causa del seu major rendiment i puntuació final en el repte ILSVRC: GoogLeNet i VGGNet. Primer, estudiem les diferents característiques de les lesions i definim els passos fonamentals del processament de les dades. Posteriorment, identifiquem el programari i el maquinari necessaris per executar les solucions d'aprenentatge profund. Finalment, investiguem els principis d'experimentació implicats per avaluar els diferents mètodes, la base de dades pública utilitzada per a les fases d'entrenament i validació i es reporta i es discuteix la precisió final de detecció amb altres mètriques importants.

Paraules clau: Classificació d'imatges, Aprenentatge profund, Retinografia, Xarxes neuronals convolucionals, Malalties oculars, Anàlisi d'imatges mèdiques.

Abstract: Retinal pathologies are the most common cause of childhood blindness worldwide. Rapid and automatic detection of diseases is critical and urgent in reducing the ophthalmologist's workload. Ophthalmologists diagnose diseases based on pattern recognition through direct or indirect visualization of the eye and its surrounding structures. Dependence on the fundus of the eye and its analysis make the field of ophthalmology perfectly suited to benefit from deep learning algorithms. Each disease has different stages of severity that can be deduced by verifying the existence of specific lesions and each lesion is characterized by certain morphological features where several lesions of different pathologies have similar characteristics. We note that patients may be simultaneously affected by various pathologies, and consequently, the detection of eye diseases has a multi-label classification with a complex resolution principle.

Two deep learning solutions are being studied for the automatic detection of multiple eye diseases. The solutions chosen are due to their higher performance and final score in the ILSVRC challenge: GoogLeNet and VGGNet. First, we study the different characteristics of lesions and define the fundamental steps of data processing. We then identify the software and hardware needed to execute deep learning solutions. Finally, we investigate the principles of experimentation involved in evaluating the various methods, the public database used for the training and validation phases, and report the final detection accuracy with other important metrics.

Keywords: Image classification, Deep learning, Retinography, Convolutional neural networks, Eye diseases, Medical imaging analysis.

2 Introducció

La retina és una capa de teixit de la part posterior de l'ull que percep la llum d'entrada i envia imatges al nostre cervell. Al seu centre existeix un teixit nerviós on es troba la màcula, que proporciona una visió nítida i que ens ajuda amb tasques tan necessàries com conduir i llegir. Aquest teixit tan valuós es pot veure afectat per diferents trastorns o malalties que poden afectar la visió. Avui dia, les patologies de la retina ja són la causa més comuna de ceguesa infantil a tot el món. A mesura que els països es fan més rics i els ingressos per càpita augmenten, la prevalença de ceguesa disminueix i les causes que produeixen la ceguesa canvien. En les nacions més pobres del món, la principal causa de ceguesa és la cataracta. En un país amb un producte interior brut mitjà com a Amèrica llatina, la principal causa de ceguesa és el glaucoma i la retinopatia diabètica. A causa de la millora econòmica, la cirurgia de la cataracta és majorment accessible i la seva incidència és menor. En països amb un PIB alt, el glaucoma i la cataracta continuen sent patologies molt habituals i importants, però la ceguesa es deu a altres malalties de la retina com la retinopatia diabètica que es pot prevenir i és tractable en les seves primeres etapes. (Gilbert, C. 2001)

La diabetis és un problema creixent als països en desenvolupament. A l'Índia, s'estima que entre el 8 i el 10% de la població és diabètica i la seva prevalença creix. Tot i que els estudis basats en la població suggereixen que la retinopatia diabètica no és una causa principal de ceguesa a l'Índia actualment, és probable que ho sigui en el futur. L'any 2010 es realitzaven al voltant de 10 milions d'operacions de cataractes a l'any a tot el món i el 2020, s'espera superar els 30 milions. Gairebé tot el creixement s'ha produït en països en desenvolupament. Més cirurgia de cataracta comporta complicacions en el segment posterior de l'ull, com és el despreniment de retina. Aquestes complicacions són molt tractables, sempre que hi hagi un cirurgià retinal qualificat i estigui ben equipat. Tenint en compte la tendència observable, és probable que les

malalties de la retina siguin ja un problema important i creixent a totes les parts del món (Yorston, D. 2003).

És cert que hi ha moltes degeneracions de la retina per a les quals no hi ha cap cura. Tot i això, els pacients poden beneficiar-se molt de rebre un diagnòstic precís, amb una explicació detallada i un pronòstic clar (sovint una malaltia afecta un ull i és possible fer la prevenció de l'altre). Per a la detecció de malalties en països desenvolupats, els oftalmòlegs utilitzen una eina estàndard d'imatge mèdica anomenada "fotografia de fons d'ull o retinografia". A través d'un procediment ràpid i senzill, el metge o especialista, és capaç d'obtenir una fotografia en color d'alta qualitat del fons de l'ull on es pot observar bé la seva morfologia i estructures (nervi òptic, vasos sanguinis, màcula, retina, etc.) com es mostra en la Figura 1, oferint una font important d'informació sobre la salut del pacient (Bonet, 2018) (Saine and Tyler, 2002).

La imatge del fons d'ull de la Figura 1 mostra la màcula en el centre de la imatge, el disc òptic situat cap al costat del nas, les artèries i les diferents venes.

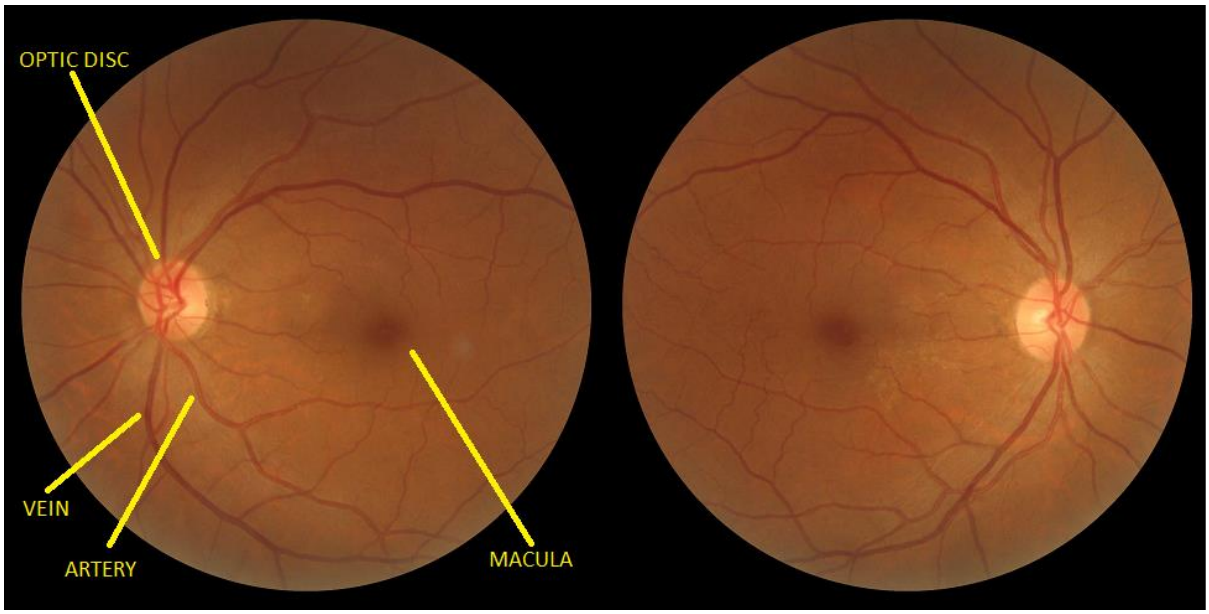


Figura 1. Retinografia ull esquerre i dret (vist des del front) sense cap anomalia.

A causa dels avenços en tecnologia, els equips per tractar malalties de la retina, tot i que encara són costosos, són ara molt més adequats per ser utilitzats en països en desenvolupament. Tanmateix, continua essent una limitació important l'escassetat de personal qualificat per afrontar els reptes futurs. Això vol dir que necessitem més oftalmòlegs amb formació subespecialitzada en malalties de la retina. (Oftalmològica, 2019).

El procediment d'obtenció de les imatges es fa mitjançant una càmera retinal tal com mostra la Figura 2. Per poder dur a terme la captura de fons d'ull, el metge aplica a l'ull unes gotes per dilatar la pupil·la i s'espera uns minuts mentre fan efecte. Al moment de realitzar l'examen, el pacient ha de mirar a la càmera, i l'especialista, pot fer la captura del fons d'ull per obtenir les imatges a analitzar posteriorment (Garcia, 2010).

Figura 2. Càmera retinal (Roletschek, 2019)

El procediment, que es pot realitzar anualment, aporta una manera eficaç, segura i econòmica d'evitar la ceguesa com més aviat millor provocada per malalties com: la diabetis, el glaucoma, la cataracta i la degeneració macular entre d'altres. L'examen del fons d'ull ha estat un aspecte clau en el diagnòstic de malalties oculars, i fonamentalment, en el diagnòstic precoç de malalties generals que es manifesten a la retina. La retina és una prolongació del cervell i tot

allò que l'afecti es manifestarà en ella, com les malalties neurodegeneratives o de tipus vascular (Oftalmològica, 2019).

Com a avantatges fonamentals que ofereix el procediment, podem considerar la facilitat d'obtenció i el ràpid diagnòstic de malalties i permetre així poder descartar l'aparició de lesions i evitar-ne la progressió. Com a desavantatge principal podem mencionar que la midriasi (que ocorre després de posar les gotes per dilatar l'ull) dura unes 4 hores, cosa que limita l'activitat del pacient fins que l'ull torna al seu estat normal.

La següent Figura 3 agrupa un conjunt de patologies comuns que es poden detectar mitjançant la prova de fons d'ull (Oftalmològica, 2019):

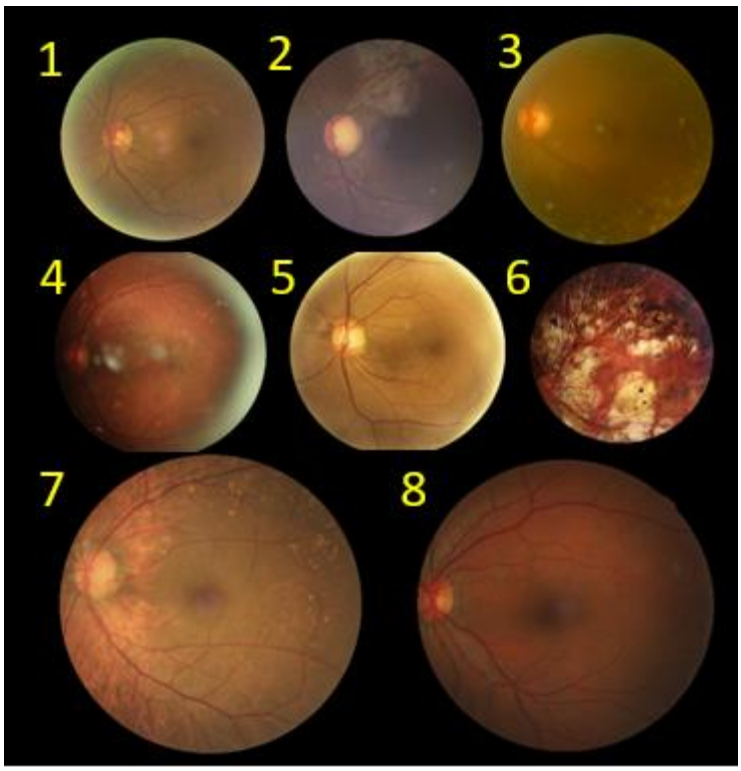


Figura 3. (1) Retinopatia diabètica, (2) Glaucoma, (3) Cataracta, (4) Degeneració de la màcula, (5) Hipertensió, (6) Miopia, (7) Druses, (8) membrana epirretiniana.

Algunes de les patologies contingudes a la Figura 3, poden provocar la ceguesa si la malaltia progressa o es complica. Les persones amb diabetis i d'una edat avançada tenen més possibilitats de desenvolupar retinopatia diabètica (DR) i si no és controlada a temps, pot derivar en ceguesa. El glaucoma és una malaltia que causa un dany progressiu en el nervi òptic a casa d'un constant increment de la pressió intraocular. El glaucoma no té cura definitiva, tot i que, mitjançant cirurgia es pot pal·liar en gran manera les seves conseqüències. Aquesta patologia és una de les principals causes de pèrdua de visió en el món.

La cataracta afecta els adults d'edat avançada i provoca una disminució de la visió a causa del canvi fisiològic de l'interior de l'ull on es desenvolupen una sèrie de pegats que dificulten la visió. La màcula és la responsable de la visió central i la visió es distorsiona si els fluids s'hi acumulen. La degeneració de la màcula (AMD), sol ocórrer en persones majors de setanta anys i normalment no sol haver-hi cap símptoma durant les primeres etapes de la malaltia, cosa que fa el fons d'ull un procés important en la detecció precoç de malalties oculars i que s'hauria de fer regularment cada dos anys.

La hipertensió és una malaltia silenciosa, però que és capaç d'afectar tot l'organisme causant destrucció amb el pas del temps. Aquesta patologia és capaç de canviar les estructures morfològiques dels vasos sanguinis, com per exemple, canvis de diàmetre, alteració de la tortuositat, i pot produir malalties vasculars cardíaques cerebrals, accidents cerebrovasculars i atacs de cor. La miopia és una alteració ocular que provoca una pèrdua de visió important a causa de l'aprimament i atenuació epitelials del pigment de la retina progressiva. Principalment altera la visió dels objectes de lluny fent-los borrosos i provoca alteracions que es poden veure al fons d'ull de manera evident donat que el nervi òptic sol estar inclinat i mostra una superfície escleròtica tal com es mostra en la Figura 3 (Garrido, 2011).

Les druses són petits sacs de material extracel·lular de color groc i que solen aparèixer a partir dels seixanta-cinc anys. Aquesta malaltia sol indicar sovint l'inici d'una degeneració macular.

La membrana epirretiniana sol formar-se davant la màcula i provoca una distorsió en la visió. Aquesta malaltia subtil crea una petita deformació davant de la màcula i genera una distorsió en la visió central. A mesura que la malaltia avança, els seus efectes acostumen a ser molt notables si no es tracta.

Les càmeres digitals de fons d'ull desen les imatges directament a l'ordinador i llavors són avaluades pels especialistes per a la generació d'un diagnòstic. L'estàndard actual per a la classificació de malalties a partir de fotografies de fons d'ull, inclou una estimació manual de les ubicacions de les lesions i l'anàlisi del seu grau de severitat, que requereixen molt de temps per part de l'oftalmòleg, incorrent també, en costos elevats en el sistema sanitari. Per tant, seria important disposar de mètodes automàtics per fer l'anàlisi.

La generació d'un diagnòstic assistit per ordinador, ens proporciona un repte interessant d'imatge mèdica per a la detecció automàtica de malalties oculars a partir de les imatges de fons d'ull. Existeixen sistemes de diagnòstic assistit per ordinador (CAD) que encara s'utilitzen en algunes situacions, per a la detecció d'anomalies a través de l'avaluació de lesions i fites anatòmiques de la imatge de fons d'ull, i que, presenten una anàlisi fiable dels resultats segons el seu context. (C.I. Sánchez et al. 2011) (C.I. Sánchez et al. 2012).

Segons el paper publicat per Zhou et al, (2019), existeixen dues àrees de recerca importants en el tractament d'imatges mèdiques de fons d'ull. La primera detalla la classificació de les malalties segons el seu grau de severitat i la segona sobre la classificació basada en la segmentació de lesions a través de l'anàlisi de les característiques de més baix nivell, és a dir, analitzant els píxels de la imatge.

Les dues àrees es poden veure com un problema genèric de classificació. Una classificació, on avui dia s'estan aplicant mètodes de deep learning, tant en Oftalmologia com en altres camps mèdics i que han suposat un gran avenç. Els oftalmòlegs diagnostiquen malalties basant-se en el reconeixement de patrons mitjançant la visualització directa o indirecta de l'ull i les seves estructures circumdants. Les tecnologies de diagnòstic proporcionen una informació d'acompanyament que ajuda al metge amb la seva presa de decisions. Aquesta dependència amb la imatge de fons d'ull i la seva anàlisi fa que el camp de l'oftalmologia sigui perfectament adequat per beneficiar-se d'algorismes de deep learning. La incorporació d'algorismes d'aprenentatge profund està començant a implementar-se en diferents àrees d'oftalmologia i pot canviar potencialment el tipus de treball realitzat pels oftalmòlegs (M.D. Abramoff, Y. Lou, A. Erginay, et al 2016) (Parampal S, et al 2018) (A. Lee et al 2017).

El deep learning s'està aplicant a fotografies de fons d'ull, tomografia de coherència òptica i altres camps visuals, aconseguint un rendiment de classificació robust en la detecció de retinopatia diabètica, retinopatia prematura, glaucoma, edema macular i degeneració macular relacionada amb l'edat. El deep learning en imatges oculars es pot utilitzar conjuntament amb la telemedicina com a possible solució per seleccionar, diagnosticar i controlar malalties oculars per a pacients en atenció primària (Ting DSW, Pasquale LR, Peng L, et al 2019).

Més concretament, mitjançant esquemes d'aprenentatge de màquines que utilitzen xarxes neuronals basades en convolució (CNNs). Les xarxes utilitzen un conjunt de filtres de processament d'imatges per a extreure'n diversos tipus de característiques on la xarxa considera l'existència de signes patològics. L'extracció de característiques ocorre després del seu aprenentatge sobre un conjunt d'entrenament i que forma part integral dels mètodes de classificació de patrons. El deep learning, doncs, es pot veure com un algorisme de força bruta

que és capaç de determinar els filtres i eines de processament d'imatges més adequats, que poden quantificar diverses característiques de les diferents malalties (Hijazi et al., 2015).

Les CNNs han esdevingut el mètode estàndard de classificació d'imatges utilitzant aprenentatge de màquina. L'ImageNet Large Scale Visual Recognition Challenge (ILSVRC), és una competició per a trobar l'arquitectura d'aprenentatge de màquina amb més precisió per la múltiple classificació d'objectes. Les CNNs han guanyat el concurs des del 2012 i la precisió de classificació ha sigut millor que la que nosaltres, els humans, podem fer des de l'any 2015. Una de les propietats més importants de l'aprenentatge profund, és la capacitat d'extreure característiques automàticament. Els avenços recents en enfocaments de xarxes neuronals s'han posat al capdavant de sistemes de reconeixement visual d'última generació (Lecun Y., 2015) (Russakovsky et al., 2015).

El nostre treball estudia dues solucions de deep learning per a la detecció automàtica de múltiples malalties oculars. Aquest problema complex d'oftalmologia es realitza mitjançant l'anàlisi d'imatges mèdiques a través de visió per computador i usant un conjunt de dades públic. Com a resultat, s'obté una categorització múltiple basada en un conjunt de malalties determinades. Es comparen dues solucions de deep learning que representen l'estat de l'art: GoogLeNet i VGGNet pels seus resultats al repte ILSVRC i perquè han sigut molt prometedores en generalitzar el contingut global d'una imatge. En els últims anys, hi ha hagut molts treballs relacionats amb la valoració d'aquests sistemes per a aplicacions mèdiques, inclosa l'anàlisi biomèdica d'imatges. GoogLeNet introdueix l'arquitectura Inception que és la guanyadora del ILSVRC 2014, mentre que VGGNet va quedar en segon lloc el mateix any (Krizhevsky, 2012) (Simonyan, 2015).

Totes dues arquitectures van obtenir molts bons resultats al concurs ImageNet i s'han escollit a causa de la seva implementació i agilitat d'entrenament. Inicialment, el principal inconvenient

en l'aplicació de CNNs era la manca de poder computacional però l'augment d'unitats gràfiques de processament (GPUs), que han superat amb escreix les capacitats computacionals de les unitats de processament centrals (CPU), ha fet que les CNNs d'aprenentatge profund es puguin implementar en maquinari d'escriptori.

D'altra banda, treballs com el realitzat per Jaworek-Korjakowska, J., et al (2019) sobre la predicció del gruix del melanoma basat en xarxes neuronals convolucionals amb l'aprenentatge de transferència de models VGG han tingut un pes important en la tria de l'arquitectura a causa dels seus bons resultats si s'utilitza amb aprenentatge transferit.

Les xarxes es desenvolupen mitjançant TensorFlow i Keras en Python (Abadi et al., 2016) i els experiments es fan sobre un maquinari d'escriptori amb una única GPU. TensorFlow és una llibreria de programari de codi font obert per computació numèrica utilitzant gràfics de flux de dades i és capaç de desplegar càlculs en una o més CPU/GPU amb una sola API. Keras és una llibreria d'aprenentatge profund d'alt nivell que s'executa per sobre de TensorFlow i que ens ofereix un nivell d'abstracció més elevat. La Figura 4 mostra un detall de què s'aconsegueix.

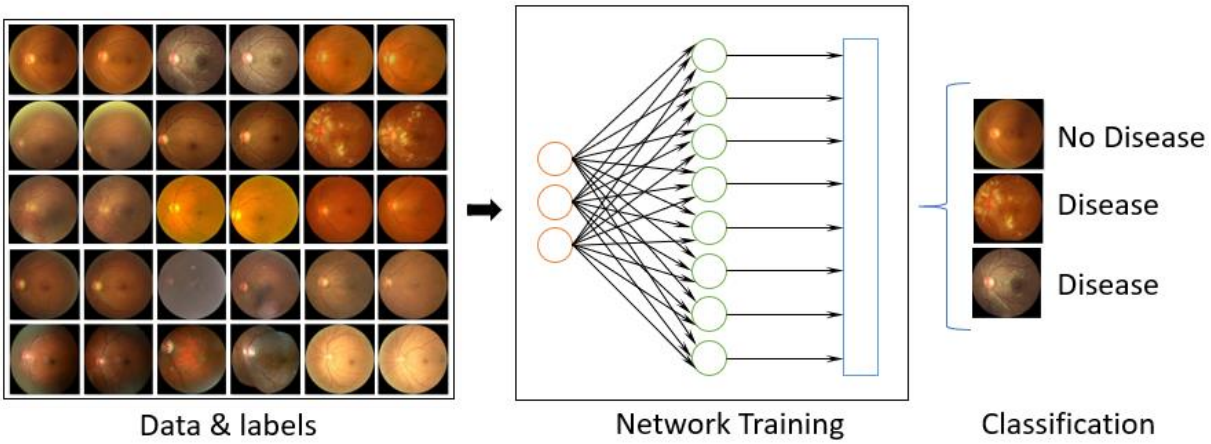


Figura 4. Classificació de malalties oculars mitjançant CNNs.

El conjunt de dades que es farà servir és el proporcionat pel repte ODIR (Ocular Disease Intelligent Recognition) de la Universitat de Pequín i que conté unes 5000 imatges estructurades de pacients amb un conjunt d'anotacions on es defineixen les diferents malalties de cada ull.

La metodologia a utilitzar es basa en l'experimentació dels diferents models, canviant i ajustant diferents paràmetres per a trobar la combinació que ens proporciona la millor precisió en la classificació.

Es presenta en la Figura 5, el diagrama de flux descrivint en detall les diferents parts que componen el nostre treball.

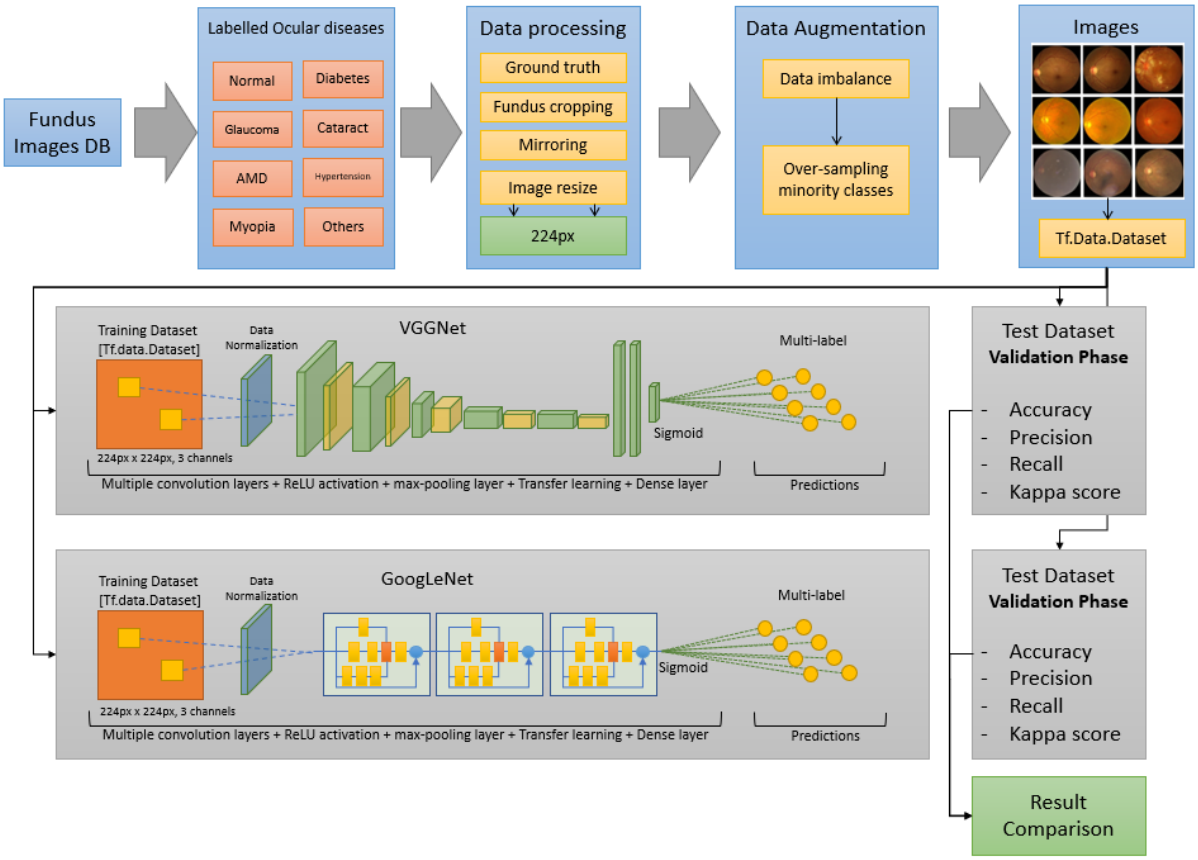


Figura 5. Representació del diagrama de flux

El diagrama de flux mostra l'estudi de les imatges del dataset i la seva descomposició en diferents patologies. Llavors es fa una anàlisi exhaustiva de les imatges i es generen una sèrie

d’algorismes per al seu tractament i que cada xarxa és capaç d’acceptar. S’afegeix també un mòdul d’augment de dades per compensar el desequilibri trobat en el conjunt de dades. Es generen vectors de dades que cada model pot consumir i s’entrenen les xarxes d’aprenentatge profund per a proporcionar la classificació d’imatges en 8 grups. La mida de les imatges i els blocs de conversió poden variar en diferents aplicacions. Les capes convolucionals intenten extreure funcions d’imatge rellevants alhora que redueixen la seva dimensionalitat. La capa Sigmoid és la responsable de l’aspecte de la presa de decisions de la xarxa.

3 Justificació i motivació del TFM

En els darrers anys, hem vist un augment notable en la manera en què els ordinadors entenen el món al seu voltant. Avui dia, existeixen cotxes capaços de ser conduïts per si mateixos amb l'agilitat suficient per a detectar objectes i obstacles i evitar-los (Lyana N. M. 2015) i telèfons mòbils que són capaços de detectar rostres (Sermanet, 2013). Els darrers avenços en intel·ligència artificial (sobretot en el camp de l'aprenentatge profund) han fet possible aquestes tecnologies. L'aprenentatge profund està basat en xarxes neuronals, un tipus d'estructura de dades inspirada en les nostres neurones. Les xarxes de neurones, estan organitzades en capes amb entrades i sortides on una capa prèvia es connectaria a l'entrada i la sortida es connectaria a una capa següent, creant un circuit complex.

Diversos científics han experimentat amb les xarxes neuronals, fins que l'any 2012 es fa realitzar un avenç molt important. Aquest avenç ens va portar a l'anomenada revolució de l'aprenentatge profund on es va descobrir que es podia obtenir un rendiment elevat si en comptes d'usar xarxes neuronals simples (amb poques capes), s'utilitzaven xarxes neuronals amb moltíssimes capes. El descobriment va ser possible amb la potència que els ordinadors oferien l'any 2012. Els investigadors Alex Krizhevsky, Ilya Sutskever i Geoffrey E. Hinton de la Universitat de Toronto van publicar un paper anomenat "ImageNet Classification with Deep Convolutional Neuronal Networks" que ha revolucionat les xarxes neuronals a través de xarxes neuronals convolucionals profundes on se suggereix que poden obtenir rendiments elevats si es combinen amb una potència de computació elevada i un conjunt de dades gran. La xarxa anomenada AlexNet va ser la guanyadora de la competició ILSVRC del 2012 amb una ràtio d'error del 16% sobre el top-5 (Krizhevsky, 2012).

Les xarxes neuronals convolucionals s'han tornat molt populars en la classificació d'imatges mèdiques a causa dels seus darrers avenços tecnològics i sobretot per la manera en què són capaces de detectar característiques específiques en les imatges. Les xarxes neuronals són capaces de fer prediccions aprenent les diferents relacions que tenen les característiques de les nostres dades, és a dir, cada capa de la xarxa actua com a filtre de detecció buscant una característica en particular i altres patrons abstractes que potser passarien desapercebuts per l'ull humà. D'aquí la meva motivació per a la realització d'una solució a un problema d'imatge mèdica com és el de l'anàlisi del fons d'ull.

La solució proposada estudia dues solucions de deep learning que formen part de l'estat de l'art i que ens serveixen per a resoldre un problema de visió per computador analitzant imatges de fons d'ull, deixant que les xarxes neuronals realitzin la detecció de característiques per a identificar patologies oculars. És doncs, un projecte molt elegant, que busca un equilibri entre l'ús de les darreres tecnologies relacionades amb aprenentatge computacional com és TensorFlow i Keras, l'ús de computació estàndard d'usuari amb una GPU genèrica de mercat, i finalment, la resolució d'un problema mèdic que pot donar peu a darrers treballs a partir dels descobriments realitzats en el nostre projecte.

4 Objectius del TFM

Els objectius del treball es poden resumir en:

- Analitzar dues solucions basades en aprenentatge profund per a la classificació de patologies oculars utilitzant una base de dades pública amb imatges de fons d'ull. La primera, Inception (GoogLeNet) amb un classificador ajustat de densa connexió i la segona, VGG-16 (VGGNet) amb un model pretractat i amb ajustament de densa connexió a la capa final. Tots dos configurats per a solucionar un problema mèdic amb múltiples etiquetes.
- Avaluar diferents mètodes de fine-tuning per a trobar una solució òptima.
- Documentar els experiments mostrar els seus resultats.

4.1 Objectiu General

El següent projecte se centrarà en els següents objectius generals:

- Revisar literatura sobre l'aprenentatge profund.
- Revisar literatura sobre anàlisi d'imatges i sobre retinografia.
- Revisar literatura sobre l'anatomia de l'ull i les seves característiques i malalties més comunes.
- Revisar i estudiar treballs relacionats en l'entorn d'aprenentatge profund.
- Estudiar la plataforma TensorFlow i fer els diferents tutorials per a la familiarització de la llibreria i de com realitzar aplicacions d'aprenentatge de màquina.
- Estudiar les diferents configuracions per a l'execució mitjançant GPU.
- Analitzar les característiques de maquinari necessari per a l'execució de problemes d'aquesta complexitat.

- Analitzar les dades de l’ODIR-5K, esbrinar com processar-les, amb quina estratègia i finalment fer la transformació final a vectors per al tractament directe a TensorFlow.
- Generació d'un subconjunt d'imatges de test.
- Analitzar el problema d’etiquetes múltiples i els diferents canvis a realitzar als models per a poder-les suportar.
- Estudiar diferents xarxes convolucionals de la literatura sobre classificació d’imatges mitjançant Deep Convolutional Neural Networks.
- Analitzar l’extracció de característiques.
- Disseny de dos prototips d’aplicació d’aprenentatge profund de la literatura relacionada i comparar-les en un únic dataset (Inception, VGG-16).
- Generació d’algorismes d'augment de dades.
- Anàlisi d’aprenentatge transferit per Inception.
- Anàlisi d’aprenentatge transferit per VGG-16
- Anàlisi de la funció Kappa Score per a l'avaluació final de la categorització d'imatges.
- Desenvolupament de diferents classes per al tractament d'imatges en Python.
- Desenvolupament d'aplicacions addicionals per a la correcta categorització d'imatges.
- Desenvolupament d'aplicacions per a l'anàlisi de resultats.
- Processament del problema concret, analitzar les dades del repte i obtenir una classificació final.
- Estudi de resultats.
- Documentació, memòria de projecte i presentació.

4.2 Objectiu Específic

Els objectius específics vénen definits per la generació de les CNNs:

- Examinar la possibilitat d'usar VGG-16 prèviament entrenat i usar entrenament transferit per a la categorització d'imatges de domini no mèdic a mèdic.
- Examinar la possibilitat d'usar Inception per a la categorització d'imatges de domini mèdic.
- Adaptar els models per a l'especificació del nostre problema incloent la clapa classificadora densament connectada (dense layer) i el mètode de pèrdua a entropia binària creuada (binary cross-entropy).
- Entrenar les xarxes usant tota la potència de maquinari disponible en una sola màquina.
- Estudiar els paràmetres d'ajust (learning curves, .fine tuning, confusion matrix, etc.)
- Utilitzar les dades finals de classificació per fer la comparació amb altres participants del repte ODIR-2019.
- Preparar format de lliurables per a l'estudi via inferència.
- Estudi de resultats i conclusions.

Un cop tinguem els resultats, analitzarem els valors obtinguts i especificarem on hem obtingut una millor classificació en els nostres experiments.

5 Estructuració del treball

El projecte estableix una planificació acurada d'una sèrie de tasques a assolir. Els següents apartats mostren la planificació i execució del treball definint una sèrie de tasques i objectius a aconseguir durant el període que durarà el projecte així com la càrrega que ens aportarà. També es mostrarà el diagrama de planificació temporal amb les diferents tasques.

5.1 Planificació i execució del treball

Es detalla a continuació la relació de les tasques i la seva planificació temporal.

5.1.1 Relació de tasques

La relació de les tasques del projecte es pot veure en la Taula 1:

Taula 1. Relació de tasques

Task Name	Duration	Start	Finish	Predecessors
Pla de treball	**12 days**	**Mon 07/10/19**	**Tue 22/10/19**	
Revisar literatura sobre aprenentatge profund.	3 days	Mon 07/10/19	Wed 09/10/19	
Revisar literatura sobre anàlisi d'imatges i sobre retinografia.	2 days	Thu 10/10/19	Fri 11/10/19	2
Revisar literatura sobre l'anatomia de l'ull i les seves característiques i malalties més comunes.	2 days	Mon 14/10/19	Tue 15/10/19	3
Revisar i estudiar treballs relacionats en l'entorn d'aprenentatge profund.	5 days	Wed 16/10/19	Tue 22/10/19	4
Estudi i anàlisi de la plataforma	**20 days**	**Wed 23/10/19**	**Tue 19/11/19**	
Estudiar la plataforma TensorFlow i fer els diferents tutorials per a la familiarització de la llibreria i les maneres de com realitzar aplicacions d'aprenentatge de màquina.	5 days	Wed 23/10/19	Tue 29/10/19	5
Estudiar les diferents configuracions per a l'execució mitjançant GPU.	1 day	Wed 30/10/19	Wed 30/10/19	7
Analitzar les característiques de maquinari necessari per a l'execució de problemes d'aquesta envergadura.	1 day	Thu 31/10/19	Thu 31/10/19	8
Analitzar les dades de l'ODIR-5K, esbrinar com processar-les, amb quina estratègia i finalment fer la transformació final a TFRecords per al tractament directe a TensorFlow.	1 day	Fri 01/11/19	Fri 01/11/19	9
Canvi d'estructura de dades de TFRecord a vector.	5 days	Mon 04/11/19	Fri 08/11/19	10
Migració de tensorFlow 1.14 a TensorFlow 2.0.	2 days	Mon 11/11/19	Tue 12/11/19	11

Generació d'un subconjunt d'imatges de test.	3 days	Wed 13/11/19	Fri 15/11/19	12
Problema multi-label.	2 days	Mon 18/11/19	Tue 19/11/19	13
Implementació	**31 days?**	**Wed 20/11/19**	**Wed 01/01/20**	
Estudiar diferents xarxes convolucionals de la literatura sobre classificació d'imatges mitjançant Deep Convolutional Neural Networks.	5 days	Wed 20/11/19	Tue 26/11/19	14
Analitzar l'extracció de característiques	2 days	Wed 27/11/19	Thu 28/11/19	16
Disseny de dos prototips d'aplicació d'aprenentatge profund de la literatura relacionada I comparar-les en un únic dataset (Inception, VGG-16).	**18 days?**	**Fri 29/11/19**	**Tue 24/12/19**	
Disseny de dos prototips d'aplicació d'aprenentatge profund de la literatura relacionada I comparar-les en un únic dataset (Inception, VGG-16).	8 days	Fri 29/11/19	Tue 10/12/19	17
Generació d'augment de dades.	2 days	Wed 11/12/19	Thu 12/12/19	19
Anàlisi de transfer learning VGG-16.	2 days	Fri 13/12/19	Mon 16/12/19	20
Anàlisi de la funció Kappa Score per a l'avaluació final de la categorització d'imatges.	1 day?	Tue 17/12/19	Tue 17/12/19	21
Desenvolupament de diferents classes per al tractament d'imatges en Python.	1 day	Wed 18/12/19	Wed 18/12/19	22
Desenvolupament d'aplicacions addicionals per a la correcta categorització d'imatges.	2 days	Thu 19/12/19	Fri 20/12/19	23
Desenvolupament d'aplicacions per a l'anàlisi de resultats.	2 days	Mon 23/12/19	Tue 24/12/19	24
Processament del problema concret, analitzar les dades del repte i obtenir una classificació final.	3 days	Wed 25/12/19	Fri 27/12/19	25
Estudi de resultats.	3 days	Mon 30/12/19	Wed 01/01/20	26
Confecció d'entregues	**6 days**	**Thu 02/01/20**	**Thu 09/01/20**	
Documentació i memòria de projecte.	3 days	Thu 02/01/20	Mon 06/01/20	27
Presentació.	3 days	Tue 07/01/20	Thu 09/01/20	29
Fi de Projecte.	0 days	Thu 09/01/20	Thu 09/01/20	30

5.1.2 Planificació temporal

La planificació temporal segons les tasques definides en el punt anterior es pot veure en la Figura 6.

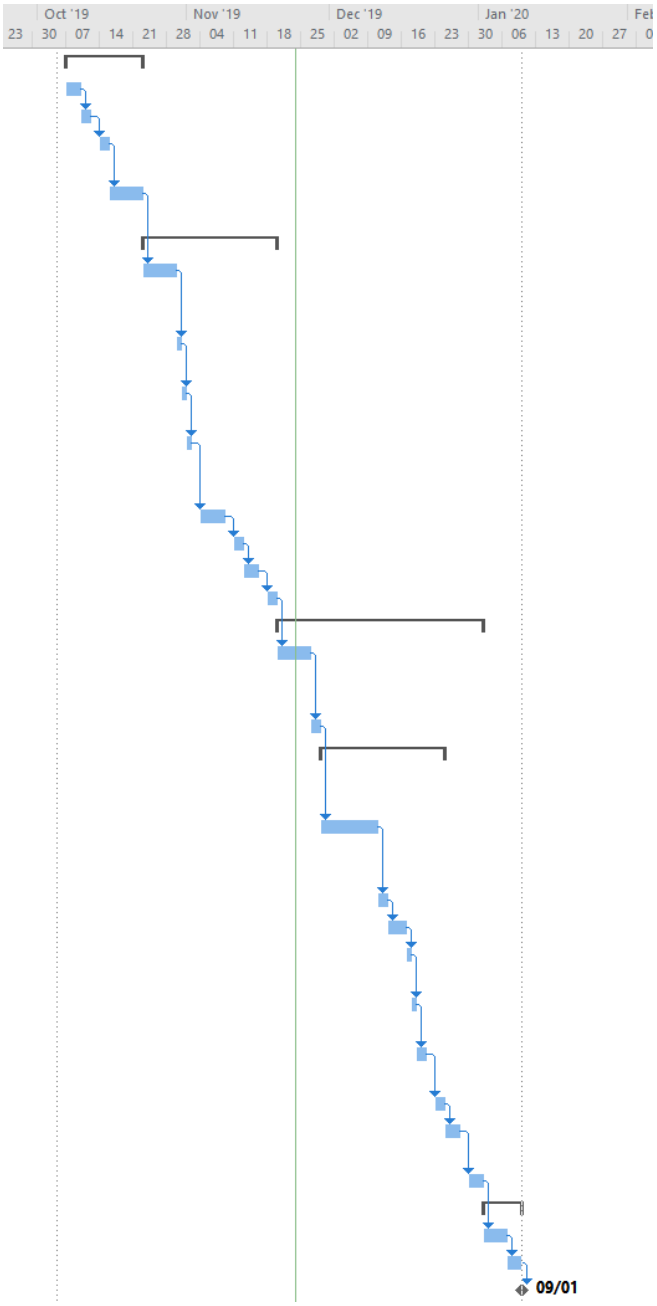


Figura 6. Diagrama de planificació temporal

La planificació és subjecte a modificacions durant la vida del projecte a causa de la manca d'experiència prèvia en tipus de treballs similars.

6 El conjunt de dades

ODIR-2019 és un dataset públic amb més de 5 mil imatges de fons d'ull (ull esquerre i dret) d'alta resolució dividides en 8 categories. Les imatges de pacients reals, són capturades mitjançant diferents càmeres de mercat (Canon, Zeiss, Kowa, etc.) per l'empresa Shanggong Medical Technology Co. Ltd., a través hospitals i centres mèdics de la Xina. Les fotografies s'han fet públiques a través del repte ODIR-2019 (Ocular Disease Intelligent Recognition, 2019) de la Universitat de Pequín.

El conjunt de dades conté un fitxer amb anotacions on s'etiqueten les imatges amb les paraules clau correctes que defineixen cada malaltia. Els especialistes anoten els pacients amb 8 etiquetes diferents que identifiquen les següents patologies: N, D, G, C, A, H, M i O. Les diferents categories són:

- [0] N: Normal. Sense cap malaltia.
- [1] D: Diabetis.
- [2] G: Glaucoma.
- [3] C: Cataracta.
- [4] A: AMD. Degeneració de la màcula.
- [5] H: Hipertensió.
- [6] M: Miopia.
- [7] O: Altres anormalitats.

Cada pacient pot estar marcat amb una o més etiquetes indicant que pot patir més d'una malaltia a la vegada. El dataset a més a més, conté informació bàsica com el sexe del pacient, la seva edat, les etiquetes de la seva patologia i algunes paraules clau que defineixen millor les

etiquetes. Seguidament podem trobar informació rellevant sobre 5 aspectes del conjunt de dades en el següent apartat:

- 1) la seva edat (Figura 7) on s'observa que el conjunt d'imatges més elevat pertany al grup de pacients d'entre quaranta-set a seixanta-vuit anys.

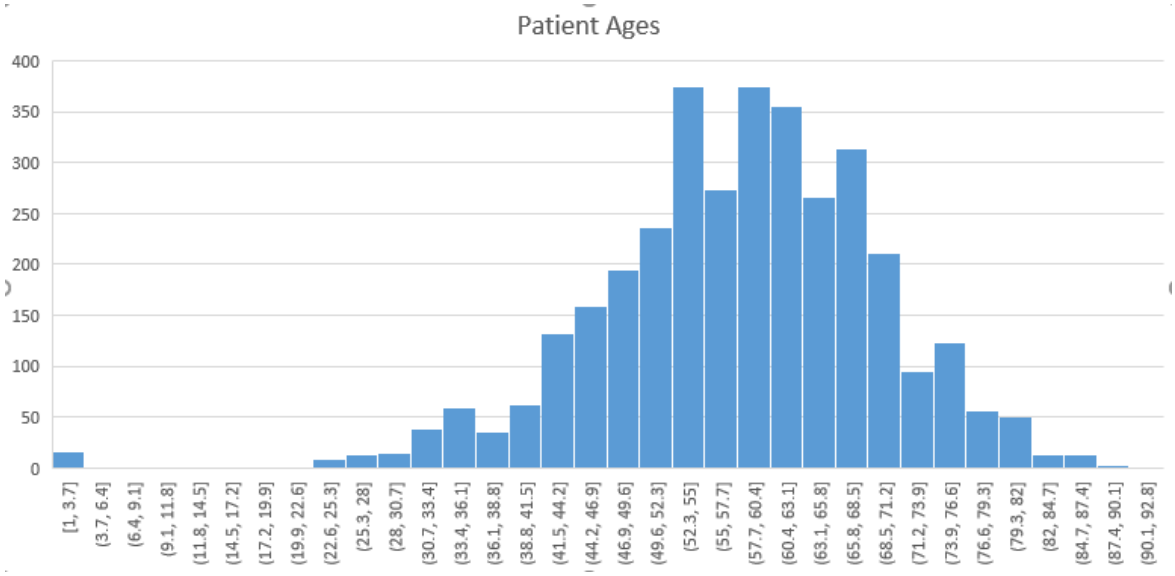


Figura 7. Histograma d'edat dels pacients

- 2) el sexe del pacient: Home o Dona.
- 3) la classificació final en una o més categories (Figura 8)

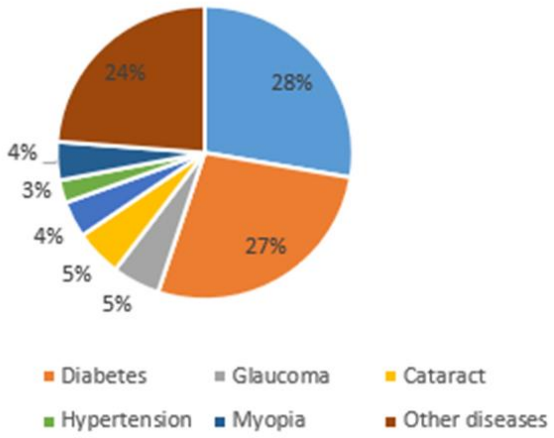


Figura 8. Classificació de les diferents malalties

- 4) retinografia d'ulls esquerre i dret en diverses mides i resolucions
- 5) paraules clau del diagnòstic fet per diferents especialistes

Un exemple de les dades proveïdes a la base de dades es pot observar en la Figura 9.

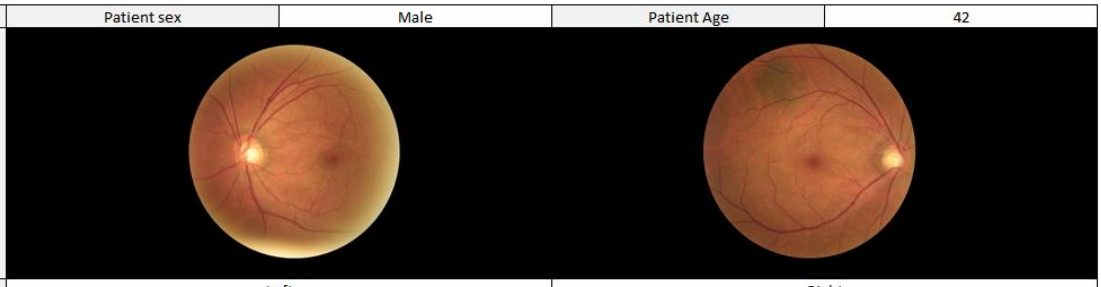
Basic Info	Patient sex		Male		Patient Age		42	
Fundus Images								
Literality	Left				Right			
Disease Labels	Normal (N)	Diabetes (D)	Glaucoma (G)	Cataract (C)	AMD(A)	Hypertension (H)	Myopia (M)	other diseases (O)
	0	1	0	0	0	0	0	1
Diagnostic Keywords	mild nonproliferative retinopathy				intraretinal microvascular abnormality			

Figura 9. Exemple de registre oftàlmic del conjunt de dades

6.1 El fitxer d'anotacions especials

El fitxer d'anotacions especials es pot trobar a l'apartat 11.2 de l'annex. El document defineix certs aspectes que hem de considerar a l'hora de processar les imatges del dataset. Cada imatge és etiquetada amb un valor clau de diagnòstic amb el qual aconseguirem definir el conjunt de dades que formen el ground truth.

Les anotacions s'han de tenir en compte perquè definiran el conjunt de dades final que utilitzarem. Seguidament podem trobar la descripció de cada anotació:

- Una imatge és classificada com a (N) Normal si la paraula clau "normal fundus" forma part del grup de paraules clau definida a la cel·la. La columna (N) només està marcada com a 1 si els dos ulls estan classificats com a Normal. Per tant, a l'hora de generar el ground truth, ens caldrà analitzar cada cel·la de les paraules clau i veure si "normal fundus" existeix allí.

- La resta de malalties o anormalitats es poden considerar correctament marcades al dataset.
- Les imatges marcades amb "*anterior segment image*" i "*no fundus image*" es poden descartar donat que no pertanyen a cap de les categories mencionades anteriorment.
- Les imatges marcades amb "*lens dust*", "*optic disk photographically invisible*", "*low image quality*" i "*image offset*" es poden descartar donat que no ens ajuden a determinar la malaltia en si.
- Trobem un llistat d'imatges definides on el color de fons és diferent de la resta i es poden descartar. La llista sencera es pot trobar a l'apartat 11.2 de l'annex adjunt a la memòria.

6.2 Imatges de validació

El dataset públic ODIR, ofereix un conjunt de 1000 imatges de validació que es poden fer servir per a comprovar la precisió que generen els nostres models. El conjunt de validació encara no està anotat amb els resultats de la classificació perquè el repte és actiu i diferents participants de tot el món encara envien els seus resultats a través de la pàgina web disponible. Per a solucionar el problema, definim un conjunt addicional de dades que hem creat específicament per a l'entrenament dels nostres models. La següent Figura 10 mostra els diferents grups d'imatges i la seva divisió final:

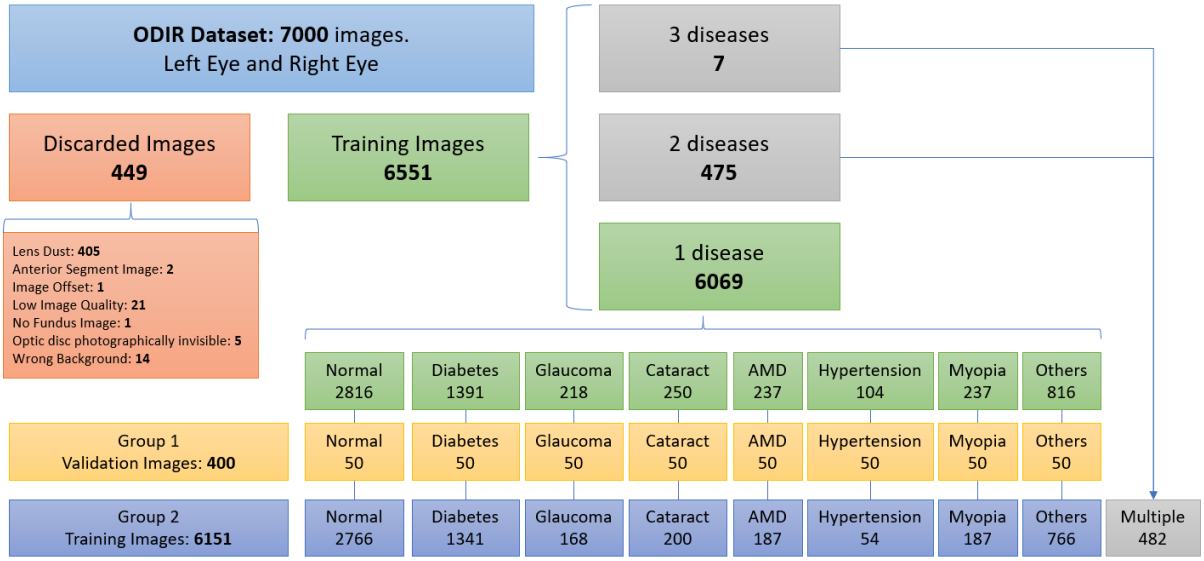


Figura 10. Representació dels grups d'imatges (entrenament i validació)

Tenim un total de 7000 imatges, 3500 imatges d'ull esquerre i 3500 imatges d'ull dret. Se'n descarten 449 pels diferents motius expressats en el fitxer d'anotacions especials perquè les imatges estan marcades amb certes paraules clau que indiquen que la imatge no és vàlida. El llistat sencer de les imatges rebutjades es pot trobar a la secció 11.4 de l'annex.

Això ens deixa amb 6551 imatges que podem utilitzar en el nostre procés. 6069 imatges estan etiquetades amb 1 de les opcions possibles, mentre que la resta està etiquetada amb 2 o 3 patologies.

Per a la composició del conjunt de dades de validació, utilitzem imatges amb 1 etiqueta i escollim un grup petit de 50 imatges per categoria que formaran el conjunt total de 400 imatges a avaluar. La resta d'imatges (5669) més les imatges marcades amb múltiple malalties (482) s'utilitzaran com a dades d'entrenament, formant així el grup sencer de 6151 imatges amb les quals podrem treballar.

Taula 2. Desequilibri de dades

No.	**Labels**	**Training Cases**	**Percentage from total**
0	N	2816	46.40%
1	D	1391	22.92%
2	G	218	3.59%
3	C	250	4.12%
4	A	237	3.91%
5	H	104	1.71%
6	M	237	3.91%
7	O	816	13.45%

La Taula 2 mostra un desequilibri important respecte les classes majoritàries. Les imatges de fons d'ull sense cap patologia representen el 46% del total de les dades, mentre que classes minoritàries com la hipertensió formen un 2% de les dades sobre el conjunt total. Això produeix un repte addicional en termes de desequilibri de dades, i per a balancejar-les, implementarem mecanismes per a garantir que els models puguin aprendre sobre les classes minoritàries (Glaucoma, Cataracta, AMD, Hipertensió i Miopia).

7 Metodologia

7.1 Preprocessament

El conjunt de dades ha de ser tractat de tal manera que tinguem un vector de TensorFlow que puguem utilitzar dins dels nostres models. Triem doncs, les estratègies de preprocessat que ens ajudaran a esbrinar la millor configuració per a l'assoliment de l'objectiu i ser capaços de classificar les imatges amb les diferents etiquetes marcades en el dataset. Definim com a estratègies específiques del conjunt de dades:

- Analitzar les 7000 imatges presents i establir quines imatges no han de pertànyer al conjunt amb les anotacions definides en l'apartat 6.1. El component ens generarà una llista d'imatges rebutjades i que no s'utilitzaran.
- Analitzar el format de les imatges i establir una resolució comuna de 224 píxels. Cada imatge en format comú JPEG té una resolució i mida diferent i les CNNs necessiten una estandardització en l'entrada. D'altra banda, la decisió de la resolució comuna bé definida per la recerca feta pel que fa a la mida de la imatge que accepten els models estudiats.
- Associar l'etiqueta de l'ull a través de l'exploració dels diferents comentaris creats per cada especialista. Analitzarem cada paraula clau i etiquetarem automàticament cada ull i després farem la comparació amb el vector original per comprovar que les dades no han variat després de fer la divisió entre ulls.
- Analitzar el procés de mirall que es pot fer a cada ull per a tenir un únic punt de vista i així evitar entrenar cada ull per separat. Tot i que és important la distinció de cada ull, tal com demostren Tan et al. (2009) en el seu paper "*Automatic Detection of Left and Right Eye in Retinal Fundus Images*", en el nostre treball es consideren imatges d'ulls sense definir explícitament si és dret o esquerre.

- D'aquí obtindrem el preprocessament de les dades obtenint les imatges de l'entrenament i les etiquetes associades a cada imatge.

7.1.1 Anàlisi del dataset i generació del ground truth

El principal repte que trobem en analitzar el conjunt de dades proporcionat, és que no veiem distinció sobre quines dades pertanyen a l'ull esquerre i a l'ull dret. Mitjançant el fitxer d'anotacions podem arribar a construir un algorisme capaç de dividir les dades correctament per a cada ull i així generar un fitxer de ground truth vàlid per a la nostra estratègia.

Com es mostra en l'anterior Figura 10, es descarten 449 imatges del conjunt original. Seguidament fem un comentari sobre totes les imatges descartades i la raó del seu rebuig.

Tenim 7 categories de rebuig i l'algorisme d'identificació del ground truth que acompanya el nostre treball, és capaç de categoritzar les imatges correctament mitjançant les paraules clau de cada ull i fa una validació exacta comparant-les amb el vector de malalties proporcionat en el dataset. El fitxer amb els vectors originals es pot trobar a l'apartat 11.1 de l'annex.

Les categories per considerar rebuig són les següents:

- Lens Dust
- Anterior Segment Image
- Image Offset
- Low Image Quality
- No Fundus Image
- Optic Disc Photographically Invisible
- Wrong Background

En la secció 11.4 de l'annex es pot trobar el llistat complet de les imatges rebutjades amb la categoria associada a cada rebuig.

Com que algunes de les imatges han sigut rebutjades, l'algorisme del ground truth ha modificat els vectors que mostra la Taula 3 per a donar un valor adient a la imatge segons les seves característiques (paraules clau). La sortida generada per l'algorisme ens permetrà, doncs, poder usar imatges de manera individual i amb el seu ground truth corresponent.

Taula 3. Ground truth per a imatges específiques

ID	Fundus	Diagnostic	Provided Vector	New Vector
65	65_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
122	122_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
124	124_left.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
183	183_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
251	251_left.jpg	moderate non proliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
380	380_left.jpg	mild nonproliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
438	438_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
465	465_right.jpg	drusen	[0,0,0,1,0,0,0,1]	[0,0,0,0,0,0,0,1]
470	470_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
558	558_left.jpg	mild nonproliferative retinopathy, mild nonproliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
594	594_right.jpg	moderate non proliferative retinopathy	[0,1,0,1,0,0,0,0]	[0,1,0,0,0,0,0,0]
664	664_right.jpg	drusen	[0,1,0,0,0,0,0,1]	[0,0,0,0,0,0,0,1]
671	671_left.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
679	679_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
698	698_left.jpg	moderate non proliferative retinopathy	[0,1,0,0,0,0,1,0]	[0,1,0,0,0,0,0,0]
802	802_right.jpg	moderate non proliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
817	817_left.jpg	mild nonproliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
842	842_left.jpg	mild nonproliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
866	866_left.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
875	875_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]

991	991_right.jpg	mild nonproliferative retinopathy	[0,1,0,0,0,0,0,1]	[0,1,0,0,0,0,0,0]
1096	1096_left.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
1121	1121_right.jpg	normal fundus	[0,0,0,0,0,0,0,1]	[1,0,0,0,0,0,0,0]
1409	1409_left.jpg	mild nonproliferative retinopathy	[0,1,1,0,0,0,0,0]	[0,1,0,0,0,0,0,0]
1657	1657_right.jpg	mild nonproliferative retinopathy	[0,1,0,0,0,0,1,0]	[0,1,0,0,0,0,0,0]
2074	2074_right.jpg	cataract, suspected cataract	[0,0,1,1,0,0,0,0]	[0,0,0,1,0,0,0,0]
2119	2119_left.jpg	normal fundus	[0,0,0,1,0,0,0,0]	[1,0,0,0,0,0,0,0]
2174	2174_left.jpg	normal fundus	[0,0,0,1,0,0,0,0]	[1,0,0,0,0,0,0,0]
2175	2175_right.jpg	normal fundus	[0,0,0,1,0,0,0,0]	[1,0,0,0,0,0,0,0]
2222	2222_right.jpg	epiretinal membrane	[0,0,0,1,0,0,0,1]	0,0,0,0,0,0,0,1]

La Taula 3 es llegeix de la següent manera. Si mirem la primera fila, indiquem que la figura de l'ull dret del pacient 65 no té cap anomalia i que l'ull esquerre s'ha descartat tal com es mostra en la sortida de l'script que es pot trobar en la secció 11.5 de l'annex. A causa d'aquesta diferència, el vector original no té sentit i hem de generar un nou vector utilitzant les paraules clau. En aquest cas, el vector original només marcava amb 1 la posició d'altres malalties que correspon a l'ull esquerre, però com que ara només tenim una sola imatge, hem de marcar la primera posició del vector amb un 1 per a indicar que és una imatge d'un ull normal.

7.1.2 Tractament de les imatges

Si observem les imatges d'entrenament com es mostra en la següent Figura 11, ens trobem amb certes dificultats:

- Totes les imatges tenen una mida i resolució diferents perquè provenen de diferent maquinari i fonts.
- Tenim imatges que no tenen una relació d'aspecte 1:1 i afegeixen un fons negre addicional a la imatge.
- Tenim imatges on el disc òptic no està centrat correctament.

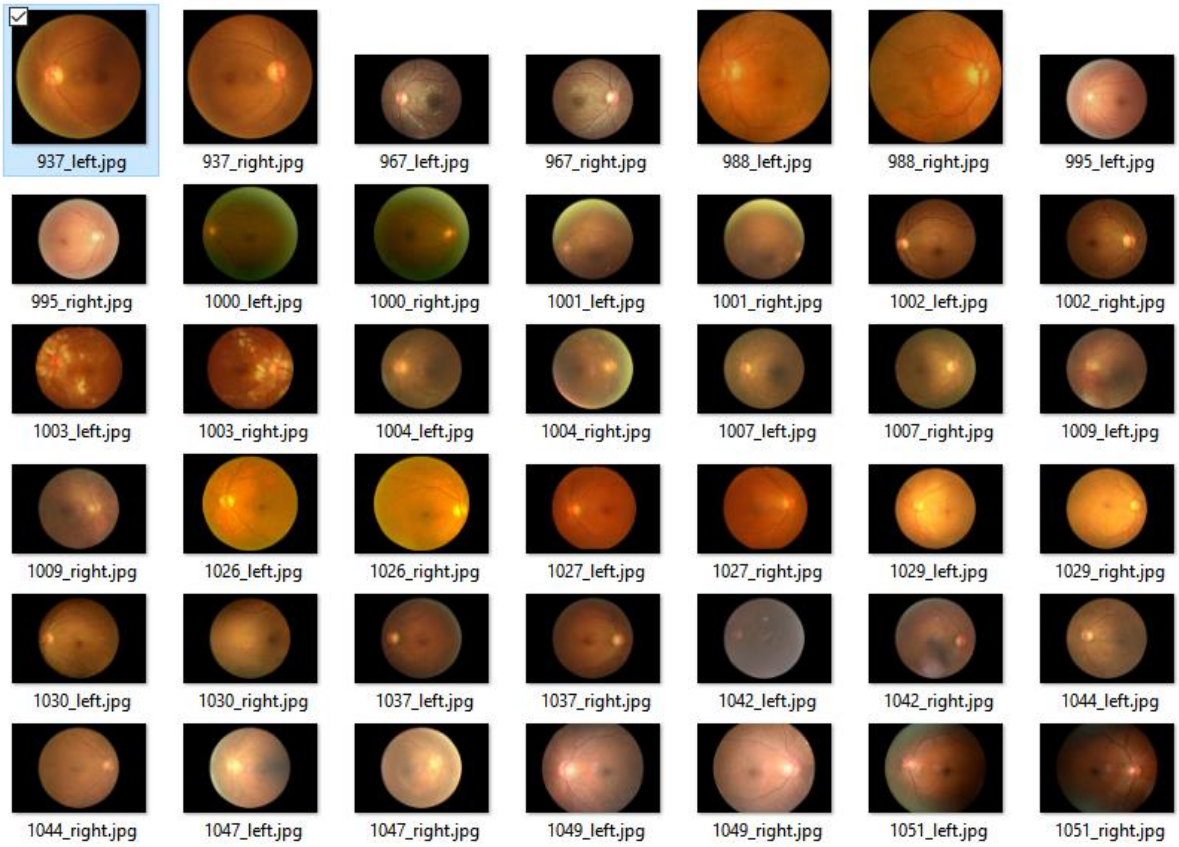


Figura 11. Exemple d’imatges de prova amb diferents formats

Seguidament podem trobar els diferents tractaments realitzats sobre les imatges per a aconseguir un format més uniforme. Com que la majoria de xarxes neuronals requereix una imatge amb una relació d’aspecte 1:1, hem de modificar-les lleugerament per a poder treballar correctament amb elles. Els següents passos mostren quines transformacions hem realitzat per a aconseguir el format d’imatge final que mostra la Figura 12.

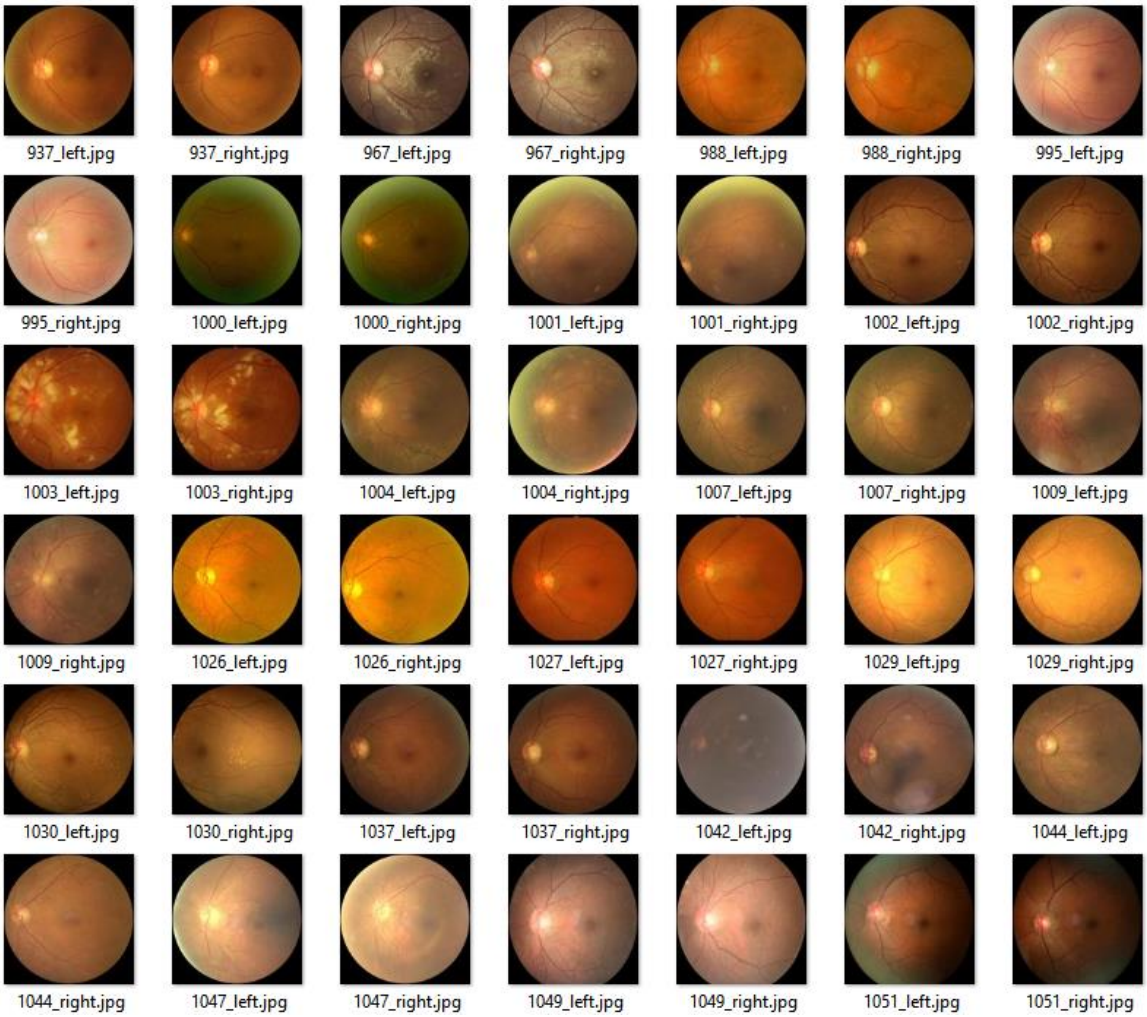


Figura 12. Llista d'imatges tractades

Com podem observar, les imatges estan correctament centrades, mostrant l'àrea que ens interessa i reduint el soroll addicional que el fons negre introdueix a la imatge. Amb aquest conjunt de dades és amb el que podem començar a treballar. Seguidament definim els passos realitzats a cada imatge.

Pas 1. Fer el retall (crop) de les imatges.

Es crea un algorisme que és capaç de retallar les imatges tal com es mostra en la Figura 13:

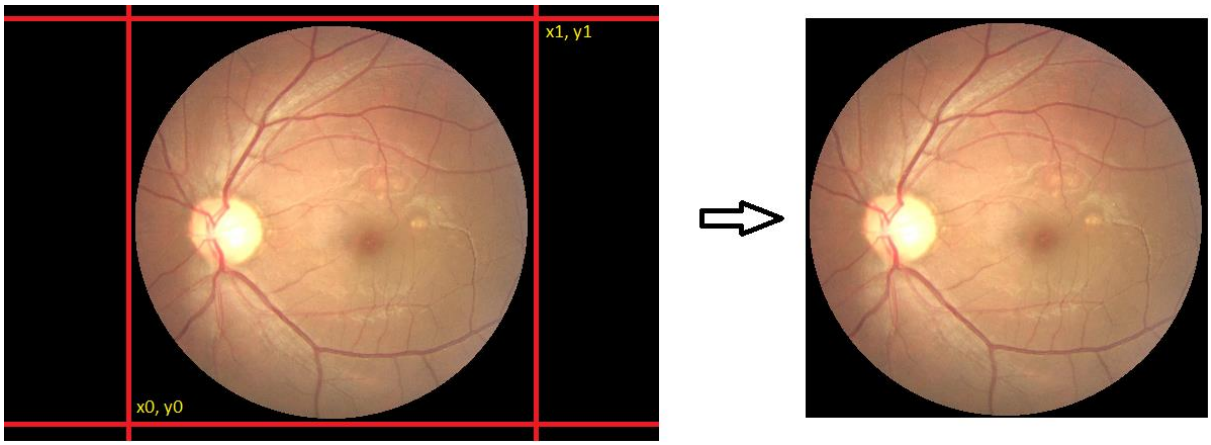


Figura 13. Algorisme de retall d'imatges

El nostre algorisme ens ajudarà a fer que les imatges tinguin una relació d'aspecte més adient. Usem la llibreria OpenCV de Python que ens permet carregar la imatge com a un vector de píxels. Un cop tenim el vector, ubiquem tots els píxels de color negre i després fem la intersecció de les coordenades (x0, y0), (x1,y1) amb els píxels de color per a obtenir el retall d'imatge que volem.

Pas 2. Fer l'efecte de mirall (mirroring) de les Imatges.

Com a estratègia específica definida en el nostre treball, es tracta cada imatge de cada pacient com una imatge individual. Per a fer-ho, es consideren que totes les imatges són d'ulls esquerre i per a realitzar el canvi, construïm un algorisme de tractament d'imatges que fa l'efecte de mirall de la imatge un cop es troba una imatge d'un ull dret. El procediment es pot veure representat en la Figura 14.

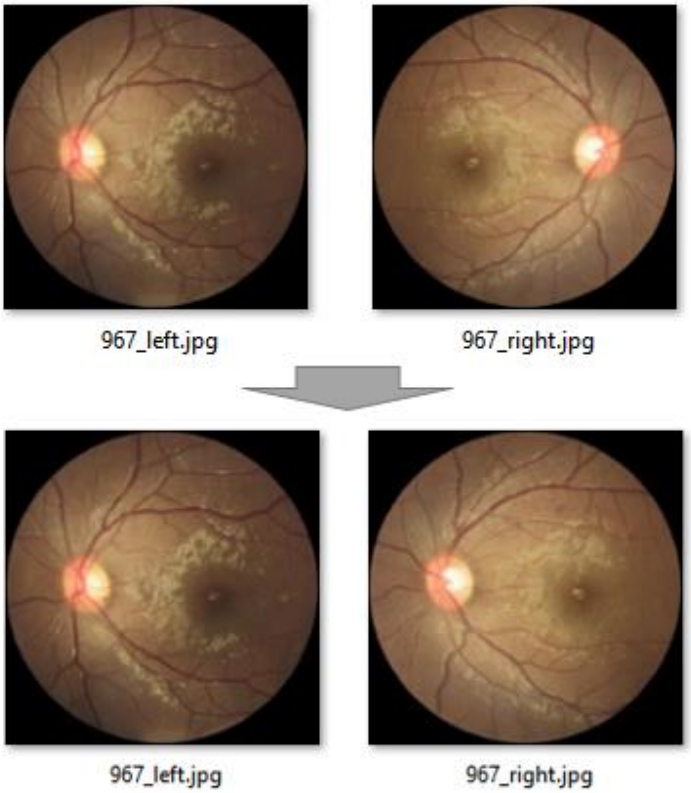


Figura 14. Efecte de mirall de les imatges

Pas 3. Fer el redimensionament (resize) de les imatges.

Les imatges del dataset tenen un format irregular i amb diverses mides i resolucions. Per a poder-les entrar dins els diversos models, les hem d'ajustar a una mida concreta. Hem triat una mida estàndard de 224x224 píxels perquè és un format comú acceptat per diversos models. L'algorisme permet un ajustament dinàmic i ens permet fer canvis addicionals per fer proves per exemple amb imatges de mida de 128 píxels. El redimensionament es pot veure il·lustrat en la Figura 15.

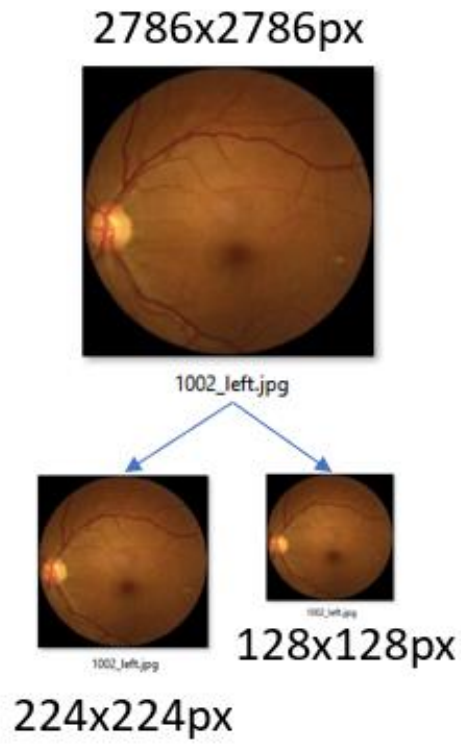


Figura 15. Redimensionat d'imatges

7.1.3 Generació de la seqüenciació d'elements

Donat el grup d'imatges que s'ha processat, ens disposem a generar un component de seqüenciació que ens ajudarà a introduir les diferents imatges en un format homogeni dins un model. El nostre component transforma les imatges en llistes de vector amb un format molt comú que TensorFlow i Keras entenen. L'algorisme que forma part de l'entrega del nostre treball, ens permetrà generar les diferents col·leccions d'imatges en funció de la mida seleccionada (128, 224, etc., píxels) i la quantitat d'imatges que volem extreure com a part validació (100, 200, etc., sobre el conjunt d'entrenament). A més a més, s'incorpora una secció addicional per a poder avaluar també el conjunt de dades de prova que el mateix repte entrega i així comprovar els resultats del nostre model amb el d'altres participants. El següent diagrama representat en la Figura 16 mostra el desglossament d'informació que es pot generar a través de l'script proporcionat:

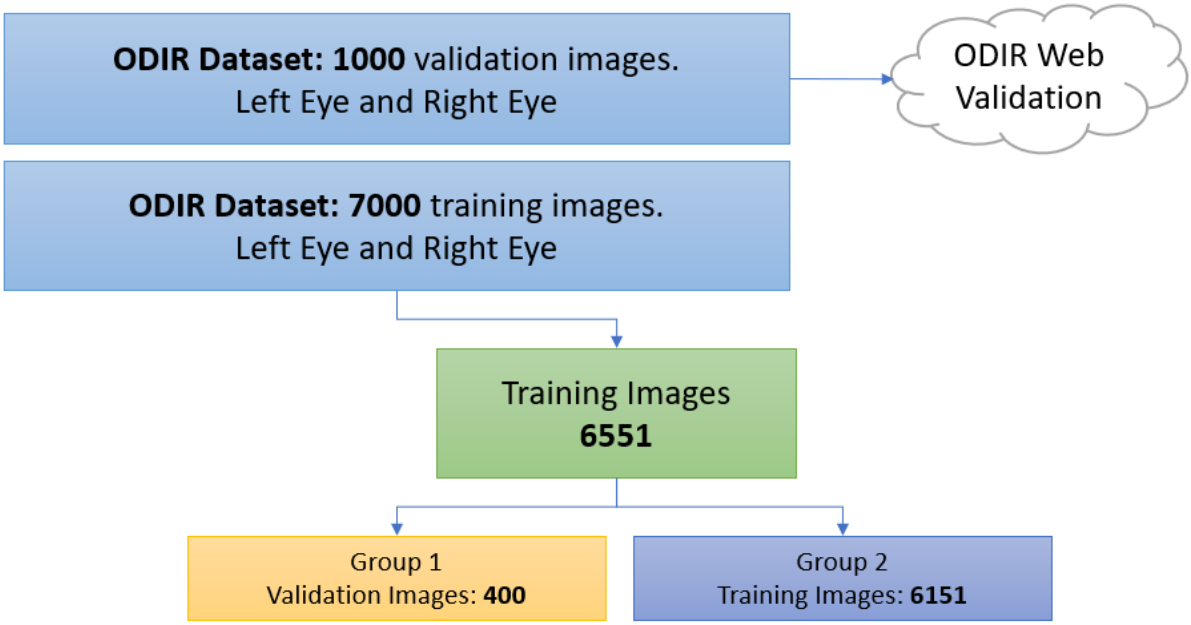


Figura 16. Desglossament de dades

El pas descrit transforma les imatges a vectors i les desa en fitxers '.npy' per al seu tractament posterior. Un cop els contenidors d'imatges s'han generat, podem fer la crida del següent codi per a l'obtenció dels vectors en memòria i que podem afegir a l'entrada de qualsevol model que accepti aquesta mida d'imatge (128 píxels o 224 píxels, per exemple):

```
# Load the data
(x_train, y_train), (x_test, y_test) = odir.load_data(image_size)
```

7.1.4 Augment de dades

És típic en problemes de classificació que totes les classes de les dades no estiguin representades amb un balanç òptim. En el nostre cas, disposem de 6551 imatges desglossades en 8 categories diferents on tenim una classe representada amb un màxim de 192 imatges. Per tant, podem observar un desequilibri important cosa que impacta l'aprenentatge que el nostre model pugui fer sobre la classe minoritària. En els capítols següents, podem observar com diferents models obtenen una fidelitat (accuracy) del 89% però la precisió és del 62%, indicant-nos que el model

no està aprenent realment sobre el problema en qüestió. Això és degut al fet que el model, de manera sistemàtica, decideix que la millor opció és predir les classes amb més dades per aconseguir una major fidelitat.

Les tàctiques que s'han usat per a combatre el desequilibri de dades són les següents:

1) Pes o parcialitat a cada classe de sortida del model

La següent Taula 4, ens mostra en detall les dades agrupades en nombre de malalties per imatge:

Taula 4. Desequilibri de dades

Number of diseases per image	Number of Images	Normal	Diabetes	Glaucoma	Cataract	AMD	Hypertension	Myopia	Others
1	6069	2816	1391	218	250	237	104	237	816
2	475	0	380	92	22	42	88	25	301
3	7	0	7	3	3	1	0	0	7
Total	6551	2816	1778	313	275	280	192	262	1124

Sent "Normal" la classe majoritària, podem calcular els pesos per la resta de classes fent una simple operació matemàtica, dividint la classe majoritària per la classe minoritària individual i obtenir un pes que utilitzarem a l'hora d'entrenar els nostres models. De la Taula 4, podem obtenir doncs la següent distribució representada en la Taula 5:

Taula 5. Càlcul dels pesos per al desequilibri

	Normal	Diabetes	Glaucoma	Cataract	AMD	Hypertension	Myopia	Others
Images	2816	1778	313	275	280	192	262	1124
Weight	1.0000	1.5838	8.9968	10.2400	10.0571	14.6667	10.7481	2.5053

D'aquí obtenim les diferents ponderacions que afegirem durant l'entrenament dels models:

```
class_weight = {0:1.,
                1:1.583802025,
                2:8.996805112,
                3:10.24,
                4:10.05714286,
                5:14.66666667,
                6:10.7480916,
                7:2.505338078}
```

Podem llegir les ponderacions com: “totes les instàncies de la classe 3 són com 10.24 instàncies de la classe 0”. Els pesos penalitzaran la recompensa de la classe majoritària, donant una ponderació més adient i que representa el conjunt de dades que tenim.

2) Mètriques addicionals

Com hem comentat anteriorment, la fidelitat o “accuracy” no és la millor mètrica quan treballem amb un dataset desequilibrat i pot portar-nos a una anàlisi incorrecte dels resultats. Hi ha mètriques que s’han dissenyat més específicament per a treballar amb desequilibri de dades. Les diferents mètriques i el seu corresponent atribut de TensorFlow Keras (si existeix) es pot trobar a la Taula 6.

Taula 6. Mètriques utilitzades

Metric	tf.Keras.Metric	Manual Calculation	Description
Precision	tf.keras.metrics.Precision	No	A measure of a classifier exactness
Recall	tf.keras.metrics.Recall	No	A measure of a classifier completeness
Kappa	n/a	Yes	Classification accuracy normalized by the imbalance of the classes in the data
F1-Score	n/a	Yes	A weighted average of precision and Recall
AUC	tf.keras.metrics.AUC	No	A measure of the area under the curve via a Riemann sum
Confusion Matrix	n/a	Yes	Breakdown prediction into a table with correct predictions in the diagonal

Com que el problema que tenim entre mans, és un problema amb múltiples etiquetes, és a dir, disposem d’imatges marcades amb múltiples patologies, els organitzadors del repte ODIR han publicat un algorisme que permet tenir en compte la complexitat sobre les diferències i així donar-nos una mètrica més adient sobre la classificació obtinguda. Els avaluadors que l’algorisme utilitza són el *coeficient Kappa de Cohen*, el *F1-score* i el *AUC* (Àrea sota la corba) que es poden usar per a calcular la puntuació final de la classificació de les imatges del concurs.

A més a més, afegim les mètriques de *Precisió*, *Recall* i *matriu de confusió* per a obtenir més dades per a la seva anàlisi i discussió. Tot seguit, introduïm breument cada mètrica:

Coeficient Kappa de Cohen:

El coeficient Kappa de Cohen mesura l'acord entre dos observadors on classifiquen un nombre N d'elements entre un grup C de categories mútuament exclusives. La seva fórmula és:

$$k = \frac{p_o - p_e}{1 - p_e}$$

On P_o és l'acord relatiu observat entre els dos observadors i P_e és la probabilitat hipotètica d'acord per atzar. Si els observadors estan completament d'acord, llavors el coeficient és 1. Si no hi ha acord, llavors cal avaluar l'atzar de les seves decisions (Carletta, 1996)

F1-Score:

El F1-Score, també anomenat valor-F, és un avaluador que mesura la precisió que té un test. La puntuació té en compte el valor de Precisió i Recall (exhaustivitat), i la seva fórmula és: (Van Rijsbergen, 1979):

$$F_1 = 2 \cdot \frac{precision \cdot \ recall}{precision + recall}$$

AUC:

L'àrea sota la corba és igual a la probabilitat que un classificador puntuï un element positiu seleccionat a l'atzar de manera més elevada que un element negatiu seleccionat a l'atzar. Assumint que un element positiu té una major puntuació que un element negatiu. El valor s'utilitza sovint durant l'aprenentatge computacional per a la comparació de models (Fawcett, 2016).

Precisió:

La precisió és la ràtio entre els elements que han sigut classificats correctament sobre una classe determinada (veritable positiu) i el nombre total d'elements de la classe determinada. Per tant si tenim un total de 8 imatges classificades com a normals i 5 han sigut correctament predites (veritable positiu) com a normals, la nostra precisió és de 5/8 (Kent et al., 1955):

$$P = \frac{tp}{tp + fp}$$

On, *tp* denota el veritable positiu i *fp* denota un fals positiu.

Recall:

Recall o exhaustivitat, és la ràtio entre els elements que han sigut classificats correctament sobre una classe determinada (veritable positiu) i el nombre total d'elements que haurien de formar part de la classe determinada (fals negatiu) (Kent et al., 1955).

$$R = \frac{tp}{tp + fn}$$

On, *tp* denota el veritable positiu i *fn* denota un fals negatiu.

Matriu de confusió:

La matriu de confusió ens mostra en forma de taula, el resum de les dades predites envers el seu valor real. La Figura 17, mostra un exemple de matriu de confusió (Fawcett, 2006):

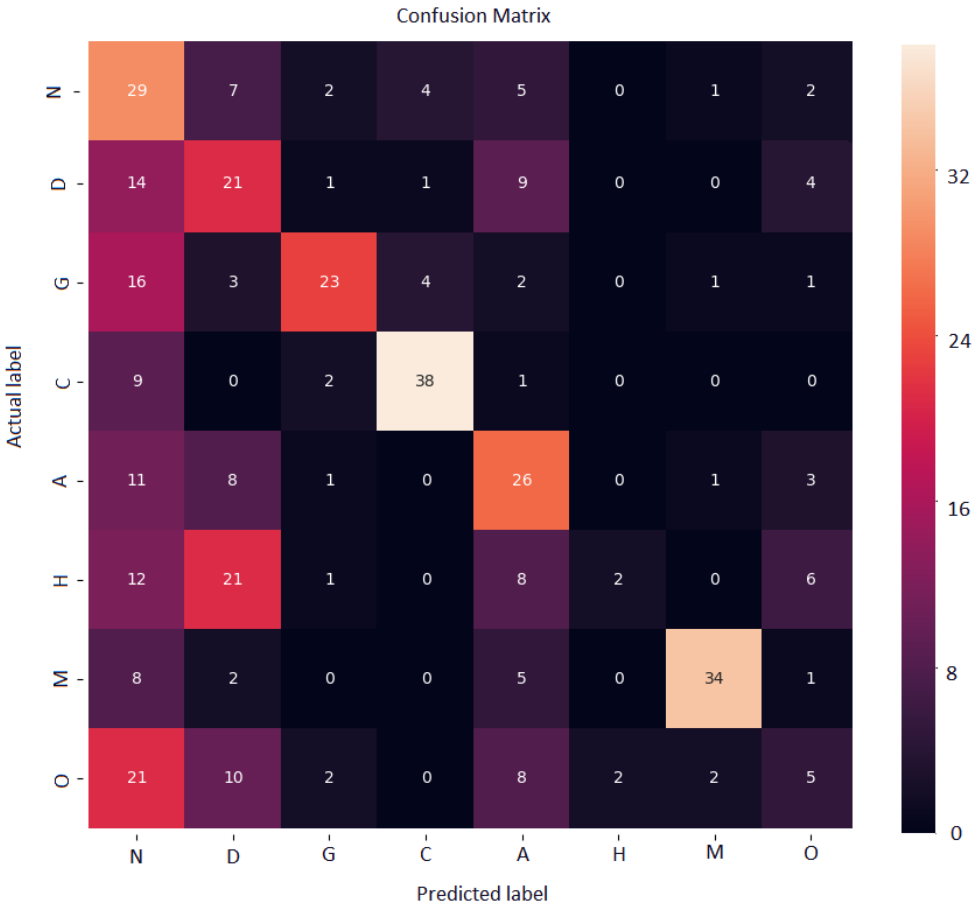


Figura 17. Exemple de matriu de confusió

Les mètriques que hem definit i que els nostres models usen:

```
defined_metrics = [
    tf.keras.metrics.BinaryAccuracy(name='accuracy'),
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall'),
    tf.keras.metrics.AUC(name='AUC'),
]
```

Un resultat d'exemple de sortida de les mètriques utilitzades i els seus valors corresponents és:

```
loss:              0.3193148720264435
accuracy:          0.8903125
precision:         0.62189054
Recall:            0.3125
AUC:               0.8111183

kappa score:       0.36268724466636404
f-1 score:         0.8903125
AUC value:         0.816071875
Final Score:       0.689690539888788
```

El valor de puntuació final és el valor mitjà fet de la suma del *coeficient Kappa de Cohen*, *F-1 score* i *AUC* tal com es mostra en l'equació següent:

$$Final_score = \frac{Kappa + F1 + AUC}{3.0}$$

El càlcul final es realitza de la següent manera tal com suggereixen en el repte ODIR:

```python
def odir_metrics(gt_data, pr_data):
    th = 0.5
    gt = gt_data.flatten()
    pr = pr_data.flatten()
    kappa = metrics.cohen_kappa_score(gt, pr > th)
    f1 = metrics.f1_score(gt, pr > th, average='micro')
    auc = metrics.roc_auc_score(gt, pr)
    final_score = (kappa + f1 + auc) / 3.0
    return kappa, f1, auc, final_score

gt_data = import_data('odir_ground_truth.csv')
pr_data = import_data('odir_predictions.csv')
kappa, f1, auc, final_score = odir_metrics(gt_data[:, 1:], pr_data[:, 1:])
print("kappa score:", kappa)
print("f-1 score:", f1)
print("AUC value:", auc)
print("Final Score:", final_score)
```

3) Generació de mostres sintètiques

La manera més adient de generar mostres sintètiques és mitjançant la creació d'imatges aleatòries de diferents atributs de les imatges que representen les classes minoritàries. Analitzant les dades de la Taula 7, observem que per a equilibrar el dataset hem de generar el següent nombre d'imatges per classe per a balancejar-lo:

Taula 7. Nombre d'Imatges a generar per classe minoritària

	Normal	**Diabetes**	**Glaucoma**	**Cataract**	**AMD**	**Hypertension**	**Myopia**	**Others**
Total	2816	1778	313	275	280	192	262	1124
Limit	0	2816	2816	2816	2816	2816	2816	2816
Images to generate	0	1038	2503	2541	2536	2624	2554	1692
Samples per image	0.0000	0.5838	7.9968	9.2400	9.0571	13.6667	9.7481	1.5053
Strategy		1038x1	313x7 312x1	275x9 66x1	280x9 16x1	192x13 128x1	262x9 196x1	1124x1 568x1

Tenim 2816 imatges de la classe majoritària i hem de balancejar el conjunt de dades per a obtenir el mateix valor per a cada classe. L'algorisme descrit a continuació utilitzarà els valors descrits a la fila "Strategy" de la Taula 7 per a generar un conjunt de dades addicionals usant augmentació de dades, fins a equilibrar completament el dataset, obtenint un conjunt de 22528 imatges (2816 imatges per classe).

Hem de considerar també que les imatges a generar han de ser específiques per al nostre problema i que siguin identificadores de la incògnita que volem resoldre. Per això, els següents algorismes definits en la Figura 18 sobre l'augment de dades, seran els usats per a la generació de dades addicionals sobre el conjunt original. Les xarxes han sigut específicament escollides per la seva naturalesa i que creiem que poden ajudar en la millora d'identificació de característiques de les diverses patologies:

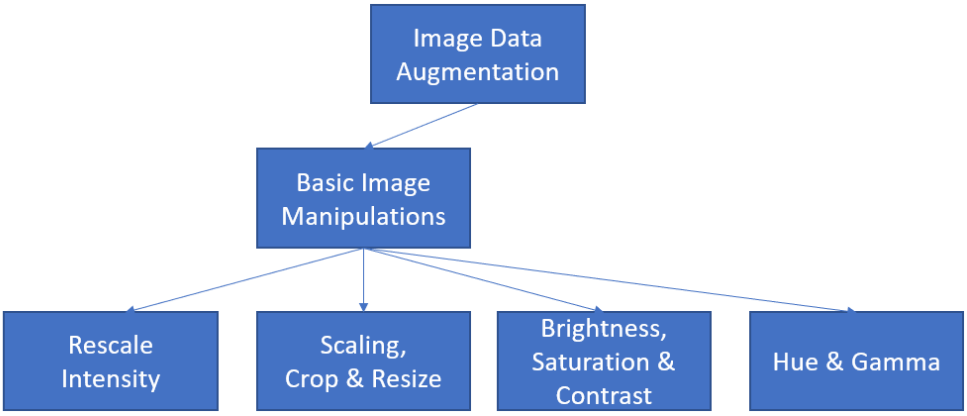


Figura 18. Algorismes d'augment de dades utilitzats

Un exemple de dades generades a través de l'augment de dades es pot observar en la Figura 19. L'algorisme realitzat és capaç de generar fins a 14 imatges aleatòries diferents per a les diferents estratègies que realitzarem segons la Taula 7.

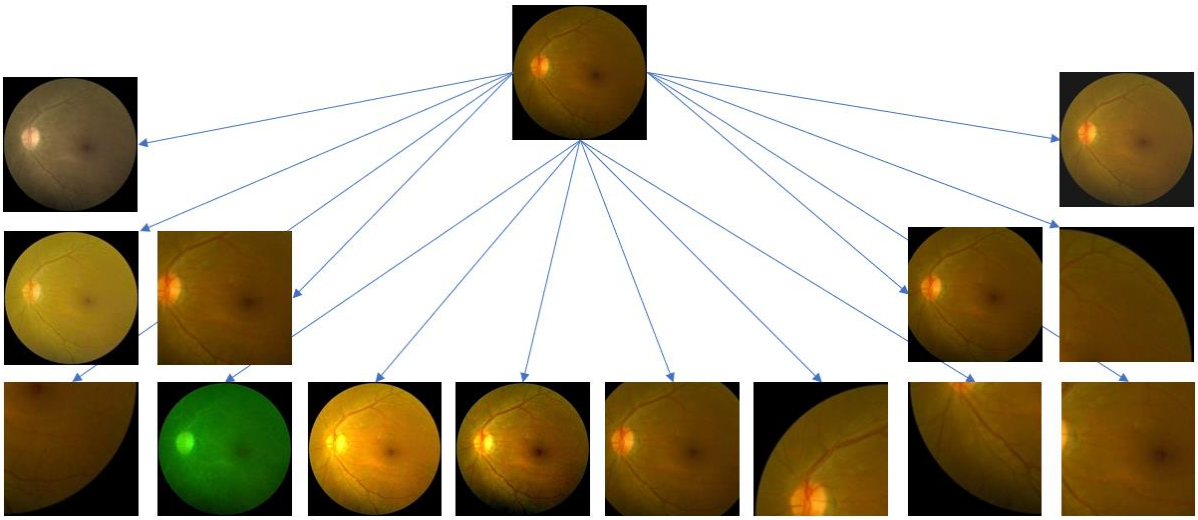


Figura 19. Augment de dades: escalat, saturació, retall, il·luminació, contrast i d'altres.

Els diferents algorismes utilitzats per a la generació automàtica de noves imatges es pot consultar als fitxers que formen part del nostre treball:

- odir_data_augmentation_generator.py
- odir_data_augmentation_strategies.py.

7.2 Model Inception

7.2.1 Descripció del model

L'arquitectura Inception també coneguda com a GoogLeNet va ser la guanyadora del ILSVRC (ImageNet Large Scale Visual Recognition Competition) l'any 2014 i que té un error relatiu menor comparant-la amb l'arquitectura VGGNet definida en el següent capítol i que és la 1a finalista del 2014. GoogLeNet (com a homenatge a Yann LeCuns que va presentar la xarxa LeNet 5) és l'anomenada Inception v1 i també podem trobar versions evolucionades com la versió 2, 3 i 4. Cada versió és una millora iterativa de l'anterior i entendre el funcionament de cada versió ens pot ajudar a esbrinar quina és la que hem d'usar per al nostre treball. En el nostre cas, s'ha triat la versió 3 donat que l'arquitectura va marcar una fita important en el desenvolupament de classificadors convolucionals, pel seu disseny eficient, pel seu entrenament afegit via ImageNet i per què es pot usar en maquinari estàndard.

La majoria de CNNs només apilaven capes i més capes convolucionals amb l'esperança d'obtenir un major rendiment, però sense resultats. La xarxa Inception, és sens dubte, una xarxa molt complexa i que ha sigut força dissenyada per a donar una empenta al rendiment en termes de velocitat i precisió.

El model construït és l'Inception v3 que es basa en la generació de capes tal com mostra la Figura 20. Inception presenta una de les xarxes de l'estat de l'art pel que fa a la classificació d'imatges mitjançant xarxes convolucionals neuronals. A primera vista, l'arquitectura no sembla gaire complicada, però el reiterat apilament dels mòduls Inception fan que sigui molt profunda (Szegedy et al., 2014).

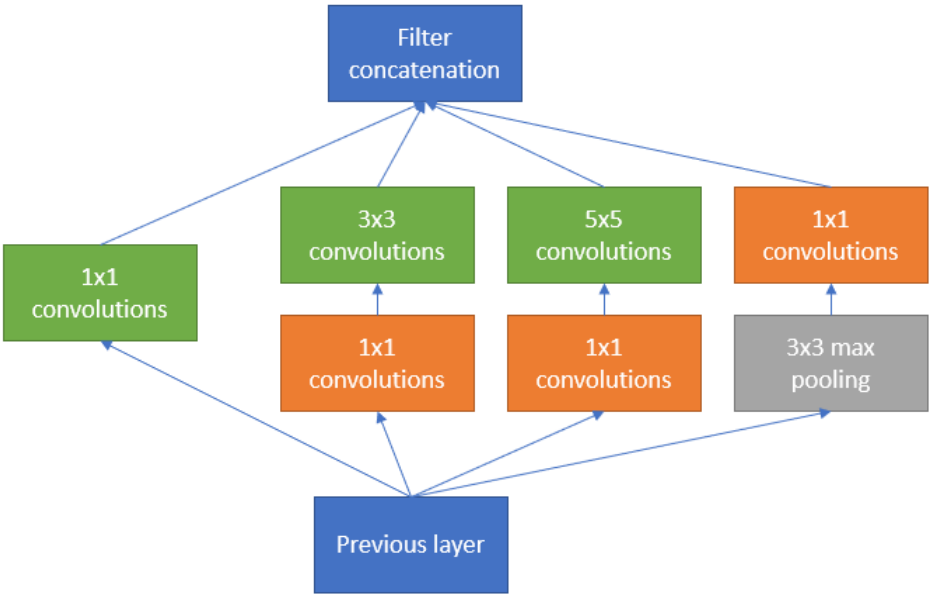


Figura 20. Model Inception

En apilar cada mòdul, aconseguim una certa profunditat, que a la vegada ens aporta una amplada considerada i que ens facilita reconèixer característiques d'imatges amb diferents escales. En termes de xarxes neuronals convolucionals, això vol dir, introduir convolucions amb filtres de diferents mides, d'aquí que puguem veure convolucions apilades de 3x3 i de 5x5. Cada convolució té un cost de computació molt elevat. Per a combatre el nostre problema s'afegeixen capes de convolució d'1x1 per a reduir la dimensionalitat abans de cada convolució com es mostra en la Figura 20.

El model que s'entrega amb el treball conté la càrrega de l'Inception v3 on es pot modificar el nombre de capes a entrenar (si és que s'utilitza entrenament transferit). Per a poder realitzar el procés, es carrega la xarxa original amb els pesos d'ImageNet i llavors es modifica l'última capa del model afegint la capa que ens permetrà treballar amb el multi-label. La Figura 21 mostra les capes que s'han afegit al model original.

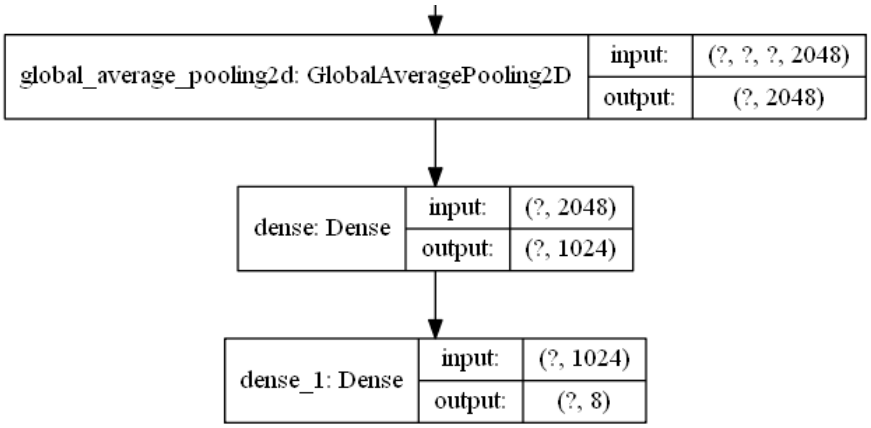


Figura 21. Capes afegides al model Inception

Les noves capes inclouen un GlobalAveragePooling2D que realitza una operació on es calcula la sortida mitjana de cada mapa de característiques que prové de la capa anterior. Amb l'operació reduïm les dades de manera significativa i preparem el model per a la capa de classificació final. La secció de classificació final conté dues capes denses, una de 1024 i l'última de 8 que representa el nombre de possibles etiquetes disponibles en el nostre sistema. L'última capa utilitza una funció Sigmoid que ens permetrà assignar pesos a cada etiqueta de manera individual.

Finalment, el model de pèrdua ha de ser ajustat al tipus d'entropia creuada binària (binary cross-entropy) perquè la pèrdua és calculada per a cada component de manera individual (recordem que la funció sigmoid anterior disposa de 8 sortides individuals i s'hi han de calcular les pèrdues per a cadascuna d'elles). El tipus de connexió definit és estàndard en problemes amb múltiple etiquetes.

El script que acompanya el nostre projecte i que conté el model realitzat es pot trobar en el següent fitxer:

- odir_inception_training.py

El model suporta imatges de 224x224 píxels i les imatges són transferides al model a través de les seves diferents capes. Entrenem el model des de l'inici i totalment (totes les capes) per a analitzar els resultats que podem obtenir-hi. El resum del model es pot trobar al fitxer inception_model_summary.md o a la secció 11.6 de l'annex.

El diagrama model (vista parcial a causa de la seva mida) es pot veure a la Figura 22:

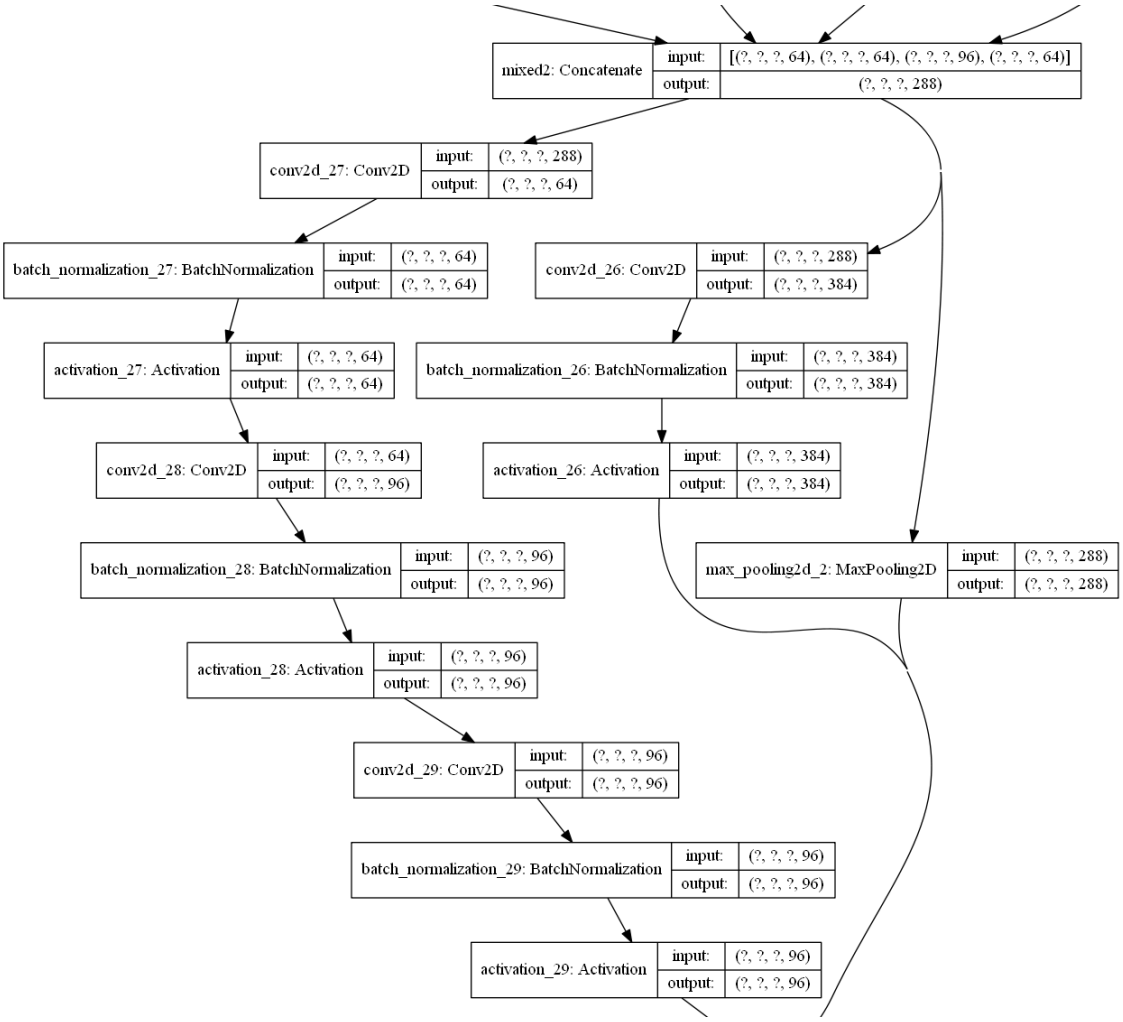


Figura 22. Model Inception

El model genera 34 milions de paràmetres amb un cost d'entrenament d'1,5h per època.

7.2.2 Procés d'entrenament

L'entrenament del model es realitza en diferents passos. Primerament, s'analitza la funció fit i en un procés més desenvolupat s'utilitza el fit_generator que ens permet l'ús de generadors per a entrar imatges sense que la memòria a consumir no es vegi tan afectada. Cada entrenament o experiment genera un codi identificador que ens permet desar-lo de manera ordenada.

El model s'inicia amb Adam (Adaptive Moment Estimation) amb un learning rate de 0.01 i durant la fase de tuning s'avalua el SGD (Stochastic Gradient Descent) amb diferents paràmetres.

En realitzar centenars de proves, han de quedar correctament registrades per a la seva posterior anàlisi. Durant l'entrenament s'usaran diferents estratègies i algorismes fins a aconseguir un resultat òptim.

Durant la fase d'entrenament amb transferència de dades, configurarem les dades d'entrada mitjançant la unitat de preprocessament que ens permetrà ajustar les imatges a les dades d'ImageNet. El procés consisteix a convertir les imatges de RGB a BGR i llavors centrar cada canal de color amb el zero al centre respecte al conjunt de dades ImageNet.

Cada experiment genera dades significants a més de diversos gràfics exploratoris que contenen valors com la precisió, la pèrdua, la precisió de validació, la pèrdua de validació, Recall i AUC entre d'altres. Les dades, com s'ha mencionat anteriorment, queden desades respectivament amb un ID únic assignat per defecte per al seu tractament i anàlisi posterior.

7.2.3 Procés de validació

Per a validar les dades, utilitzarem les mètriques definides anteriorment i usarem també els components que l'ODIR defineix per a obtenir una puntuació general sobre l'estat de la classificació final. El procés es realitza mitjançant la càrrega en memòria d'un model

prèviament entrenat i que conté el seu graf d'execució. Llavors, s'entra la imatge que volem comprovar (realitzant les transformacions pertinents perquè sigui compatible amb el model) i s'obté el resultat de la categorització.

El procés executa els passos interns necessaris i dóna un resultat a interpretar. Quan rebem els resultats del model, hem d'analitzar-los en la manera en què hem dissenyat la sortida. En el nostre cas, hem de tenir en compte que estem treballant amb un problema multi-label i per tant podem obtenir imatges amb diverses etiquetes marcades.

La validació es realitza d'inici a fi. Primer, comprovem que les dades d'entrada son realment imatges del nostre dataset, i que, estan correctament formatades i llavors, analitzem la sortida de la mateixa manera, marcant la imatge amb les diferents classes. La següent Figura 23, mostra les dades d'entrada d'entrenament i la seva classificació segons el ground truth.

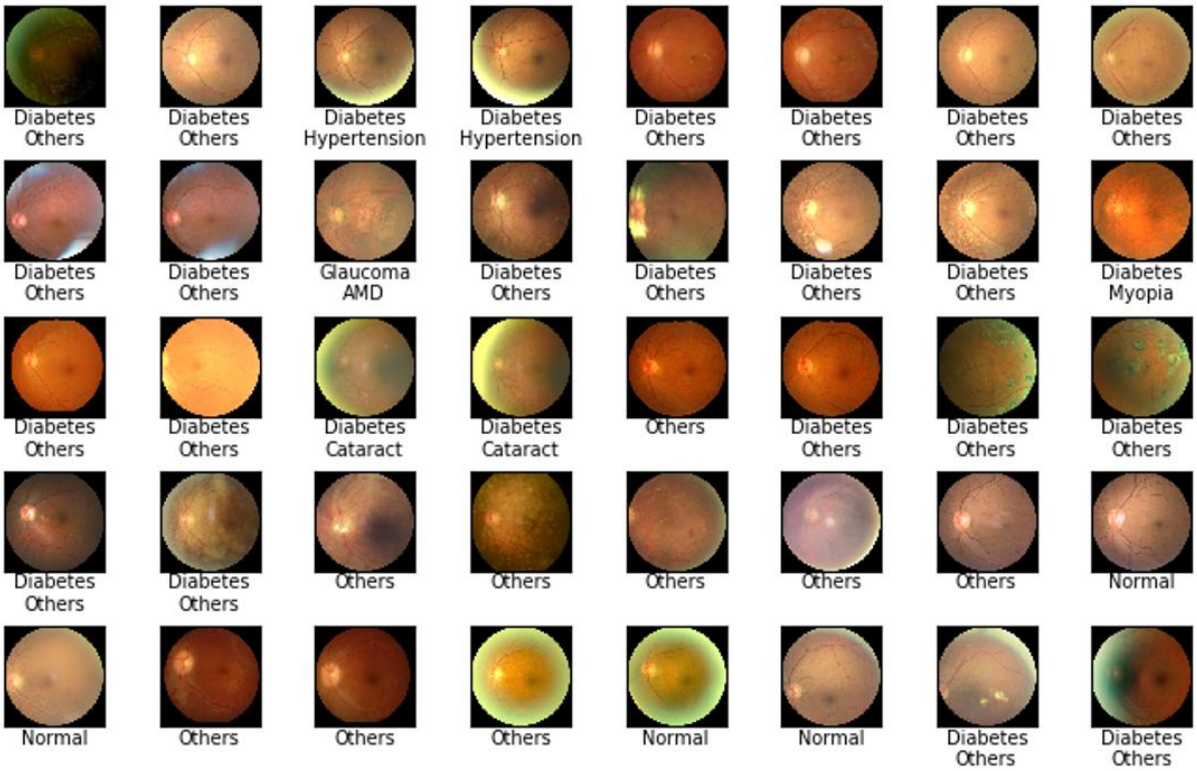


Figura 23. Mostra de dades d'entrada etiquetades

Com es pot observar, hi ha imatges classificades amb diverses patologies. Som capaços de comprovar que les dades entren de manera adient i estan associades correctament amb les seves corresponents etiquetes. La crida de codi per representar l'entrada de dades es pot veure a continuació, on s'utilitza la classe interna *Plotter* per a compondre totes les imatges d'una manera eficient i elegant:

```
plotter.plot_metrics(history, os.path.join(newfolder, 'plot1.png'), 2)
```

Un cop volem comprovar una imatge o múltiples, podem cridar l'entrenament desat anteriorment en un dels experiments i comprovar la sortida que el model ens dóna envers l'entrada. La Figura 24 mostra un exemple d'execució de la sortida generada i la seva classificació. En vermell es mostra la predicció i en verd el ground truth. Si el ground truth correspon amb la predicció (considerant totes les possibles opcions) llavors apareix en verd:

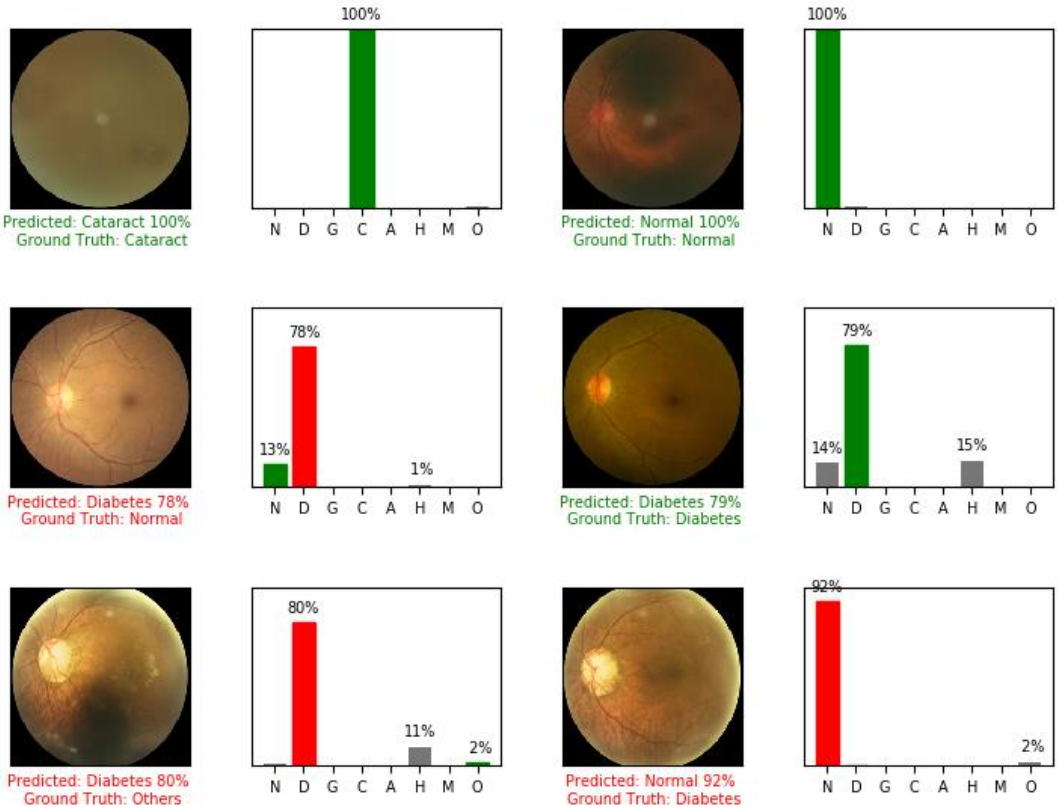


Figura 24. Exemple de sortida del model per a la seva validació visual

D'altra banda, diferents mètriques es mostren a la sortida de l'execució de l'avaluació del model per a l'anàlisi més profund dels valors. Cadascuna de les propietats es mostren i s'analitzen en detall en els experiments.

Un exemple de sortida de l'avaluació amb les diferents mètriques que el model genera:

```
400/1 - 1600s - loss: 10.1270 - tp: 46.0000 - fp: 263.0000 - tn: 2537.0000 - fn:
354.0000 - accuracy: 0.8072 - precision: 0.1489 - Recall: 0.1150 - AUC: 0.5101
loss :  9.516395740509033
tp :  46.0
fp :  263.0
tn :  2537.0
fn :  354.0
accuracy :  0.8071875
precision :  0.14886731
Recall :  0.115
AUC :  0.5101478

Kappa score: 0.02334784329244166
F-1 score: 0.8071875
AUC value: 0.5193276785714286
Final Score: 0.4499543406212901
```

Finalment, es mostra la Figura 25 amb la matriu de confusió, amb una confirmació visual d'on se situen les prediccions:

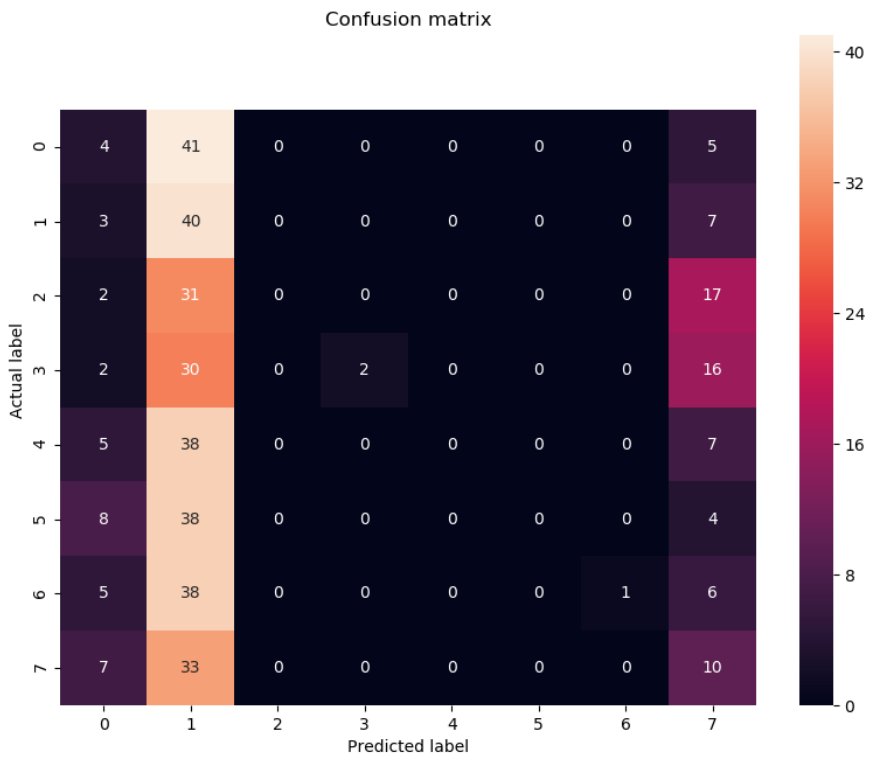


Figura 25. Exemple de matriu de confusió

El següent fitxer ens permet avaluar imatges via inferència pel model Inception:

- odir_inception_testing_inference.py

7.2.4 Experiments

Els experiments inicials es realitzen amb unes imatges d'entrada de 224x224 píxels on el conjunt d'entrenament és de 6151 imatges i el conjunt de validació de 400 per defecte. En els experiments amb augment de dades, el nombre d'imatges d'entrenament varia i es fa menció del nombre d'imatges generades sintèticament. A més a més, iniciem els experiments amb:

- Aprenentatge transferit usant les dades d'ImageNet.
- S'entrenen totes les capes.
- Optimitzador: Adam amb una ràtio d'aprenentatge de 0.01 (Si l'optimitzador és diferent, s'indica a l'experiment).

Cada experiment comença amb una taula indicant els paràmetres que s'han utilitzat per al seu correcte seguiment.

7.2.4.1 Experiment inicial

La configuració d'aquest experiment inicial és per a tenir un contacte inicial amb el model Inception i a partir d'aquí fer els ajustaments necessaris per a trobar la configuració que més s'adapti a les nostres necessitats. La Taula 8 mostra aquesta configuració.

Taula 8. Configuració de l'experiment

Training Detail	Inception
Data Augmentation	No
Transfer Learning	Yes
Weights	Pre-trained on ImageNet
Last Layer	GlobalAveragePooling2D Dense (1024, activation='relu') Dense (8, activation='sigmoid')
Feature Extraction Enabled	Yes
Classification Enabled	Yes
Optimizer	Adam lr=0.01

Loss function	Binary Cross-Entropy
Early Stopping patience	None
Number of Parameters	23,909,160
Number of trainable Parameters	23,874,728

Els primers resultats sense cap ajust, és a dir, sense afegir res al model, ens donen un comportament amb un valor de *accuracy* que sembla elevat. La Taula 9 mostra els diferents valors de l'entrenament bàsic sense cap ajustament i sense mètriques addicionals.

Taula 9. Resultats bàsics amb la xarxa Inception

loss	**accuracy**	**Val loss**	**Val accuracy**
0.2728	0.8759	0.4595	0.8453
0.229	0.8969	0,5704	0.8191
0.0653	0.9759	0.7999	0.8194

La Taula 9 mostra les diferents execucions del model Inception sobre les dades d'entrenament i validació. Sense cap ajustament, aconseguim un *accuracy* del 81.94% sobre el conjunt de validació. Tot i tenir un *accuracy* elevat, hem d'analitzar més mètriques pel fet que amb desequilibri és habitual obtenir aquest tipus de resultats, i potser, no està reflectint bé la classificació.

Si analitzem les Figures 26 i 27 que mostren els valors de *accuracy* del model, *accuracy* de la validació, pèrdua i pèrdua de validació podem veure com el model s'està comportant:

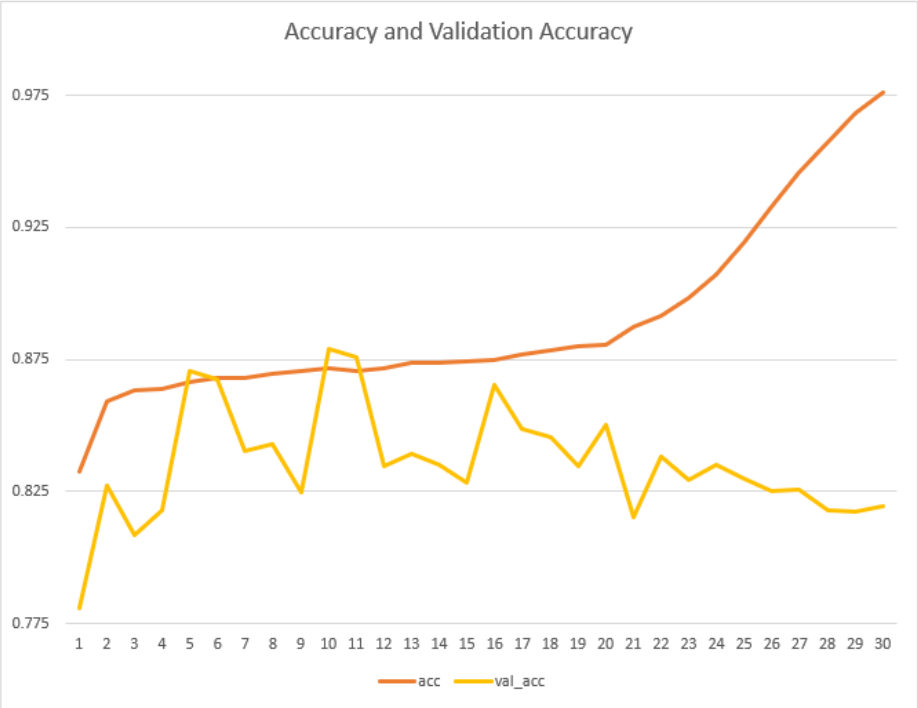


Figura 26. Accuracy d'entrenament i Accuracy de validació

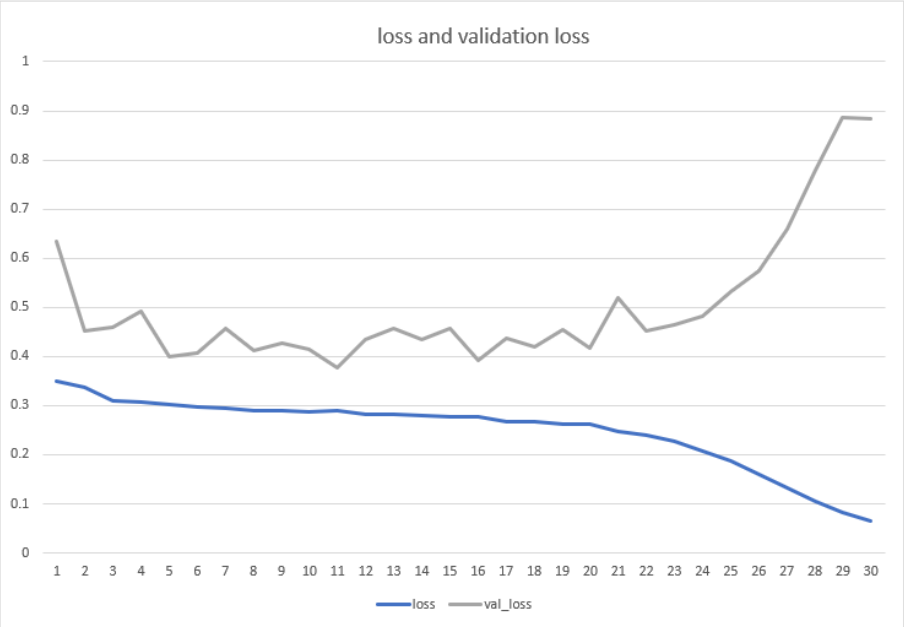


Figura 27. Pèrdua de l'entrenament i pèrdua de la validació

De les anteriors figures podem entendre que la pèrdua disminueix a mesura que l'entrenament està en execució. Si la pèrdua de l'entrenament disminueix, llavors l'entrenament està funcionant correctament.

L'accuracy de validació mesura com bona és la predicció del nostre model. Si el model està aprenent, el valor de *accuracy* s'incrementa. En el nostre cas, la pèrdua d'entrenament està disminuint però també ho fa *l'accuracy* de validació cosa que és un senyal d'overfitting. En el nostre cas, haurem d'experimentar amb diferents tècniques per a combatre el problema que s'observa.

A l'entrenar massa, arribem a un punt on la pèrdua en el conjunt de validació deixa de baixar i comença a pujar (overfitting). Hem d'afegir doncs, una condició d'aturada, ja que si no el model es converteix extremadament bo per la classificació de dades d'entrenament però generalitzant de mala manera i provocant que la classificació de les dades de validació empitjori. La solució per a reduir aquest soroll de dades de l'entrenament és aturant l'entrenament quan comenci a augmentar l'error de validació.

Si analitzem en detall la matriu de confusió de la Figura 28 de l'experiment en curs, podem observar com el sistema es torna mandrós i intenta obtenir un valor d'*accuracy* elevat només categoritzant les imatges en les classes majoritàries (0-Normal, 1-Diabetis):

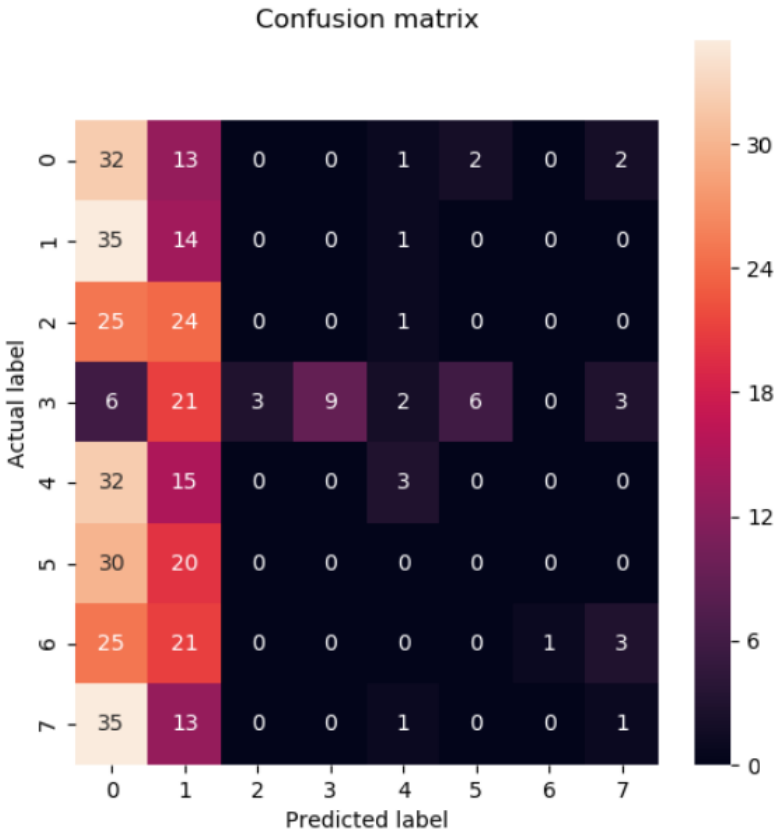


Figura 28. Matriu de confusió de l'experiment inicial

7.2.4.2 Experiment amb pesos o parcialitat en cada classe de sortida

La configuració d'aquest experiment es pot trobar a la Taula 10:

Taula 10. Configuració de l'experiment

Training Detail	Inception
Data Augmentation	No
Transfer Learning	Yes
Weights	Pre-trained on ImageNet
Last Layer	GlobalAveragePooling2D Dense (1024, activation='relu') Dense (8, activation='sigmoid')
Class Weights	Yes
Feature Extraction Enabled	Yes
Classification Enabled	Yes
Optimizer	Adam lr=0.01
Loss function	Binary Cross-Entropy
Early Stopping patience	8 steps for validation loss, type [min]
Number of Parameters	23,909,160
Number of trainable Parameters	23,874,728

Aquests experiments consisteixen a modificar els pesos introduïts al model per a forçar l'algorisme a tractar certes instàncies del model amb un pes diferent i així donar un significat major a classes minoritàries. A més a més, la xarxa es carrega amb els pesos d'ImageNet però s'entrenen totes les capes.

Com hem definit anteriorment en l'apartat que especifica com combatre el desequilibri de dades, tenim els següents pesos que introduirem al nostre model:

```
class_weight = {0:1.,
                1:1.583802025,
                2:8.996805112,
                3:10.24,
                4:10.05714286,
                5:14.66666667,
                6:10.7480916,
                7:2.505338078}
```

A causa dels canvis realitzats en les mètriques afegides, observem nous valors afegits a la Taula 11 i 12. Aquestes mètriques ens aporten un valor afegit i ens ajuden a entendre si el nostre model està aprenent o no.

Taula 11. Resultats sobre les dades d'entrenament amb pesos amb la xarxa Inception

loss	**accuracy**	**precision**	**Recall**	**AUC**
0.5148	0.8808	0.6183	0.3054	0.8994

Taula 12. Resultats sobre les dades de validació amb pesos amb la xarxa Inception

Val loss	**Val accuracy**	**Val precision**	**Val Recall**	**Val AUC**
0.2633	0.8963	0.6288	0.4150	0.8426

La Taula 11 mostra les dades sobre el conjunt d'entrenament i s'observa una precisió del 61% i un Recall del 30% indicant-nos que el model no està aprenent sobre les classes minoritàries. Així i tot, la precisió de les nostres imatges aconsegueix un 62% amb un Recall del 41%, cosa que ens mostra que tenim un model de partida a millorar. Amb el valor de Recall obtenim les

imatges que s'han marcat com a positiu i que realment són positives (true positive). Com que estem treballant amb un problema mèdic, els falsos negatius tenen un cost elevat per als pacients, ja que podríem classificar incorrectament una malaltia. Per tant, Recall és una mètrica que podem utilitzar per a seleccionar el millor model.

La Figura 29 mostra el resultat de l'execució de l'experiment com a reforç de les dades exposades a les taules 8 i 9.

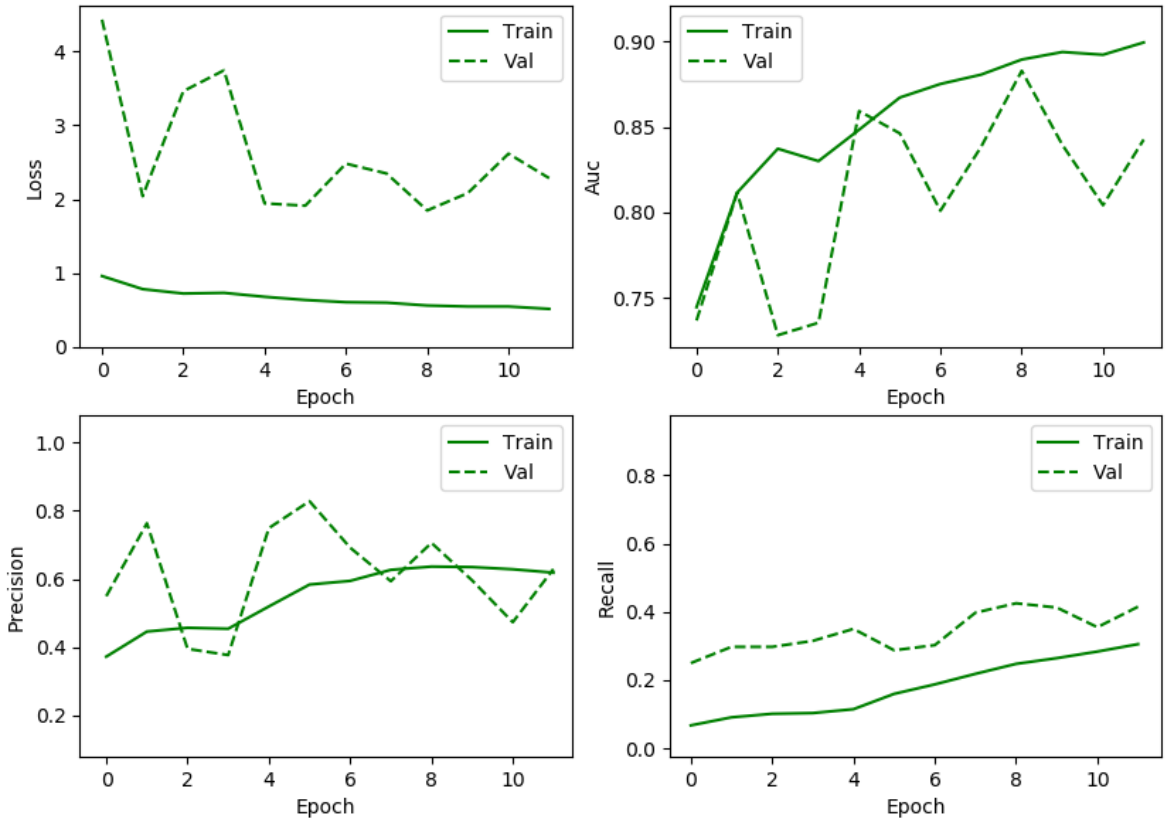


Figura 29. Execució de l'experiment amb pesos model Inception

La Figura 30 mostra el valor de *l'accuracy* d'entrenament i *accuracy* de validació:

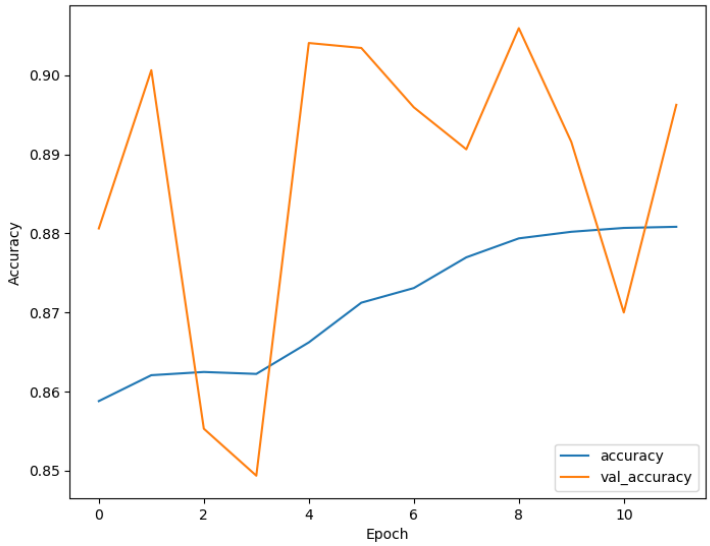


Figura 30. Accuracy d'entrenament i Accuracy de validació

La matriu de confusió de la Figura 31, mostra una diferencia significant amb la matriu anterior i és que, en aquest cas, aconseguim una variació significant en el model i decideix ubicar correctament algunes de les imatges a classificar com podem observar en la diagonal de la matriu:

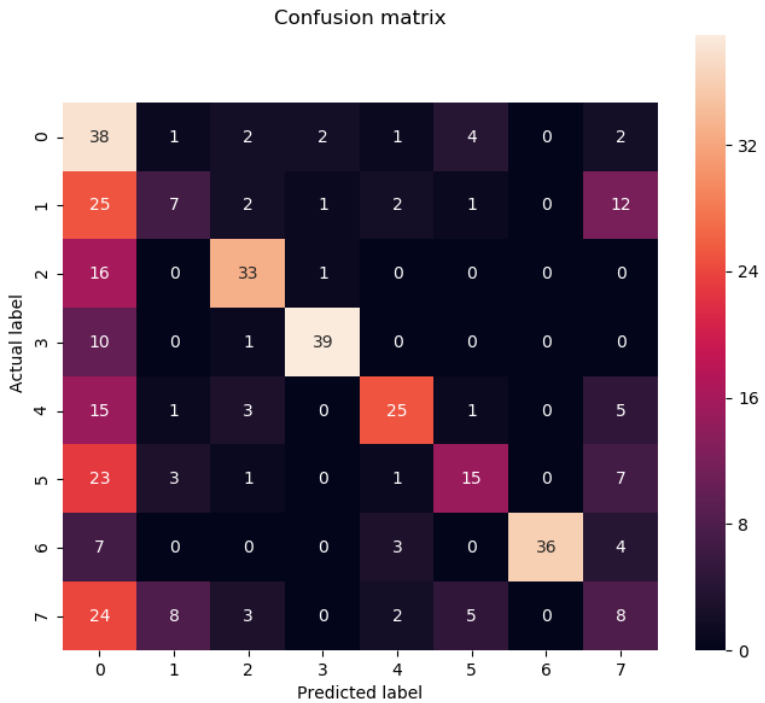


Figura 31. Matriu de confusió

Com a resultats addicionals, tenim el càlcul del F-1 score i el coeficient Kappa de Cohen:

Taula 13. Resultats sobre les dades de validació amb pesos amb la xarxa Inception

Kappa score	F-1 score	AUC value	Final Score
0.034	0.8184	0.5147	**0.45588**

Aquest model obté una puntuació de **0.45** segons mostra la Taula 13 en finalitzar la seva execució i afegint tots els valors calculats anteriorment.

La Figura 32 mostra una confirmació visual del resultat obtingut pel classificador sobre les dades de validació (verd: encerts, vermell: errors):

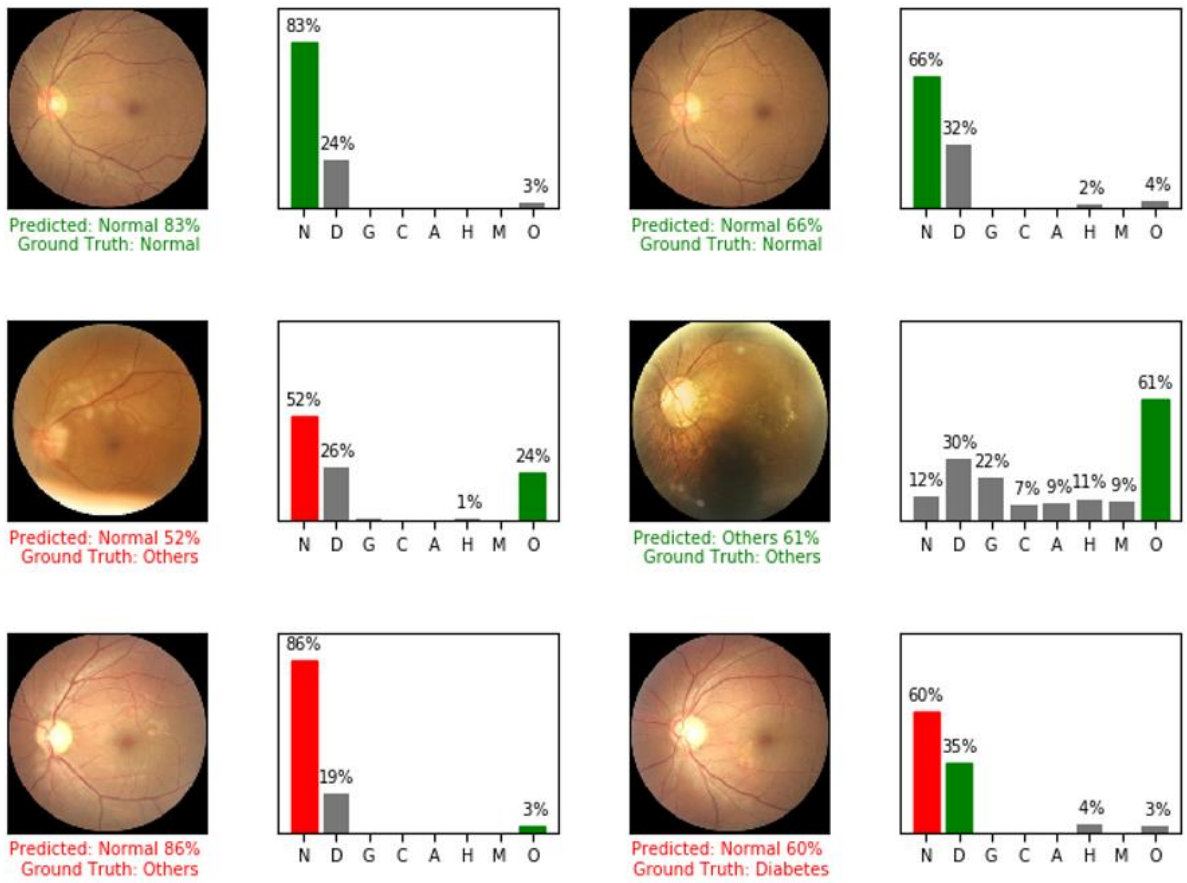


Figura 32. Exemple de resultat del classificador

7.2.4.3 Experiment amb Augment de dades

La configuració de l'experiment es pot trobar a la Taula 14:

Taula 14. Configuració de l'experiment

Training Detail	Inception
Data Augmentation	Yes
Transfer Learning	Yes
Weights	Pre-trained on ImageNet
Last Layer	GlobalAveragePooling2D Dense (1024, activation='relu') Dense (8, activation='sigmoid')
Class Weights	No
Feature Extraction Enabled	Yes
Classification Enabled	Yes
Optimizer	Adam lr=0.01
Loss function	Binary Cross-Entropy
Early Stopping patience	8 steps for validation loss, type [min]
Number of Parameters	23,909,160
Number of trainable Parameters	23,874,728

Aquest experiment consisteix a generar mostres sintètiques per a cobrir la mancança de dades que tenim de les classes minoritàries. És a dir, es generen imatges reflectint aspectes que volem destacar sobre les imatges originals perquè el classificador tingui un altre punt de vista sobre la imatge original. D'aquesta manera augmentem les dades d'entrada generant-ne de noves, utilitzant les mateixes imatges com a font d'origen. A més a més, continuem amb pesos d'ImageNet i entrenant totes les capes. Es generen moltes més imatges (més de 20 mil) i el model s'entrena totalment.

Els resultats de l'experiment es poden trobar a les Taules 15 i 16:

Taula 15. Resultats sobre les dades d'entrenament amb pesos amb la xarxa Inception

loss	accuracy	precision	Recall	AUC
0.0212	0.9924	0.9774	0.9708	0.9987

Taula 16. Resultats sobre les dades de validació amb pesos amb la xarxa Inception

Val loss	Val accuracy	Val precision	Val Recall	Val AUC
0.6558	0.87625	0.5050505	0.5	0.7949938

Aquí notem un salt de qualitat positiu. En termes de dades d'entrenament, aconseguim una precisió del 97% cosa que ens indica que el model està aprenent. Si analitzem les dades de validació, observem que el model és capaç de categoritzar correctament el 50%.

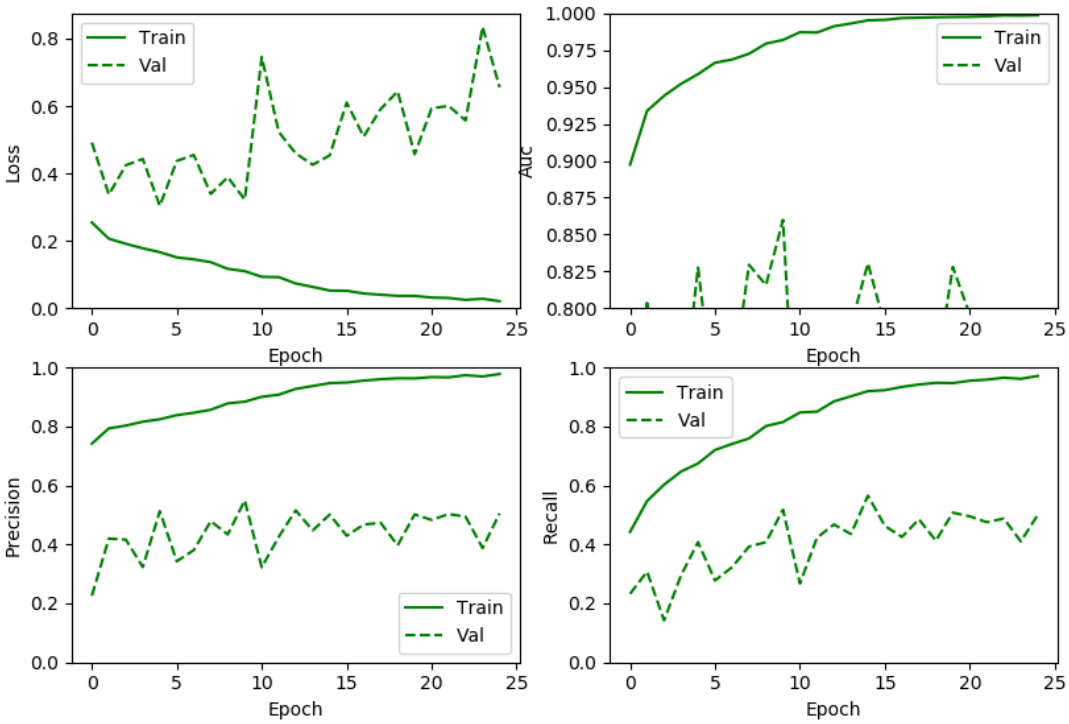


Figura 33. Resultat de l'execució del model

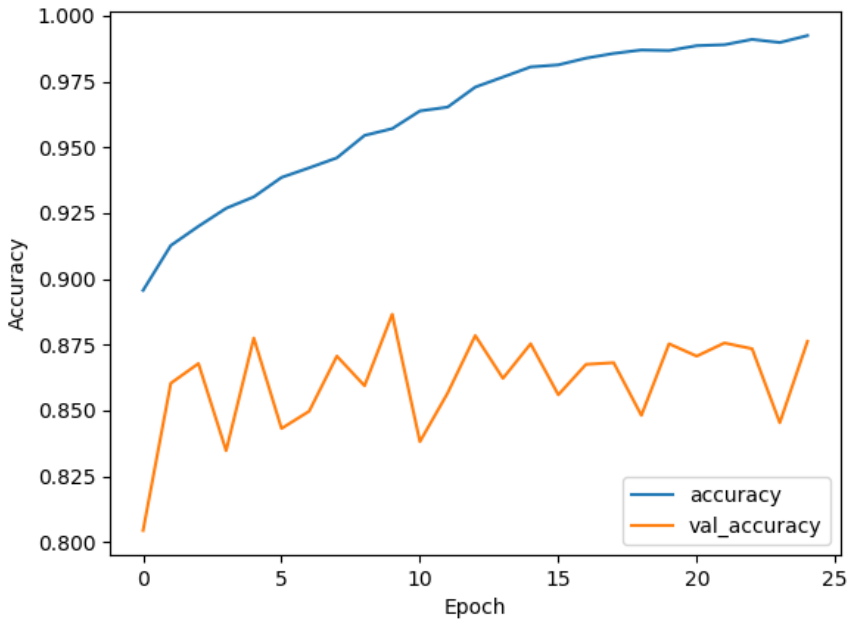


Figura 34. Accuracy d'entrenament i Accuracy de validació

La Figura 34 ens mostra *l'accuracy* final del nostre model i la Figura 35 ens mostra la ubicació de les imatges de validació dins la matriu de confusió. Com podem observar tenim algunes imatges ubicades a la diagonal donant-nos un bon senyal visual sobre la seva correcta classificació. A més a més, podem observar com el model decideix classificar la resta de manera incorrecta en una de les categories majoritàries.

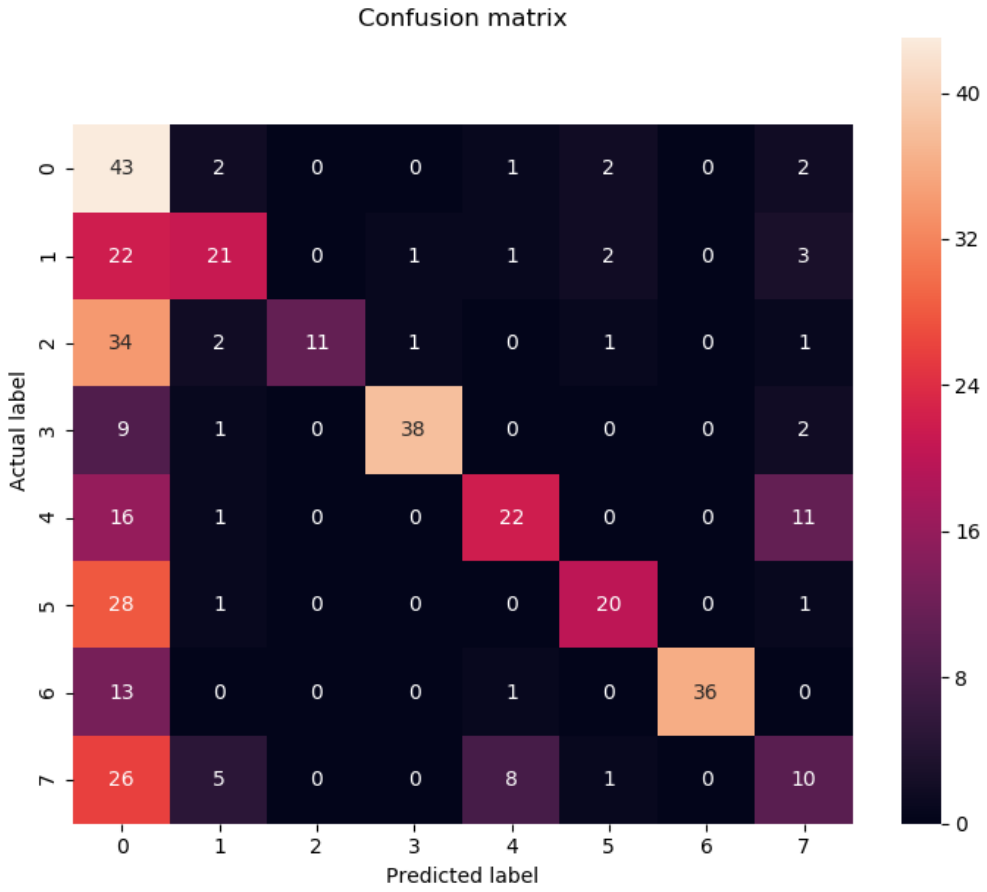


Figura 35. Matriu de confusió

La Taula 17 ens mostra la puntuació final del model, on obtenim una puntuació elevada de **0.71**.

Taula 17. Resultats puntuació final

Kappa score	**F-1 score**	**AUC value**	**Final Score**
0.4318	0.87625	0.8406	**0.7162**

La Figura 36 mostra una confirmació visual del resultat obtingut pel classificador sobre les dades de validació (verd: encerts, vermell: errors):

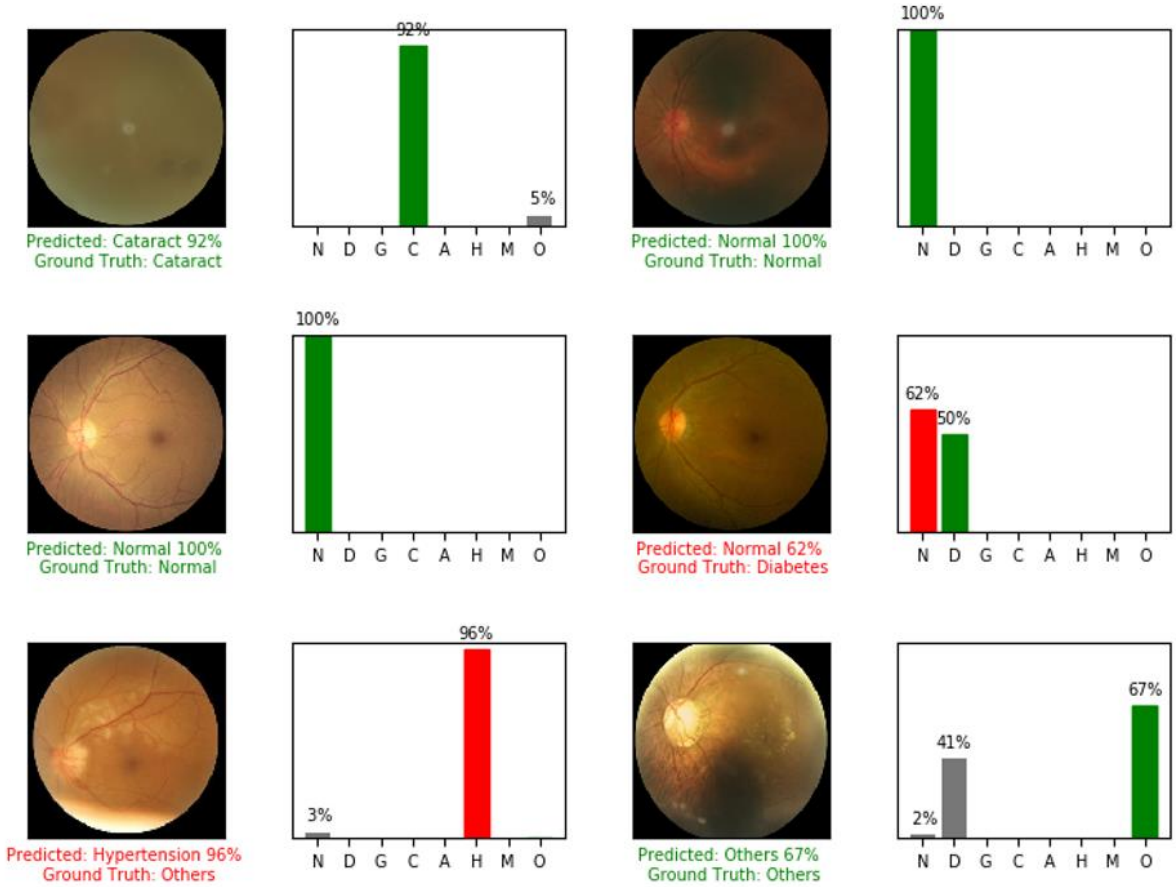


Figura 36. Exemple de sortida de les dades de validació

7.2.4.4 Experiment amb aprenentatge transferit

La configuració de l'experiment es pot trobar a la Taula 18:

Taula 18. Configuració de l'experiment

Training Detail	Inception
Data Augmentation	Yes
Transfer Learning	Yes
Weights	Pre-trained on ImageNet
Last Layer	GlobalAveragePooling2D Dense (1024, activation='relu') Dense (8, activation='sigmoid')
Class Weights	No
Feature Extraction Enabled	No
Classification Enabled	Yes
Optimizer	Adam lr=0.01
Loss function	Binary Cross-Entropy

Early Stopping patience	8 steps for validation loss, type [min]
Number of Parameters	23,909,160
Number of trainable Parameters	2,106,784

Fins ara, els entrenaments s'han realitzat entrenant la xarxa totalment (carregant els pesos de l'entrenament original amb les dades d'ImageNet). El procés és costós ja que es triguen dies a entrenar la xarxa amb 23 milions de paràmetres. Existeix doncs, la possibilitat d'usar la xarxa Inception amb un aprenentatge realitzat sobre les dades d'ImageNet on congelem les capes que no volem modificar i només entrenem l'última capa que hem modificat per a la nostra classificació multi-label.

Aquest procés s'aconsegueix mitjançant la propietat *trainable* de cada capa del model:

```
for layer in base_model.layers:
    layer.trainable = False
```

Mitjançant aquest mètode, informem el model de què no volem entrenar les capes originals que s'han entrenat per a les dades d'ImageNet i deixem que l'aprenentatge transferit faci la seva feina sobre les dades que volem usar per al nostre entrenament. La Figura 37 ens mostra el resultat de les mètriques després de l'execució del model.

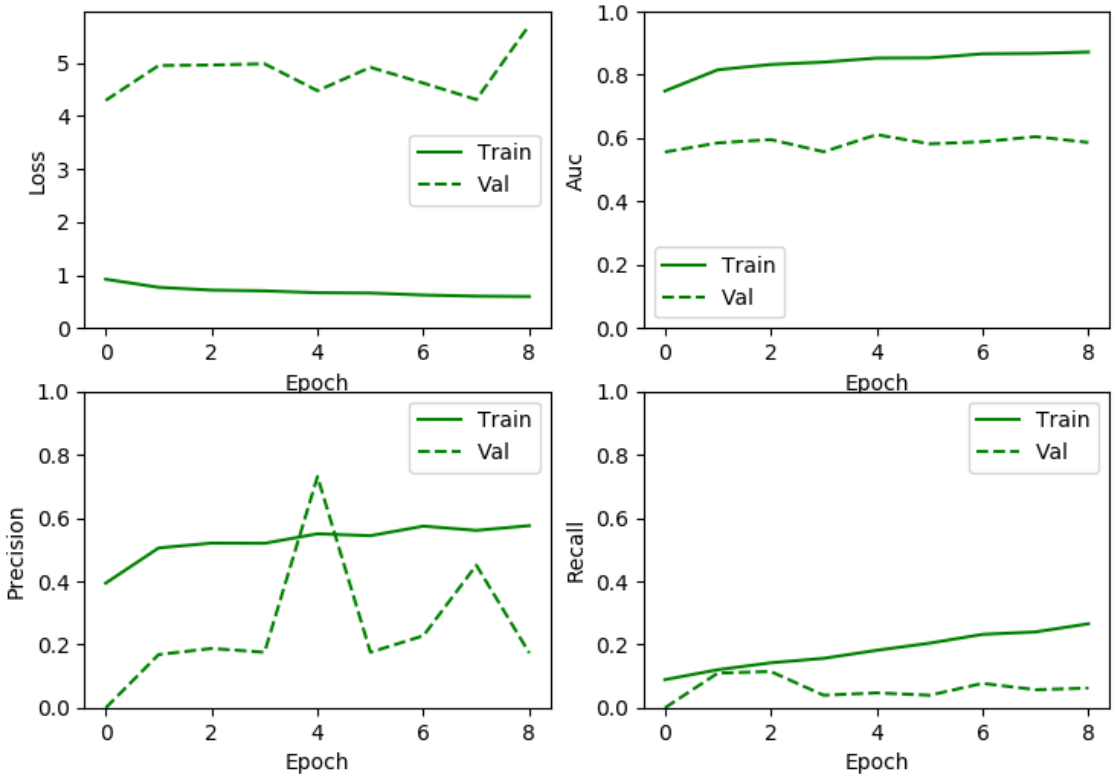


Figura 37. Sortida de l'execució amb aprenentatge transferit

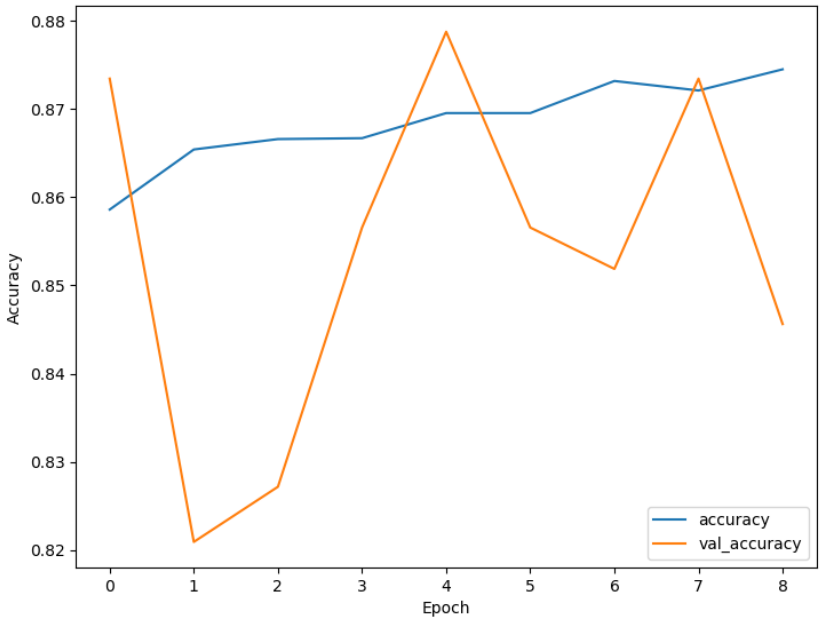


Figura 38. Accuracy d’entrenament i Accuracy de validació

Les figures 38 i 39 ens mostren les dades de *l'accuracy* del model i la ubicació de les imatges envers la classificació dins la matriu de confusió.

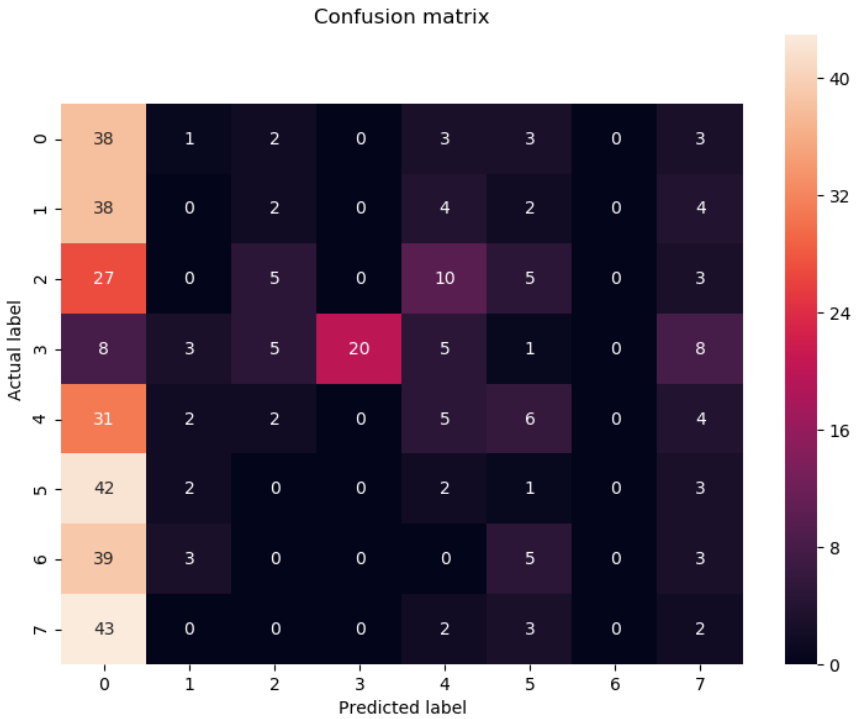


Figura 39. Matriu de confusió

Es pot observar com l'aprenentatge transferit no ens ha aportat el benefici esperat amb la classificació final, donant-nos una puntuació més baixa que en altres experiments.

Els resultats de l'experiment es poden trobar a les Taules 19 i 20:

Taula 19. Resultats sobre les dades d'entrenament amb aprenentatge transferit

loss	**accuracy**	**precision**	**Recall**	**AUC**
0.5681	0.8762	0.5881	0.2750	0.8797

Taula 20. Resultats sobre les dades de validació amb aprenentatge transferit

Val loss	**Val accuracy**	**Val precision**	**Val Recall**	**Val AUC**
0.669	0.8259	0.1572	0.09	0.566

El resultat amb la puntuació final es pot trobar a la Taula 21.

Taula 21. Resultats puntuació final

Kappa score	F-1 score	AUC value	Final Score
0.02579	0.8259	0.5918	**0.4811**

7.2.4.5 Fine-Tuning

En aquesta secció s'utilitzen diferents canvis progressius en diversos paràmetres del model per a intentar millorar la seva sortida. El millor model fins ara és el de l'augment de dades on hem obtingut una precisió i Recall del 50% sobre les dades de validació. Per tant, l'usarem com a model de partida i farem diverses variacions per a veure si obtenim un millor resultat.

Per tant, la configuració de cada execució és: Augment de dades, càrrega de pesos d'ImageNet i entrenament de totes les capes tal com mostra la Taula 22:

Taula 22. Configuració de l'experiment

Training Detail	Inception
Data Augmentation	Yes
Transfer Learning	Yes
Weights	Pre-trained on ImageNet
Last Layer	GlobalAveragePooling2D Dense (1024, activation='relu') Dense (8, activation='sigmoid')
Class Weights	No
Feature Extraction Enabled	No
Classification Enabled	Yes
Optimizer	*To be selected in Table 23
Loss function	Binary Cross-Entropy
Early Stopping patience	8 steps for validation loss, type [min]
Number of Parameters	23,909,160
Number of trainable Parameters	23,874,728

La Taula 23 mostra els resultats de manera resumida amb la primera fila que conté l'execució a millorar com a dada de partida:

Taula 23. Resultats dels experiments mitjançant Fine-tuning

Details: Data Augmentation, Imagenet weights, all layers trained	**Val loss**	**Val accuracy**	**Val precision**	**Val Recall**	**Val AUC**	**Kappa score**	**F-1 score**	**AUC value**	**Final Score**
Optimizer = Adam lr=0.01	0.6558	0.87625	0.5050	0.5	0.79499	0.4318	0.87625	0.8406	**0.7162**
Optimizer = rmsprop lr=0.01	0.3684	0.8825	0.5340	0.47	0.8333	0.4737	0.8825	0.8375	**0.7179**
Optimizer = SGD lr=0.01, decay=0, momentum=0, nesterov=False	0.5913	0.8621	0.4422	0.3925	0.7657	0.3380	0.8621	0.7926	**0.6643**
Optimizer = SGD lr=0.001, decay=1e-6, momentum=0.9, nesterov=True	0.3820	0.8703	0.4794	0.4375	0.8490	0.3840	0.8703	0.8548	**0.7030**
Optimizer = SGD lr=0.01, decay=1e-6, momentum=0.9, nesterov=True	0.3769	0.8984	**0.6021**	**0.552**	0.855	0.5186	0.8984	0.8838	**0.7669**
Optimizer = SGD lr=0.1, decay=1e-6, momentum=0.9, nesterov=True	0.5553	0.8856	0.5420	0.5475	0.8187	0.4793	0.8856	0.8629	**0.7426**

Com podem observar, després de diversos experiments, hem trobat un optimitzador basat en SGD (Stochastic Gradient Descent) que aconsegueix una precisió del 60% amb un Recall del 55%. Aquests doncs, és el nostre guanyador usant el model Inception. La Figura 40 ens mostra la matriu de confusió on podem veure que la majoria dels elements s'han categoritzat correctament (elements en la diagonal):

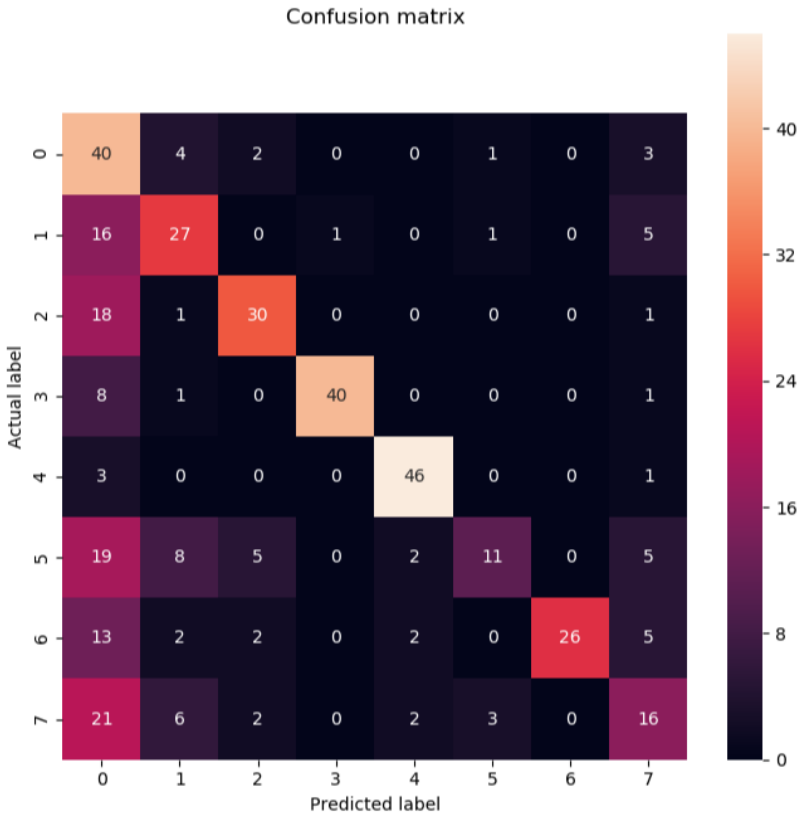


Figura 40. Matriu de confusió final

El model que s'entrega com a part d'aquest treball en format binary "h5" es pot trobar com a lliurable al següent repositori:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/releases/tag/v0.1

7.3 Model VGG-16

7.3.1 Descripció del model

L'arquitectura del model VGGNet va ser introduïda pels investigadors Simonyan i Zisserman l'any 2014 en el paper anomenat "Very Deep Convolutional Networks for Large Scale Image Recognition". Aquesta xarxa es caracteritza per la seva simplicitat, usant només convolucions de 3x3 apilades una sobre de l'altra incrementant la profunditat del model. La reducció de la mida del volum es realitza mitjançant un Max pooling. Les últimes capes estan formades de dues capes totalment connectades amb 4096 nodes i seguit d'un classificador softmax que alterem per a la descomposició del nostre problema en 8 classes diferents, mitjançant una funció sigmoid (Simonyan, 2015).

El nombre 16 indica el nombre de pesos de capes en la xarxa i aquesta és considerada del tipus molt profund. Com que aquest tipus de xarxa és lenta d'entrenar, es recomana fer ús d'aprenentatge transferit, carregant els pesos originals del dataset ImageNet, bloquejant les diferents capes perquè no s'entrenin un altre cop i afegint noves capes per al nostre problema específic i entrenar només aquesta part. La decisió d'usar aquest model és definida pels seus bons resultats en problemes de classificació d'imatges i pel resultat obtingut en treballs com els de Jaworek-Korjakowska, J., et al (2019) per a la detecció de melanomes malignes via VGG19.

El model, que suporta imatges d'entrada de 224x224 píxels és construït amb TensorFlow i Keras i el disseny d'aquest es pot trobar al següent fitxer:

- odir_model_vgg16.py

El resum del model es pot trobar al fitxer vgg_model_summary.md o a la secció 11.7 de l'annex.

El diagrama complet del model es pot veure a la Figura 41:

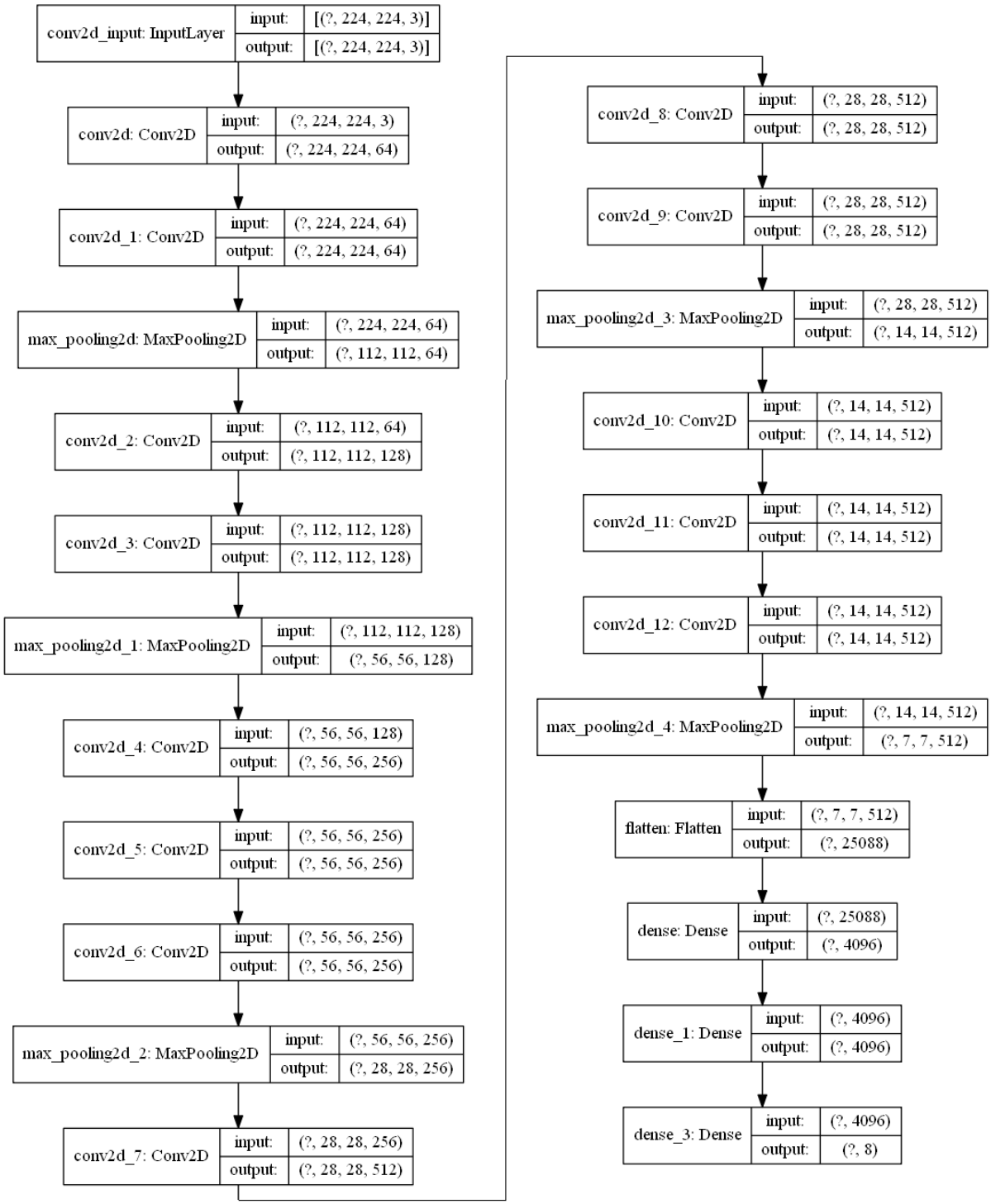


Figura 41. Model VGG16

Aquest model genera 134 milions de paràmetres amb un cost d'entrenament de 1.5h per època.

7.3.2 Procés d'entrenament

L'entrenament del model es realitza en diferents passos. Primerament, s'analitza la funció fit i en un procés més desenvolupat s'utilitza el fit_generator que ens permet l'ús de generadors per a entrar imatges sense que la memòria a consumir no es vegi tan afectada. Cada entrenament o experiment genera un codi identificador que ens permet desar-lo de manera ordenada.

El model s'inicia amb Adam (Adaptive Moment Estimation) amb un learning rate de 0.01 i durant la fase de tuning s'avalua el SGD (Stochastic Gradient Descent) amb diferents paràmetres.

En realitzar centenars de proves, aquestes han de quedar correctament registrades per la seva posterior anàlisi. Durant l'entrenament s'usaran diferents estratègies i algorismes fins a aconseguir un resultat òptim.

Durant la fase d'entrenament amb transferència de dades, configurarem les dades d'entrada mitjançant la unitat de preprocessament que ens permetrà ajustar les imatges a les dades d'ImageNet. Aquest procés consisteix a convertir les imatges de RGB a BGR i llavors centrar cada canal de color amb el zero al centre respecte al conjunt de dades ImageNet.

Cada experiment genera dades significants a més de diversos gràfics exploratoris que contenen valors com la precisió, la pèrdua, la precisió de validació, la pèrdua de validació, Recall i AUC entre d'altres. Aquestes dades, com s'ha mencionat anteriorment, queden desades respectivament amb un ID únic assignat per defecte per al seu tractament i anàlisi posterior.

7.3.3 Procés de validació

Per a validar les dades, utilitzarem les mètriques definides anteriorment i usarem també els components que l'ODIR defineix per a obtenir una puntuació general sobre l'estat de la classificació final. El procés es realitza mitjançant la càrrega en memòria d'un model prèviament entrenat i que conté el seu graf d'execució. Llavors, s'entra la imatge que volem comprovar (realitzant les transformacions pertinents perquè sigui compatible amb el model) i s'obté el resultat de la categorització.

El procés executa els passos interns necessaris i dóna un resultat a interpretar. Quan rebem els resultats del model, hem d'analitzar-los en la manera en què hem dissenyat la sortida. En el nostre cas, hem de tenir en compte que estem treballant amb un problema multi-label i per tant podem obtenir imatges amb diverses etiquetes marcades.

La validació es realitza d'inici a fi. Primer, comprovem que les dades d'entrada son realment imatges del nostre dataset, i que, estan correctament formatades i llavors, analitzem la sortida de la mateixa manera, marcant la imatge amb les diferents classes.

Un cop volem comprovar una imatge o múltiples, podem cridar l'entrenament desat anteriorment en un dels experiments i comprovar la sortida que el model ens dóna envers l'entrada. En finalitzar l'execució, es mostra la predicció envers el ground truth per a la confirmació visual de resultats.

Diferents mètriques es mostren a la sortida de l'execució de l'avaluació del model per a l'anàlisi més profund dels valors. Cadascuna d'aquestes propietats es mostren i s'analitzen en detall en els experiments.

El següent fitxer ens permet avaluar imatges via inferència pel model VGG:

- odir_vgg_testing_inference.py

7.3.4 Experiments

Els experiments inicials es realitzen amb unes imatges d'entrada de 224x224 píxels on el conjunt d'entrenament és de 6151 imatges i el conjunt de validació de 400 per defecte. En els experiments amb augment de dades, el nombre d'imatges d'entrenament varia i es fa menció del nombre d'imatges generades sintèticament. A més a més, iniciem els experiments amb:

- Aprenentatge transferit usant les dades d'ImageNet.
- Només s'entrenen les capes afegides.
- Optimitzador: Adam amb una ràtio d'aprenentatge de 0.01 (Si l'optimitzador és diferent, s'indica a l'experiment).

Cada experiment comença amb una taula indicant els paràmetres que s'han utilitzat per al seu correcte seguiment.

7.3.4.1 Experiment inicial

La configuració d'aquest experiment inicial és per a tenir un contacte inicial amb el model VGG i a partir d'aquí fer els ajustaments necessaris per a trobar la configuració que més s'adapti a les nostres necessitats. La Taula 24 mostra aquesta configuració.

Taula 24. Configuració de l'experiment

Training Detail	VGG
Data Augmentation	No
Transfer Learning	No
Weights	None
Last Layer	Dense (8, activation='sigmoid')
Feature Extraction Enabled	Yes
Classification Enabled	Yes
Optimizer	Adam lr=0.01
Loss function	Binary Cross-Entropy
Early Stopping patience	None
Number of Parameters	134,293,320
Number of trainable Parameters	134,293,320

Els primers resultats sense cap ajust, és a dir, sense afegir res al model, ens donen un comportament amb un valor de *accuracy* que sembla elevat. La Taula 25 mostra els diferents valors de l'entrenament bàsic sense cap ajustament i sense mètriques addicionals.

Taula 25. Resultats bàsics amb la xarxa VGG-16

loss	**accuracy**	**Val loss**	**Val accuracy**
0.8074	0.8633	2.3519	**0.8938**
0.7345	0.8649	2.1178	0.8934
0.6803	0.8674	2.6367	0.8844

La Taula 25 mostra les diferents execucions del model VGG-16 sobre les dades d'entrenament i validació. Sense cap ajustament, aconseguim un *accuracy* del 89.38% sobre el conjunt de validació. Tot i tenir un *accuracy* elevat, hem d'analitzar més mètriques perquè amb desequilibri de dades és habitual obtenir aquest tipus de resultats, i potser, no està reflectint bé la classificació.

Si analitzem les Figures 42 i 43 que mostren els valors de *accuracy* del model, *accuracy* de la validació, pèrdua i pèrdua de validació podem veure com el model s'està comportant:

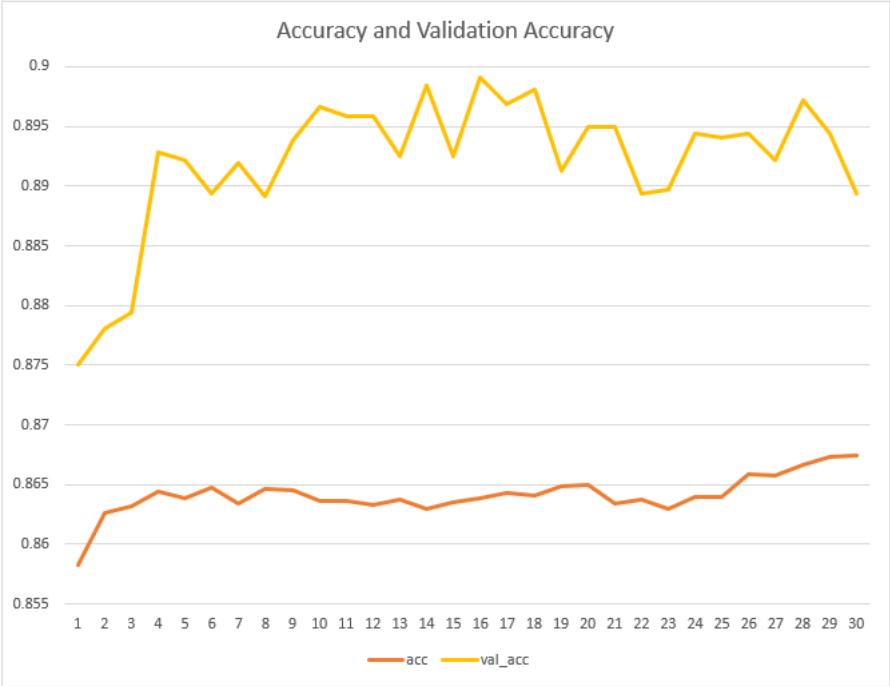


Figura 42. Precisió i Precisió de validació

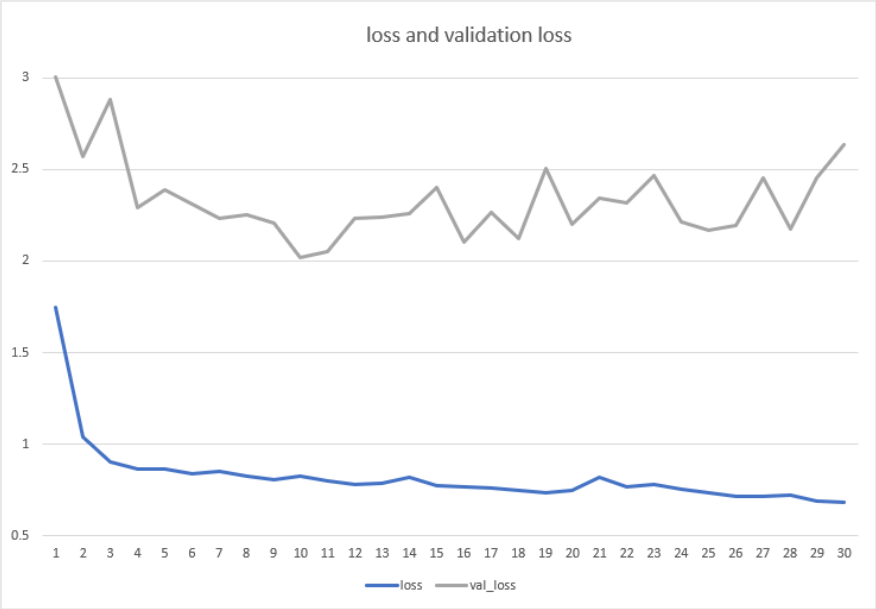


Figura 43. Pèrdua i pèrdua de validació

De les figures anteriors podem entendre que la pèrdua disminueix a mesura que l'entrenament està en execució. Si la pèrdua disminueix, llavors l'entrenament està funcionant correctament.

La precisió de validació mesura com bona és la predicció del nostre model. Si el model està aprenent, el valor de precisió s'incrementa. A diferència del model anterior, podem veure que està aprenent a un ritme més lent però incrementant la seva precisió a cada pas.

La manera més eficient que tenim és la d'analitzar la matriu de confusió. La Figura 44 mostra la matriu de confusió amb els resultats que estàvem sospitant. Les classes majoritàries són les que el classificador decideix utilitzar.

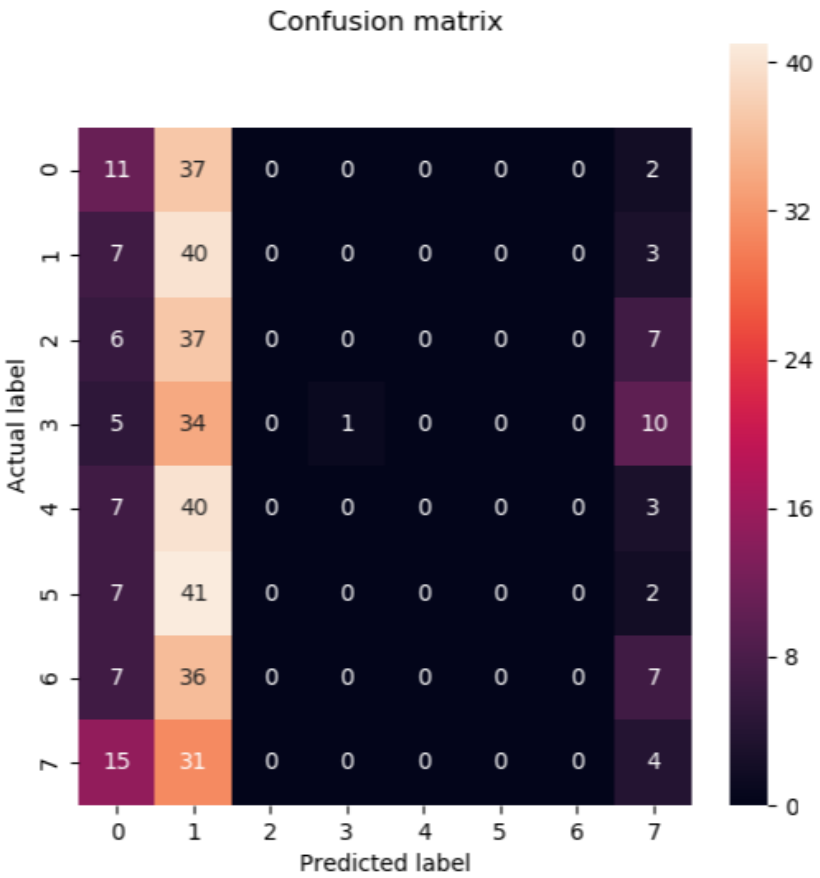


Figura 44. Matriu de confusió

En els següents experiments, s'analitzen diverses estratègies per a la millora de la precisió final.

7.3.4.2 Experiment amb aprenentatge transferit i augment de dades

L'experiment té la següent configuració (Taula 26):

Taula 26. Configuració de l'experiment

Training Detail	VGG
Data Augmentation	Yes
Transfer Learning	Yes
Weights	Pre-trained on ImageNet
Last Layer	Dense (8, activation='sigmoid')
Feature Extraction Enabled	No
Classification Enabled	Yes
Optimizer	Adam lr=0.01
Loss function	Binary Cross-Entropy
Early Stopping patience	None
Number of Parameters	134,293,320
Number of trainable Parameters	32,776

Aquest aprenentatge transferit té dos vessants. La primera és la d'usar aprenentatge transferit, carregant l'aprenentatge d'ImageNet i on només s'entrenen les capes afegides. I la segona, d'utilitzar augment de dades per ajudar al model a entendre les classes minoritàries. Els experiments són més eficients en aquest apartat a causa de l'experiència amb el model anterior.

A causa dels canvis realitzats en les mètriques afegides, observem nous valors afegits a la Taula 27 i 28. Aquestes mètriques ens aporten un valor afegit i ens ajuden a entendre si el nostre model està aprenent o no.

Taula 27. Resultats sobre les dades d'entrenament amb pesos amb la xarxa VGG-16

loss	accuracy	precision	Recall	AUC
0.6745	0.8964	0.6896	0.5425	0.9229

Taula 28. Resultats sobre les dades de validació amb pesos amb la xarxa VGG-16

Val loss	Val accuracy	Val precision	Val Recall	Val AUC
0.3517	0.8768	0.5094	0.4025	0.8014

La Taula 27 mostra les dades sobre el conjunt d'entrenament i s'observa una precisió del 68% i un Recall del 54% indicant-nos que el model no està aprenent sobre les classes minoritàries. Així i tot, la precisió de les nostres imatges aconsegueix un 50% amb un Recall del 40%, cosa que ens mostra que tenim un model de partida a millorar. Amb el valor de Recall obtenim les imatges que s'han marcat com a positiu i que realment són positives (true positive). Com que estem treballant amb un problema mèdic, els falsos negatius tenen un cost elevat per als pacients, ja que podríem classificar incorrectament una malaltia. Per tant, Recall és una mètrica que podem utilitzar per a seleccionar el millor model.

La Figura 45 mostra el resultat de l'execució de l'experiment com a reforç de les dades exposades a les Taules 27 i 28.

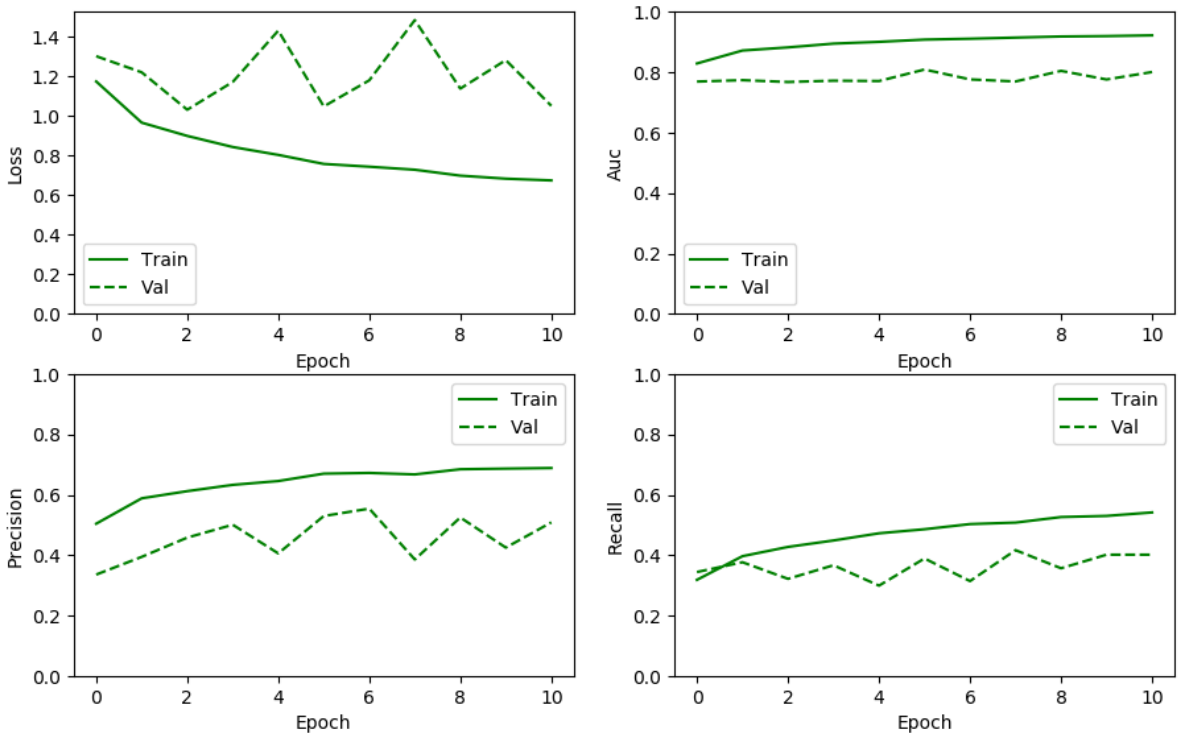


Figura 45. Execució de l'experiment amb aprenentatge transferit i model VGG16

La Figura 46 mostra el valor de *l'accuracy* d'entrenament i *accuracy* de validació:

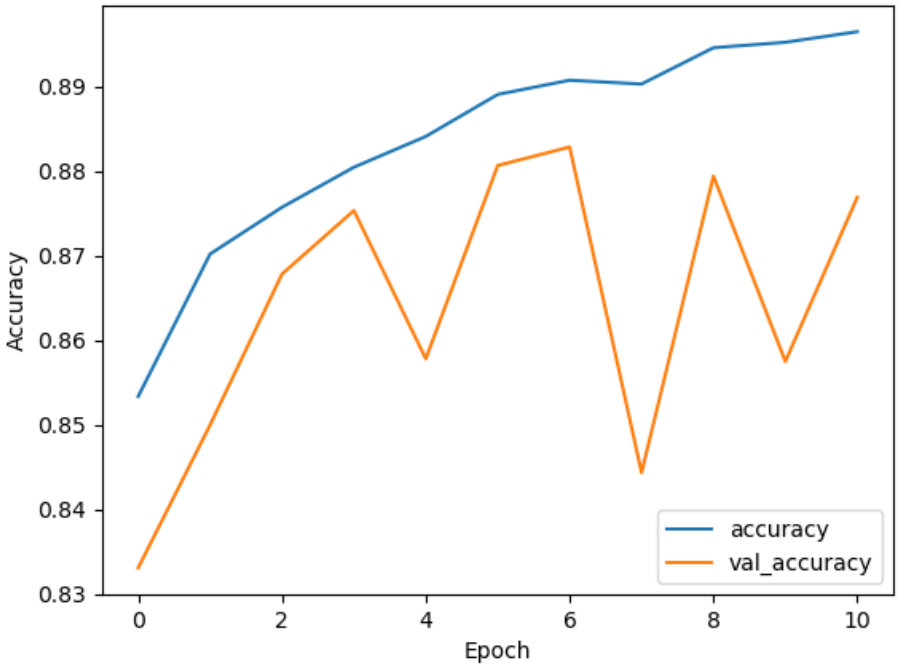


Figura 46. Accuracy d'entrenament i Accuracy de validació

La matriu de confusió de la Figura 47, mostra una diferència significant amb la matriu anterior i és que, en aquest cas, aconseguim una variació significant en el model i decideix ubicar correctament algunes de les imatges a classificar com podem observar en la diagonal de la matriu:

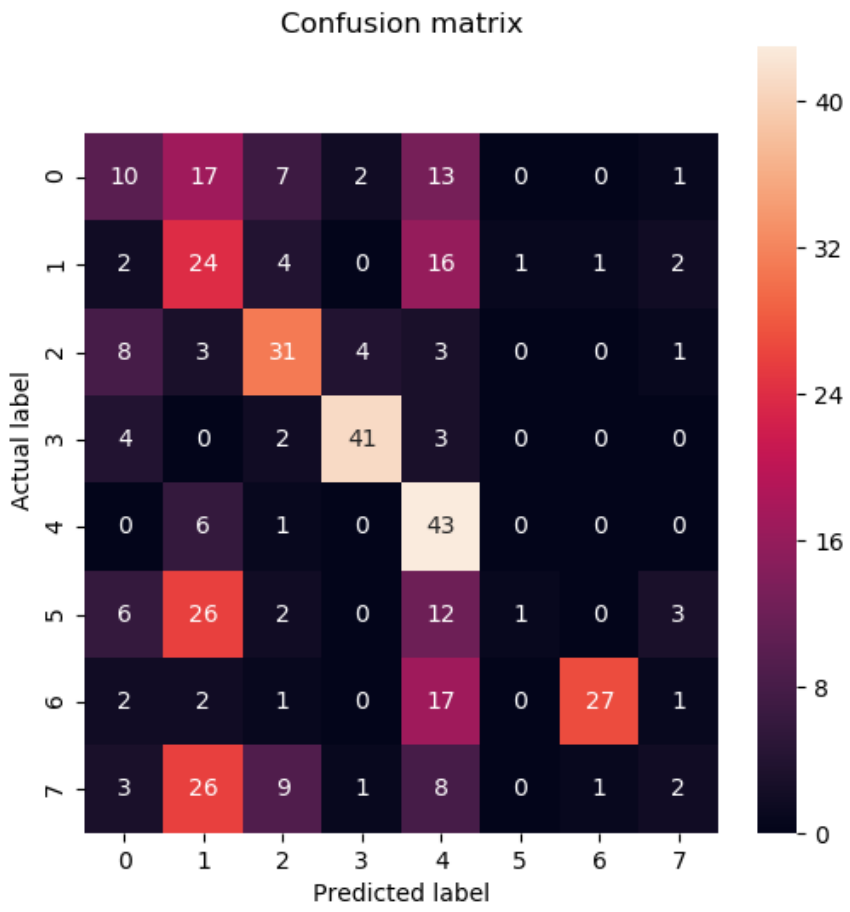


Figura 47. Matriu de confusió

Com a resultats addicionals, tenim el càlcul del F-1 score i el coeficient Kappa de Cohen:

Taula 29. Resultats sobre les dades de validació amb pesos amb la xarxa VGG-16

Kappa score	**F-1 score**	**AUC value**	**Final Score**
0.3814	0.8768	0.8073	**0.6885**

Aquest model obté una puntuació de **0.68** segons mostra la Taula 29 en finalitzar la seva execució i afegint tots els valors calculats anteriorment.

7.3.4.3 Fine-Tuning

Donat i que encara no som capaços de classificar més del 50% de les mostres de validació correctament, haurem de realitzar un fine-tuning addicional del model per aconseguir incrementar la classificació ajudant al model a aprendre sobre les classes minoritàries. Els següents experiments es resumeixen tot seguit:

Taula 30. Resultats dels experiments mitjançant Fine-tuning

Details: Data Augmentation, Imagenet weights, added layers trained	**Val loss**	**Val accuracy**	**Val precision**	**Val Recall**	**Val AUC**	**Kappa score**	**F-1 score**	**AUC value**	**Final Score**
Optimizer = Adam lr=0.01	0.3517	0.8768	0.5094	0.4025	0.8014	0.3814	0.8768	0.8073	0.6885
Optimizer = rmsprop lr=0.01	5.6789	0.8406	0.3788	0.43	0.6756	0.3112	0.8406	0.7032	0.6183
Optimizer = SGD lr=0.01, decay=0, momentum=0, nesterov=False	0.4153	0.8443	0.3607	0.3175	0.7840	0.25	0.8443	0.7930	0.6291
Optimizer = SGD lr=0.001, decay=1e-6, momentum=0.9, nesterov=True	**0.3137**	**0.8871**	**0.5776**	**0.3625**	**0.8140**	**0.3863**	**0.8871**	**0.8176**	**0.6970**
Optimizer = SGD lr=0.01, decay=1e-6, momentum=0.9, nesterov=True	0.4759	0.8656	0.4617	0.4525	0.8138	0.3804	0.8656	0.8359	0.6939
Optimizer = SGD lr=0.1, decay=1e-6, momentum=0.9, nesterov=True	0.4536	0.8687	0.4705	0.4	0.7791	0.3587	0.8687	0.7919	0.6731

Com podem observar, després de diversos experiments, hem trobat un optimitzador basat en SGD (Stochastic Gradient Descent) que aconsegueix una precisió del 57% amb un Recall del 36%. Aquests doncs, és el nostre guanyador usant el model VGG. La Figura 48 ens mostra la matriu de confusió on podem veure que la majoria dels elements s'han categoritzat correctament (elements en la diagonal):

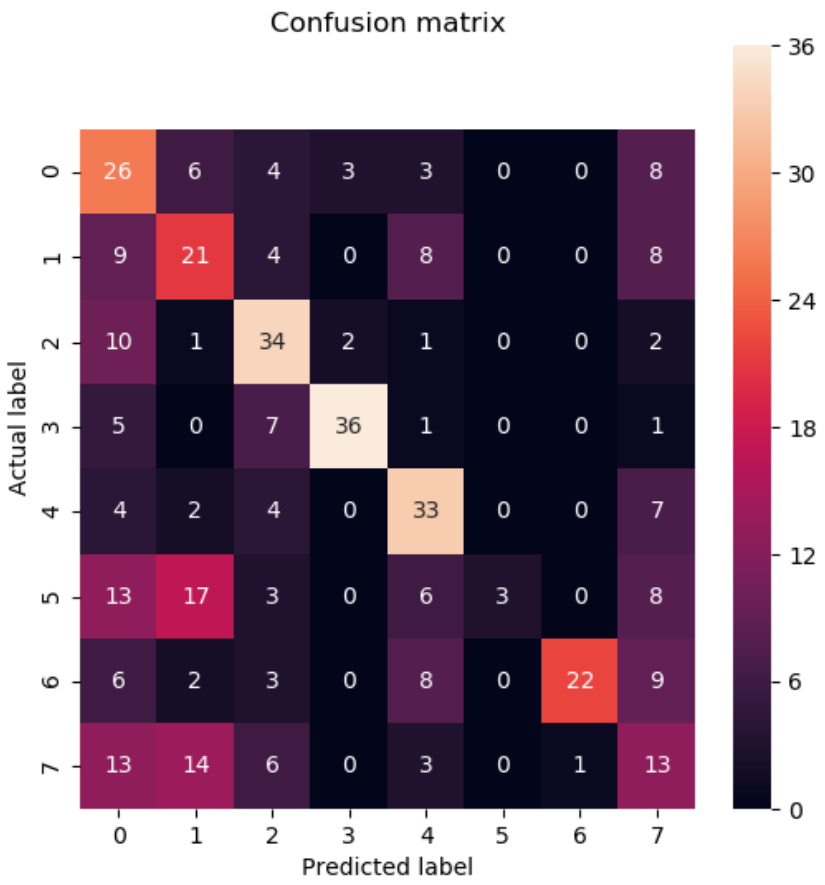


Figura 48. Matriu de confusió final

El model que s'entrega com a part d'aquest treball en format binary "h5" es pot trobar com a lliurable al següent repositori:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/releases/tag/v0.2

8 Discussió

Els resultats dels experiments dels dos models de deep learning seleccionats demostren certa efectivitat en la detecció de patologies oculars aconseguint un 60% de precisió de validació en el model Inception i un 55% de precisió de validació en el model VGG. Aquesta precisió ens indica el percentatge de resultats que són rellevants. D'altra banda, hem d'observar també el valor de Recall perquè tenim un problema amb múltiples etiquetes entre mans i aquest valor ens ofereix el percentatge de resultats rellevants totals classificats correctament pels models. En el model Inception aconseguim un Recall del 55% mentre que en VGG aconseguim un Recall del 36%.

Aquestes dues mètriques d'avaluació del model són extremadament importants. Malauradament, no és possible maximitzar aquestes dues mètriques al mateix temps, ja que una depèn de l'altra. Existeix però, una mètrica disponible anomenada F-1 score que és un mitjà harmònic de precisió i Recall. En el nostre cas, aquesta mètrica ha sigut important a l'hora de seleccionar els models finals donant-nos una mètrica molt similar (89% per Inception i 88% per VGG) per cada model.

Com que les dades provenen d'un dataset públic, els organitzadors suggereixen que s'utilitzi el coeficient Kappa Cohen i el valor AUC també per a obtenir una puntuació més adient amb la complexitat del problema que tenim entre mans. La Taula 31 ens mostra els resultats obtinguts per a cada model amb els valors descrits per F1-score, AUC i Kappa amb la puntuació final en l'última columna.

Taula 31. Resultat dels models

Model	**Val loss**	**Val accuracy**	**Val precision**	**Val Recall**	**Val AUC**	**Kappa score**	**F-1 score**	**AUC value**	**Final Score**
Inception v3	0.3769	0.8984	**0.6021**	**0.552**	0.855	0.5186	**0.8984**	0.8838	**0.7669**
VGG-16	0.3137	0.8871	**0.5776**	**0.3625**	0.8140	0.3863	**0.8871**	0.8176	**0.6970**

La complexitat amb la puntuació final forma part de la sortida amb múltiples etiquetes tal com mostra la Figura 49. Podem observar com el model classifica de manera correcta l'ull en la categoria Normal, però a més a més l'afegeix en la categoria Others fent que la classificació final sigui incorrecta.

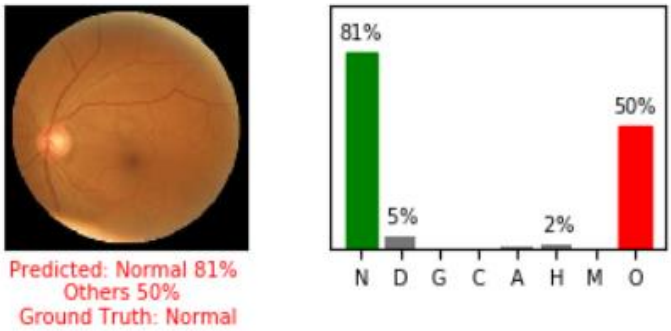


Figura 49. Predicció de sortida

La Taula 32 mostra la configuració dels models finals seleccionats com a part d'aquest treball i que ens han donat un resultat que trobem interessant.

Taula 32. Configuració dels models

Training Detail	Inception	VGG
Data Augmentation	Yes	Yes
Transfer Learning	Yes	Yes
Weights	Pre-trained on ImageNet	Pre-trained on ImageNet
Last Layer	GlobalAveragePooling2D Dense (1024, activation='relu') Dense (8, activation='**sigmoid**')	Dense (8, activation='**sigmoid**')
Feature Extraction Enabled	Yes	No
Classification Enabled	Yes	Yes
Optimizer	SGD lr=0.01, decay=1e-6, momentum=0.9, nesterov=True	SGD lr=0.001, decay=1e-6, momentum=0.9, nesterov=True
Loss function	Binary Cross-Entropy	Binary Cross-Entropy
Early Stopping patience	8 steps for validation loss, type [min]	8 steps for validation loss, type [min]
Number of Parameters	23,909,160	134,293,320
Number of trainable Parameters	**23,874,728**	**32,776**

Cada model té dues parts, una referent a l'extracció de característiques i l'altra de classificació. Com podem observar a la Taula 32, l'única diferència entre els dos models utilitzats és que amb

Inception, hem aconseguit un millor resultat habilitant l'extracció de característiques mentre que amb VGG només ens ha calgut entrenar la part del classificador.

Una altra mètrica que ens pot ajudar a comprovar de manera visual com s'estan comportant els models, és mitjançant la matriu de confusió que és mostra a la Figura 50.

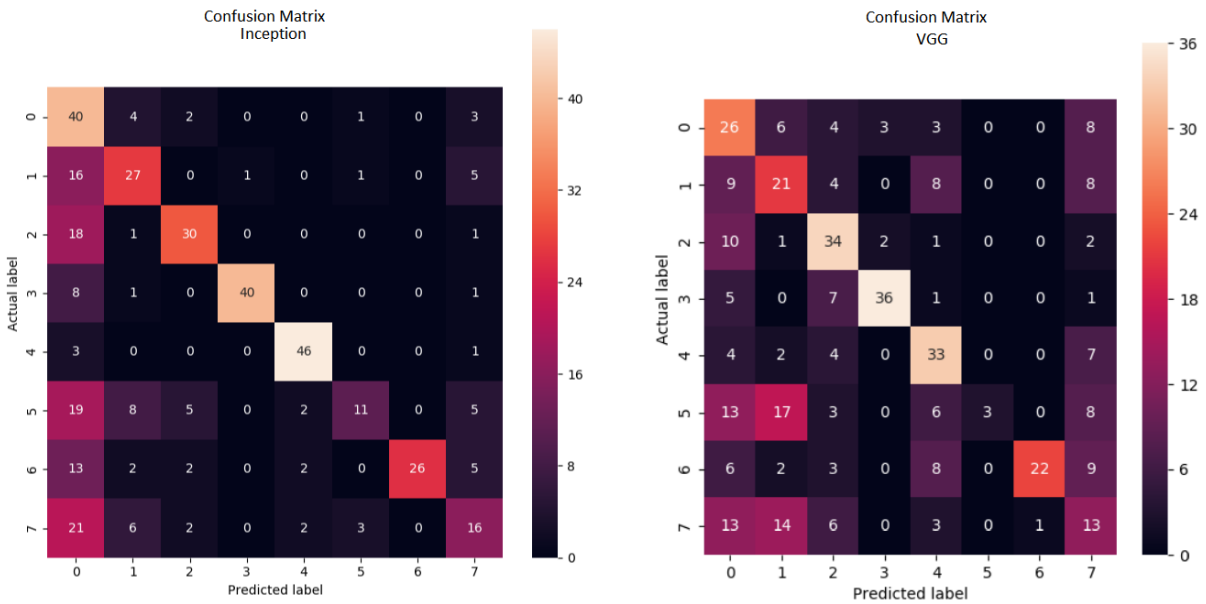


Figura 50. Matrius de confusió (Inception vs. VGG)

Tots dos models estan classificant patologies de manera correcta, ubicant les imatges en la categoria correcta. Aquest factor es pot observar en la diagonal de la matriu de confusió. Hi ha un total de 400 imatges a categoritzar en el conjunt de validació i s'observa com la part d'augment de dades ha ajudat a fer que els models aprenguessin sobre les classes minoritàries.

Ara bé, cal un treball addicional per a perfeccionar aquesta categorització mitjançant altres mètodes. El problema de categoritzar múltiples patologies és d'una complexitat molt elevada i d'aquí que la literatura referent a deep learning i malalties oculars només tracti de cobrir una sola patologia. Com que aquest conjunt de dades forma part d'un repte, seria interessant veure com els guanyadors d'aquest han solucionat els diferents problemes amb els quals m'he trobat

durant l'anàlisi dels experiments. Per a comprovar l'eficàcia del millor model trobat en aquest treball (Inception), he decidit enviar els resultats de predicció al concurs amb els següents resultats obtinguts mostrats en la Figura 51:

Result

User

jordi.corbilla

Challenge

odir2019

Submission created

Dec. 26, 2019, 2:32 p.m.

Result created

Dec. 26, 2019, 2:32 p.m.

Position on leaderboard

151

Visibility

This result is published on the leaderboard

Metrics


```
{
  "Score": {
    "kappa": 0.44652105446357304,
    "AUC_vlaue": 0.8464614317378988,
    "f-1_score": 0.8554999999999999,
    "Final_Score": 0.716160828733824
  }
}
```


Figura 51. Resultats ODIR amb model Inception

S'observa doncs que obtenim una puntuació final del 71.6% sobre 1000 imatges on no coneixem el resultat d'aquestes, ja que formen part del repte. El guanyador del repte obté una puntuació final del 89%.

En resum, les dues solucions de deep learning proposades donen resultats raonables en la detecció de lesions i la seva classificació. Això i tot, encara hi ha algunes imperfeccions com es mostra en les figures anteriors. Cal doncs, millorar les mostres d'entrenament i estudiar altres

solucions o definir altres estratègies per a cobrir totes les patologies. Potser seria més adient realitzar un model per patologia i per ull. I després analitzar els resultats al final. Tot i sonar més dificultòs, pot ser que ens doni millor resultats. No obstant això, estem limitats pel temps i els recursos actuals i hem intentat tot el possible per assegurar la correctesa de les mostres i que els resultats dels models fossin raonables.

9 Conclusió

Els nostres resultats mostren la viabilitat de classificar múltiples malalties oculars mitjançant solucions estàndard de deep learning amb l'ús d'aprenentatge transferit amb models carregats amb aprenentatge no mèdic i l'ús d'augment de dades per a corregir un problema de desequilibri trobat en les dades. Els resultats obtinguts a cada model s'observen a la Figura 52 amb els valors finals d'entrenament i classificació de validació.

Inception v3:	VGG-16
Training:	**Training:**
• loss: 0.0485 • accuracy: 0.9812 • precision: 0.9460 • recall: 0.9252 • AUC: 0.9969	• loss: 0.7054 • accuracy: 0.8929 • precision: 0.6998 • recall: 0.4799 • AUC: 0.9168
Validation:	**Validation:**
• loss : 0.37697273850440977 • accuracy : 0.8984375 • precision : 0.6021798 • recall : 0.5525 • AUC : 0.85563624	• loss : 0.31377036929130553 • accuracy : 0.8871875 • precision : 0.57768923 • recall : 0.3625 • AUC : 0.8140241
Final Score:	**Final Score:**
• Kappa score: 0.5186967789707515 • F-1 score: 0.8984375 • AUC value: 0.8838098214285715 • Final Score: 0.7669813667997744	• Kappa score: 0.38631534211644714 • F-1 score: 0.8871875 • AUC value: 0.8176205357142858 • Final Score: 0.6970411259435777

Figura 52. Resum de resultats

En conclusió, podem dir que aquest és un dels primers treballs en la classificació múltiple de malalties oculars mitjançant deep learning i que demostra que l'ús de models de l'estat de l'art poden ser suficients per a les tasques de predicció de patologies mèdiques en general si les patologies estan correctament representades en el conjunt de dades inicial. També podem dir

que hem gaudit molt d'aquest projecte primer perquè ens ha obert les portes cap a un món molt interessant com és el deep learning i l'aprenentatge de màquina i perquè és un treball que pot ajudar en un futur a altres treballs relacionats amb l'oftalmologia i l'anàlisi del fons d'ull.

9.1 Dificultats

Els projectes d'aprenentatge de màquina són en si dificultosos a causa de totes les incògnites que ens anem trobant i totes les proves que hem de realitzar fins a trobar una cosa que funciona. En el nostre cas, vam començar des de zero. Aprenent sobre el deep learning, mirant vídeos i tutorials i recursos accessibles a través de la Universitat de Stanford i llavors iniciant-nos en la plataforma TensorFlow i Keras.

La corba d'aprenentatge és significant aquí, ja que per a poder començar amb el primer classificador, primer hem d'aprendre tots els conceptes bàsics, llegir la literatura sobre els problemes mèdics i llavors entendre com es construeixen els models en Python i com es configura el maquinari per a usar la GPU.

La següent dificultat a mencionar és la del tractament del conjunt de dades ODIR. Aquest conjunt de dades és de pacients reals i totes les dades tenen formats i resolucions diferents cosa que va dificultar poder treballar amb elles. La selecció d'estratègies per a treballar amb aquestes dades és també un aspecte a mencionar. Tenim un temps limitat a fer el projecte, per tant, fer l'elecció de les estratègies a seguir a l'inici del projecte té un impacte molt gran. L'elecció de tractar cada imatge com a única pot haver sigut encertada perquè només hem hagut de generar un model que és capaç de consumir imatges d'ulls, sense considerar si és l'ull esquerre o l'ull dret. D'altra banda, pot comportar problemes, ja que cada ull és subtilment diferent i hi ha estudis que ho demostren.

Creiem que l'elecció d'analitzar dos models no ha sigut encertada a causa de la quantitat de temps que es triga a construir el model, entendre'l i entrenar-lo. Però si ens ha servit per a comprovar quin dels dos era millor en funció dels resultats de sortida. Així i tot, creiem que hauríem d'haver fet un projecte més senzill donat la quantitat de temps i recursos disponibles. Per exemple, avaluar només les malalties de les classes majoritàries o només avaluar un dels models de l'estat de l'art.

Tot i que els models s'han seleccionat a causa de la possibilitat d'execució sobre maquinari d'escriptori, ens hem trobat sovint amb errors de memòria plena i hem hagut de buscar alternatives per a poder continuar amb l'entrenament dels models.

En conclusió, l'aprenentatge de màquina és realment això. Arribar a un punt on tenim tots els models configurats i funcionant i llavors començar a modificar diversos paràmetres per ajustar els valors de sortida per a veure quin ens dóna una classificació millor.

9.2 Objectius assolits

Creiem que hem assolit tots els objectius definits en aquest treball. Primerament hem fet l'aprenentatge inicial que aquest projecte ens requereix i llavors hem estudiat el problema a fons i proposat dues solucions de deep learning per a la solució del problema mèdic.

Finalment, s'han desenvolupat els models, entrenat i realitzat diversos experiments fins a trobar una sortida adient basant-nos en la complexitat del problema i considerant el temps i recursos disponibles.

10 Tecnologia

10.1 Frameworks utilitzats

La plataforma utilitzada per a realitzar el nostre treball és TensorFlow 2.0 amb la API Keras 2.3.1. El llistat de llibreries utilitzades complet és:

```
-  tensorboard-2.0.0
-  tensorflow-2.0.0
-  tensorflow-estimator-2.0.1
-  tensorflow-gpu-2.0
-  matplotlib-3.1.1
-  keras-applications-1.0.8
-  keras-preprocessing-1.0.5
-  opencv-python-4.1.1.26
-  django-2.2.6
-  image-1.5.27
-  pillow-6.2.0
-  sqlparse-0.3.0
-  IPython-7.8.0
-  keras-2.3.1
-  scikit-learn-0.21.3
-  pydot-1.4.1
-  graphviz-0.13.2
-  pylint-2.4.4
-  imbalanced-learn-0.5.0
-  seaborn-0.9.0
-  scikit-image-0.16.2
```

10.2 Preparació del maquinari

Totes les proves s'han realitzat en un maquinari d'escriptori estàndard amb una única GPU. Per a la seva configuració, s'ha instal·lat TensorFlow GPU i s'han realitzat diferents proves per garantir que les dades es carreguessin en la GPU.

El maquinari té la següent composició:

- **Processador**: Intel(R) Core(TM) i7-4720HQ CPU @ 2.60

Instal·lar el driver correcte per a la versió corresponent de la targeta gràfica, sistema operatiu i versió de TensorFlow:

- https://developer.nvidia.com/cuda-10.0-download-archive?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exenetwork

Configuració del Path:

```
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\libnvvp;C:\Program
Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin;
```

La comprovació del seu funcionament es pot observar en el fixer gpu_configuration_output.md:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/blob/master/gpu_configuration_output.md

Al final de l'execució del script podem observar com la GPU és accessible i TensorFlow l'està utilitzant per a carregar les dades.

11 Annexos

11.1 Fitxer d'anotacions de les dades

S'annexa amb la memòria el fitxer original de les anotacions per a les imatges que el repte ODIR entrega dins el seu concurs públic.

- ODIR-5K_Training_Annotations(Updated)_V2.xlsx

11.2 Fitxer d'explicacions especials

Special Explanation on ODIR-5K Database and Annotations

1. The annotated classification labels are determined by the following rules
(1) The classification labels of one patient depends on left and right fundus images and corresponding diagnosis keywords;
(2) One patient is classified as normal if and only if both left and right diagnosis keywords are "normal fundus";
(3) The classification labels are decided by the other fundus image when one of fundus images is marked as "normal fundus";
(4) Treat all suspected diseases or abnormalities as diagnosed diseases or abnormalities.

2. Special words that appeared in diagnostic keywords
(1) The appearance of the two keywords "anterior segment image" and "no fundus image" are not classified into any of the eight categories in this competition.
For example, there are two anterior segment images in ODIR-5K database, 1706_left.jpg and 1710_right.jpg.
In this case, the patient's classification labels are only judged by the other fundus image of the same patient.
In addition, it is very important to note that the diagnostic keyword for 4580_left.jpg image is "no fundus image".
Because this image is actually not the left fundus image of this patient, it is from a rotation of right fundus image.
The introduction of these two diagnostic keywords can also be one of the challenges in this competition.
(2) The keywords "lens dust", "optic disk photographically invisible", "low image quality" and "image offset" do not play a decisive role in determining patient's labels.

3. The background of the following images is quite different from the rest ones. They are fundus images uploaded from the hospital.
We are sure that these images are pre-processed. You can decide by yourself whether or not to train these images in the model.
These images include
2174_right.jpg
2175_left.jpg

```
2176_left.jpg
2177_left.jpg
2177_right.jpg
2178_right.jpg
2179_left.jpg
2179_right.jpg
2180_left.jpg
2180_right.jpg
2181_left.jpg
2181_right.jpg
2182_left.jpg
2182_right.jpg
2957_left.jpg
2957_right.jpg
```

11.3 Codi Font

El codi font del projecte s'ha entregat en un fitxer zip com a part de la memòria i també es pot trobar el següent repositori en línia:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow

11.4 Llistat d'imatges descartades

El llistat complert d'imatges descartades es pot trobar al fitxer list_discarded_images.md com a part dels fitxers entregats, o en línia a:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/blob/master/list_discarded_images.md

11.5 Llistat d'imatges amb un nou ground truth

La sortida de l'execució de l'algorisme generat es pot trobar al fitxer generated_ground_truth.md com a part dels fitxers entregats, on en línia a:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/blob/master/generated_ground_truth.md.

11.6 Resum del model Inception

La sortida del model és disponible en el següent fitxer:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/blob/master/inception_model_summary.md

11.7 Resum del model VGG

La sortida del model és disponible en el següent fitxer:

- https://github.com/JordiCorbilla/ocular-disease-image-recognition-tensorflow/blob/master/vgg_model_summary.md

12 Glossari

Per ordre d'aparició:

- **PIB** – Producte Interior Brut
- **ILSVRC** - ImageNet Large Scale Visual Recognition Challenge
- **CAD** - computer aided diagnostic
- **IIAI** - Inception Institute of Artificial Intelligence
- **JPEG -** Joint Photographic Experts Group
- **CNN** - Convolutional Neural Networks
- **ODIR** - Ocular Disease Intelligent Recognition
- **AMD** – Age-related macular degeneration
- **AUC** – Area Under the curve

13 Bibliografia

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., and, ... (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

A. Lee, M. Seattle, P. Taylor, U. Kingdom. Machine learning has arrived! Ophthalmology. (2017), pp. 1726-1728.

Bengio, Y., Courville, A. and Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), pp.1798-1828.

Bonet, E. (2018). *Què és un fons d'ull?* Servei d'Oftalmologia. Fundació Hospital de nens de Barcelona, p.1.

Carletta, Jean. (1996) Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics, 22(2), pp. 249–254

C.I. Sánchez, M. Niemeijer, I. Išgum, A.V. Dumitrescu, M.S.A. Suttorp-Schulten, M.D. Abràmoff and B. van Ginneken. "Contextual computer-aided detection: Improving bright lesion detection in retinal images and coronary calcification identification in CT scans", Medical Image Analysis 2012;16(1):50-62

C.I. Sánchez, M. Niemeijer, A.V. Dumitrescu, M.S.A. Suttorp-Schulten, M.D. Abràmoff and B. van Ginneken. "Evaluation of a Computer-Aided Diagnosis system for Diabetic Retinopathy screening on public data", Investigative Ophthalmology and Visual Science 2011;52:4866-4871.

Fawcett, Tom (2006) "An Introduction to ROC Analysis"(PDF). Pattern Recognition Letters. 27 (8): 861–874.

García, B., De Juana, P., Hidalgo, F and Bermejo, T. (2010). *Oftalmología*. Farmacia hospitalaria Tomo II. Publicado por la SEFH. Capítulo 15.

Garrido, R. (2011). Epidemiología descriptiva del estado refractiva en estudiantes universitarios. *Universidad Complutense de Madrid*, p.339.

Gilbert, C., Foster A. (2001). Childhood blindness in the context of VISION 2020 – the right sight. Bull World Health Organ, 79(3):227-32.

Hijazi, S., Kumar, R. Rowen, C. Using Convolutional Neural Networks for Image Recognition. 2015. Cadence.

Jaworek-Korjakowska, J., Kleczek, P., Gorgon, M. Melanoma Thickness Prediction Based on Convolutional Neural Network With VGG-19 Model Transfer Learning. (2019) The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Kent, Allen; Berry, Madeline M.; Luehrs, Jr., Fred U.; Perry, J.W. (1955). "Machine literature searching VIII. Operational criteria for designing information retrieval systems". *American Documentation*. 6 (2): 93. doi:10.1002/asi.5090060209. arXiv preprint arXiv: 1409.4842

Krizhevsky, A., Sutskever, I. and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp.84-90.

Kusumoto D, Yuasa S. The application of convolutional neural network to stem cell biology. Inflamm Regen. 2019 39:14. Published 2019 Jul 5. doi:10.1186/s41232-019-0103-3.

Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–444.

Lyana N. M., & Norshahrizan, N., (2015). The Self-Driving Car.

M.D. Abramoff, Y. Lou, A. Erginay, et al. Improved automated setection of diabetic retinopathy on a publicly available dataset through integration of deep learning Invest Ophthalmol Vis Sci. (2016), pp. 5200-5206

Ocular Disease Intelligent Recognition. (2019). *ODIR-2019 - Grand Challenge*. [online] Available at: https://odir2019.grand-challenge.org/introduction/ [Accessed 27 Sep. 2019].

Oftalmològica. (2019). *Tecnologia per a la revisió de la retina - Àrea Oftalmològica Avançada.*

Parampal S. Grewal, Faraz Oloumi, Uriel Rubin, Matthew T.S. Tennant, Deep learning in ophthalmology: a review, Canadian Journal of Ophthalmology, Volume 53, Issue 4, 2018, Pages 309-313, ISSN 0008-4182,

Roletschek, R., (2019). [image] Available at: Fahrradtechnik auf fahrradmonteur.de [FAL or GFDL 1.2 license] [Accessed 3 Nov. 2019].

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), pp.211-252.

Saine, P. and Tyler, M. (2002). *Ophthalmic photography*. Boston [Mass.]: Butterworth-Heinemann.

Sermanet, Pierre, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. *arXiv:1312.6229* [Cs], December, 2013.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Published as a conference paper at ICLR 2015*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erham, D., Vanhoucke, V., Rabinovich, A. (2014). Going Deeper with Convolutions.

Ting DSW, Pasquale LR, Peng L, et al Artificial intelligence and deep learning in ophthalmology British Journal of Ophthalmology 2019;103:167-175.

Tan, N. M., Liu, J., Wong, D. W. K., Lim, J. H., Li, H., Patil, S. B., Yu, W., Wong, T. Y. (2009). Automatic Detection of Left and Right Eye in Retinal Fundus Images. Springer Berlin Heidelberg. pp 610—614.

Van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). Butterworth-Heinemann.

Xie, S., Girshick, R. and Dollár, P. and Tu, P. and He, K. (2016). Aggregated Residual Transformations for Deep Neural Networks. *Cornell University*.

Yorston, D. (2003). Retinal Diseases and VISION 2020. Community Eye Health. 2003;16(46):19–20.

Zhou, Y., He, X., Huang, L., Liu, L., Zhu, F., Cui, S., Shao, L. (2019). Collaborative Learning of Semi-Supervised Segmentation and Classification for Medical Images. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).