



UNIVERSITAT DE
BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques i Informàtica
Universitat de Barcelona**

**PROGAN PER A LA GENERACIÓ DE
RESSONÀNCIA MAGNÈTICA DE MAMA**

Autor: Carlos Arranz Sánchez

Director: Dr. Oliver Fernando Díaz Montesdeoca

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 12 de juny de 2022

Resum

Actualment, han estat implementats múltiples models d'aprenentatge automàtic per a l'assistència del personal de radiologia en el procés de presa de decisions durant l'anàlisi d'imatges mèdiques a la recerca de lesions canceroses, tant benignes com malignes. Malgrat això, aquests models en la seva majoria no poden ser entrenats amb suficients imatges, a causa de les complicacions ètiques i legals per a l'obtenció d'aquestes.

Per a això, l'objectiu d'aquest estudi ha estat definit com la implementació d'un sistema d'un model d'aprenentatge profund, més concretament una Progressively Growing GAN, mitjançant el qual puguin ser generades imatges mèdiques bidimensionals de ressonàncies magnètiques de mama a partir d'un conjunt d'altres imatges mèdiques bidimensionals que van ser subministrades pel Consorci Corporació Sanitària Parc Taulí.

Malgrat la falta de coneixement radiològic i capacitat computacional, els resultats del model poden ser vists com a satisfactoris, pel fet que l'estructura de les mames és conservada, juntament amb una aproximació dels teixits d'aquestes.

No obstant això, si aquestes imatges pretenen ser usades en un altre projecte, el model hauria de ser entrenat en una màquina amb major capacitat computacional mitjançant la qual pugui ser entrenat el model durant més temps.

Resumen

Actualmente, han sido implementados múltiples modelos de aprendizaje automático para la asistencia del personal de radiología en el proceso de toma de decisiones durante el análisis de imágenes médicas en busca de lesiones cancerosas, tanto benignas como malignas. Pese a ello, dichos modelos en su mayoría no pueden ser entrenados con suficientes imágenes, debido a las complicaciones éticas y legales para la obtención de las mismas.

Para ello, el objetivo de este estudio ha sido definido como la implementación de un sistema de un modelo de aprendizaje profundo, más concretamente una Progressively Growing GAN, mediante el cual puedan ser generadas imágenes médicas bidimensionales de resonancias magnéticas de mama a partir de un conjunto de otras imágenes médicas bidimensionales que fueron suministradas por el Consorci Corporació Sanitària Parc Taulí.

Pese a la falta de conocimiento radiológico y capacidad computacional, los resultados del modelo pueden ser vistos como satisfactorios, debido a que la estructura de las mamas es conservada, junto con una aproximación de los tejidos de las mismas.

No obstante, si estas imágenes pretenden ser usadas en otro proyecto, el modelo debería ser entrenado en una máquina con mayor capacidad computacional mediante la cual pueda ser entrenado el modelo durante más tiempo.

Abstract

Currently, multiple machine learning models have been implemented to assist radiology staff in the decision-making process during the analysis of medical images for cancerous lesions, both benign and malignant. Despite this, such models mostly cannot be trained with sufficient images, due to the ethical and legal complications in obtaining the images.

For this purpose, the objective of this study has been defined as the implementation of a deep learning model system, more specifically a Progressively Growing GAN, by means of which two-dimensional medical images of breast MRIs can be generated from a set of other two-dimensional medical images that were provided by the Consorci Corporació Sanitària Parc Taulí.

Despite the lack of radiological knowledge and computational capacity, the results of the model can be seen as satisfactory, because the structure of the breasts is preserved, together with an approximation of the breast tissues.

However, if these images are intended to be used in another project, the model should be trained on a machine with higher computational capacity by which the model can be trained for a longer time.

Agraïments

Principalment, voldria expressar el meu més profund agraïment al Dr. Oliver Fernando Díaz Montesdeoca, qui ha estat el meu tutor en el present treball. Aquest ha realitzat una excel·lent labor guiant-me durant la realització del treball, a més de que ha creat un clima de treball molt sa i desenfadat, la qual cosa li agraeixo molt.

També desitjo mostrar el meu més profund agraïment a Richard Osuala, qui també ha estat com un guia per a mi amb tot el seu coneixement sobre les Generative Adversarial Networks i tot el relacionat amb elles.

També volia fer especial menció als meus amics de la universitat i a la meva família per animar-me a seguir endavant i a donar-me forces per donar el millor de mi mateix durant la realització d'aquest treball i aguantar com donava la xapa sempre amb el mateix tema.

Gràcies eternes a tota aquesta gent que ha fet aquest treball possible. You are the real MVP.

Índex

1	Introducció	8
1.1	Contextualització	8
1.2	Objectius i motivació	10
1.3	Estructura del document	11
2	Estat de l'art	13
2.1	Anàlisi de diferents arquitectures GAN	13
2.1.1	Conditional GAN	13
2.1.2	Auxiliary Classifier GAN	14
2.1.3	Deep Convolutional GAN	14
2.1.4	Wasserstein GAN	15
2.1.5	Progressively Growing GAN	16
2.2	Conclusions	17
3	Materials i mètodes	18
3.1	Materials	18
3.1.1	Base de dades	18
3.1.1.1	Preprocessat de la base de dades	18
3.1.1.1.1	Conversió del format DICOM a imatges bidimensionals	18
3.1.1.1.2	Optimització de l'estructura de subdirectoris	20
3.1.2	Llenguatge de programació Python	20
3.1.2.1	PyTorch	21
3.1.2.2	Python Imaging Library	21
3.1.2.3	TensorBoard	21
3.2	Metodologia	22
3.2.1	Algorisme DFS per a subarbres expandits	22
3.2.2	Filtres d'imatge	22
3.2.3	Deep Learning	23
3.2.3.1	Funció d'activació	23
3.2.3.1.1	Rectified linear unit	23
3.2.3.1.2	Leaky rectified linear unit	24
3.2.3.2	Optimitzadors	24
3.2.4	Generative Adversarial Networks	25
3.2.4.1	Topologia del model	25

3.2.4.1.1	Generador	26
3.2.4.1.2	Discriminador	27
3.2.4.2	Entrenament	28
3.2.5	Creixement progressiu de les GAN	30
3.2.5.1	Arquitectura multiescala	30
3.2.5.2	Incorporament progressiu de noves capes	30
3.2.5.3	Desviació estàndar del minibatch	31
3.2.5.4	Normalització de generador i discriminador	33
3.2.5.4.1	Tassa d'aprenentatge igualada	33
3.2.5.4.2	Normalització del feature vector del generador	33
3.2.6	Mètriques d'evaluació	34
3.2.6.1	Matriu de confusió	34
3.2.6.2	Fréchet inception distance	35
4	Implementació	37
4.1	Funcions auxiliars	37
4.1.1	Iterador de directoris comprimits	37
4.1.1.1	Descompressió d'un arxiu comprimit	37
4.1.1.2	Recorregut del subarbre de directoris ampliat	37
4.2	Preprocessat de les dades	38
4.3	Model de Progressive Growing of GAN's	40
4.3.1	Hiperparàmetres	40
4.3.2	Xarxes neuronals auxiliars	41
4.3.2.1	Weighted scale convolution	41
4.3.2.2	Normalització de píxels	41
4.3.2.3	Bloc de convolució personalitzat	41
4.3.3	Xarxa neuronal generativa	43
4.3.4	Xarxa neuronal discriminativa	44
5	Resultats i discussió	45
5.1	Calibratge de hiperparàmetres	45
5.2	Experiment a gran escala	50
6	Conclusions	56
6.1	Problemes trobats	56
6.1.1	Capacitat computacional	56

6.1.2	Escalabilitat del model	56
6.2	Possibles ampliacions de la implementació	57
6.2.1	Processat directe d'imatges DICOM	57
	Referències	58

1 Introducció

A manera introductòria al problema que serà abordat, és realitzada una contextualització de tot el relacionat amb aquest, per a després ser exposades les motivacions per a la realització del present treball i, per a acabar, són definits els objectius a ser complerts que han estat proposats.

1.1 Contextualització

Actualment, gràcies a unes substancials millores en la recerca clínica, l'atenció mèdica i el diagnòstic mitjançant l'anàlisi d'imatges mèdiques, han estat aconseguits uns importants augments en les taxes de supervivència de diverses malalties físiques, més concretament els diferents tipus de càncer. Malgrat aquest progrés, les morts causades per aquesta mena de dolències continuen estant posicionades com una de les principals causes de mort en l'actualitat.

No obstant això, moltes de les morts a conseqüència d'aquests patiments poden ser evitades mitjançant una ràpida i efectiva detecció precoç i, posteriorment, tractament concorde al tipus específic de lesió cancerosa. Per tant, han de ser efectuats nombrosos esforços en el perfeccionament de les tècniques d'anàlisi d'imatges mèdiques.

Un dels majors problemes concernents al diagnòstic mitjançant l'anàlisi d'imatges mèdiques és la naturalesa costosa i difícil d'aquest, pel fet que aquestes anàlisis han de ser realitzats exclusivament per personal expert en radiologia que, malgrat una dilatadíssima carrera i una més que ressenyable experiència en la detecció de qualsevol mena de patró associat a tumors cancerosos, també resulten ser susceptibles a biaixos a causa dels quals siguin condicionats, de manera inconscient i indirecta, el seu judici.

A manera d'assistència al judici d'aquest personal expert en radiologia, han estat implementades diverses col·leccions d'eines automatitzades de presa de decisions per a l'anàlisi d'imatges mèdiques. Pel fet que aquestes eines han estat basades en algorismes d'intel·ligència artificial, és generada la necessitat d'una gran i heterogènia base de dades d'imatges mèdiques perquè pugui ser perfeccionat l'entrenament de les mateixes i amb aquest, el potencial futur d'aquests models.

Malgrat que aquestes eines automatitzades puguin ser considerades com la panacea de tots els mals d'aquest àmbit, ha existit sempre un inexorable obstacle a superar, que és caracteritzat per la intrínseca dificultat d'obtenció de les imatges mèdiques amb les quals

pot ser entrenat el model i , en última instància, que pugui ser garantida la solidesa del judici d'aquest. Aquesta dificultat de recopilació d'informació de qualitat és deguda a un cúmul de motius d'allò més heterogenis, tant legals com ètics, per a l'obtenció de les imatges mèdiques, al costat de la variabilitat del format de les imatges mèdiques, sent necessitades complexes solucions algorítmiques a manera de pal·liatiu per a aquestes discordances.

En aquest escenari són introduïdes les arquitectures GAN (Generative Adversarial Networks), que poden ser definides com un sistema de dues intel·ligències artificials en el qual aquestes són confrontades, amb la finalitat última d'una millor generació de resultats, en aquest cas, imatges mèdiques. Més concretament, una GAN és composta per una xarxa neuronal generativa G , a la qual se li són subministrades unes imatges reals com a entrada, sobre les característiques de les quals han de ser basades les imatges que siguin generades per aquesta xarxa. Per l'altre costat, ha estat creada una xarxa neuronal discriminativa D , la qual ha de ser capacitada per a discernir entre si les imatges que li són subministrades han estat obtingudes directament del conjunt de dades d'entrenament o, per contra, han estat generades per la xarxa generativa. La topologia a escala de blocs d'aquesta arquitectura és mostrada en la Figura 1.

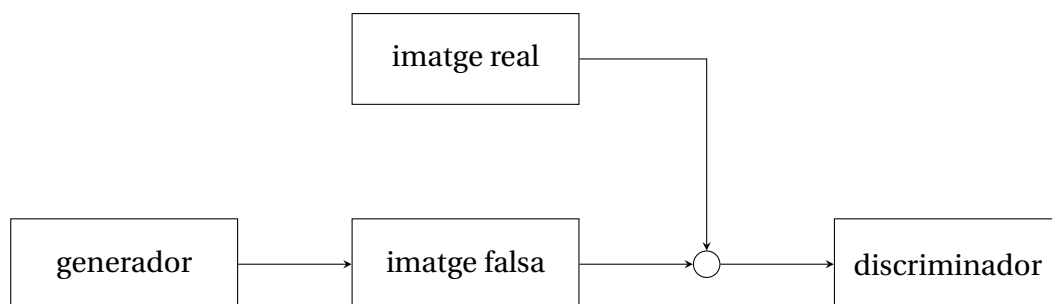


Figura 1: Esbós de la col·locació i connexions dels blocs d'una arquitectura GAN.

El potencial que és ofert per les arquitectures GAN pot ser vist com un avanç prometedor en el tractament i generació imatges mèdiques del càncer, però amb aquestes han de ser superats diversos desafiaments consubstancials a la naturalesa de tot el procés de recol·lecció de dades i de les lesions canceroses, més concretament ha de ser tingut en compte l'heterogeneïtat, bel·licositat i dificultat de detecció de les lesions canceroses i l'alt desequilibri en el format de les dades que han estat recol·lectats.

Si bé múltiples problemes que són trobats pel personal de radiologia en el diagnòstic poden ser solucionats, totalment o parcialment, mitjançant l'ús de GAN, el cas d'ús que serà estudiat en el present treball serà l'anàlisi del rendiment d'un model basat en GAN per a la generació

d'imatges de ressonàncies magnètiques (MRI) de mama.

A manera d'apèndix a la prèvia explicació, les ressonàncies magnètiques són definides com una tècnica de visualització interna del cos humà amb la qual són obtingudes imatges mèdiques d'alta qualitat, per tant, són considerades com una metodologia de gran fiabilitat per al personal de radiologia. Per a l'obtenció d'aquestes imatges, el subjecte és introduït en un escàner com el de la Figura 2.



Figura 2: Aspecte físic d'un escàner per a la realització de MRI

1.2 Objectius i motivació

L'objectiu que ha estat proposat per al present treball ha estat definit per la implementació d'un model d'aprenentatge automàtic basat en GAN mitjançant el qual puguin ser generades un conjunt d'imatges mèdiques MRI de mama mitjançant l'entrenament amb una encara més gran col·lecció d'imatges mèdiques de prova completament reals. Més concretament, dins de tot el ventall d'arquitectures derivades de models GAN que han estat creats fins a l'actualitat, la implementació a ser realitzada serà basada en la metodologia Progressive Growing of GAN's.

En el concernent a la motivació per a la realització d'aquest treball ha de ser destacat principalment:

- L'interès pel funcionament i diferents models de xarxes neuronals, més concretament els diferents tipus d'arquitectures derivades de GAN.

- Esperança que les imatges mèdiques que seran creades puguin ser utilitzades per a models d'intel·ligència artificial que puguin ser implementats en un futur.

A manera de colofó, ha de ser esmentat que aquest projecte ha estat desenvolupat sota el paraigua d'un projecte europeu que ha estat coordinat pel Barcelona Artificial Intelligence in Medicine lab (BCN-AIM lab) [1] de la Universitat de Barcelona que és anomenat EuCanImage [2], el qual és definit principalment per la seva gran dimensionalitat d'actuació, ja que són inclosos més de 20 institucions d'11 països diferents per a la creació d'una gran plataforma europea amb conjunts de dades de càncer de mama de gran qualitat i escala mitjançant els quals puguin ser millorats els models d'intel·ligència artificial amb aplicació en l'oncologia. A causa de la pertinença a un projecte de semblant envergadura i que aquest treball pugui ser ampliat i continuat en un futur, això també és considerat com una de les grans motivacions per al desenvolupament d'aquest projecte.

1.3 Estructura del document

El present document ha estat estructurat segons l'esquema següent:

1. **Introducció:** A manera introductòria al problema que ha estat abordat, és realitzada una contextualització de tot el relacionat amb aquest, per a després ser exposades les motivacions per a la realització del present treball i, per a acabar, són definits els objectius a ser complerts que han estat proposats.
2. **Estat de l'art:** És exposat l'últim nivell de desenvolupament, en el moment present, de les metodologies amb les quals pretén ser solucionat el problema a tractar, mitjançant un estudi, acarat i exposició dels articles científics que van ser escrits sobre la utilització d'arquitectures basades en GAN per a la generació i anàlisi d'imatges mèdiques de càncer.
3. **Materials i mètodes:** En aquesta secció són descrits ordenada i detalladament la totalitat dels materials que han estat utilitzats per a la realització del present estudi, al costat de les metodologies i procediments que han estat duts a terme. Aquesta exposició és realitzada perquè pugui ser avaluada la idoneïtat de la recerca, la fiabilitat i validesa dels resultats, a més que també, en un futur, pugui ser replicada la recerca de la mateixa manera en la qual va ser realitzada.
4. **Implementació:** En aquest lloc és detallat el procediment amb el qual ha estat implementat el model a partir dels materials i mitjançant les metodologies que han estat

exposades anteriorment. Aquí seran exposades també les característiques principals de la implementació de cadascun dels components sobre els quals ha estat creat el model, juntament amb una explicació del funcionament, quant a codi, dels principals algorismes que van ser implementats.

- 5. Resultats i discussió:** En aquesta secció són reportats els resultats de la recerca que ha estat realitzada a partir dels materials i mitjançant les metodologies que han estat exposades anteriorment. Els resultats seran exposats de forma completament objectiva, sense cap biaix o interpretació, amb l'objectiu últim que pugui ser confirmada la hipòtesi inicial o, per contra, que sigui desmentida la hipòtesi inicial i puguin ser considerades noves perspectives d'enfocament mitjançant les quals pugui ser articulada una nova recerca.
- 6. Conclusions:** Aquí són exposats i reiterats els punts principals de la recerca a manera de resum, juntament amb les conclusions que han estat obtingudes a través d'una anàlisi dels resultats que van ser obtinguts anteriorment. Mitjançant aquesta exposició són articulades unes consideracions, tant a manera de problemes que han estat trobats i les seves respectives solucions, juntament amb unes possibles extensions futures del present treball.

2 Estat de l'art

Avui dia, diversos models basats en extensions de Generative Adversarial Networks han estat utilitzats de manera reeixida per a diverses tasques concernents a la generació d'imatges mèdiques, tant la generació sota condicions flexibles com la generació de dades discretes. A més de la generació d'imatges a partir de soroll, també han estat emprades per a la traducció d'imatge a imatge, adaptació del domini, augment de dades, pintura d'imatges, augment de resolució i registre i reconstrucció d'imatges.

2.1 Anàlisi de diferents arquitectures GAN

A causa de la nombrosa quantitat de metodologies i arquitectures basades en GAN que han estat inventades fins avui i a l'ús tan extens que els ha estat donat dins del gran camp de la imatge mèdica, a continuació seran exposats, arran del treball de síntesi que va ser realitzat per Osuala et al. [3], un subconjunt de les arquitectures que són aplicades amb major freqüència en la imatge mèdica del càncer, sent aquest el tema concret que va ser investigat.

2.1.1 Conditional GAN

L'arquitectura Conditional GAN, tal i com va ser definida per Mirza, M. and Osindero [4], és definida com una extensió de la Vanilla GAN mitjançant la qual és subministrada una etiqueta discreta tant a la xarxa neuronal generativa com a la xarxa neuronal discriminativa per a la generació de mostres condicionals d'una classe. L'estructura d'aquest model, a escala de blocs, pot ser vist en la Figura 3.

En el treball de Chaitanya et al., 2021 [31] van ser implementats dos models basats en aquesta arquitectura per a la segmentació d'imatges mèdiques de cor, pròstata i pàncrees, amb un especial èmfasi en la captura de les variacions de forma i intensitat de diferents grups poblacionals. D'aquest treball va ser observat que el realisme de les imatges resultants d'aquests models va resultar petit, en comparació amb el realisme major de les segmentacions que van ser obtingudes.

Mentrestant, en el treball de Shin et al., 2018 [32] va ser implementat un model basat en aquesta arquitectura per a la generació sintètica d'imatges mèdiques de pòlips de colonoscòpies. Aquest model és caracteritzat per l'aplicació de convolucions dilatades en les capes de codificació i el redimensionament d'imatges en les capes de descodificació. Les imatges que van ser generades per aquest model van resultar ser de gran realisme, tant pel que fa a

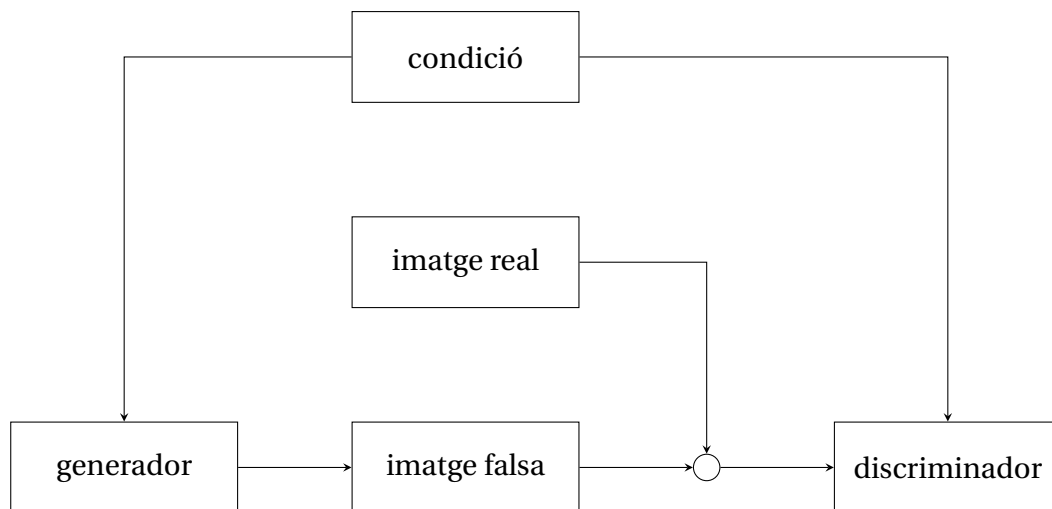


Figura 3: Esbós de la col·locació i connexions dels blocs d'una arquitectura Conditional GAN.

resolució com d'estructura de les lesions, que aquelles que van ser generades per una xarxa neuronal de referència.

2.1.2 Auxiliary Classifier GAN

L'arquitectura Auxiliary Conditional GAN és definida com una extensió de la Vanilla GAN mitjançant la qual és subministrada una etiqueta de classe únicament a la xarxa neuronal generativa, mentre que la xarxa neuronal discriminativa és utilitzada per a la correcta classificació tant de l'etiqueta de classe com la classificació de les imatges que li són subministrades en funció de si aquestes han estat creades per la xarxa neuronal generativa o si han estat obtingudes de la base de dades real. L'estructura d'aquest model, a escala de blocs, pot ser vist en la Figura 4.

En el treball de Kapil et al., 2018 [30] va ser implementat un model basat en aquesta arquitectura focalitzat en l'anàlisi de la variabilitat inter-observador d'un model d'aprenentatge automàtic, el qual va ser concordant amb la puntuació que va ser donada pels patòlegs per al mateix conjunt de dades. En línies generals, gràcies a aquest treball va ser demostrat que el rendiment d'un model basat en l'arquitectura Auxiliary Conditional GAN podia igualar i superar al d'un model completament supervisat.

2.1.3 Deep Convolutional GAN

L'arquitectura Deep Convolutional GAN és definida com una extensió de Vanilla GAN mitjançant la qual són generades mostres realistes gràcies a la utilització d'una arquitectura de

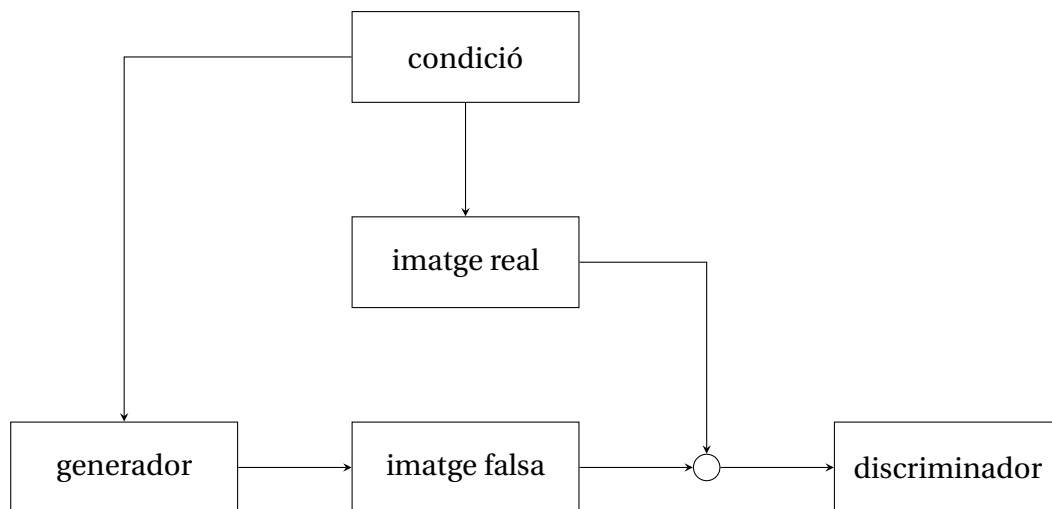


Figura 4: Esquema bàsic dels blocs d'una Auxiliary Conditional GAN.

xarxa neuronal convolucional amb *batch normalization*, tal i com va ser descrita per Ioffe et al., 2015 [5] tant en la xarxa neuronal generativa com en la xarxa neuronal discriminativa. En conjunció amb l'anteriorment esmentada també és augmentada de manera progressiva la dimensió espacial de les capes del generador mitjançant convolució transposada, tal i com va ser definida per Radford et al., 2015 [6].

En el treball de Desai et al., 2020 [33], va ser implementat basat en aquesta arquitectura mitjançant el qual pretenen ser solucionats els problemes relatius als etiquetats limitats per a la classificació d'imatges mèdiques mamogràfiques de càncer de mama. A partir de les imatges que van ser generades per aquest model va poder ser alimentat un altre model de Convolutional Neural Network. A partir d'això va poder ser corroborada la viabilitat de l'ús de GAN en entorns d'escassetat de dades i de limitació de dades d'anotacions.

2.1.4 Wasserstein GAN

L'arquitectura Wasserstein GAN és definida com una extensió de Vanilla GAN en la qual és reemplaçada la funció de pèrdua binary sigmoid cross entropy per la funció Wasserstein. Aquesta substitució en la funció de pèrdua és motivada per l'intent d'aproximació entre la distribució que ha estat generada artificialment i la distribució de les dades reals, ja que, mitjançant l'aplicació de la fórmula de pèrdua Wasserstein el discriminador maximitza, per la qual cosa el generador minimitza en conseqüència, aquesta diferència entre les distribucions entre tots dos conjunts de dades.

Per al treball de Han et al., 2018 [28] va ser implementat un model basat en l'arquitectura

Wasserstein GAN mitjançant el qual van ser generades imatges de ressonàncies magnètiques cerebrals multisequència de 128x128 píxels de resolució tan realistes que les imatges reals i les falses no van poder ser distingides ni un metge expert. Aquest èxit va ser atribuït a la capacitat de generalització dels models basats en aquesta arquitectura.

D'altra banda, en el treball de Liu et al., 2019 [29] va ser implementat un model basat en aquesta arquitectura per a identificació de lesions canceroses en imatges de resolucions reduïdes mitjançant diferents indicadors de precisió. En aquest treball va ser demostrat que, arran de l'ús d'aquesta metodologia, van ser generades moltes més imatges sintètiques de qualitat que amb altres enfocaments d'aprenentatge automàtic i mètodes de mostreig excessiu.

2.1.5 Progressively Growing GAN

L'arquitectura Progressively Growing GAN és definida originalment per Karras et al., 2017 [9] com una extensió de Vanilla GAN en la qual, tant la xarxa neuronal generativa com la xarxa neuronal discriminativa, són configurades inicialment per a la generació d'imatges de baixa resolució, per a posteriorment ser augmentada aquesta resolució mitjançant l'addició de noves capes de convolució a totes dues xarxes neuronals. La fiabilitat d'aquesta implementació és justificada en la idea de la ràpida convergència de les capes de resolucions més baixes amb les quals són obtingudes imatges poc detallades, perquè després siguin afegides les capes més grans la convergència de les quals és més lenta i, per conseqüència, el perfeccionament de les imatges serà major.

En el treball de Foroozandeh and Eklund, 2020 [26], va ser implementada aquesta arquitectura per a la generació de dades sintètiques d'entrenament d'un model de segmentació d'imatges de tumors cerebrals, on va ser demostrat que l'augment, tant en volum com en variabilitat de les dades, de les dades d'entrenament del model de segmentació pot ser reduïda dràsticament, sense afectar el rendiment del model, la quantitat de segmentacions manuals que li eren subministrades al model.

Mentrestant, el treball de Zhang et al., 2020 [27] va ser implementada aquesta arquitectura per a la segmentació semàntica d'imatges tridimensionals de pelvis masculines en un model semisupervisat amb un model d'augmentat de dades, també implementat amb GAN's. En aquest treball va ser demostrat que aquesta segmentació semàntica, tant en les imatges anotades com en les no anotades, pot ser obtinguda, i amb uns excel·lents resultats, per un model basat en l'arquitectura Progressively Growing GAN.

2.2 Conclusions

Havent estat realitzada una enumeració d'algunes de les arquitectures basades en Generative Adversarial Networks juntament amb els usos, dins de la imatge mèdica, que els han estat donats, el present treball estarà basat en la utilització d'un model de Progressively Growing GAN per a la generació d'imatges mèdiques a partir de mamografies, tant amb tumors cancerosos com sense la presència d'aquests.

3 Materials i mètodes

En aquesta secció són descrits ordenada i detalladament la totalitat dels materials que han estat utilitzats per a la realització del present estudi, al costat de les metodologies i procediments que han estat duts a terme. Aquesta exposició és realitzada perquè pugui ser avaluada la idoneïtat de la recerca, la fiabilitat i validesa dels resultats, a més que també, en un futur, pugui ser replicada la recerca de la mateixa manera en la qual va ser realitzada aquí.

3.1 Materials

Aquí seran exposats de manera concreta aquelles eines de programari, recursos de computació i sistemes de bases de dades que seran utilitzats en aquest projecte.

3.1.1 Base de dades

La base de dades que ha estat utilitzada per a l'entrenament del model de Machine Learning que ha estat aquí implementat ha estat íntegrament conformada per un conjunt de 30 ressonàncies magnètiques pertanyents al Consorci Corporació Sanitària Parc Taulí de dones amb una mitjana d'edat de 57 ± 5 anys amb alguna malaltia cancerosa en el procés previ a la quimioteràpia. Cadascuna d'aquestes ressonàncies ha estat emmagatzemada en format DICOM (Digital Imaging and Communications in Medicine), per la qual cosa, per a cada ressonància són emmagatzemats múltiples arxius en una carpeta, relatius als diferents formats de dades i protocols del format DICOM.

3.1.1.1 Preprocessat de la base de dades

Pel fet que no ha pogut ser implementada fins avui una metodologia gràcies a la qual pugui ser establerta una compatibilitat entre el format DICOM i el tipus de dades d'entrada del model de Machine Learning aquí implementat, ha estat necessària la realització d'un preprocessament d'aquestes imatges mèdiques en format DICOM.

3.1.1.1.1 Conversió del format DICOM a imatges bidimensionals

Dins de cadascuna de les imatges DICOM han estat emmagatzemats diversos protocols de visualització d'imatges mèdiques, per la qual cosa ha de ser triat un d'ells per a la representació de totes les imatges mèdiques que conformaran el dataset preprocessament. Després d'una anàlisi de la mena de visualització de molts d'aquests protocols mitjançant l'ús del programari ITKSnap [13], finalment ha estat triat el `t1_f13d_cor_dyn`, pel fet que en aquest

pot ser vista amb relativa facilitat la mamografia, sent aquesta acolorida en una escala de grises, mentre que el fons de la imatge és acolorit en negre.

Havent estat triat el protocol, han estat convertides cadascuna de les imatges mèdiques tridimensionals en format DICOM en una successió d'imatges mèdiques bidimensionals mostrades en perspectiva axial, també mitjançant l'ús del programari ITKSnap. Cadascuna d'aquestes representacions com a successió d'imatges bidimensionals ha estat emmagatzemada en un directori i, tots aquests directoris de successions bidimensionals han estat emmagatzemats en un únic directori arrel. Per tant, ha estat creada una estructura de directoris similar a la de la Figura 5.

Pel que concerneix l'aspecte de cadascuna de les imatges bidimensionals que han estat generades, aquestes són mostrades com en la Figura 6.

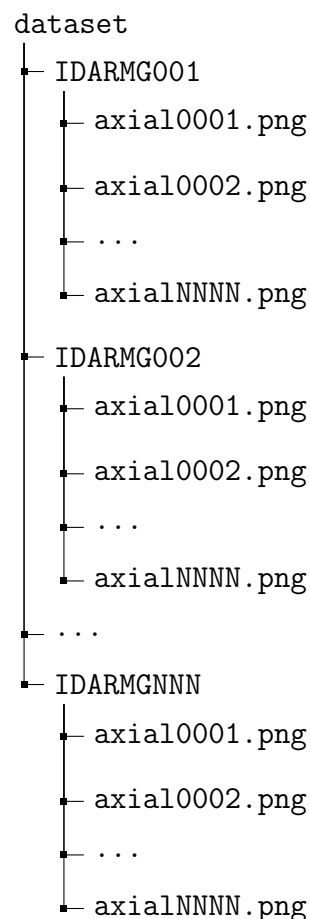


Figura 5: Estructura de directoris de la conversió d'una imatge tridimensional en una successió d'imatges bidimensionals des de la perspectiva axial per cada imatge mèdica.

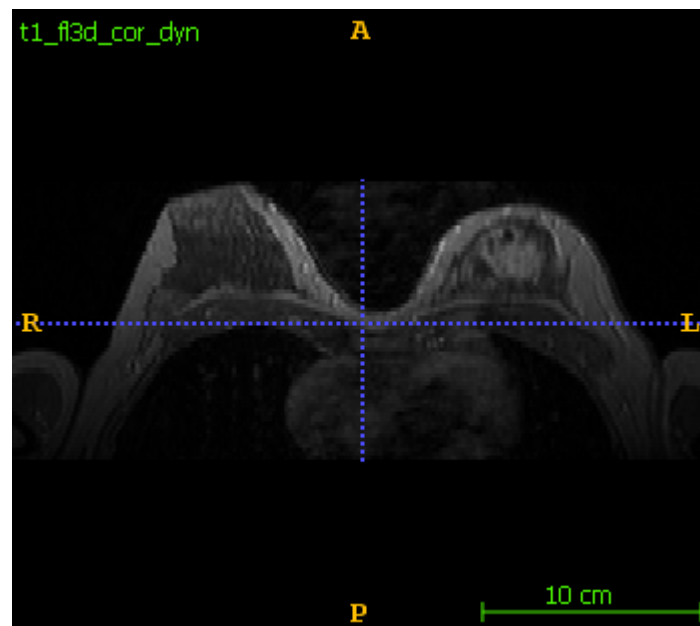


Figura 6: Format .png de descàrrega d'una imatge mèdica amb les metadades incloses.

3.1.1.1.2 Optimització de l'estructura de subdirectoris

A conseqüència de la gran quantitat de fitxers que han estat emmagatzemats en una gran quantitat de subdirectoris, els subdirectoris han estat comprimits perquè així sigui obtingut un menor consum de la memòria de la màquina, sent obtingut així una estructura de directoris més optimitzada, com la de la Figura 7.

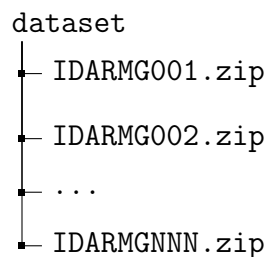


Figura 7: Estructura de directoris optimitzada de la conversió d'una imatge tridimensional en una successió d'imatges bidimensionals des de la perspectiva axial per cada imatge mèdica

3.1.2 Llenguatge de programació Python

La implementació de la totalitat del codi font serà escrit en el llenguatge de programació Python. L'elecció d'aquest llenguatge sobre altres opcions disponibles ha estat majoritàriament deguda a la simplicitat sintàctica d'aquest llenguatge, gràcies a la qual cosa és maximitzada

la velocitat de desenvolupament del codi. A més de l'anteriorment esmentat, han estat implementades i estan actualment mantingudes múltiples biblioteques de programari especialitzades per a Python especialitzades en la programació científica i en el confeccionat de models d'aprenentatge automàtic, mitjançant l'ús de les quals poden ser creats aquests models sent especificades únicament les capes més altes d'abstracció, sent delegada la implementació de diversos algorismes o mètodes matemàtics a les mateixes funcions de la biblioteca.

3.1.2.1 PyTorch

Dins de tot el ventall de biblioteques de programari per a Python especialitzades en l'aprenentatge automàtic ha estat triada la biblioteca PyTorch, degut, entre altres motius, al caràcter completament *open-source* d'aquesta, gràcies a la qual cosa pot ser implementat el codi de forma completament gratuïta.

En conjunció amb l'anteriorment exposada, la documentació d'aquesta biblioteca està mantinguda avui dia i és prou clara perquè el pendent de la seva corba d'aprenentatge sigui minimitzada a un nivell acceptable.

3.1.2.2 Python Imaging Library

Durant la realització del present treball ha estat utilitzada la biblioteca de programari Python Imaging Library (PIL per les seves sigles en anglès) [24] per al preprocessament de les imatges mèdiques bidimensionals. Entre les funcionalitats d'aquesta biblioteca estan l'obertura, processament, manipulat i guardat d'imatges de diferents formats.

L'elecció d'aquesta biblioteca per a aquesta finalitat és justificada a causa de la seva naturalesa completament *open-source* i a la seva facilitat d'ús, gràcies a la qual no ha estat necessària cap mena de coneixement previ al tractament, característiques internes ni metadades de cap format d'imatge.

3.1.2.3 TensorBoard

Durant el procés d'experimentació amb el model de Machine Learning ha estat necessària la visualització en directe del comportament del model durant l'entrenament així que, a causa de la incapacitat, tant en l'àmbit temporal com de coneixement d'implementació d'alguna mena de visualitzador d'aquestes característiques, han estat delegades totes aquestes funcionalitats a una biblioteca de programari.

Actualment, han estat implementades diverses d'aquestes funcionalitats per part de múltiples biblioteques, però per al present treball ha estat utilitzada TensorBoard [25]. El motiu darrere de l'elecció d'aquesta biblioteca abans que unes altres ha estat definida per la gran quantitat de funcionalitats que en ella han estat implementades, com ara el seguiment de diverses mètriques, visualització de gràfics, histogrames per al model.

3.2 Metodologia

Per al complet enteniment del marc teòric del present treball, és necessari que siguin exposats una sèrie de conceptes clau relatius al *framework* de *machine learning* que serà utilitzat.

3.2.1 Algorisme DFS per a subarbres expandits

Tal com ha estat exposat en la secció 3.1.1.1, la base de dades preprocessada és conformada per un directori arrel dins del qual són emmagatzemats diferents arxius comprimits, sent emmagatzemats de cadascun d'ells la successió d'imatges bidimensionals equivalent a la representació en format DICOM de cadascuna de les imatges mèdiques.

Perquè pugui ser automatitzat aquest procés de descomprimint d'arxius per un únic algorisme, ha estat implementada una variant de l'algorisme DFS mitjançant la qual una estructura de directoris és recorreguda i, quan és obtingut un fitxer comprimit com a fitxer actual, aquest arxiu comprimit és descomprimint i és explorat l'interior d'aquest nou directori descomprimint.

De manera complementària a l'anterior funcionalitat, per aquest algorisme és retornat un generador [Referència 1], mitjançant el qual, és obtinguda la ruta de cadascun dels arxius no comprimits que van sent trobats durant l'expedició del subdirectori.

Gràcies a la implementació d'una metodologia d'exploració d'arxius descomprimits dins d'un directori, aquest algorisme pot ser utilitzat per al recorregut de qualsevol estructura de directoris, independentment de la complexitat o composició d'arxius comprimits dins d'aquest.

3.2.2 Filtres d'imatge

Tal com ha estat exposat en la secció 3.1.1, el format de les imatges en format DICOM que han estat cedides pel Consorci Corporació Sanitària Parc Taulí va ser considerat incompatible respecte al format d'entrada d'un model de Machine Learning basat en el processament de múltiples imatges bidimensionals, per la qual cosa aquestes imatges inicials van haver de ser preprocessades per a així poder resultar d'utilitat per a aquest model.

Tot el referent als canvis de dimensionalitat, és a dir, el canvi d'una imatge en format DICOM tridimensional a una successió d'imatges bidimensionals va ser realitzat mitjançant l'ús del programari ITKSnap, sent obtinguts uns resultats com els de la Figura 6.

Malgrat que hagi estat solucionat el problema de la dimensionalitat de les dades, ha estat trobat un altre problema amb les dades que són mostrats en cadascuna de les imatges bidimensionals. Més concretament, és mostrat un eix de coordenades enmig de cada imatge, al costat d'unes metadades, sent així contaminada la imatge amb píxels de colors que poden ser interpretats erròniament pel model de Machine Learning.

3.2.3 Deep Learning

El Deep Learning pot ser definit com un subconjunt d'aquells algorismes de Machine Learning a través dels quals una intel·ligència artificial pot ser entrenada per a la realització d'una tasca específica.

Aquest entrenament, en ser dut a terme mitjançant un sistema de xarxes neuronals de diverses capes, pot ser completament no supervisat per cap ésser humà en cap moment, sent delegades al mateix algorisme totes les tasques de predicció i anàlisi dels resultats.

3.2.3.1 Funció d'activació

La incapacitat de realització de càlculs complexos és presa una característica intrínseca de la naturalesa d'una xarxa neuronal completament lineal, per la qual cosa és minvat considerablement el ventall d'aplicacions d'aquesta. L'única manera que una xarxa neuronal sigui capacitada per a la realització de tasques i càlculs més complexos és l'addició de components no lineals a aquesta, com ara les funcions d'activació.

3.2.3.1.1 Rectified linear unit

Una de les funcions d'activació més simples i més utilitzades per la gran velocitat d'aprenentatge que és obtinguda a xarxes neuronals on hi és implementada és la funció Rectified Linear Unit (RELU).

Tal com pot ser apreciat a les equacions 1 i 2, si li és subministrat a la funció un nombre estrictament major a 0, és retornat com a valor de sortida aquest mateix nombre. Si a la funció li és subministrat un nombre menor o igual a 0, és retornat com a valor de sortida un 0.

$$\text{RELU}(x) = \max(0, x) \quad (1)$$

$$\text{RELU}(x) = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \leq 0 \end{cases} \quad (2)$$

On:

- x = Valor d'entrada de la funció RELU()

3.2.3.1.2 Leaky rectified linear unit

Tal i com va ser definit per (Lu, Shin, Su and Em Karniadakis, 2019) [12], el Dying RELU és catalogat com un dels majors problemes que poden ser esdevinguts amb l'aplicació de la funció d'activació RELU. Aquest problema és caracteritzat per la desactivació de les neurones d'una xarxa neuronal, sent retornat únicament un 0 en tot moment com a valor de sortida d'aquesta neurona, a conseqüència del càlcul del gradient en el punt 0 de manera permanent. Com pot ser deduït, quan aquest problema és detectat en una neurona, aquesta no pot ser recuperada.

A manera de prevenció d'aquest problema, la funció RELU ha estat modificada amb la forma de les equacions 3 i 4, sent retornat, en lloc de 0, un valor ínfim lleugerament superior de 0 quan li és subministrat un valor estrictament menor a 0, sent aquesta conversió batejada com a funció Leaky Rectified Linear Unit (Leaky RELU).

$$\text{LeakyRELU}(x) = \max(0, x) + \alpha * \min(0, x) \quad (3)$$

$$\text{LeakyRELU}(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha * x & \text{if } x < 0 \end{cases} \quad (4)$$

On:

- x = Valor d'entrada de la funció RELU()
- α = valor petit per que sigui evitat el Dying RELU (normalment $\alpha = 1 \times 10^{-2}$).

3.2.3.2 Optimitzadors

Per tal que sigui minimitzada la diferència entre les imatges generades pel model i les imatges reals, és necessari que sigui minimitzada la funció de pèrdua del model.

Per l'assoliment d'aquest objectiu de minimització de la funció de pèrdua, han sigut implementats els optimitzadors, que són definits com a funcions que, mitjançant la connexió entre els valors resultants de la funció de pèrdua amb els pesos de la xarxa neuronal i la taxa d'aprenentatge, pot ser implementada una actualització de forma dinàmica d'aquests dos paràmetres, sent obtingut com a conseqüència un canvi en els valors de la funció de pèrdua. Aquesta possibilitat d'alteració dels valors de la funció de pèrdua mitjançant aquest mecanisme de connexió és la base comuna de tots els tipus d'optimitzadors que han sigut implementats fins al moment. Mentrestant, cada tipus d'optimitzador és diferenciat de la resta pel seu algorisme de càlcul del punt mínim de la funció de pèrdua.

3.2.4 Generative Adversarial Networks

El modelatge generatiu és definit, dins de l'aprenentatge automàtic, com una tasca d'aprenentatge no supervisat en funció del descobriment de regularitats o patrons en les dades d'entrada, de tal manera que puguin ser generats nous exemples que plausiblement podrien haver estat extrets del conjunt de dades original.

Les Generative Adversarial Networks, o GAN per les seves sigles en anglès, són definides com una aproximació al modelatge generatiu mitjançant la utilització d'un nou framework d'aprenentatge profund.

A manera d'ampliació a la succinta introducció a les Generative Adversarial Networks que ha estat explicada en la secció 1.1, aquí seran exposats els detalls més concrets de tots els components i metodologies sobre els quals estan basades les GAN.

3.2.4.1 Topologia del model

Per a la implementació d'un model d'aquestes característiques, primerament, és col·locada una xarxa generativa $G(z)$, la qual és alimentada amb una distribució aleatòria de les dades del conjunt d'entrada $P_{data(x)}$, a partir de la qual és generada com a sortida de G la distribució del generador P_z .

Posteriorment, és col·locada darrere de la xarxa generativa una xarxa discriminativa ($D(x)$), la qual pot ser alimentada de forma equitativament probable a partir d'una mostra de les dades d'entrada x o, per contra, pot ser alimentada per una mostra de les dades generades per la xarxa generativa (z). D'aquesta xarxa discriminativa és generat un conjunt de dos

valors relatius a la probabilitat que la imatge que ha estat analitzada sigui real o falsa, és a dir, generada per la xarxa generativa.

3.2.4.1.1 Generador

Específicament, la topologia interna amb la qual ha estat construïda la xarxa neuronal generativa dins del framework de les GAN és conformada per una concatenació de blocs, on cada un dels quals és conformat per una capa de convolució, mitjançant la qual és augmentat la mida del vector de característiques que és processat, per a ser canalitzada aquesta sortida a una capa de batch normalization, on el vector de característiques és normalitzat, per a, en últim lloc, ser passat aquest vector a la funció d'activació. D'una forma més esquemàtica, l'estructura anteriorment descrita de cada bloc pot ser visualitzada a la Figura 8.

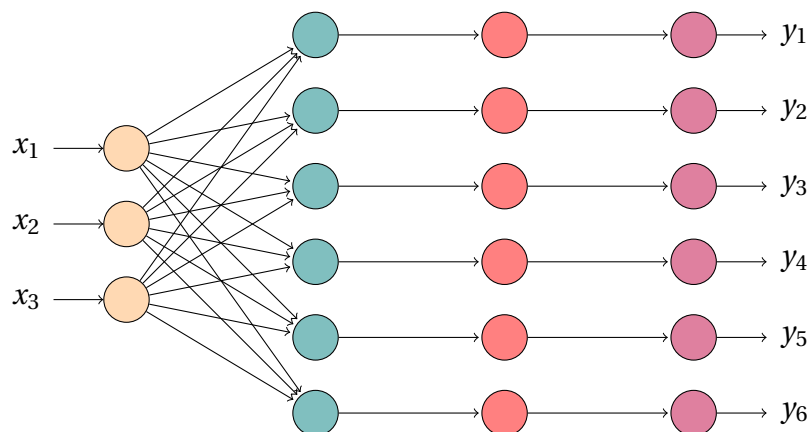


Figura 8: Topologia d'un bloc d'exemple de la xarxa neuronal generativa, on amb el color groc és representada la capa d'entrada, el color blau la capa de convolució, el color vermell la capa de batch normalization i en lila la capa d'activació.

Per tant, tal com ha estat introduït anteriorment, la topologia global de la xarxa neuronal generativa és conformada per la concatenació de tots els blocs que han estat definits anteriorment a la Figura 8 mitjançant els quals va sent doblada la dimensionalitat del vector de característiques de les capes internes.

No obstant això, en primer lloc, ha de ser incorporada una primera capa de convolució mitjançant la qual és transformada la dimensionalitat del vector de característiques des de la dimensió del vector de noise fins a la dimensionalitat del tensor de la primera capa internes.

A més d'això, és col·locada al final una capa de convolució mitjançant la qual és reduïda la dimensionalitat del vector de característiques fins que sigui obtinguda la dimensionalitat de

la imatge que ha estat tractada i, a continuació és afegida una altra capa on és implementada una funció d'activació. Aquesta topologia global pot ser visualitzada en la Figura 9.

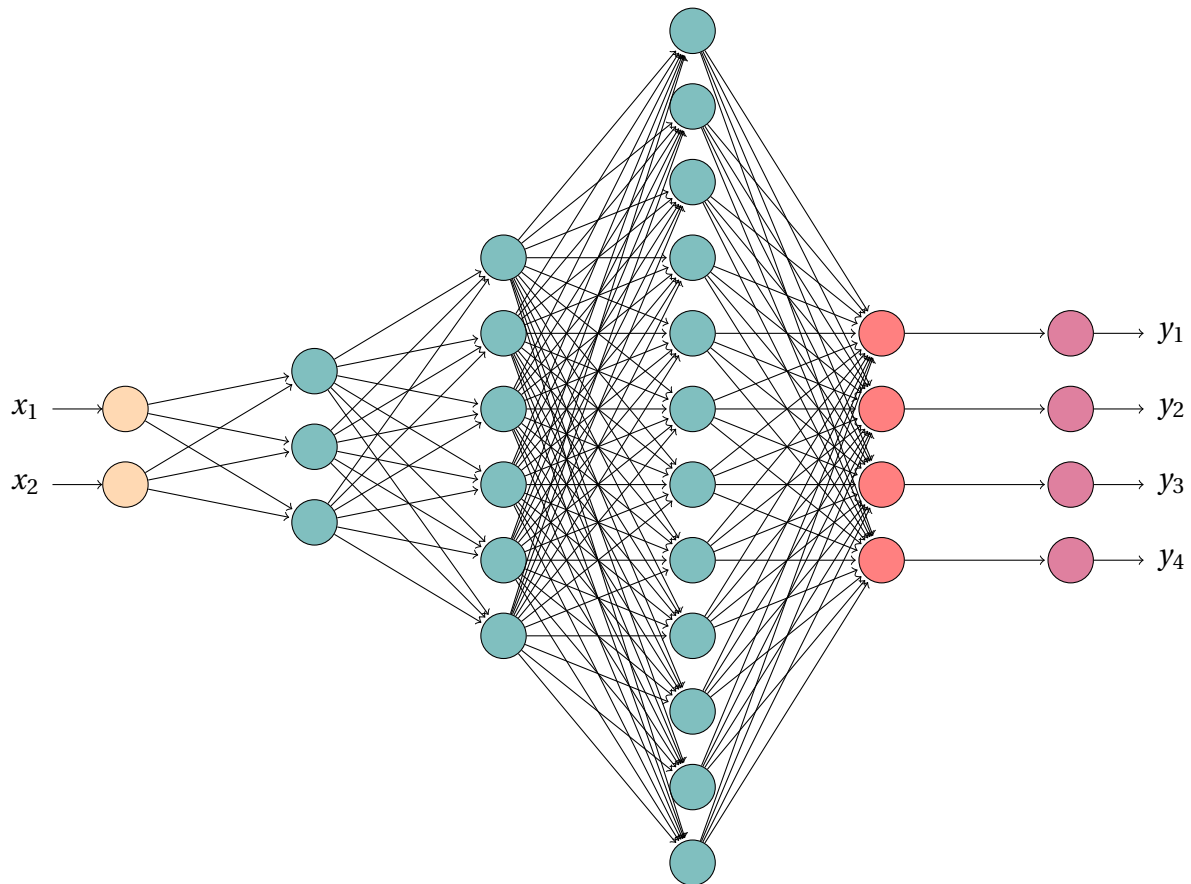


Figura 9: Topologia global d'exemple de la xarxa neuronal generativa, on amb el color groc és representada la capa d'entrada, el color blau les múltiples capes de convolució per al doblat de la dimensionalitat del vector de característiques, el color vermell la capa de redimensionat a la mida de la imatge a ser tractada i en lila la capa d'activació.

3.2.4.1.2 Discriminador

De manera complementària a la topologia mitjançant la qual ha estat construïda la xarxa neuronal generativa, la xarxa neuronal discriminativa és construïda també mitjançant una concatenació de blocs predefinitos mitjançant els quals és transformada la dimensionalitat del vector de característiques, però, en el present cas, la dimensionalitat del vector de característiques de les capes internes és dividit a la meitat en cada pas, a més que és omesa la capa de normalització del batch, sent col·locada directament la capa de la funció d'activació. D'una forma més esquemàtica, l'estructura anteriorment descrita de cada bloc pot ser visualitzada a la Figura 10.

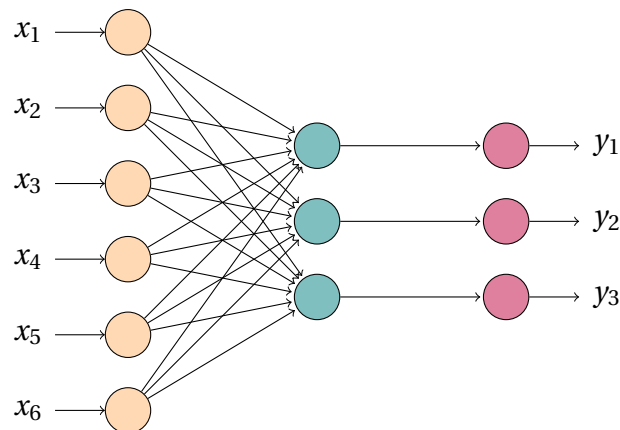


Figura 10: Topologia d'un bloc d'exemple de la xarxa neuronal discriminativa, on amb el color groc és representada la capa d'entrada, el color blau la capa de convolució i en lila la capa d'activació.

Havent estat exposada la topologia d'un únic bloc de la xarxa neuronal discriminativa, llavors pot ser explicada la topologia global de tota aquesta xarxa. Aquesta, tal com ha estat dit anteriorment, és conformada per una concatenació de blocs equivalents al de la Figura 10.

No obstant això, primer de tot ha de ser col·locada una primera capa de convolució mitjançant la qual és transformada la dimensionalitat del vector de característiques de la imatge a ser tractada a la dimensionalitat de les capes internes.

A causa de la naturalesa binària de la resposta de la xarxa neuronal discriminativa, la dimensionalitat del vector de característiques al final d'aquesta xarxa ha de ser d'1, per la qual cosa és col·locada en últim lloc una capa mitjançant la qual és reduïda a 1 aquesta dimensionalitat.

La topologia global d'aquesta xarxa discriminativa pot ser vista en la Figura 11.

3.2.4.2 Entrenament

En allò referent al procés d'entrenament de la GAN, inicialment, abans que comenci l'entrenament, la sortida falsa del generador és molt fàcil de reconèixer per al discriminador.

Com que la sortida del generador s'alimenta directament al discriminador com a entrada, això significa que quan el discriminador classifica una sortida del generador, podem aplicar l'algorisme de retropropagació a tot el sistema i actualitzar els pesos del generador.

Amb el temps, la sortida del generador es torna més realista i el generador millora en enganyar el discriminador. Finalment, les sortides del generador són tan realistes, que el discriminador és incapaç de distingir-les dels exemples reals.

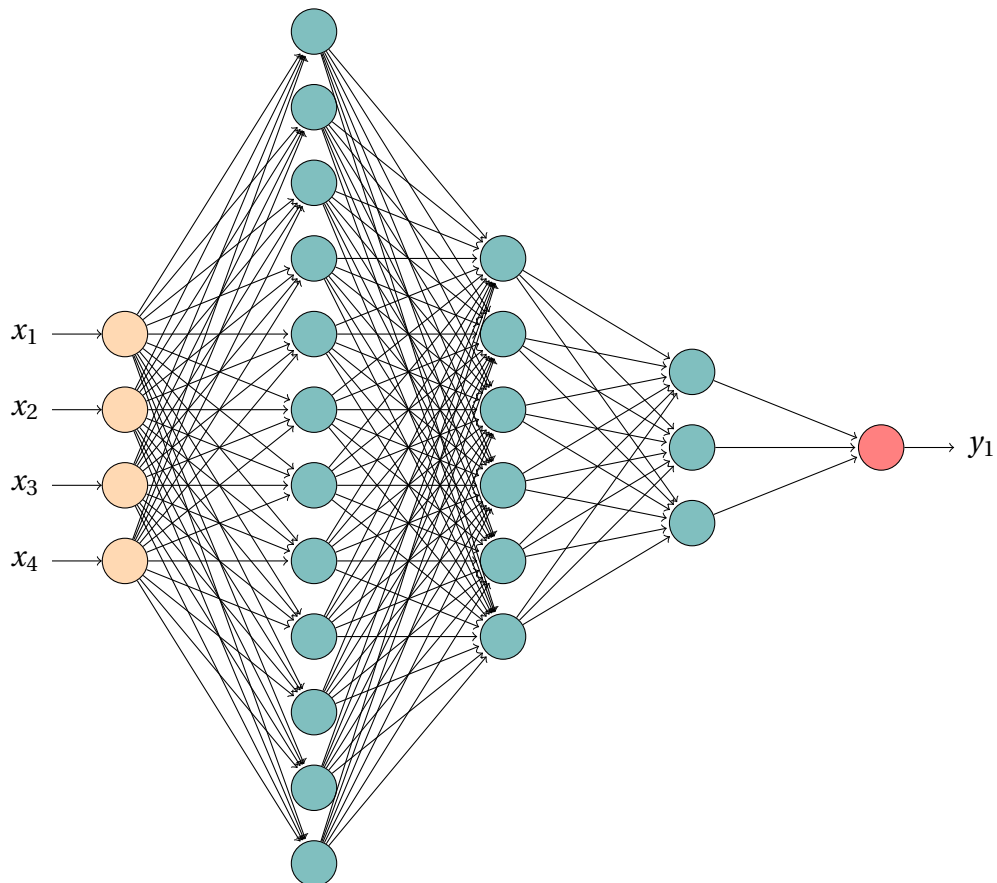


Figura 11: Topologia global d'exemple de la xarxa neuronal generativa, on amb el color groc és representada la capa d'entrada, el color blau les múltiples capes de convolució per al reduït a la meitat de la dimensionalitat del vector de característiques, el color vermell la capa de redimensionat a una sortida binària.

En estar totes dues xarxes confrontades, la GAN pot ser formalitzada com un joc minimax de suma 0 entre la xarxa generativa i la xarxa discriminativa, ja que l'objectiu de D és la minimització de la funció de pèrdua L_d , mentre que per part de G és perseguit l'objectiu de la minimització de L_d . A conseqüència d'això, la funció de valor $V(D, G)$ relativa a la GAN pot ser definida segons la forma de l'equació 5:

$$\min_G \max_D V(D, G) = \min_G \max_D [\mathbb{E}_{x \sim p_{data}} \log D(x) + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]] \quad (5)$$

3.2.5 Creixement progressiu de les GAN

La metodologia Progressive Growing of GAN's pot ser definida com una nova metodologia d'entrenament per als models basats en Generative Adversarial Networks, mitjançant el qual poden ser obtingudes imatges d'una major resolució que aquelles que van ser generades per altres models amb diferents metodologies d'entrenament.

3.2.5.1 Arquitectura multiescala

L'arquitectura interna sobre la qual ha estat construïda la metodologia Progressive Growing of GAN és definida per la creació, tant com per al generador com per al discriminador, d'un sistema de blocs de Convolutional Neural Networks on, primerament és introduït i entrenat un bloc de manera síncrona i simultània en totes dues subxarxes, mitjançant el qual seran obtingudes les imatges d'una resolució mínima (4^2 bits). Havent estat completat el procés d'entrenament d'aquesta capa per a totes dues xarxes, és afegida una nova capa, també de manera síncrona i simultània, per a la generació d'imatges d'una resolució major (8^2 bits). Per a l'entrenament de cadascuna de les capes noves no ha de ser entrenades de nou les capes antigues, sinó que, per contra, es conserven els valors finals de l'entrenament previ d'aquestes.

Tal com pot ser vist en la Figura 12 les imatges d'entrenament mitjançant les quals és entrenat el model, són baixades de resolució perquè sigui adequat aquesta grandària a les resolucions d'entrenament de l'última capa i, a mesura que van sent afegides més capes a l'arquitectura multicapa, les imatges d'entrada van sent baixades de resolució a una menor magnitud.

3.2.5.2 Incorporament progressiu de noves capes

Tal com havia estat esmentat en la secció 3.2.5.1, per al procés d'entrenament global d'una xarxa neuronal en la qual ha estat implementada la metodologia Progressive Growing of

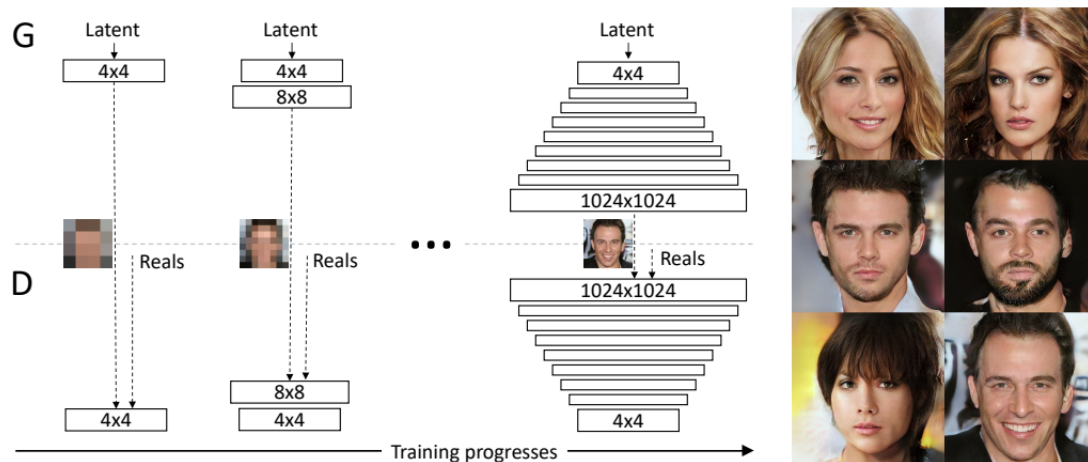


Figura 12: Arquitectura multiescala de ProGAN

GAN's no han de ser tornades a entrenar les capes de Convolutional Neural Networks que van ser entrenades prèviament, sinó que només han de ser entrenades les que van ser afegides en últim lloc. Perquè l'afegit d'aquestes noves capes sense entrenar no sigui vist com un problema de distorsió de l'entrenament, aquestes capes són introduïdes progressivament a la xarxa.

Més concretament, i tal com pot ser vist en la Figura 13, és iniciat aquest procés en l'escenari (a), on encara no ha estat afegida una nova capa de 32x32 com a continuació de la capa 16x16 ni al generador ni al discriminador.

Posteriorment, en l'escenari (b), és afegida síncronament i simultàniament una capa de 32x32. En aquest moment, la sortida de la capa de 16x16 és projectada a la dimensió 32x32 mitjançant una Nearest-neighbor interpolation. La sortida de la capa projectada és multiplicada per $1-\alpha$, on α és un número entre 0 i 1 que escala linealment, i és concatenada amb la nova capa de sortida. Aquest augment progressiu del valor de α és realitzat fins que sigui aconseguit el valor 1, mitjançant el qual la Nearest-neighbor interpolation serà anul·lada completament.

Havent estat completats aquests dos passos, és obtingut un escenari com el (c), en el qual la capa de 32x32 ha estat afegida correctament.

3.2.5.3 Desviació estàndard del minibatch

A manera de solució al problema consubstancial de les GAN relatiu a la incapacitat de captura

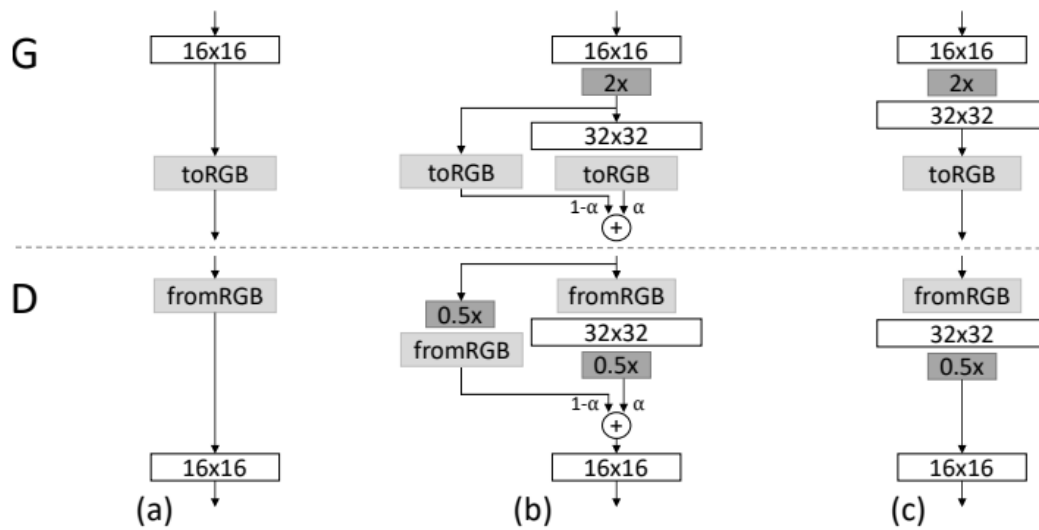


Figura 13: Incorporament progressiu de noves capes

de la variació de tot el conjunt de dades d'entrenament i, com a conseqüència final d'això, la incapacitat d'obtenció d'imatges generades pel model amb una distribució prou semblant a les imatges reals, ha estat proposada per Salimans et al. (2016) una discriminació de les dades d'entrada en minibatches.

Més concretament, en la implementació d'aquesta tècnica són calculades les estadístiques de les imatges de manera individual i les estadístiques grupals de totes les imatges que són incloses en un minibatch. Aquests càlculs són calculats en una capa de minibatches que són col·locats al final del discriminador, on aquest càlcul d'estadístiques és concatenat amb la mateixa sortida de la capa. Gràcies a això les estadístiques poden ser utilitzades internament pel discriminador.

D'una forma més específica, el càlcul de la desviació estàndard del minibatch és calculant la mitjana de totes les desviacions estàndard (Fòrmula 6) corresponents a cadascuna de les imatges del minibatch.

$$\bar{x} = \frac{\sum_{i=1}^N \sqrt{\frac{\sum_{i=1}^N (x - \bar{x})^2}{N-1}}}{N} \quad (6)$$

On:

- S_X = Desviació estàndard

- N = número de punts de dades

3.2.5.4 Normalització de generador i discriminador

Un descontrol de l'escalat de les magnituds dels senyals pot ser categoritzat com un dels perills de la competició insana entre la xarxa generativa i la xarxa discriminativa d'una GAN. A manera de protecció envers aquest fenomen, han estat implementades diverses solucions de normalització d'aquests senyals.

Més concretament, dins de la metodologia del Progressive Growing of GAN's aquesta normalització ha estat implementada mitjançant una igualació entre les tasses d'aprenentatge del generador i discriminador, a més d'una normalització del feature vector del generador.

3.2.5.4.1 Tassa d'aprenentatge igualada

En la implementació d'alguns algorismes optimitzadors moderns, com és el cas d'Adam o RMSProp, les actualitzacions dels gradients són normalitzades segons la seva desviació estàndard estimada, gràcies a la qual cosa el càlcul d'aquest valor és independitzat de l'escala dels pesos de la xarxa.

Gràcies a això, la inicialització dels pesos de la xarxa pot ser inicialitzada trivialment $\mathcal{N}(0, 1)$, la qual cosa resulta computacionalment més eficient per als inicialitzadors moderns que una inicialització estàtica dels pesos, per a després ser escalada explícitament durant l'execució amb uns valors dins d'un rang dinàmic comú per a tots els pesos.

A conseqüència de l'existència d'un únic rang de pesos per a tota la xarxa, és garantida una mateixa velocitat d'aprenentatge, tant per al generador, com per al discriminador.

3.2.5.4.2 Normalització del feature vector del generador

Una altra de les tècniques de normalització utilitzades a manera de prevenció davant una possible distorsió dels resultats a conseqüència de l'escalat de valors a conseqüència d'una competició insana entre generador i discriminador és definida per la normalització, a nivell de píxels, del vector de característiques del generador després de cada capa convolucional. Aquesta normalització és realitzada segons la fórmula 7.

$$b_{x,y} = \frac{a_{x,y}}{\sqrt{\frac{\sum_{j=0}^{N-1} (a_{x,y}^j)^2}{N} + \epsilon}} \quad (7)$$

On:

- x, y = posició del píxel
- $\epsilon = 1 \times 10^{-8}$
- N = número de mapes de característiques
- $a_{x,y}$ = valors originals del vector de característiques
- $b_{x,y}$ = valors normalitzats del vector de característiques

3.2.6 Mètriques d'evaluació

Havent estat implementada la totalitat d'un model de Machine Learning capaç de la generació de prediccions a partir d'un conjunt de dades, ha de ser creada alguna manera mitjançant la qual pugui ser avaluada la qualitat de les prediccions del model i, en última instància, la qualitat del model mateix.

Actualment, han estat inventades diverses formes de quantificació de la qualitat dels resultats d'un model de Machine Learning, com ara el càlcul de la ràtio de prediccions encertades dins del total de dades de mostra o el càlcul d'una penalització per cada predicció incorrecta que ha estat realitzada.

3.2.6.1 Matriu de confusió

La matriu de confusió és definida com una mètrica d'avaluació en la qual els resultats de la qualitat del model són exposats en una matriu quadrada de N elements, on N és el nombre de classes a ser predites. Per al cas d'un model de classificació binària com el que serà implementat en aquest projecte, llavors en la matriu de confusió resultant únicament seran necessàries 2 files i 2 columnes.

Dins de cadascuna de les cel·les d'aquesta matriu és emmagatzemat un valor específic amb una finalitat concreta, tal com pot ser vist en la Taula 1, on el significat de cadascun d'aquests valors és definit a continuació.

- **Exactitud:** proporció del nombre total de prediccions que eren correctes.
- **Valor predictiu positiu o precisió:** proporció de casos positius identificats correctament.

- **Valor predictiu negatiu:** proporció de casos negatius que s'han identificat correctament.
- **Sensibilitat o record:** proporció de casos positius reals que s'identifiquen correctament.
- **Especificitat:** proporció de casos negatius reals identificats correctament.

		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	$\frac{a}{a+b}$
	Negative	c	d	Negative Predictive Value	$\frac{d}{c+d}$
		Sensitivity	Specificity	Accuracy = $\frac{a+d}{a+b+c+d}$	
		$\frac{a}{a+c}$	$\frac{d}{b+d}$		

Taula 1: Matriu de confusió

3.2.6.2 Fréchet inception distance

La Fréchet Inception Distance (FID) és definida com una mètrica d'avaluació de les imatges que han estat generades per un model generatiu, per la qual cosa la seva utilització és recomanada per a qualsevol implementació d'una arquitectura basada en Generative Adversarial Networks. Per al càlcul d'aquesta distància, representat per un escalar, ha de ser calculada la similitud, mitjançant l'elevació al quadrat de la mètrica de Wasserstein, que és representada a l'equació 8, entre la distribució Gausiana multidimensional del conjunt de dades d'entrada $\mathcal{N}(\mu_w, \Sigma_w)$ i la distribució Gausiana multidimensional de les dades que han estat generats per la xarxa neuronal generativa $\mathcal{N}(\mu, \Sigma)$.

$$W_2(\mu_1, \mu_2)^2 = |\mu_1 - \mu_2|_2^2 + \text{tr} \left(C_1 + C_2 - 2\sqrt{\sqrt{C_2}C_1\sqrt{C_2}} \right) \quad (8)$$

On:

- C_n = Matriu de covariància
- $\mu_1 = \mathcal{N}(m_1, C_1)$
- $\mu_2 = \mathcal{N}(m_2, C_2)$

A causa de les capacitats de la xarxa neuronal mitjançant la qual és calculada habitualment dita mètrica (Inception V3) la fórmula resultant del càlcul d'aquesta mètrica és calculada segons la mitjana i la desviació estàndard d'algunes capes pròximes a la capa final d'aquesta

xarxa, en comptes del càlcul a escala de píxels, tal i com pot ser vist a l'equació 9. A conseqüència d'aquesta variància de càlcul, aquesta mètrica i la percepció humana tendeixen a ser similars.

$$FID = \|m_1 - m_2\|_2^2 + \text{trace} \left(\Sigma + \sum_w -2 \sqrt{\sqrt{\Sigma} \Sigma_w \sqrt{\Sigma}} \right) \quad (9)$$

4 Implementació

La implementació del codi font d'aquest projecte ha estat escrit íntegrament en el llenguatge de programació Python, en haver sigut confeccionades per aquest llenguatge múltiples biblioteques de programari com a abstracció de sistemes d'intel·ligència artificial, amb les quals poden ser desenvolupats aquests models a alt nivell, sent delegades a aquestes biblioteques les implementacions i optimitzacions de baix nivell.

4.1 Funcions auxiliars

Per a la implementació d'una gran part de les funcionalitats d'aquest projecte han estat utilitzades diverses biblioteques de programari, gràcies a les quals no han necessitat ser implementats diversos algorismes o procediments. Malgrat això, també han hagut de ser implementades manualment diverses rutines per a parts concretes de l'aplicació.

4.1.1 Iterador de directoris comprimits

La implementació de la variant de l'algorisme DFS que és especificat en la secció [1], mitjançant la qual és possibilitada, per a la seva execució sobre una estructura de fitxers, a més de l'obtenció de les rutes de cadascun dels fitxers que són trobats, la capacitat de descompressió i expedició d'arxius comprimits ha estat implementada en els següents arxius:

- `src/utis/file_management/directory_iterator.py`: En aquest arxiu ha estat implementat l'algorisme DFS amb les modificacions per al reconeixement i processament d'arxius comprimits.
- `src/utis/file_management/file_decompressor.py`: En aquest arxiu ha estat implementada la descompressió d'un únic arxiu comprimit.

4.1.1.1 Descompressió d'un arxiu comprimit

Tal com pot ser visualitzat en la Figura 14 per a la implementació del descomprimit d'un arxiu és subministrada la ruta del fitxer a descomprimir. Primerament, és obtinguda la ruta del directori on és emmagatzemat el fitxer comprimit, per a després ser descomprimit el fitxer i eliminat el fitxer comprimit, respectivament.

4.1.1.2 Recorregut del subarbre de directoris ampliat

```
def decompress_file(cls, compressed_file_path: os.PathLike) -> None:
    directory_path, _ = os.path.split(compressed_file_path)

    with zipfile.ZipFile(compressed_file_path, 'r') as zip_file:
        zip_file.extractall(directory_path)

    os.remove(compressed_file_path)
```

Figura 14: Descompressió d'un arxiu comprimit

Tal com pot ser visualitzat en la Figura 15, per a la implementació del recorregut del subarbre de directoris ampliat és subministrada la ruta del directori arrel a recórrer.

Les rutes de fitxers sense visitar són emmagatzemades en una llista, a la qual li seran extrets un a un els elements fins que no hi hagi cap més i, llavors, es donarà per finalitzat el recorregut al subarbre.

Per a cadascuna de les iteracions és tractada una ruta de fitxer (`directory_item_path`). Primerament, és comprovat que aquesta no hagi estat visitada i, si és així, es marca com visitada. A continuació és examinat el tipus de fitxer que és albergat en aquesta ruta.

- Si és un arxiu comprimit aquest és descomprimit i la ruta de l'element que és tractat en aquesta iteració (`directory_item_path`) serà substituïda per la ruta de la carpeta descomprimida resultant.
- Si és un directori, aquest serà recorregut en el seu primer nivell i totes les rutes dels elements que siguin trobats seran guardades en la llista d'elements no visitats (`unvisited_paths`).
- Si és un arxiu no comprimit, la ruta d'aquest és llançada mitjançant un generador.

4.2 Preprocessat de les dades

La totalitat del codi per al preprocessament de les imatges bidimensionals que ha sigut especificat a la secció 3.2.2 ha sigut implementat dins de la carpeta `src/data_preprocessing`.

Per a això, han estat eliminats tots els píxels corresponents a eixos de coordenades i metadades i han estat substituïts pel color mitjà de tots els píxels adjacents on no han estat descoberts els colors en els quals aquesta informació irrellevant va ser col·locada.

```
def expanded_subtree(cls, root_directory_path: os.PathLike) -> \
    Iterator[os.PathLike]:
    visited_paths, unvisited_paths = set(), [root_directory_path]

    while unvisited_paths:
        directory_item_path = unvisited_paths.pop()
        if directory_item_path not in visited_paths:
            visited_paths.add(directory_item_path)

            if cls.__is_compressed_file(directory_item_path):
                FileDecompressor.decompress_file(directory_item_path)
                decompressed_directory_path, _ = os.path \
                    .splitext(directory_item_path)
                directory_item_path = decompressed_directory_path

            if os.path.isdir(directory_item_path):
                directory_items_paths = set(cls \
                    .subtree_first_level(directory_item_path))
                directory_unvisited_paths = directory_items_paths - \
                    visited_paths
                unvisited_paths.extend(directory_unvisited_paths)

            if os.path.isfile(directory_item_path):
                yield directory_item_path
```

Figura 15: Recorregut del subarbre de directoris ampliat

4.3 Model de Progressive Growing of GAN's

La totalitat del codi font relatiu a la Generative Adversarial Network amb la variant de creixement progressiu incorporada ha sigut implementat dins del directori `src/gan`. Aquesta implementació del codi ha estat basada en el treball de BakingBrains [34], on va ser implementada l'arquitectura Progressively Growing GAN tal com va ser descrita en l'article original d'aquesta arquitectura Karras et al., 2017 [9], a partir de la qual han estat realitzades múltiples modificacions per a l'adaptació al format i naturalesa de les imatges mèdiques i el cas a tractar.

4.3.1 Hiperparàmetres

El model de Machine Learning que ha estat implementat requereix l'especificació d'un seguit de paràmetres específics abans del mateix entrenament, mitjançant els quals pot ser modificat i controlat el ritme i rendiment de l'entrenament del model a nivell global. Més concretament, els hiperparàmetres més importants que han estat especificats són enumerats a continuació.

- `CHANNELS_IMG`: Nombre de colors primaris mitjançant els quals seran construïts els colors finals de cada píxel de la imatge.
- `DEVICE`: Tipus d'unitat de processament (CPU, GPU, TPU...) on hi seran emmagatzemades i processades les dades.
- `FACTORS`: Llista on són guardats els progressius factors de reducció dels canals d'entrada de cada capa de convolució, tant del generador com el discriminador.
- `FIXED_NOISE`: Array de so aleatori, gràcies a la naturalesa estàtica d'aquest, pot ser garantida la solidesa dels càlculs on aquest ha estat utilitzat.
- `LEARNING_RATE`: Velocitat a la qual, per part de l'algorisme, són actualitzades o estudiades les estimacions dels paràmetres.
- `NUM_WORKERS`: Nombre de subprocessos que poden ser utilitzats per al carregat de les dades del dataset.
- `PATH_DATASET`: Ruta al directori en el qual són emmagatzemats les dades d'entrada amb les quals el model és entrenat.
- `PROGRESSIVE_EPOCHS`: Nombre de passades de les dades d'entrada per l'algorisme.

4.3.2 Xarxes neuronals auxiliars

Malgrat que la xarxa neuronal generativa i la xarxa neuronal discriminativa siguin considerades les peces fonamentals de qualsevol model de Machine Learning que hagi estat basat en Generative Adversarial Networks, per a la implementació d'aquesta mena de models basats en la metodologia Progressive Growing of GAN's també han de ser implementades altres xarxes neuronals auxiliars per a la realització d'alguns càlculs o algorismes pertanyents a la xarxa neuronal generativa, la xarxa neuronal discriminativa o ambdues.

4.3.2.1 Weighted scale convolution

A conseqüència de la inexistència d'una escala d'imatges estàtica durant tot el procés d'entrenament dins de les Progressive Growing of GAN's, aquest dinamisme ha de ser tingut en compte, sent implementades unes capes convolucionals especials per a això, on el càlcul del valor de sortida de cada neurona és calculat mitjançant el sumatori d'una convolució estàndard i diversos paràmetres de bias.

4.3.2.2 Normalització de píxels

La xarxa neuronal mitjançant la qual seran normalitzats els valors dels píxels ha estat implementada en el fitxer `src/gan/model/pixel_norm.py`.

Per a cadascuna de les cel·les pertanyents a aquesta xarxa neuronal és realitzat el càlcul que ha sigut especificat a l'equació 7, per tant, en la seva funció `forward()` serà realitzat aquest càlcul, tal com pot ser comprovat a la Figura 16.

```
def forward(self, x: torch.Tensor, epsilon: float=1e-8) -> torch.Tensor:
    mean_of_squares = torch.mean(x**2, dim=1, keepdim=True)
    normalization_factor = 1 / torch.sqrt(mean_of_squares + epsilon)

    return x * normalization_factor
```

Figura 16: Codi font de la xarxa neuronal de normalització de píxels

4.3.2.3 Bloc de convolució personalitzat

La implementació de l'algorisme que serà descrit a continuació ha estat realitzada en l'arxiu `src/gan/model/components/cnn_block.py`.

Dins de la topologia de les xarxes neuronals, tant generativa com discriminativa, han estat implementades diverses capes de convolució, mitjançant les quals és alterada la dimensiona-

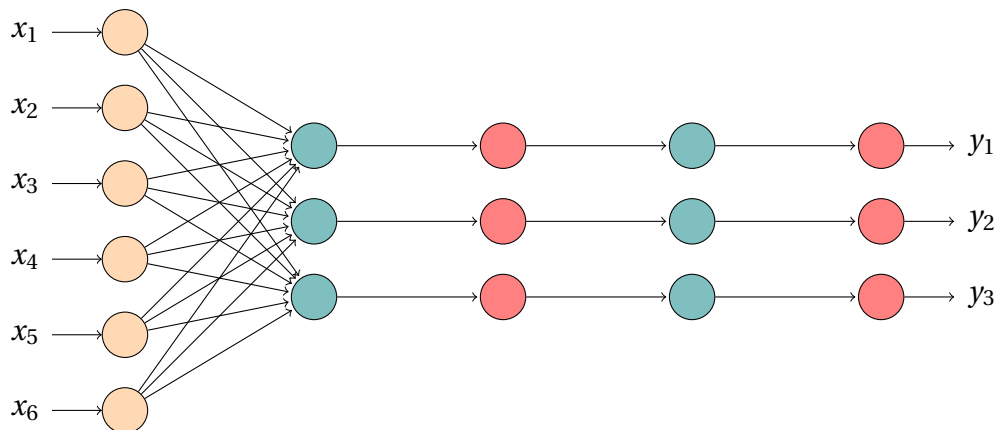


Figura 18: Topologia del bloc de convolució personalitzat per a la xarxa neuronal discriminativa, on amb el color groc és representada la capa d'entrada, el color blau la capa de *Weighted Scale Convolution*, el color vermell la capa d'activació.

4.3.3 Xarxa neuronal generativa

L'estructura de la xarxa neuronal generativa ha estat definida mitjançant una concatenació de capes de redimensionat d'imatge, més concretament és doblegada la resolució d'aquesta, juntament amb una capa de *Weighted Scale Convolution*, respectivament. Més concretament, aquesta concatenació de convolucions a diferents escales és mostrada a la Taula 2.

Operació	Funció d'activació	Dimensió d'imatge de sortida
Upsample		3 x 8 x 8
WSConv 3x3	Leaky RELU	3 x 8 x 8
Upsample		3 x 16 x 16
WSConv 3x3	Leaky RELU	3 x 16 x 16
Upsample		3 x 32 x 32
WSConv 3x3	Leaky RELU	3 x 32 x 32
Upsample		3 x 64 x 64
WSConv 3x3	Leaky RELU	3 x 64 x 64
Upsample		3 x 128 x 128
WSConv 3x3	Leaky RELU	3 x 128 x 128
Upsample		3 x 256 x 256
WSConv 3x3	Leaky RELU	3 x 256 x 256
Upsample		3 x 512 x 512
WSConv 3x3	Leaky RELU	3 x 512 x 512
Upsample		3 x 1024 x 1024
WSConv 3x3	Leaky RELU	3 x 1024 x 1024

Taula 2: Estructura del model de la xarxa neuronal generativa.

4.3.4 Xarxa neuronal discriminativa

L'estructura de la xarxa neuronal discriminativa, i de forma contrària a la xarxa neuronal generativa, ha estat definida mitjançant una concatenació de capes de Weighted Scale Convolution, juntament amb una capa de redimensionat d'imatge, més concretament és reduïda a la meitat la resolució d'aquesta, respectivament. Més concretament, aquesta concatenació de convolucions a diferents escales és mostrada a la Taula 3.

Operació	Funció d'activació	Dimensió d'imatge de sortida
WSConv 3x3	Leaky RELU	3 x 1024 x 1024
Downsample		3 x 1024 x 1024
WSConv 3x3	Leaky RELU	3 x 512 x 512
Downsample		3 x 512 x 512
WSConv 3x3	Leaky RELU	3 x 256 x 256
Downsample		3 x 256 x 256
WSConv 3x3	Leaky RELU	3 x 128 x 128
Downsample		3 x 128 x 128
WSConv 3x3	Leaky RELU	3 x 64 x 64
Downsample		3 x 64 x 64
WSConv 3x3	Leaky RELU	3 x 32 x 32
Downsample		3 x 32 x 32
WSConv 3x3	Leaky RELU	3 x 16 x 16
Downsample		3 x 16 x 16
WSConv 3x3	Leaky RELU	3 x 8 x 8
Downsample		3 x 8 x 8

Taula 3: Estructura del model de la xarxa neuronal discriminativa.

5 Resultats i discussió

En aquesta secció són reportats els resultats de la recerca que ha estat realitzada a partir dels materials i mitjançant les metodologies que han estat exposades anteriorment. Els resultats seran exposats de forma completament objectiva, sense cap biaix o interpretació, amb l'objectiu últim que pugui ser confirmada la hipòtesi inicial o, per contra, que sigui desmentida la hipòtesi inicial i puguin ser considerades noves perspectives d'enfocament mitjançant les quals pugui ser articulada una nova recerca.

5.1 Calibratge de hiperparàmetres

En primer lloc, han estat realitzats per una màquina amb les especificacions de la Taula 4 un conjunt d'experiments per al calibratge d'alguns dels hiperparàmetres del model que ha estat implementat. Per això, l'entrenament que serà fet en els següents experiments no serà complet, únicament sent buscat unes configuracions concretes amb les quals pugui ser obtinguda una estabilitat de les funcions de pèrdua, tant de la xarxa neuronal generativa com la discriminativa. Més concretament, només seran generades imatges de 16x16 píxels.

Típus de component	Model utilitzat
CPU	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz (x8)
GPU	Intel(R) Iris(R) Xe Graphics
GPU	NVIDIA GeForce MX450
RAM	16,0 GB (15,8 GB usable)

Taula 4: Recursos de hardware que han sigut utilitzats.

En un primer experiment han estat configurats els hiperparàmetres com en la Taula 5 i han estat aconseguits uns resultats a la funció de loss del generador que poden ser vists a la Figura 19 i al discriminador que poden ser vists a la Figura 20.

Posteriorment, en un segon experiment han estat configurats els hiperparàmetres com en la Taula 6 i han estat aconseguits uns resultats a la funció de loss del generador que poden ser vists a la Figura 21 i al discriminador que poden ser vists a la Figura 22.

En últim lloc, en un tercer experiment han estat configurats els hiperparàmetres com en la Taula 7 i han estat obtinguts els següents resultats i han estat aconseguits uns resultats a la funció de loss del generador que poden ser vists a la Figura 23 i al discriminador que poden ser vists a la Figura 24.

Com pot ser apreciat, amb les configuracions de hiperparàmetres de l'experiment 1! és

Hiperparàmetre	Valor
START_TRAIN_AT_IMG_SIZE	16
PATH_DATASET	C:/Users/carra/OneDrive/Escritorio/dataset
PATH_GENERATOR_CHECKPOINT	generator.pth
PATH_DISCRIMINATOR_CHECKPOINT	discriminator.pth
DEVICE	'cuda' if torch.cuda.is_available() else 'cpu'
SAVE_MODEL	True
LOAD_MODEL	True
LEARNING_RATE	1e-3
BATCH_SIZES	[32, 32, 32, 16, 16, 16, 16, 8, 4]
IMAGE_SIZE	512
CHANNELS_IMG	3
Z_DIM	256
IN_CHANNELS	256
CRITIC_ITERATIONS	1
LAMBDA_GP	10
NUM_STEPS	$\text{int}(\log_2(\text{IMAGE_SIZE} / 4)) + 1$
PROGRESSIVE_EPOCHS	[10] * len(BATCH_SIZES)
FIXED_NOISE	<code>torch.randn(8, Z_DIM, 1, 1).to(DEVICE)</code>
NUM_WORKERS	4
FACTORS	[1, 1, 1, 1, 1/2, 1/4, 1/8, 1/16, 1/32]

Taula 5: Valors dels hiperparàmetres que han sigut establerts al primer experiment.

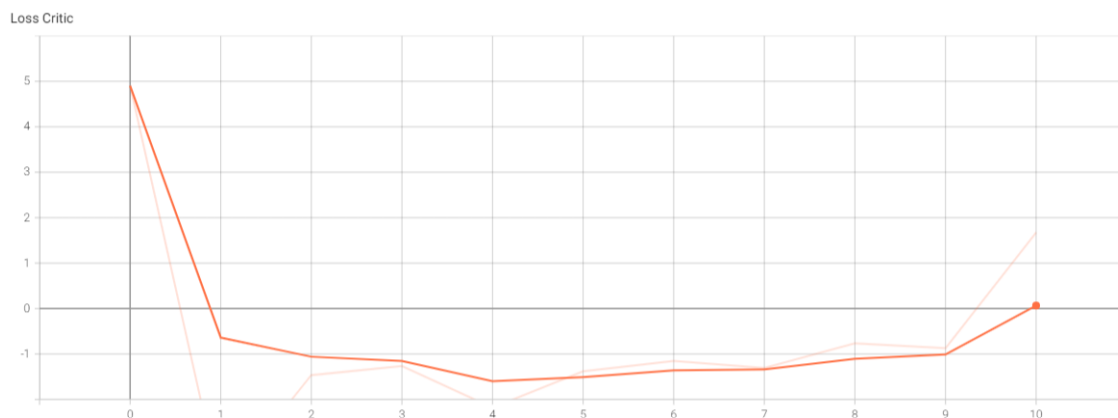


Figura 19: Resultats de la funció de loss del generador a l'experiment 1. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

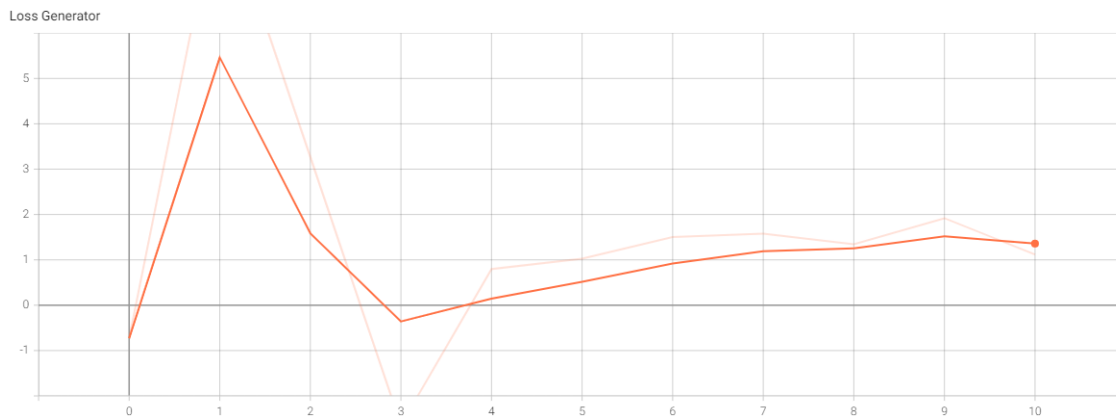


Figura 20: Resultats de la funció de loss del discriminador a l'experiment 1. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

Hiperparàmetre	Valor
START_TRAIN_AT_IMG_SIZE	16
PATH_DATASET	C:/Users/carra/OneDrive/Escritorio/dataset
PATH_GENERATOR_CHECKPOINT	generator.pth
PATH_DISCRIMINATOR_CHECKPOINT	discriminator.pth
DEVICE	'cuda' if torch.cuda.is_available() else 'cpu'
SAVE_MODEL	True
LOAD_MODEL	True
LEARNING_RATE	1e-2
BATCH_SIZES	[32, 32, 32, 16, 16, 16, 16, 8, 4]
IMAGE_SIZE	512
CHANNELS_IMG	3
Z_DIM	256
IN_CHANNELS	256
CRITIC_ITERATIONS	1
LAMBDA_GP	10
NUM_STEPS	$\text{int}(\log_2(\text{IMAGE_SIZE} / 4)) + 1$
PROGRESSIVE_EPOCHS	[10] * len(BATCH_SIZES)
FIXED_NOISE	<code>torch.randn(8, Z_DIM, 1, 1).to(DEVICE)</code>
NUM_WORKERS	4
FACTORS	[1, 1, 1, 1, 1/2, 1/4, 1/8, 1/16, 1/32]

Taula 6: Valors dels hiperparàmetres que han sigut establerts a l'experiment 2.

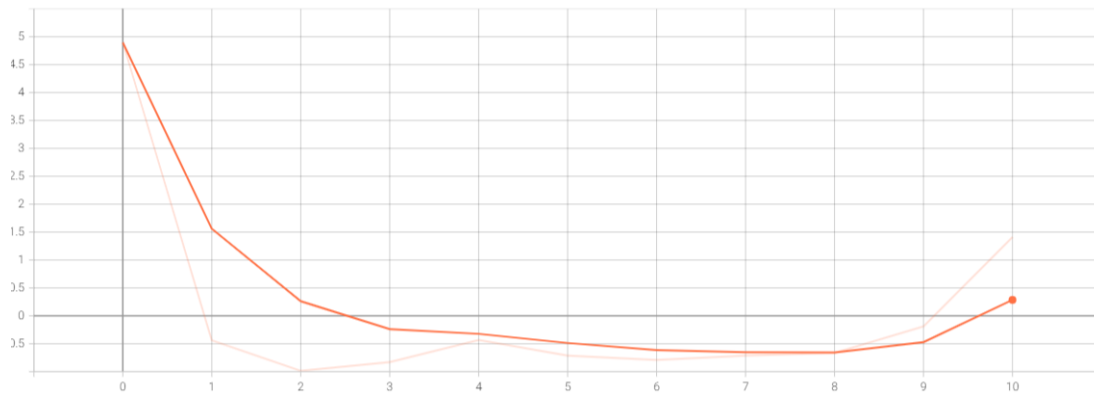


Figura 21: Resultats de la funció de loss del generador a l'experiment 2. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.



Figura 22: Resultats de la funció de loss del discriminador a l'experiment 2. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

Hiperparàmetre	Valor
START_TRAIN_AT_IMG_SIZE	16
PATH_DATASET	C:/Users/carra/OneDrive/Escritorio/dataset
PATH_GENERATOR_CHECKPOINT	generator.pth
PATH_DISCRIMINATOR_CHECKPOINT	discriminator.pth
DEVICE	'cuda' if torch.cuda.is_available() else 'cpu'
SAVE_MODEL	True
LOAD_MODEL	True
LEARNING_RATE	1e-1
BATCH_SIZES	[32, 32, 32, 16, 16, 16, 16, 8, 4]
IMAGE_SIZE	512
CHANNELS_IMG	3
Z_DIM	256
IN_CHANNELS	256
CRITIC_ITERATIONS	1
LAMBDA_GP	10
NUM_STEPS	$\text{int}(\log_2(\text{IMAGE_SIZE} / 4)) + 1$
PROGRESSIVE_EPOCHS	$[10] * \text{len}(\text{BATCH_SIZES})$
FIXED_NOISE	<code>torch.randn(8, Z_DIM, 1, 1).to(DEVICE)</code>
NUM_WORKERS	4
FACTORS	[1, 1, 1, 1, 1/2, 1/4, 1/8, 1/16, 1/32]

Taula 7: Valors dels hiperparàmetres que han sigut establerts a l'experiment 3.

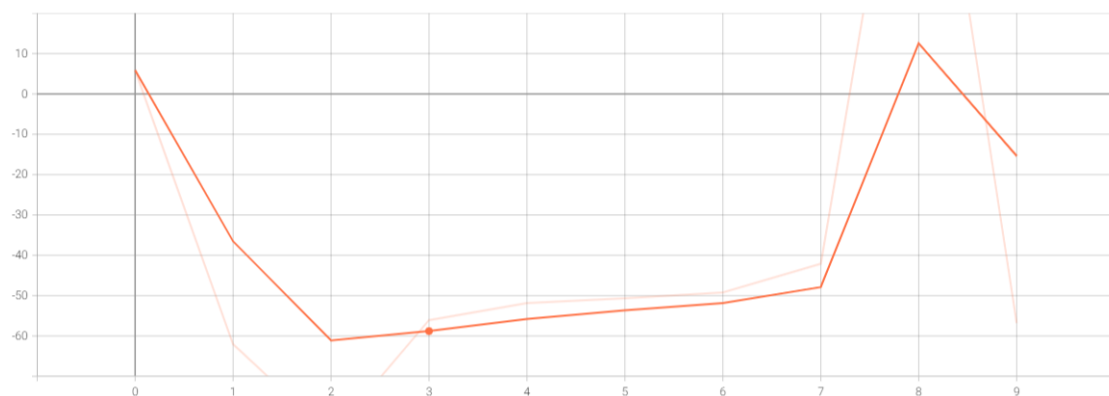


Figura 23: Resultats de la funció de loss del generador a l'experiment 3. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

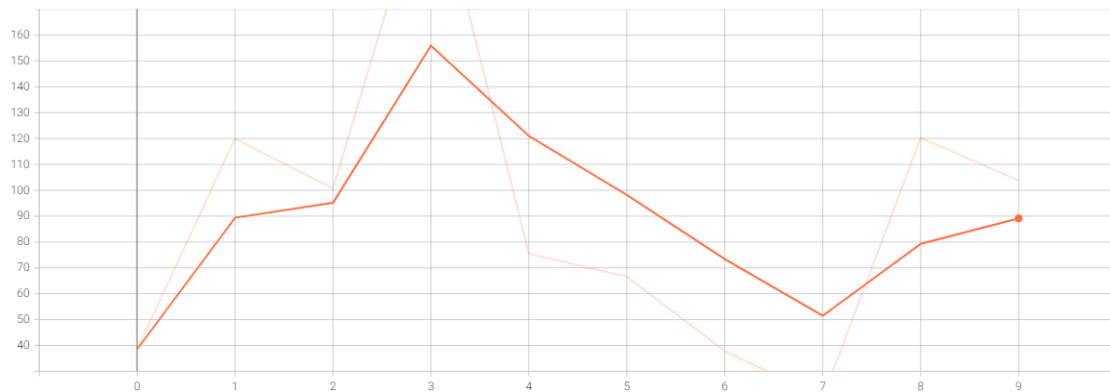


Figura 24: Resultats de la funció de loss del discriminador a l'experiment 3. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

aconseguit un entrenament més estable, per la qual cosa aquestes seran conservades per a experiments a major escala.

5.2 Experiment a gran escala

Tal com havia estat especificat anteriorment, per a aquest experiment són utilitzades les configuracions de hiperparàmetres de la Taula 5. A més d'això, la màquina les especificacions de la qual han estat especificades en la Taula 4 continua sent emprada per a aquest experiment.

Pel que concerneix les resolucions d'imatges que pretenen ser generades en aquest experiment, en l'entrenament seran incloses les resolucions 16x16, 32x32 i 64x64. La idea de la generació d'imatges amb resolucions més altes ha estat rebutjada a causa de la insuficiència computacional que ha estat trobada per a això.

En el present experiment han estat aconseguits uns resultats a la funció de loss del generador que poden ser vists a la Figura 25 i al discriminador que poden ser vists a la Figura 26.

Dins de les Figures 25 i 26 han estat executats 10 epochs per a cadascuna de les resolucions, per la qual cosa cada 10 passos en totes dues figures són trobats alguns pics, a causa de la nova resolució d'imatge a ser processada i, per tant, uns nous càlculs que han de ser realitzats.

A més d'això, com han estat executades dues execucions completes per resolució, més concretament han estat computades totalment els càlculs de les resolucions 16x16 i 32x32, pot ser obtingut una variació en la Fréchet Inception Distance que és calculada per a cada resolució. Més concretament, en la Figura 27 pot ser benvolgut que aquest coeficient i les

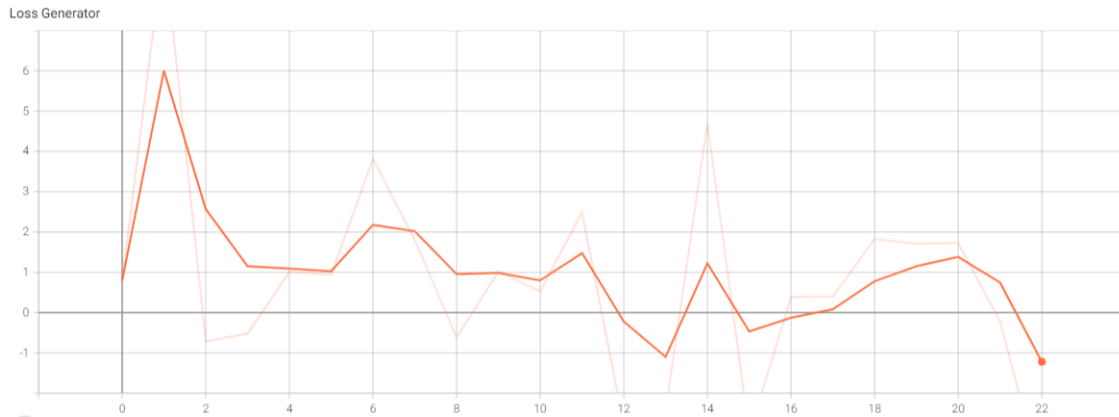


Figura 25: Resultats de la funció de loss del generador a l'experiment 4. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

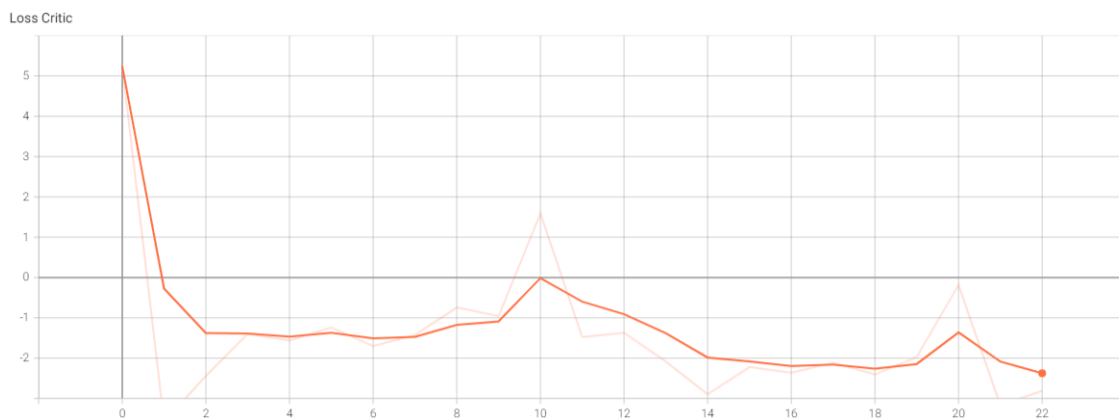


Figura 26: Resultats de la funció de loss del discriminador a l'experiment 4. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

resolucions són augmentats de manera directament proporcional.

Aquesta relació de proporcionalitat entre aquestes dues magnituds pot ser deguda al fet que, com més píxels són inclosos en una imatge, més probabilitats hi ha que les estructures de la forma d'una mama de la imatge puguin ser diferents de les estructures del conjunt de dades original.

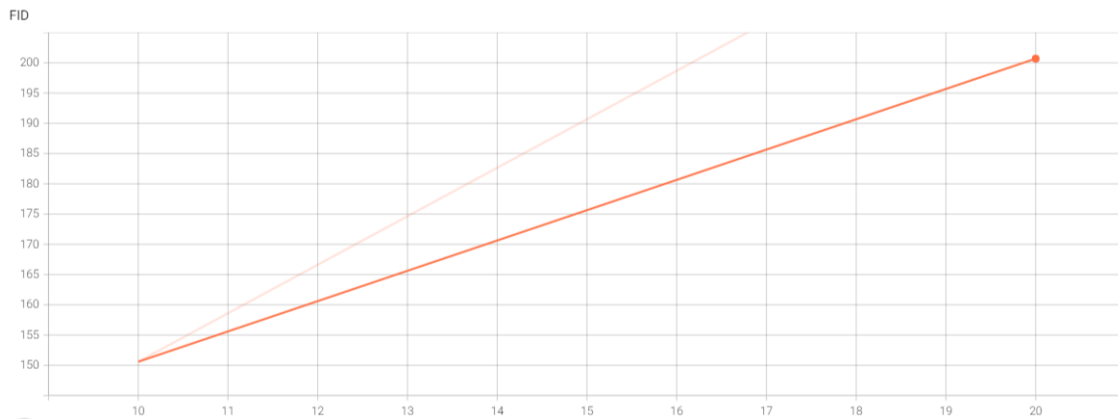


Figura 27: Resultats del progrés de la Fréchet Inception Distance a l'experiment 4. En l'eix horitzontal són representats els epochs, mentre que en l'eix vertical és representat el valor a ser mesurat. Mentrestant, per la línia taronja intensa és representat el valor smoothed del valor mesurat, mentre que pel color taronja feble és representat aquest mateix valor sense el suavitzat.

En el relacionat a exemples de les imatges que han estat generades, poden ser visualitzades en les figures 28, 29, 30 diversos samples de les imatges de 16x16, 32x32 i 64x64 píxels de resolució, respectivament.

A causa de la insuficiència de judici i coneixement radiològic per a la correcta i rigorosa avaluació de les imatges mèdiques que han estat generades, el rendiment del model pot ser qualificat d'acceptable gràcies a l'estructura relativament similar de les imatges mèdiques generades automàticament en contraposició amb les del conjunt de dades original.

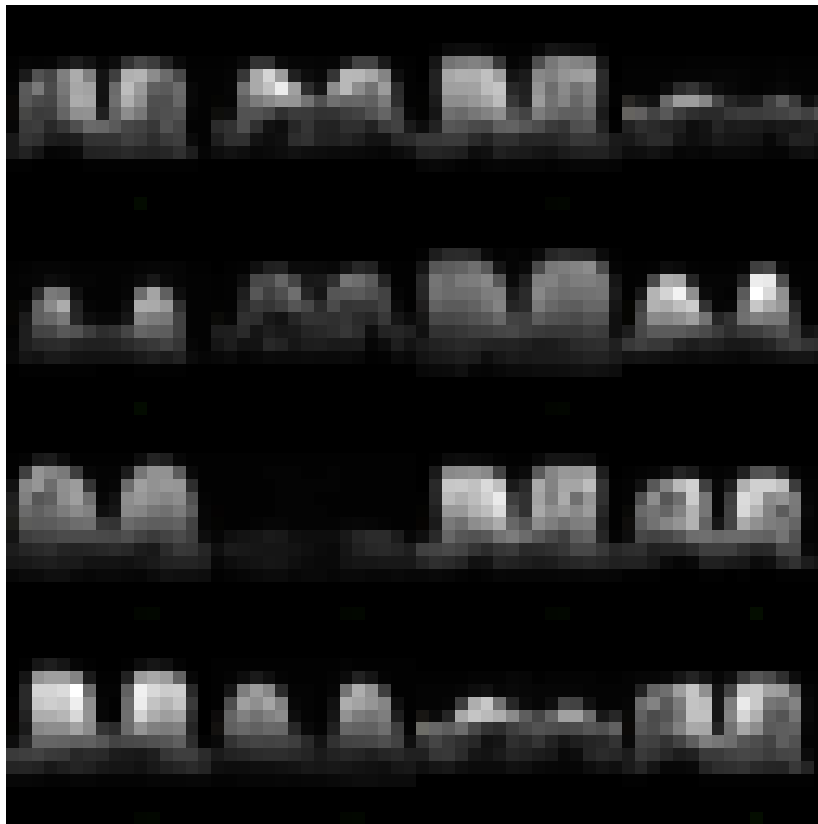


Figura 28: Subconjunt de les imatges de 16x16 píxels que han sigut generades.

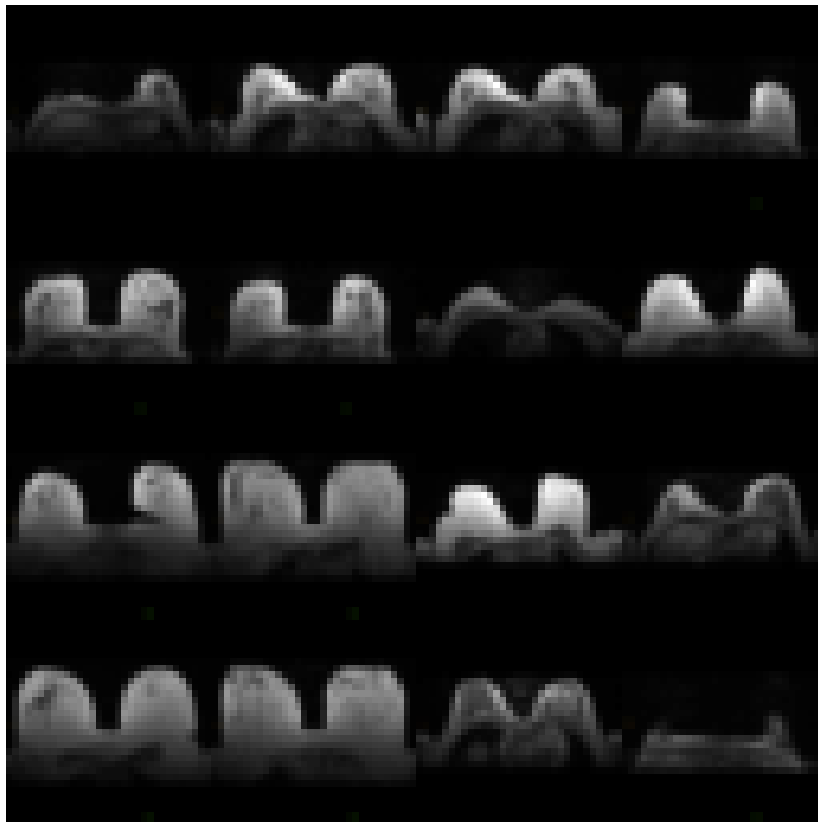


Figura 29: Subconjunt de les imatges de 32x32 píxels que han sigut generades.

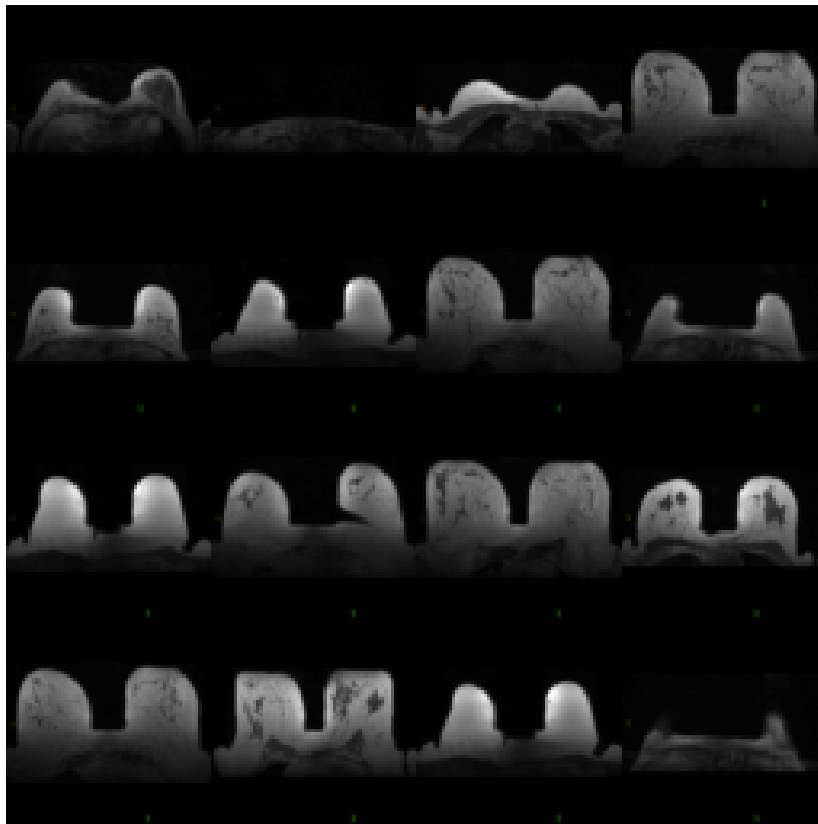


Figura 30: Subconjunt de les imatges de 64x64 píxels que han sigut generades.

6 Conclusions

Durant el procés de realització del present projecte han estat observades diverses problemàtiques a solucionar, juntament amb possibles ampliacions de la implementació i recerca dels models de Machine Learning aplicats a aquest àmbit.

6.1 Problemes trobats

El desenvolupament d'aquest projecte ha estat caracteritzat per l'afrontament de diversos problemes relatius, alguns d'ells podent ser vistos com a obstacles inexorables en projectes d'aquest àmbit, encara que uns altres podrien haver estat evitats amb una millor planificació.

6.1.1 Capacitat computacional

La insuficiència de capacitat computacional ha estat catalogada com un dels grans problemes del desenvolupament del present projecte, sent notada aquesta majorment a mesura que la resolució de les imatges de l'entrenament segons la metodologia Progressive Growing of GAN's era incrementat. Aquesta insuficiència de capacitat computacional per a l'entrenament del model ha estat causada pels grans requisits de maquinari per a l'entrenament de models basats en aquesta metodologia, en composició amb la reduïda capacitat de còmput de les màquines en les quals va ser entrenat el model.

Una possible solució a la problemàtica d'insuficiència de maquinari per a l'entrenament d'aquest model pot estar basada en la utilització d'un maquinari més potent i especialitzat en els càlculs amb tensors, els quals són l'estructura de dades bàsica dels models de Machine Learning. Actualment, han sigut creats diversos sistemes de maquinari basats en múltiples instàncies de Tensor Processing Units per part de Google [10], per tant, l'entrenament de models com l'implementat en aquest projecte podria ser optimitzat i, en conseqüència, executat completament en una menor quantitat de temps.

6.1.2 Escalabilitat del model

La limitada escalabilitat del model també ha estat catalogada com un altre dels grans problemes consubstancials a la implementació d'aquest projecte. Aquesta problemàtica ha estat causada per la poca variabilitat de les dades, més concretament pel que concerneix l'origen de les dades, pel fet que tots van ser proveïts pel Consorci Corporació Sanitària Parc Taulí i, per tant, ha estat introduït involuntàriament un biaix per culpa de l'homogeneïtat de les màquines amb les quals han estat preses les imatges mèdiques.

A manera de solució d'aquest problema de baixa escalabilitat del model a conseqüència de l'homogeneïtat de les dades d'entrada amb els quals aquest és entrenat, podria ser obtinguda una base de dades d'imatges mèdiques molt més heterogènia en el respecte al centre mèdic d'origen de les dades, encara que pot ser que els tràmits burocràtics per a l'accés a múltiples fonts de dades sigui considerat com un obstacle massa gran per a ser solucionat. Malgrat això, podria ser implementat un sistema d'aprenentatge federat, és a dir, una instància d'aquest model especialitzat en les dades d'un centre mèdic en concret que seria únicament utilitzat per aquest centre mèdic.

6.2 Possibles ampliacions de la implementació

El model basat en la metodologia de Progressive Growing of GAN's que ha estat implementat per al present treball pot ser millorat de diverses formes perquè pugui ser usat de diferents formes, com en un possible *pipeline* d'una composició de models de Machine Learning amb els quals puguin ser obtinguts uns resultats de major qualitat.

6.2.1 Processat directe d'imatges DICOM

Una de les possibles ampliacions del present treball podria estar protagonitzada per la capacitat de processament d'una imatge mèdica tridimensional en format DICOM de manera directa, en contraposició amb la necessitat de desglossat de la imatge tridimensional en una concatenació de múltiples imatges bidimensionals a les quals són aplicades filtres d'imatge. Amb la implementació d'aquesta millora podria ser obtingut un entrenament uniforme i únic de la xarxa neuronal, a més de la possibilitat d'obtenció d'una imatge DICOM com a tipus de dada de sortida de la xarxa, el qual podria ser processat directament per programes especialitzats d'anàlisi d'imatges mèdiques, com ara ITKSnap.

Referències

- [1] Artificial Intelligence in Medicine Lab
<https://www.bcn-aim.org/>
- [2] EuCanImage
<https://eucanimage.eu/>
- [3] Osuala, R., Kushibar, K., Garrucho, L., Linardos, A., Szafranowska, Z., Klein, S., Glocker, B., Diaz, O. and Lekadir, K., 2021. A Review of Generative Adversarial Networks in Cancer Imaging: New Applications, New Solutions.
<https://arxiv.org/abs/2107.09543v1>
- [4] Mirza, M. and Osindero, S., 2014. Conditional Generative Adversarial Nets.
<https://arxiv.org/abs/1411.1784>
- [5] Ioffe, S. and Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
<https://arxiv.org/abs/1502.03167>
- [6] Radford, A., Metz, L. and Chintala, S., 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.
<https://arxiv.org/abs/1511.06434>
- [7] O. Diaz, K. Kushibar, R. Osuala, A. Linardos, L. Garrucho, L. Igual, P. Radeva, F. Prior, P. Gkontra, K. Lekadir, Data preparation for artificial intelligence in medical imaging: A comprehensive guide to open-access platforms and tools, *Physica Medica* 83 (2021) 25–37.
[https://www.physicamedica.com/article/S1120-1797\(21\)00095-8/fulltext](https://www.physicamedica.com/article/S1120-1797(21)00095-8/fulltext)
- [8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative Adversarial Networks.
<https://arxiv.org/abs/1406.2661>
- [9] Karras, T., Aila, T., Laine, S. and Lehtinen, J., 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation.
<https://arxiv.org/abs/1710.10196>
- [10] Sato, K., Young, C. and Patterson, D., 2017. An in-depth look at Google's first Tensor Processing Unit (TPU) | Google Cloud Blog. [online] Google Cloud Blog.

<https://cloud.google.com/blog/products/ai-machine-learning/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>

- [11] O'Shea, K. and Nash, R., 2015. An Introduction to Convolutional Neural Networks.
<https://arxiv.org/pdf/1511.08458.pdf>
- [12] Lu, L., Shin, Y., Su, Y. and Em Karniadakis, G., 2019. Dying ReLU and Initialization: Theory and Numerical Examples.
<https://arxiv.org/abs/1903.06733>
- [13] Itksnap.org. 2022. ITK-SNAP Home.
<http://www.itksnap.org/pmwiki/pmwiki.php>
- [14] Generative Adversarial Networks
<https://deepai.org/machine-learning-glossary-and-terms/generative-adversarial-network>
- [15] LeakyRELU - PyTorch documentation
<https://pytorch.org/docs/stable/generated/torch.nn.LeakyReLU.html>
- [16] Complete Guide to Adam Optimization
<https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d>
- [17] Progressively-Growing GANs
<https://towardsdatascience.com/progressively-growing-gans-9cb795caebee>
- [18] Activation Function - Wikipedia
https://en.wikipedia.org/wiki/Activation_function
- [19] Activation Function Definition | DeepAI
<https://deepai.org/machine-learning-glossary-and-terms/activation-function>
- [20] 11 Important Model Evaluation Metrics for Machine Learning Everyone should know
<https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/>
- [21] Yield in Python: An Ultimate Tutorial on Yield Keyword in Python
<https://www.simplilearn.com/tutorials/python-tutorial/yield-in-python>

- [22] <https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/>
- [23] <https://towardsdatascience.com/handwritten-digit-mnist-pytorch-977b5338e627>
- [24] Python Imaging Library - Wikipedia
https://en.wikipedia.org/wiki/Python_Imaging_Library
- [25] TensorBoard | TensorFlow
<https://www.tensorflow.org/tensorboard/>
- [26] Foroozandeh, M. and Eklund, A., 2020. Synthesizing brain tumor images and annotations by combining progressive growing GAN and SPADE.
<https://arxiv.org/abs/2009.05946v1>
- [27] Zhang, Z., Zhao, T., Gay, H., Sun, B. and Zhang, W., 2020. Semi-supervised Semantic Segmentation of Prostate and Organs-at-Risk on 3D Pelvic CT Images. [online] arXiv.org.
<https://arxiv.org/abs/2009.09571>
- [28] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, H. Nakayama, Gan-based synthetic brain mr image generation, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE, 2018
- [29] Y. Liu, Y. Zhou, X. Liu, F. Dong, C. Wang, Z. Wang, Wasserstein gan-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology, *Engineering* 5 (1) (2019) 156–163.
- [30] A. Kapil, A. Meier, A. Zuraw, K. E. Steele, M. C. Rebelatto, G. Schmidt, N. Brieu, Deep semi supervised generative learning for automated tumor proportion scoring on nsclc tissue needle biopsies, *Scientific reports* 8 (1) (2018) 1–10
- [31] K. Chaitanya, N. Karani, C. F. Baumgartner, E. Erdil, A. Becker, O. Donati, E. Konukoglu, Semi-supervised task-driven data augmentation for medical image segmentation, *Medical Image Analysis* 68 (2021) 101934.
- [32] Y. Shin, H. A. Qadir, I. Balasingham, Abnormal colon polyp image synthesis using conditional adversarial networks for improved detection performance, *IEEE Access* 6 (2018) 56007–56017

- [33] S. D. Desai, S. Giraddi, N. Verma, P. Gupta, S. Ramya, Breast cancer detection using gan for limited labeled dataset, in: 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), IEEE, 2020, pp. 34–39.
- [34] BakingBrains/Progressive_GAN-ProGAN-_implementation
https://github.com/BakingBrains/Progressive_GAN-ProGAN-_implementation
- [35] Fréchet Inception Distance (FID) for Pytorch
<https://github.com/hukkelas/pytorch-frechet-inception-distance>