UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Models de Deep Learning per fer front a la COVID-19 a partir d'imatges mèdiques

TREBALL FI DE GRAU

Grau en Enginyeria Informàtica

Autor: Javier Martínez Bernia

Tutor: Jon Ander Gómez Adrián

Curs 2019-2020

Resum

En aquests moments estem vivint una pandèmia mundial causada pel SARS-CoV-2, un virus que produeix una enfermetat infecciosa coneguda com COVID-19 que provoca, entre altres, malalties respiratòries, com pneumònia en casos greus. Aquest projecte es centra en l'anàlisi d'imatge mèdica per a la detecció de pneumònia a partir de radiografies pulmonars per tal d'ajudar a la lluita contra la pandèmia.

Durant el projecte s'utilitzaran tècniques d'aprenentatge profund per a construir diversos classificadors amb l'ús de xarxes neuronals convolucionals que siguen capaços de detectar la infecció. La finalitat és que aquests classificadors puguen ajudar al personal mèdic a la criva de pacients que presenten simptomatologies durant la pandèmia. S'utilitzarà un conjunt d'imatges de radiografies pulmonars proporcionat pel Banc digital d'Imatge Mèdica de la Comunitat Valenciana -BIMCV-, les quals entrenaran els models construïts.

Paraules clau: aprenentatge automàtic, aprenentatge profund, xarxes neuronals, processament d'imatge, imatge mèdica

Resumen

En estos momentos estamos viviendo una pandemia mundial causada por el SARS-CoV-2, un virus que produce una enfermedad infecciosa que se conoce como COVID-19, la cual provoca, entre otras, enfermedades respiratorias, como neumonía en casos graves. Este proyecto se centra en el análisis de imagen médica para la detección de neumonía a partir de radiografías pulmonares con tal de ayudar en la lucha contra la pandemia.

Durante el proyecto se usarán técnicas de aprendizaje profundo para construir diversos clasificadores utilizando redes neuronales convolucionales los cuales sean capaces de detectar la infección. La finalidad es que estos clasificadores puedan ayudar al personal médico en el triaje de pacientes que presenten sintomatologías durante la pandemia. Se utilizará un conjunto de imágenes de radiografías pulmonares proporcionado por el Banco digital de Imagen Médica de la Comunidad Valenciana -BIMCV-, las cuales entrenarán los modelos construidos.

Palabras clave: aprendizaje automático, aprendizaje profundo, redes neuronales, procesamiento de imagen, imagen médica

Abstract

Nowadays we are experiencing a worldwide pandemic caused by the SARS-CoV-2, a virus that produces an infectious disease known as COVID-19, which causes, among others, respiratory diseases, such as pneumonia in severe cases. This project is focused on medical image analysis for the detection of pneumonia from lung X-ray images in order to fight against the pandemic.

During the project, deep learning techniques will be used to build various classifiers that can detect the infection. The purpose is that these classifiers will

be able to help medical personnel at the screening of patients presenting symptomatologies during the pandemic. A set of lung X-ray images provided by the Medical Imaging Databank of the Valencia Region -BIMCV- will be used to train the built models.

Key words: machine learning, deep learning, neural networks, image processing, medical image

Índex

Índex de figures

Índex de taules

CAPÍTOL 1
Introducció

En l'actualitat, gràcies a l'avançament dels dipositius d'adquisició d'imatge la quantitat de dades disponible és molt gran, cosa que fa de l'anàlisi d'imatge tot un repte i una disciplina molt interessant, sobre tot en el camp de la medicina. Aquest creixement en la quantitat d'imatges mèdiques requereixen d'un gran esforç per part dels professionals sanitaris per a l'anàlisi, el qual és subjectiu, propens a l'error humà i amb grans variacions entre distints experts.

La Intel·ligència Artificial ha progressat ràpidament els últims anys. Hui en dia, les tècniques d'Aprenentatge Automàtic (*Machine Learning*) i Aprenentatge Profund (*Deep Learning*) tenen un rol important en el camp de la medicina, com per exemple el processament d'imatge mèdica, diagnosi asistida per computadors, interpretació d'imatge o segmentació d'imatge.

1.1 Motivació

La COVID-19 (o enfermetat per coronavirus) és una enfermetat infecciosa causada per el virus SARS-CoV-2 [1]. Es va detectar per primera vegada a la ciutat xinesa de Wuhan a Desembre de 2019 després de diagnosticar a un grup persones amb pneumònia de causa desconeguda. L'Organització Mundial de la Salut -OMS- va reconéixer l'enfermetat com a pandèmia global el dia 11 de Març de 2020 [2]. El nombre de casos va continuar creixent fins alcançar una totalitat de $3,407,747$ casos confirmats i 238,198 casos de mort a dia 4 de Maig de 2020 [3].

L'enfermetat produeix símptomes semblants als de la grip, entre els quals inclouen febre, tos seca, dispnea, miàlgia i fatiga. En casos greus es caracteritza per produir pneumònia, síndrome del destret respiratori agut -SDRA-, sèpsia i xoc sèptic. Els símptomes apareixen entre dos i catorze dies després de l'exposició al virus, i hi ha evidència limitada de que el virus podria transmetre's uns dies abans d'experimentar símptomes [4]. Els països més afectats han entrat en quarantena, i 72% dels països han tancat les fronteres al turisme [5].

El projecte va sorgir perquè l'equip del Pattern Recognition and Human Language Technology -PRHLT-, experimentat en tasques relacionades amb classificació d'imatge mèdica i segmentació semàntica utilitzant *Deep Learning*, va decidir reorientar el seu esforç en desenvolupar classificadors basats en *Deep Neural*

Networks (DNNs) per fer front a la COVID-19, degut a l'impacte que estava tenint l'enfermetat.

1.2 Objectius

Els doctors prenen decisions considerant diverses imatges d'entrada. Les imatges de raigs X són una d'aquestes entrades les quals es poden obtindre de manera més ràpida que altres com la Tomografia Computada (TC) o les Imatges de Ressonància Magnètica (IRM). Per a fer front a la pandèmia hem decidit dissenyar i avaluar diversos classificadors binaris basats en *Convolutional Neural Networks (CNNs)*. L'eixida de classificadors podria ser utilitzada com a puntuació o mesura de confiança de la detecció de la infecció acompanyant les imatges mèdiques.

L'adopció de nous indicadors en protocols mèdics és un llarg procés que ha de passar per molts processos de validació i pel veredicte de comitès especialitzats. Baix les circumstàncies causades per la COVID-19 eixos processos podrien alleugerar-se, o simplement alguns indicadors podrien ser acceptats temporalment.

Malgrat la incertesa de l'ús de puntuacions en escenaris reals, els nostres esforços s'han centrat en l'entrenament i l'avaluació de classificadors basats en *DNNs* (per exemple, amb diferents topologies) que puguen proveir una mesura de confiança de que el pacient tinga pneumònia donada una radiografia pulmonar.

L'objectiu no tècnic d'aquest treball és contribuir amb un classificador basat en *DNNs* que poguera ser adoptat per doctors per a prendre decisions a la criva de pacients que presenten simptomatologies compatibles amb pneumònia durant la pandèmia. No obstant, som conscients de que és una remota possibilitat que depen en criteri mèdic i comitès ètics abans d'acceptar el seu ús a escenaris reals.

1.3 Estructura de la memòria

Aquesta memòria s'estructura en cinc capítols. Al primer, s'ha fet una introducció al projecte i al context en el qual s'ha desenvolupat. Al segon capítol es comenta l'estat de l'art, exposant altres projectes similars. Al tercer, es fa una descripció del problema i de les tècniques i ferramentes utilitzades. A continuació es detallen els experiments realitzats i es mostren els resultats obtinguts al quart capítol. Finalment, a l'utim capítol s'exposen les conclusions a les que s'ha arribat amb el projecte.

1.4 Col·laboració

Aquest projecte s'ha desenvolupat en col·laboració amb l'equip del projecte DeepHealth[1] del centre d'investigació Pattern Recognition and Human Language Technology, el qual ha intentat aportar models de *Deep Learning* per al problema

[1] Web del projecte DeepHealth: https://deephealth-project.eu/

plantejat pel Banc digital d'Imatge Mèdica de la Comunitat Valenciana, tal com estaven fent altres equips.

La generació dels models s'ha dividit entre els membres així com les proves. Mentre que uns membres feien proves amb l'eina Keras i TensorFlow altres han estat treballant amb la EDDL[2], una llibreria de codi obert per a entrenament distribuït de models de *Deep Learning*, la qual es troba en desenvolupament per l'equip. Personalment m'he dedicat a la generació de models i proves amb Keras i TensorFlow.

[2]Github del projecte EDDL: https://github.com/deephealthproject/eddl

CAPÍTOL 2
Estat de l'art

Degut a l'emergència de la pandèmia, s'estan duent a terme molts estudis utilitzant imatges de raigs X dels pulmons i de Tomografia Computada. Encara que les radiografies són més complicades per a aconseguir bons resultats que les de Tomografia Computada, són preferibles perquè són accessibles més fàcilment. La majoria de radiografies per a tasques de COVID-19 que s'han utilitzat als treballs publicats fins ara s'han obtingut majoritàriament de [6], i els conjunts de dades solen augmentar-se amb imatges de [7]. La majoria d'aquests estudis duen a terme la tasca de classificació d'imatges entre COVID-19 o no-COVID-19 [8]. Altres intenten diferenciar entre diversos tipus de pneumònia o entre pacients sans i pacients amb algun tipus de pneumònia [9].

En [10], s'ha utilitzat una tècnica anomenada *transfer learning* [1] per a un *dataset* de solament 146 imatges. Després d'aplicar *fine tuning*[2], la precisió va augmentar de 86% a 89.7%, encara que els autors afirmen que la xarxa estava sobreentrenant.

El *dataset* utilitzat en [9] està format per 306 imatges etiquetades en quatre classes: sa, pneumònia vírica, pneumònia bacteriana i pneumònia COVID-19. Es va aconseguir un 80% de precisió utilitzant la xarxa preentrenada *VGG16* intentant classificar en dues classes: sa i pneumònia vírica. Resultats similars es van obtindre al intentar classificar en tres classes: sa, pneumònia bacteriana i pneumònia COVID-19. No obstant això, quan es va introduir la classe pneumònia vírica al model multiclasse, la precissió va baixar al 60% degut a que l'enfermetat COVID-19 és causada per un virus. Aquests resultats mostren d'alguna manera la dificultat del model per a distingir entre pneumònia causada per COVID-19 i pneumònia vírica regular.

La tècnica *transfer learning* nomenada abans també s'ha utilitzat en [11], on s'han utilitzat dos models ja entrenats: *ResNet50* [12] entrenat amb el *dataset ImageNet* [13] i *Inception-ResNetV2* [14] entrenat amb *ImageNet-2012* [15]. Els resultats es van obtindre amb validació encreuada. El model que utilitzava la *ResNet50* va arribar al 98% de precissió, 96% de sensibilitat i 100% d'especificitat.

En [16], els autors van agafar el model preentrenat *ResNet50V2* com a base per al seu model proposat. Van afegir capes *Fully connected* amb *dropweights* al model preentrenat. Com que el seu objectiu era evitar els falsos negatius, van

[1] És una tècnica que es basa en utilitzar un model ja entrenat per a un altre problema.

[2] Es tracta d'unes tècniques de *transfer learning* que intenten acoplar el model preentrenat al problema en el qual s'està utilitzant

definir una funció de pèrdua que penalitzava més fortament als errors en imatges de COVID-19 que en imatges que no eren de COVID-19. El seu model també semblava mostrar incertesa.

Una idea similar es va aplicar en [8] per reduir la proporció de falsos negatius. Els autors van introduir un llindar que controlava l'intercanvi entre sensibilitat (proporció de vertaders positius) i l'especificitat (proporció de vertaders negatius). En [17] es va proposar un classificador multi-classe que és capaç de diferenciar imatges sense infecció d'imatges amb infecció distinta de COVID-19 i imatges amb COVID-19. Els autors van aplicar dues estratègies: una política de reducció del *learning rate* quan l'aprenentatge es quedava estancat durant un període de temps i un balanceig dels *batches*[3] per a equilibrar millor la distribució de cada tipus d'infecció a nivell de *batch*. El model va aconseguir un 92.40% d'*accuracy* amb una sensibilitat del 80% i una *precision* de 88.90% per a la classe COVID-19, que significa una proporció de falsos positius molt baixa a la detecció de COVID-19.

[3]Un *batch* és un grup d'imatges que se li dona d'entrada a la xarxa. Quan treballem amb *batches* s'alimenta l'entrada de la xarxa amb no una sinó diverses imatges cada volta.

CAPÍTOL 3
Anàlisi del problema i recursos utilitzats

Per a dur a terme els experiments disposem del conjunt de dades PADCHEST [18]. En aquest capítol descriurem el *dataset* i les tècniques i recursos que utilitzarem per a la resolució del problema.

3.1 Descripció del *dataset* PADCHEST

Aquest conjunt ha sigut proporcionat pel Banc digital d'Imatge Mèdica de la Comunitat Valenciana -BIMCV-, gràcies a la col·laboració entre el centre d'investigació Pattern Recognition and Human Language Technology -PRHLT- i la Fundació per al Foment de la Investigació Sanitària i Biomèdica de la Comunitat Valenciana -FISABIO-.

Està format per un total de 160,868 imatges corresponents a 67,625 pacients. 39,039 imatges de les 160,868 estan etiquetades manualment. Després d'un anàlisi d'exploració, només 23,244 de les 39,039 eren vàlides per al nostre propòsit, de les quals 11,989 s'han utilitzat, corresponents a la radiografia en vista Posterior-Anterior. Les imatges estan etiquetades en quatre classes: Control (**C**) -pacients sans-, Pneumònia (**N**) -pacients amb pneumònia-, Infiltració (**I**) -pacients amb infiltrats pulmonars- i Pneumònia i Infiltració (**NI**) -pacients amb ambdós diagnòstics-. S'ha decidit utilitzar aquest conjunt de dades donada la relació entre la infecció de pneumònia i la COVID-19, la qual provoca pneumònia, mentre esperem l'alliberació d'un *dataset* de radiografies específicament de COVID-19 pel BIMCV.

Als experiments utilitzarem el conjunt d'imatges dividit en tres subgrups: entrenament, validació i test. Utilitzarem el conjunt d'entrenament per a entrenar els models, el de validació per a controlar l'entrenament, i finalment el conjunt de test per a provar el model. Aquestes particions estaven definides pels autors del *dataset*. A la taula 3.1 podem trobar un resum de les particions i grups de les imatges.

Per a aquest *dataset* s'han definit dues tasques de classificació: Control (**C**) front a Pneumònia (**N**) i Control (**C**) front a Pneumònia(**N**) i Pneumònia i Infiltració (**NI**). La classe 0 està formada per les imatges del grup **C** (Control) en ambdós

Taula 3.1: Distribució de les imatges del *dataset* PADCHEST corresponent a les particions d'entrenament, validació i test i a les classes definides pels radiòlegs.

	Partició			
Classe (id)	Entrenament	Validació	Test	Total
Control (C)	4,095	1,371	1,369	6,835
Pneumònia (N)	1,420	456	473	2,349
Infiltració (I)	1,074	374	371	1,819
N & I (NI)	598	195	193	986
Total	7,187	2,396	2,406	11,989

casos, i la classe 1 està formada pel grup **N** a la primera tasca i pels grups **N** i **NI** a la segona tasca. Les imatges del grup **I** (Infiltració) no s'han utilitzat per suggerència dels radiòlegs, ja que segons ells les imatges amb infiltrats pulmonars són indistingibles d'imatges de control en alguns casos, i indistingibles d'imatges de pneumònia en altres casos.

3.2 *Tècniques de* Deep Learning

Per a entendre el projecte, és de vital importància entendre què és un model basat en *Deep Learning* i quines són les tècniques que es van a utilitzar per al tractament de la imatge mèdica. En aquesta secció entrarem en els conceptes teòrics més importants per poder entendre la memòria.

3.2.1. Model

Segons defineixen els autors a [19], el *Representation Learning* són un conjunt de mètodes que, donades unes dades d'entrada, permeten a la màquina aprendre automàticament les representacions necessàries sobre les dades per poder classificar-les o detectar-les. Un model de *Deep Learning* és un conjunt de mètodes de *Representation Learning* amb múltiples nivells de representació, que s'obtenen amb la composició de mòduls simples però no lineals els quals transformen la representació d'un nivell a una altra representació a un nivell superior i un poc més abstracte. Amb la composició de suficients transformacions es poden aprendre funcions més complexes. Una imatge, per exemple, es dóna d'entrada representada en forma de vector de píxels, i les característiques que s'aprenen en la primera capa de representació típicament representen la presència o absència de vores en zones particulars de l'imatge. A la segona capa es solen detectar dissenys amb la detecció d'ordenacions de vores. La tercera capa assemblaria els dissenys en combinacions que corresponen a objectes familiars, i les subsegüents capes detectarien objectes com a combinació d'eixes parts. L'aspecte clau del *Deep Learning* és que les capes de característiques no estan dissenyades per enginyers humans: són apreses a partir de les dades utilitzant un procediment d'aprenentatge. A aquestes construccions és al que ens referim quan parlem de model basat en *Deep Learning*.

Aquests models es construeixen amb xarxes neuronals artificials. Hi han molts tipus d'arquitectures de xarxes neuronals, de les quals entrarem en profunditat més endavant en les *Convolutional Neural Networks*, que són les que utilitzarem al nostre projecte.

3.2.2. Classificador

Amb els models el que es pretén construir són classificadors. Un classificador es podria definir com un algoritme que donades unes dades d'entrada, ens torna com a eixida a quin grup (o classe) dels possibles grups pertanyen les dades que li hem donat d'entrada. Segons el nombre de classes a les quals volem classificar, els models tindran un nombre de neurones determinat a la capa d'eixida, cadascuna representant a cada classe. Al nostre problema, com anem a construir classificadors binaris, a l'eixida tindrem dos neurones, és a dir, el classificador donarà com a eixida dos valors. A la capa d'eixida aplicarem una funció d'activació *Softmax*. Aquesta funció agafa els valors de l'eixida i els torna com a un vector els valors dels quals sumen 1. Ho fa calculant els exponents de cada valor i normalitzant amb la suma d'aquests exponents. Aquests valors es poden interpretar com a probabilitats i, la classe a la qual s'han classificat les dades d'entrada seria la de major valor. A més, com anem a classificar imatges en pacients sans o infectats, podrem utilitzar el valor per a la classe infectats com a mesura de confiança, de manera que els doctors tindrien un valor de puntuació a les imatges que podria ajudar-los a l'hora de determinar si un pacient té la infecció.

3.2.3. *Convolutional Neural Networks*

Per construir els classificadors utilitzarem xarxes neuronals artificials. Concretament, arquitectures de xarxes neuronals convolucionals, ja que són capaces de detectar dependències espacials i temporals a les dades d'entrada, molt important per a classificar imatges. L'arquitectura d'una *Convolutional Neural Network* típica consta d'una serie de fases. Les primeres fases consten de dos tipus de capes: convolucionals i de *pooling*.

Les capes convolucionals extrauen característiques de les imatges, mitjançant l'aplicació d'uns filtres. Agafant la imatge com a matriu de dos dimensions, els filtres (o *kernels*) són unes submatrius que es posicionen sobre la imatge. El que es fa és multiplicar els valors del filtre amb els valors superposats a la imatge per a obtindre una nova matriu del tamany del filtre. Posteriorment es fa una suma dels valors d'aquesta matriu (de vegades es normalitza) i s'obté un nombre que es guarda en una altra matriu. A continuació, es mou el filtre a una altra posició de la imatge i es fa el mateix càlcul de nou per a obtindre un altre valor, el qual es guarda també en la mateixa matriu que l'anterior. Aquest procés es repeteix per a cada filtre i cada possible posició a la imatge fins aconseguir una matriu per cada filtre aplicat que indica la presència o abscència de característiques a la imatge. Les matrius de cada filtre tenen de tamany el nombre de posicions distintes on hem aplicat el filtre, és a dir, tantes files com posicions hem mogut el filtre en vertical i tantes columnes com posicions hem mogut el filtre en horitzontal. Aquest procés es coneix com a convolució, i el que s'obté s'anomena mapa de

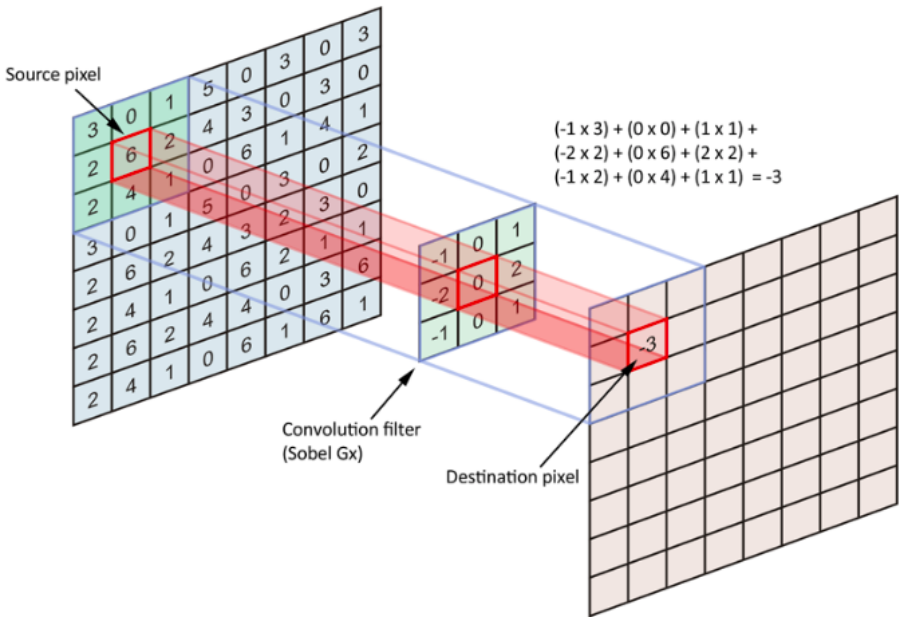


Figura 3.1: Funcionament d'una capa convolucional. El filtre es multiplica pels valors de la imatge per a obtindre un nou valor, el qual es guarda en una altra matriu a la posició central de la matriu del filtre. Imatge extreta de [20].

característiques (*feature map*). A la imatge 3.1 es pot observar el funcionament de les convolucions. Les capes convolucionals reben com a entrada una imatge (una matriu) i tornen a l'eixida un conjunt de *feature maps* (un conjunt de matrius). Als *feature maps* s'aplica la funció d'activació *ReLU* per a eliminar la linealitat.

Després de les convolucions, s'apliquen les capes de tipus *pooling*. El que fan és una reducció de la dimensionalitat dels *feature maps*, la qual cosa fa que la xarxa tinga menys paràmetres i puga estalviar temps d'entrenament i evitar l'*overfitting*. Aquestes capes redueixen cada *feature map* per separat, reduïnt l'amplària i l'altura. El mètode més comú de *pooling* és el *MaxPooling*, el qual agafa una finestra més menuda que la matriu i la passa per tota la matriu quedant-se amb el màxim valor dins de la finestra. A la figura 3.2 tenim un exemple d'aplicació. Aquesta tècnica ens permet reduir la dimensionalitat de la matriu sense perdre informació important.

A les arquitectures de xarxes convolucionals les capes convolucionals es solen encadenar al principi, i les capes de *pooling* es solen ficar entre algunes convolucionals. D'aquesta manera les característiques extretes es fan més complexes i les imatges es fan més compactes. És al que anomenem blocs de reducció. Després d'eixa fase inicial de convolucions i *poolings* es passa a una segona fase totalment connectada. S'afegeixen capes *Fully Connected* per a classificar les característiques extretes a la primera fase. Hi poden haver diverses capes *Fully Connected* amb nombre d'unitats variant, però l'ultima capa ha de tenir tantes unitats com classes, per poder dur a terme el problema de classificació. Com que les capes *Fully Connected* esperen rebre un vector d'una dimensió, es passa tot per una capa *Flatten* la qual transforma els filtres 2D en vectors d'una dimensió. També hi ha la opció de passar els filtres per una capa *GlobalMaxPooling*, que fa un *pooling* global a tots els filtres i dóna com a resultat un vector unidimensional. A la següent secció es parlarà d'aquestes capes.

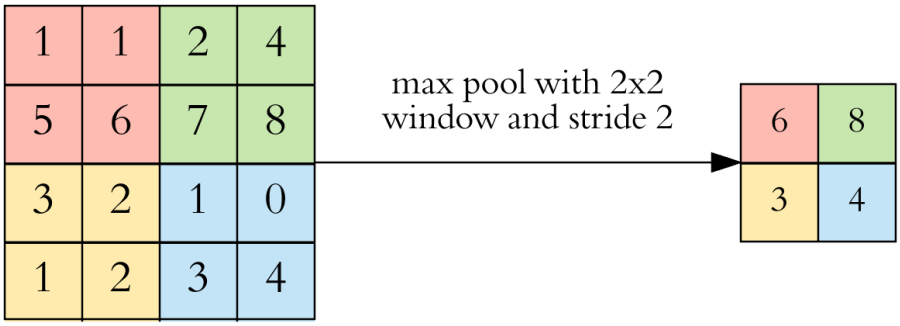
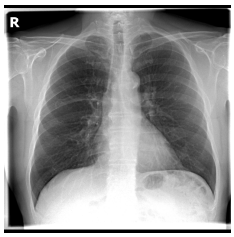


Figura 3.2: Funcionament d'una capa *MaxPooling* amb una finestra de 2x2 i *strides* de 2. Els *strides* indiquen el nombre d'elements que es mou la finestra en cada direcció. Imatge extreta de [21].

3.2.4. Capes utilitzades

En aquesta secció parlarem sobre les distintes capes que ofereix el software que utilitzarem per a construir les xarxes neuronals que utilitzarem com a classificadors.

- *Dense*: És una capa totalment connectada la qual calcula el producte del *input* per la matriu de pesos i li suma el *bias*. També ens podem referir a aquesta capa com a *Fully Connected*.
- *Flatten*: Com el seu nom indica aplana les dades que se li passen d'entrada. És útil després de les convolucions abans de passar a l'etapa *Fully Connected*, per tal de tenir les dades en un únic vector unidimensional.
- *Conv2D*: És la capa que fa les convolucions en dades de dues dimensions. Se li dóna com a paràmetres el nombre de filtres que es volen extraure i el tamany, entre altres. Dóna com a eixida un conjunt de matrius anomenat *Feature Maps*.
- *Concatenate*: Com indica el seu nom, aquesta capa concatena diversos tensors en un. La utilitzarem quan treballem amb diverses branques i vulguem agrupar-les en una.
- *Activation*: Aquesta capa aplica una funció d'activació a l'entrada i retorna el resultat. La funció a aplicar se li indica per paràmetre.
- *AveragePooling2D*: És una capa que realitza el *pooling* calculant la mitjana dels valors de la finestra aplicada a cada *feature map*. '2D' significa que treballa en dades de dues dimensions.
- *MaxPooling2D*: És una capa que realitza el *pooling* calculant el màxim dels valors de la finestra aplicada a cada *feature map*. '2D' significa que treballa en dades de dues dimensions.
- *GlobalMaxPooling2D*: És una capa que realitza un *pooling* global a tots els *feature maps* i torna com a resultat un vector unidimensional per cada element del *batch*. '2D' significa que treballa en dades de dues dimensions.



(a) Imatge original

(b) Imatge despés d'aplicar una rotació de 7.8 graus en sentit antihorari

Figura 3.3: Exemple de rotació aplicada a una imatge del *dataset* PADCHEST.

- *Dropout*: És una capa que aplica *Dropout* als valors d'entrada de la capa. El *Dropout* consisteix a posar a zero alguns valors dels que se li donen d'entrada. Es seleccionen de forma aleatòria amb una freqüència que es passa com a paràmetre. Els valors que no es modifiquen a zero són escalats per a que la suma de tots els valors no varie. Només s'aplica al procés d'entrenament i ajuda a previndre l'*overfitting*.

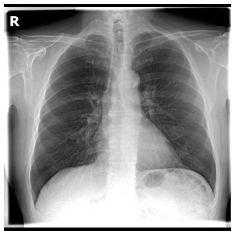
- *SpatialDropout2D*: Aquesta capa aplica el *Dropout* a *feature maps* complets en compte d'aplicar-ho a elements individuals.

- *BatchNormalization*: Aquesta capa normalitza les activacions de la capa anterior a cada *batch*, de manera que manté la mitjana pròxima a zero i la desviació estàndard pròxima a 1 abans d'entrar a la següent capa.

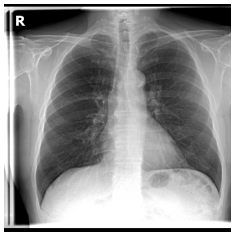
3.2.5. *Data Augmentation*

En aquestes tasques d'imatge mèdica, i en la majoria de models que utilitzen xarxes profundes, la quantitat de paràmetres que s'han d'aprendre són de l'ordre de milions. Per aquesta raó, per a tindre més mostres a l'hora d'entrenar la xarxa hem decidit aplicar *Data Augmentation* a les dades.

El *Data Augmentation* és un conjunt de tècniques que s'apliquen als conjunts de dades que augmenten el tamany d'aquests. Es basa en aplicar transformacions a les imatges del *dataset* per tal d'obtindre'n de noves. D'aquesta manera s'aconsegueix augmentar el conjunt, cosa que ens pot donar beneficis com evitar l'*overfitting* o tindre un nombre d'exemples adequat al nombre de paràmetres que ha d'aprendre el model. A més, també pot ajudar a augmentar el nombre de dades rellevants, cosa que facilita i millora l'aprenentatge. Al projecte aplicarem quatre tipus de transformacions: rotació, desplaçament, *zoom* i variació d'intensitat. La rotació gira la imatge un nombre de graus respecte d'un eix perpendicular i al centre de la imatge. El desplaçament mou el centre de la imatge un cert nombre de píxels verticals i horitzontals. El *zoom* aproxima o allunya la imatge sense variar el seu tamany. De fet, quan s'allunya la imatge els píxels que queden fora es plenen de color negre per a conservar el tamany. La variació d'intensitat augmenta o disminueix la intensitat segons un factor. A les figures 3.3, 3.4, 3.5 i 3.6 podem trobar exemples d'aquestes transformacions.

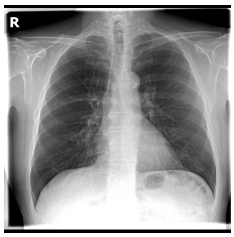


(a) Imatge original

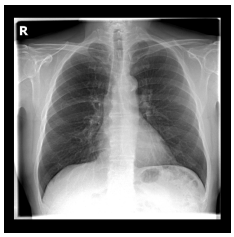


(b) Imatge després d'aplicar un desplaçament de -1 píxels en vertical i 13 en horitzontal

Figura 3.4: Exemple de desplaçament aplicat a una imatge del *dataset* PADCHEST.

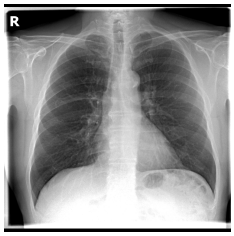


(a) Imatge original

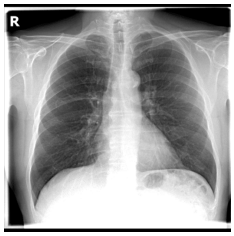


(b) Imatge despés d'aplicar un *zoom* de 0.91

Figura 3.5: Exemple de *zoom* aplicat a una imatge del *dataset* PADCHEST.



(a) Imatge original



(b) Imatge despés de variar la intensitat en un factor de 1.13

Figura 3.6: Exemple de variació d'intensitat d'una imatge del *dataset* PADCHEST.

3.3 Recursos *Hardware*

Per a treballar amb xarxes neuronals és necessària molta potència de càlcul, ja que els experiments fan moltes operacions i sense un hardware suficientment potent els experiments podrien tardar setmanes. Aquesta potència ens la ofereixen les GPUs, les quals tenen un gran potencial per a realitzar operacions matricials, molt requerides en aquestes tasques.

Al nostre problema, com que estem treballant en imatges 2D i necessitàvem molts recursos, hem utilitzat un total de 15 màquines pertanyents al centre d'investigació PRHLT i equipades cadascuna amb els següents components:

- CPU Intel(R) Core(TM) i7-7800X @ 3.50GHz (6 nuclis amb *hyper-threading*).
- 128 GB de memòria principal.
- 2 GPUs NVIDIA GeForce RTX 2080 amb 8 GB cada GPU.

3.4 Recursos *Software*

El llenguatge de programació triat per a desenvolupar el projecte ha sigut Python, ja que és un llenguatge d'alt nivell que és simple, consistent, i comprensible per als humans, per la qual cosa s'utilitza en la majoria de projectes d'Intel·ligència Artificial, ja que fa més fàcil la construcció de models d'aprenentatge automàtic. És un llenguatge independent de la plataforma i a més, disposa d'una llarga extensió de llibreries i *frameworks*, per la qual cosa s'ha convertit en un dels llenguatges de programació més populars en l'actualitat.

3.4.1. TensorFlow

TensorFlow és una plataforma de codi obert per a l'aprenentatge automàtic que ens facilita tant la compilació com la implementació de models [22]. Està desenvolupada per Google i és una plataforma molt popular i amb una gran comunitat al darrere. Proporciona APIs estables per a diversos llenguatges de programació com C, Python o JavaScript, entre altres. Hem escollit aquesta plataforma amb l'API de Python perquè ens permetrà utilitzar la llibreria Keras.

3.4.2. Keras

Keras és una API de *Deep Learning* escrita en Python, que s'executa sobre la plataforma TensorFlow [23]. Aquesta aplica una capa d'abstracció per damunt de TensorFlow a un nivell superior, la qual cosa fa més facil l'experimentació. Està dissenyada per a ser simple i consistent, a més de reduir les accions a l'usuari necessàries per als casos d'ús més comuns i proporcionar una bona i clara retroalimentació davant dels errors. Ens proporciona unes capes ja creades que ens faciliten la construcció de les xarxes neuronals, les quals hem explicat anteriorment. És l'eina software més utilitzada pels millors equips a les competicions

de Kaggle[1]. Per aquestes raons, com que necessitem fer molts experiments amb diversos models, hem decidit utilitzar l'API Keras amb la plataforma TensorFlow.

[1]Plataforma web on hi han conjunts de dades i on s'organitzen competicions. https://www.kaggle.com/

CAPÍTOL 4
Experiments

En aquest capítol entrarem en detall en els experiments realitzats al projecte. En primer lloc, farem una descripció de les topologies dels models proposats per a la resolució del problema. A continuació, parlarem de com s'ha preparat l'entrenament de les xarxes. Finalment, mostrarem i comentarem els resultats obtinguts després de realitzar les proves.

4.1 Models proposats

Les topologies de xarxes neuronals profundes proposades són classificadors binaris, on una classe està formada per imatges corresponents a Controls (pacients sans), típicament la classe 0, i la classe 1 està formada per imatges de pacients amb algun tipus de pneumònia.

Els models que anem a mostrar s'identifiquen per un nombre i una lletra. Per exemple, la lletra 'b' del model 6*b* fa referència a una de les possibles variants del model 6. Tots els models amb el mateix nombre utilitzen el mateix tipus de bloc per a les reduccions graduals i tenen les mateixes capes després de la seqüència de blocs de reducció. En general, les distintes variants dins d'un mateix model corresponen al tamany del *kernel* aplicat a l'imatge d'entrada en les capes convolucionals, encara que alguns models no segueixen aquestes condicions. Cal dir que els models s'han anat desenvolupant i provant al mateix temps, de manera que els nous models que s'anaven construint intentaven millorar els resultats amb la informació dels models que havien funcionat bé i els que no havien donat resultats bons.

Com a última observació abans de mostrar els models, a les següents subseccions veurem que hi ha variants que apliquen distintes capes en paral·lel a l'entrada. Aquestes estaven provant una arquitectura coneguda com a *Inception*, que es basa principalment en extraure filtres de distint tamany a la imatge d'entrada utilitzant una branca en paral·lel per cada tamany de filtre. S'ha utilitzat aquesta tècnica per a intentar aconseguir millors classificadors, encara que al treballar en branques en paral·lel augmentem enormement el nombre de recursos de computació a l'hora d'entrenar la xarxa.

4.1.1. Model 4

Aquest és el primer model proposat per a la resolució del problema. Es tracta d'un model amb quatre variants, que corresponen a les lletres *a*, *b*, *c* i *d*. Aquests models es caracteritzen per tenir diverses capes convolucionals aplicades en paral·lel a l'entrada, les quals es concatenen just després.

En el cas del 4*a*, es comença amb tres capes convolucionals aplicades en paral·lel a la imatge d'entrada, amb *kernels* que extrauen filtres de $7x7$, $9x9$ i $11x11$. Després, es concatenen les tres capes i s'entra a una etapa seqüencial de blocs de reducció, formada per capes convolucionals i activacions *ReLU*, on s'extrauen filtres més menuts de $1x1$ i $3x3$. A continuació, es fa un *GlobalMaxPooling* i es connecta la xarxa amb una capa *FullyConnected* formada per 256 neurones. Finalment, s'arriba a la capa d'eixida, amb dues neurones, corresponents a les dues classes on volem classificar. A la figura 4.1 podem veure la topologia del model.

Els models 4*b*, 4*c* i 4*d*, a diferència del 4*a*, comencen amb dues capes convolucionals en paral·lel en compte de tres. Aquestes, extrauen filtres de $5x5$ i $7x7$ en el cas del model 4*b*, de $7x7$ i $9x9$ en el cas del 4*c*, i de $9x9$ i $11x11$ en el model 4*d*. Després d'aquestes capes inicials, en els tres models entren en una seqüència de blocs de reducció semblant a la del model 4*a*. En el 4*b* la seqüència és més llarga que en els models 4*c* i 4*d*, però amb convolucions que extrauen menys filtres. Els filtres a aquesta part són, com en el model 4*a*, més menuts, ja que es tracta d'una part on es busca una reducció en el tamany dels filtres, per poder detectar característiques més específiques a la radiografia. Finalment, els tres models continuen com el model 4*a* després dels blocs de reducció fins a la capa d'eixida. Podem trobar la topologia d'aquests models a l'apèndix A.

4.1.2. Model 5

El model 5 és el segon tipus de model proposat per al problema. Consta de tres variants, 5*a*, 5*b* i 5*c*. La variant 5*a* porta dues branques en paral·lel des de l'entrada. Una comença amb l'extracció de filtres de $7x7$ píxels i l'altra de $9x9$. A continuació, s'entra en una seqüència de blocs de reducció semblant a les de les variants del model 4. La diferència és que no es concatenen les branques fins a la fi dels blocs de reducció, després d'una capa *GlobalMaxPooling2D*, de manera que es treballa amb dues línies al mateix temps. Després, es concatenen les dues línies i es connecta tot a una capa *FullyConnected* de 256 neurones la qual es connecta a la capa d'eixida de la mateixa manera que al model 4. A la figura 4.2 es pot observar la topologia.

Quant a les variants 5*b* i 5*c*, seria com la separació de les linies paral·leles que tenia el model 5*a* en dos models individuals, de manera que no hi ha cap paral·lelisme a les topologies, són dos models amb capes seqüencials. El 5*b* és la part que comença amb els filtres de $7x7$ i el 5*c* el que comença amb filtres de $9x9$. Després dels filtres, fins a la concatenació és tot igual que a les linies paral·leles del 5*a* exceptuant que en aquests casos no es concatena res al final. En compte d'anar connectat a una capa *FullyConnected*, es connecta a una capa *Flatten*, ja que no s'aplica la capa *GlobalMaxPooling2D* després de les convolucions, pel que necessitem aplanar les matrius perquè la capa *Dense* espera rebre dades en forma

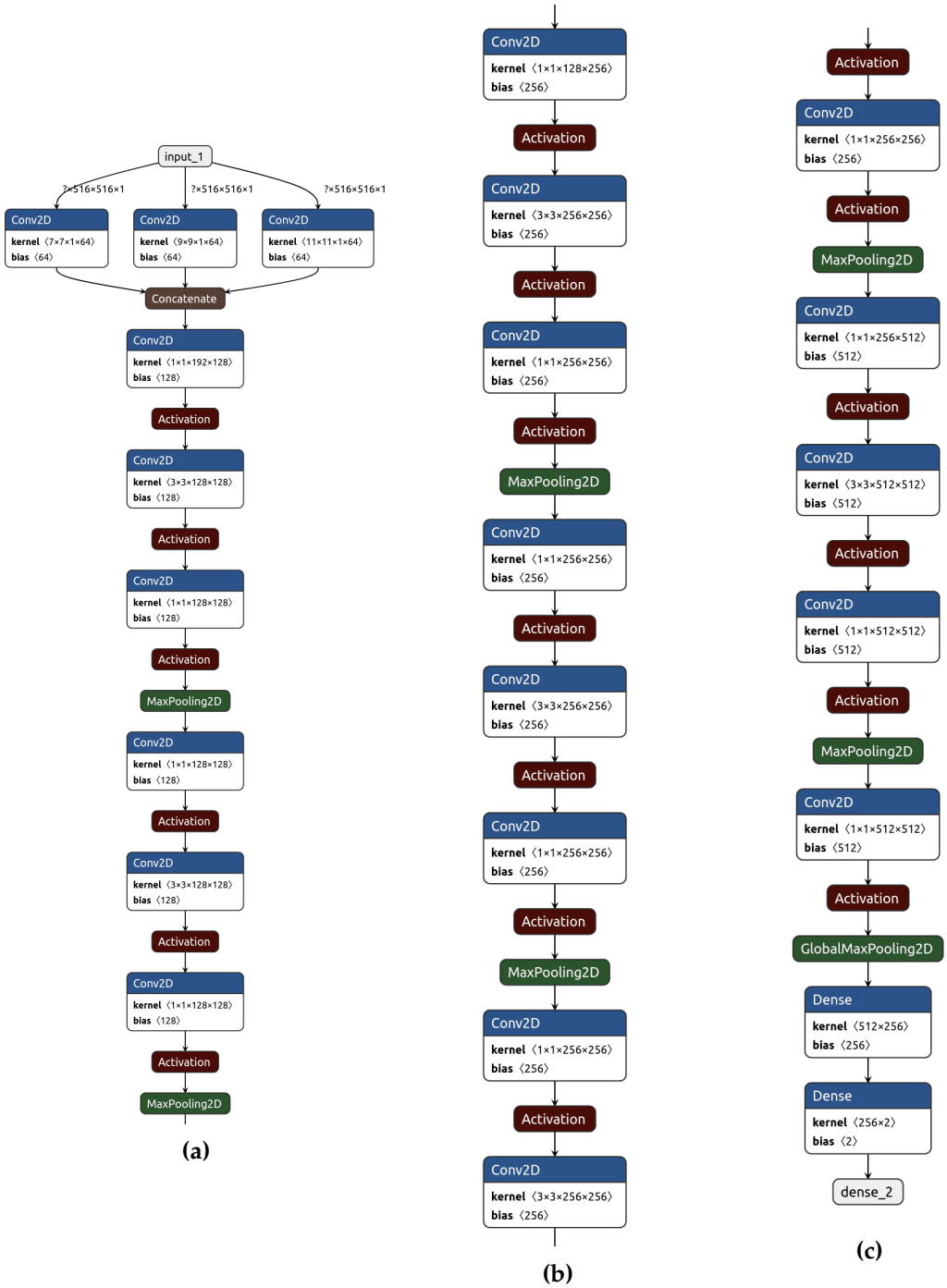


Figura 4.1: Topologia del model 4*a*.

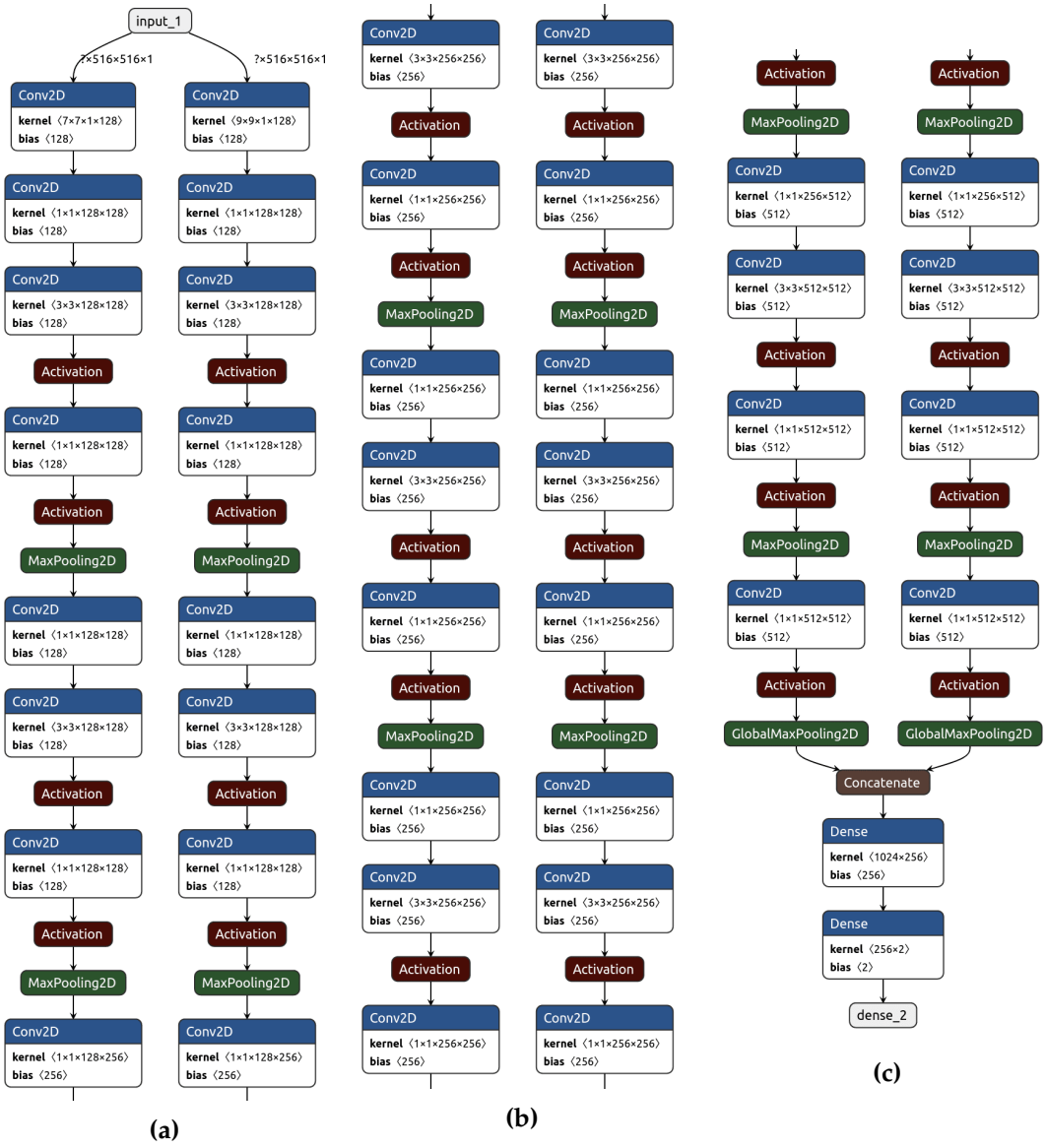


Figura 4.2: Topologia del model 5*a*.

de vector unidimensional. Finalment, es connecta tot a l'eixida. A l'apèndix A es poden trobar les topologies dels models 5*b* i 5*c*.

4.1.3. Model 6

El model 6 és un poc especial i distint als models vists fins ara. Consta de tres variants, de les quals la 6*a* és la més diferent. Aquesta variant treballa amb tres branques de capes convolucionals en paral·lel, començant cadascuna amb distints tamanys de *kernel* per als filtres i amb activacions de tipus *ReLU*. Aquestes tres branques es concatenen i es connecten a una capa *Flatten* que connecta amb la capa d'eixida. Quant a les altres dues variants, el model 6*b* es tracta del mateix model 6*a* però sols amb una branca de les tres, la qual comença amb un tamany de *kernel* de 7*x*7. La variant 6*c* seria el mateix que la 6*a* però només amb la branca que comença amb tamany de *kernel* de 9*x*9. En aquestes dues variants se li han afegit capes *BatchNormalization* entre les convolucionals. Podem observar la topologia de la variant 6a a la figura 4.3, i les de les variants 6b i 6c a l'apèndix A.

4.1.4. Model 7

El model 7 consta de tres variants que són prou semblants. Al model 7*a*, s'apliquen quinze capes convolucionals que extrauen filtres de 5*x*5 i 3*x*3 píxels. Després, es fa un *GlobalMaxPooling2D* i es connecta directament a l'eixida del classificador. Quant a les variants 7*b* i 7*c*, la diferència principal front a la 7*a* és que es fa una normalització utilitzant una capa *BatchNormalization* després de cada capa convolucional, i l'activació *ReLU* que es feia en les capes convolucionals es fa ara en una capa *Activation* després de la normalització. La diferència entre les variants 7*b* i 7*c* és que al model 7*b* la capa convolucional que s'aplica a la imatge d'entrada extrau 64 filtres de 7*x*7 i al 7*c* s'extrauen 64 filtres de 9*x*9. Després dels blocs convolucionals es fa un *GlobalMaxPooling* i es connecta tot a una capa *FullyConnected* que acaba connectant amb l'eixida. Podem observar la topologia del model 7*c* a la figura 4.4, i les dels models 7*a* i 7*b* a l'apèndix A.

4.1.5. Model 8

Aquest model consta de 4 variants, les quals s'han format cadascuna partir d'un altre model afegint noves capes o característiques. En primer lloc, el model 8*a* és una còpia del model 7*b* a la qual se li ha afegit normalització i una activació després de la capa *FullyConnected*. En el cas de la variant 8*b*, s'ha format afegint una capa *Dropout* amb freqüència de 0.4 després de la capa de normalització i l'activació afegides al 8*a*. Quant al model 8*c*, s'ha format afegint unes capes *SpatialDropout* amb freqüència 0.3 als blocs de reducció a partir del model 8*b*. Finalment, la variant 8*d* és semblant a l'anterior però amb menys capes de *SpatialDropout* i amb una freqüència menor a totes les capes amb *dropout*. A la figura 4.5 podem observar la topologia de la variant 8*b*. Les topologies dels altres models les podem trobar a l'apèndix A.

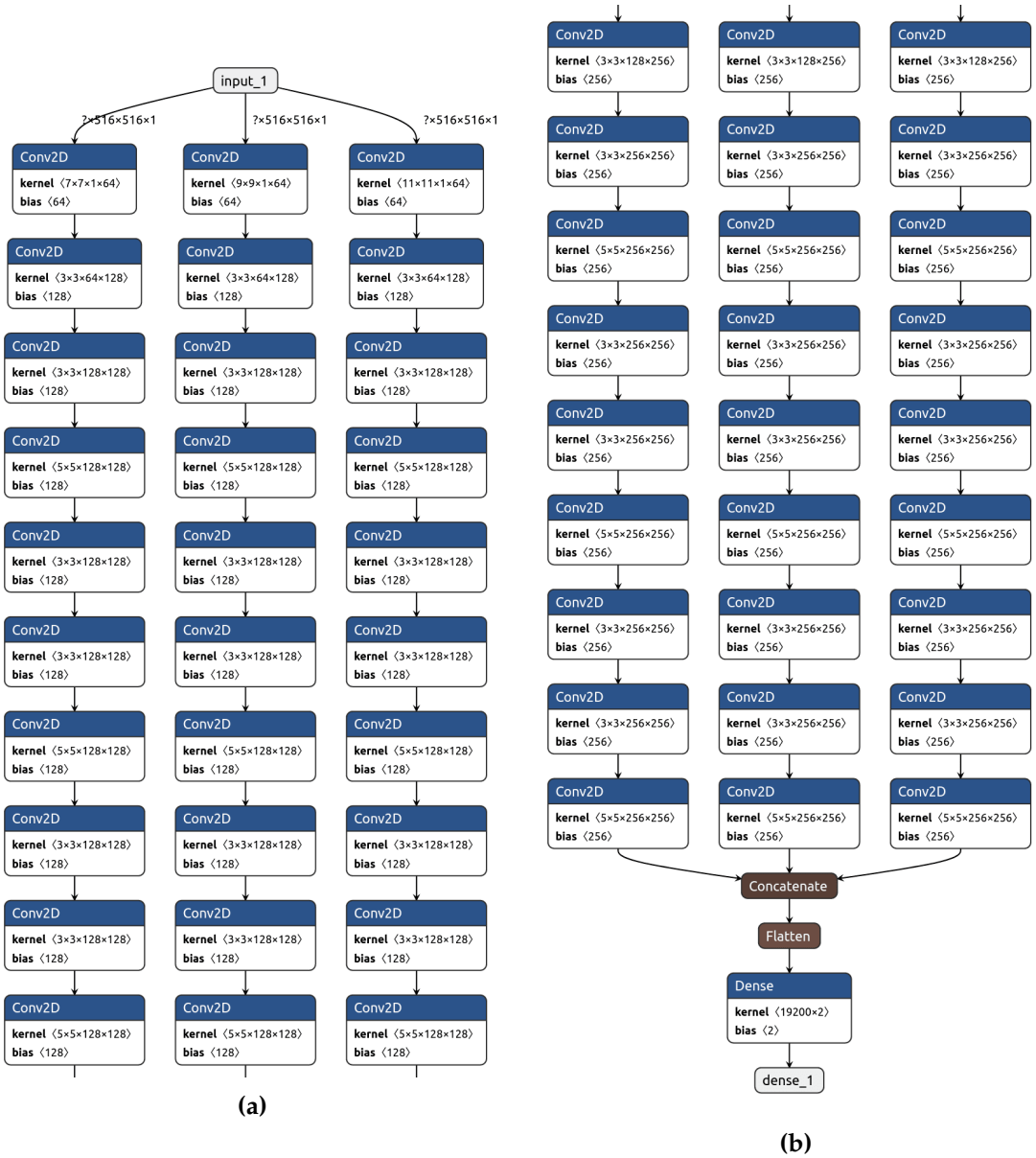


Figura 4.3: Topologia del model 6*a*.

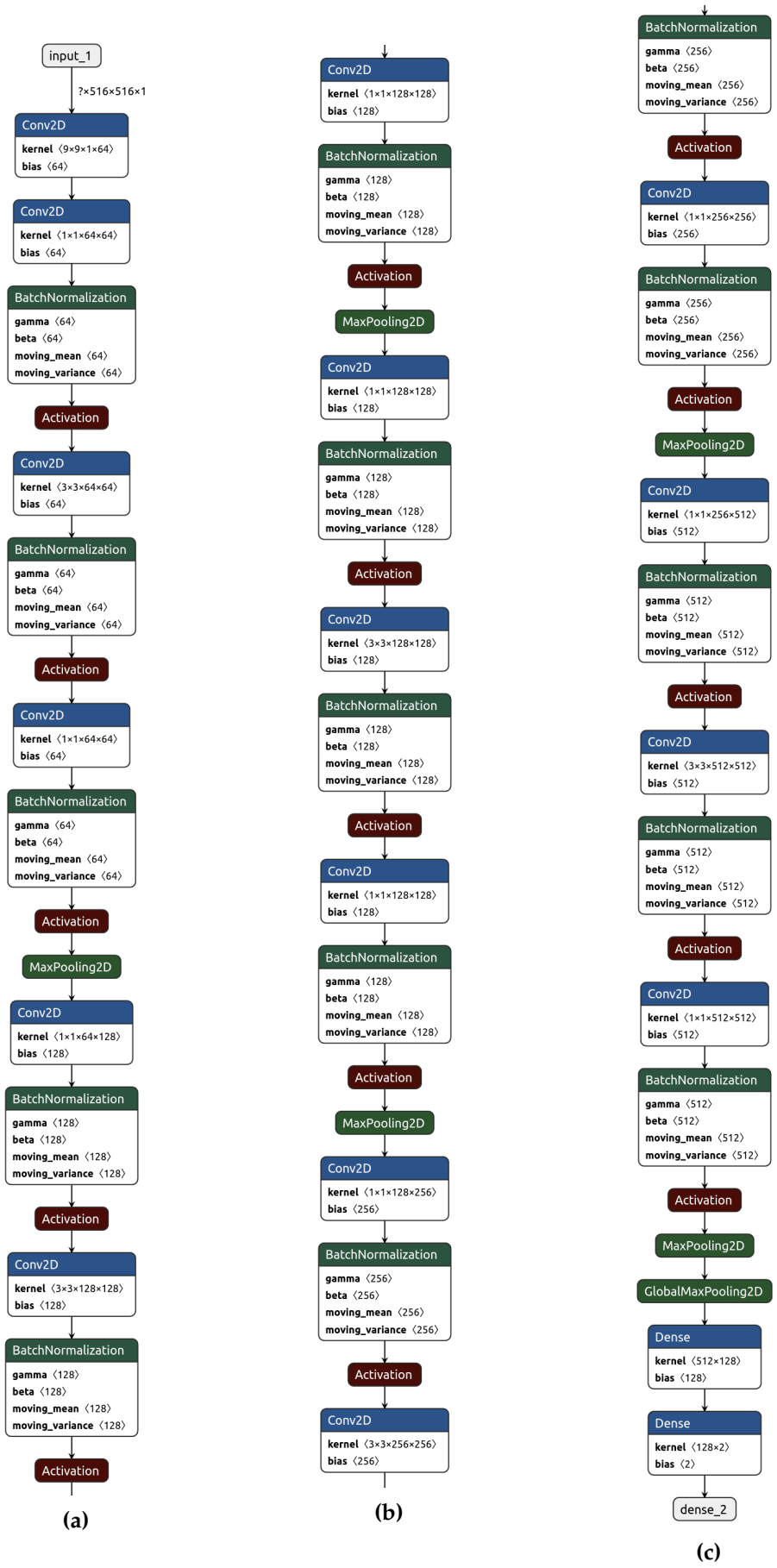


Figura 4.4: Topologia del model 7*c*.

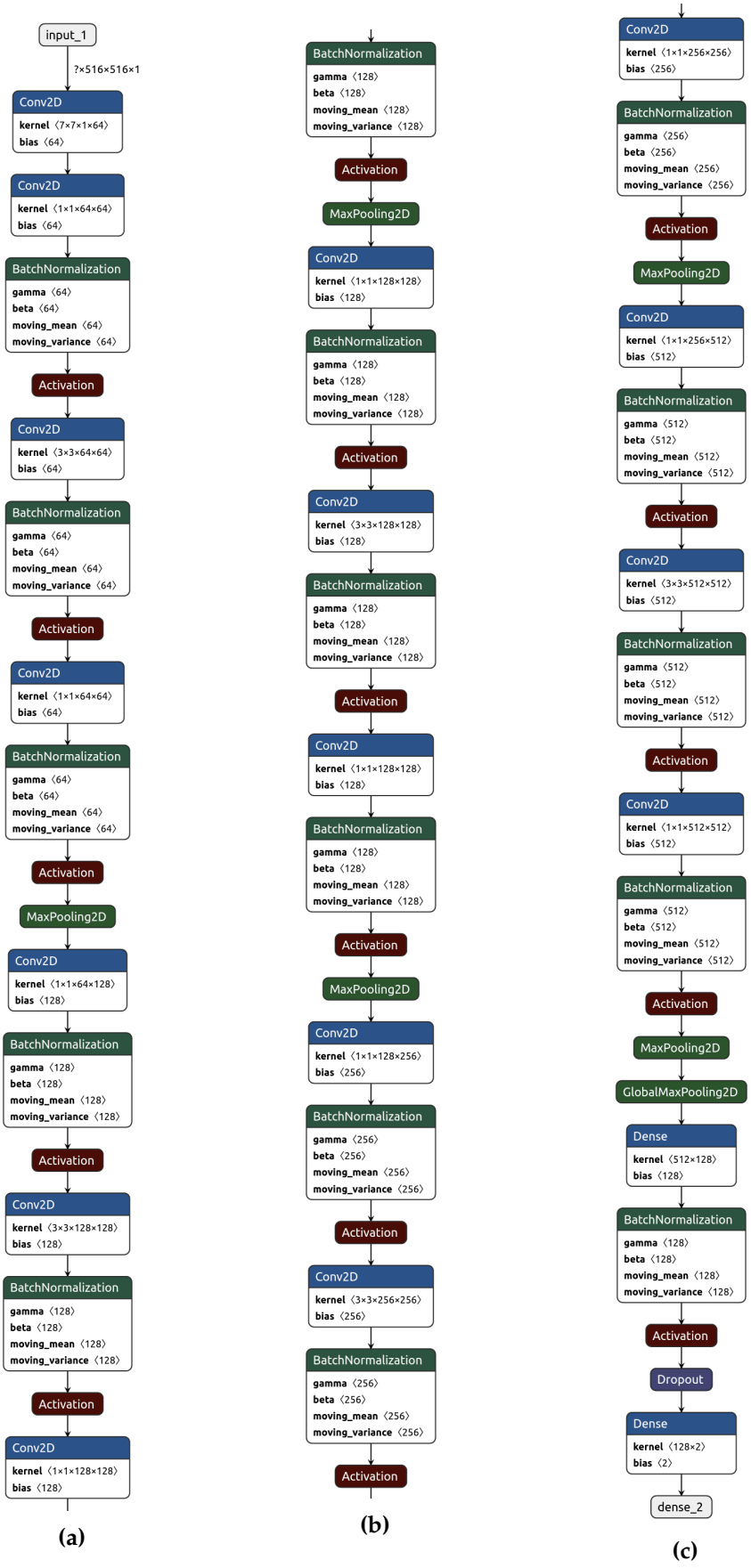


Figura 4.5: Topologia del model 8*b*.

4.2 Preparació de les dades

A l'hora de dur a terme els experiments necessitem entrenar els models i, per tant, necessitem tindre les dades preparades per a passar-les com a entrada a les xarxes. Com hem comentat, hem aplicat tècniques de *Data Augmentation* a les imatges del conjunt d'entrenament per a tenir més mostres que alimenten la xarxa i evitar l'*overfitting*. S'han utilitzat les següents transformacions aplicades aleatòriament a cadascuna de les imatges dels conjunts d'entrenament:

- Rotació aleatòria de α graus en el rang $[-20, 20]$.
- Desplaçament aleatori de y píxels verticals i x píxels horitzontals en el rang $[-15, 15]$.
- *Zoom in/out* aleatori segons un factor dins del rang $[0.9, 1.1]$.
- Variació d'intensitat aleatòria segons un factor dins del rang $[0.8, 1.2]$.

Com es pot observar a la taula 3.1, les particions d'entrenament, validació i test del *dataset* no estan balancejades. Per a la tasca de Control front a Pneumònia (*C vs. N*) hi ha aproximadament un 75% de mostres per a la classe C i un 25% aproximadament per a la classe N. Per a la tasca de Control front a Pneumònia i Pneumònia i Infiltració (*C vs. N+NI*) la classe 0 (C) representa aproximadament el 67% de les mostres i la classe 1 (N+NI) aproximadament el 33%. Per aquesta raó, i amb l'objectiu de millorar els entrenaments de tots els models en les dues tasques, s'ha decidit implementar una estratègia de balanceig de *batches* que poguera garantir que cada *batch* contenia el mateix nombre de mostres de cada classe. Açò s'ha implementat amb una classe *DataGenerator* que genera els *batches* a partir de les dades i els passa a la xarxa. Aquesta manera de passar les dades aprofita també la memòria, ja que no s'han de tenir en memòria principal totes les imatges carregades, sinó que es van carregant segons fan falta.

4.3 Mètriques per avaluar els resultats

Per avaluar els models utilitzarem diverses mètriques que calcularem sobre el conjunt de test. Aquestes són *accuracy*, *precision*, *recall* (o *sensitivity*) també coneguda com a la relació de vertaders positius (*TPR*), *f1-score* definida com la mitjana harmònica de *precision* i *recall*, i l'àrea per baix de la curva *ROC* (*auc*). Adicionalment, també s'ha utilitzat la proporció de falsos negatius (*FNR*). Per definir-les, utilitzarem els següents quatre conceptes a les equacions:

- TP (*true positive*): Mostres classificades a un grup que pertanyen a eixe grup.
- FP (*false positive*): Mostres classificades a un grup que no pertanyen a eixe grup.
- TN (*true negative*): Mostres no classificades a un grup que no pertanyen a eixe grup.

- FN (*false negative*): Mostres no classificades a un grup però sí que pertanyen a eixe grup.

En primer lloc, *accuracy* fa una mesura de la proporció de prediccions que el model ha fet correctament al classificar. No depèn de cap classe, és una mesura global del model.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

Quan es mesura *precision*, estem calculant quantes mostres classificades en un grup estan ben classificades. És la proporció de mostres que s'han classificat bé a una classe i es calcula de la següent manera:

$$precision = \frac{TP}{TP + FP} \tag{4.2}$$

Com que depen d'una classe, al nostre problema ho calcularem respecte a la classe 1, és a dir, la que conté pacients amb algun tipus de pneumònia, ja que el nostre objectiu és detectar de la millor manera els pacients amb pneumònia per a que puguen rebre l'ajuda mèdica que necessiten i prenguen les mesures adequades. És una mètrica important a tindre en compte perquè volem que els classificadors minimitzen els falsos positius ja que pot estalviar temps i recursos.

La mètrica *recall* (o *sensitivity*), també coneguda com la relació de vertaders positius (*TPR*), és la proporció de mostres d'una classe que s'han classificat finalment de manera correcta.

$$recall = sensitivity = TPR = \frac{TP}{TP + FN} \tag{4.3}$$

Al nostre problema, si mesurem *recall* sobre la classe 1, estaríem calculant la proporció de casos amb algun tipus de pneumònia que hem detectat sobre el total de les mostres de pneumònia que teníem. Aquesta mesura també és molt important perquè volem reduir al mínim possible els falsos negatius (presents al denominador), és a dir, els casos de pacients amb infecció que es classifiquen com a sans, ja que les conseqüències poden ser molt perilloses i no podem assumir aquest tipus d'errades als nostres classificadors.

El *f1-score* és la mitjana harmònica de *precision* i *recall*. Aquesta mètrica ens permet avaluar les dues en una mateixa expressió i es calcula de la manera següent:

$$Fscore = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.4}$$

També calcularem l'àrea sota la corba ROC, la qual es pot mostrar gràficament i ens diu cóm de bé distingeix les classes el model. Quan més alt és aquest valor, millor distingeix el model entre pacients amb infecció i sense infecció.

Finalment, utilitzarem la proporció de falsos negatius (FNR) com a mesura amb l'objectiu d'analitzar si es poden fixar uns llindars que puguen reduir les proporcions anteriors a unes que siguen acceptables. Concretament, ens referim a utilitzar aquesta mesura per poder classificar les mostres directament a una classe si el resultat que ens dona el model per a la classe objectiu no supera un valor llindar. Recordem que el model ens dona a l'eixida un valor per a cada

Taula 4.1: Resultats obtinguts pel model 4 amb la partició de test corresponents a la tasca *C vs. N* amb el *dataset* PADCHEST; *precision*, *recall* i *f1-score* estan donades respecte a la classe pneumònia (N).

Model	Accuracy	Precision	Recall	F1-Score	AUC
4*a*	83.51%	0.72	0.58	0.64	0.853
4*b*	81.56%	0.69	0.50	0.58	0.843
4*c*	82.54%	0.73	0.50	0.60	0.846
4*d*	82.86%	0.71	0.55	0.62	0.851

classe que podem prendre com a probabilitat encara que no ho és estrictament. Si calculem el FNR respecte a la classe 1, ens indicarà la proporció de mostres de pneumònia que no s'han classificat com a pneumònia. És un objectiu principal minimitzar aquesta proporció, ja que com hem comentat, volem evitar els falsos negatius a la classe pneumònia.

$$FNR = \frac{FN}{FN + TP} \tag{4.5}$$

4.4 Experiments i resultats Padchest

Recordem que per a aquest *dataset* s'havien definit dues tasques de classificació: Control (**C**) *vs.* Pneumònia (**N**) i Control (**C**) *vs.* Pneumònia i Pneumònia i Infiltració (**N+NI**). Llavors, l'eixida dels classificadors seran dues classes: la classe 0, formada per les imatges del grup **C** (Control) en ambdós casos, i la classe 1, formada pel grup **N** a la primera tasca i pels grups **N** i **NI** a la segona tasca.

Els experiments s'han fet utilitzant com a optimitzador l'algoritme *Stochastic Gradient Descent* (SGD) amb un *learning rate* (o factor d'aprenentatge) inicial de 10^{-3} i un *scheduler* per reduir-lo per un factor cada 100 *epochs*. Cada procés d'entrenament s'ha fet amb un nombre màxim d'*epochs* de 500 i un factor de reducció del *learning rate* de 0.7, és a dir, cada 100 *epochs* el *learning rate* disminueix en un 30%.

En primer lloc, s'ha entrenat el model 4 i les seues variants en les dues tasques de classificació. Els resultats per a la tasca *C vs. N* es poden observar a la taula 4.1, i els de la tasca *C vs. N+NI* a la taula 4.2. L'*accuracy* màxima per a *C vs. N* ha sigut de 83.51% i s'ha obtingut a la variant 4*a*. Per a *C vs. N+NI* s'ha obtingut una *accuracy* màxima de 83.57% al model 4*d*. Aquests resultats es podrien millorar perquè la *precision* i sobre tot el *recall* no ens donen molt bons resultats per a la classe objectiu. Estem classificant malament aproximadament un 40% de les mostres de pneumònia en ambdues tasques, i el nostre objectiu és minimitzar l'error a aquesta classe. A la figura 4.6 podem observar l'àrea sota la corba ROC del model 4*a*.

Quant al model 5, podem trobar els resultats per a la tasca *C vs. N* a la taula 4.3 i per a la tasca *C vs. N+NI* a la taula 4.4. Per a la tasca *C vs. N*, la millor *accuracy* ha sigut de 84.39% i s'ha aconseguit amb el model 5*a*, amb una *precision* de 0.66

Taula 4.2: Resultats obtinguts pel model 4 amb la partició de test corresponents a la tasca *C vs. N+NI* amb el *dataset* PADCHEST; *precision*, *recall* i *f1-score* estan donades respecte a la classe (N+NI).

Model	Accuracy	Precision	Recall	F1-Score	AUC
4*a*	81.22%	0.76	0.62	0.68	0.856
4*b*	80.93%	0.77	0.59	0.67	0.859
4*c*	81.32%	0.78	0.60	0.68	0.859
4*d*	83.57%	0.73	0.66	0.69	0.865

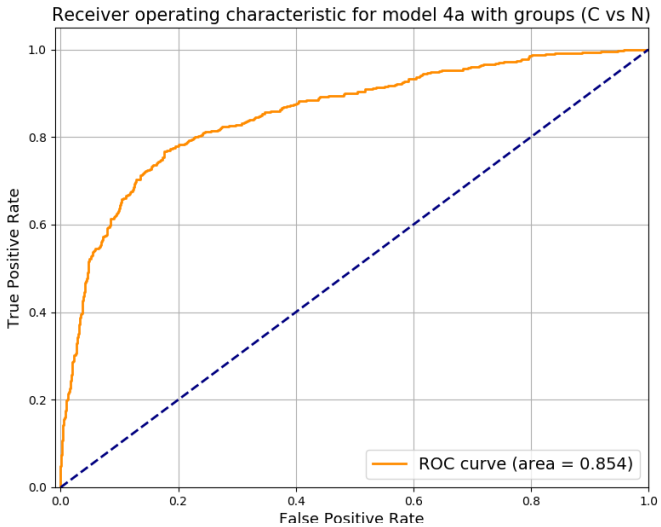


Figura 4.6: Corba ROC per al classificador binari basat en el model 4*a* per a la tasca *C vs. N*. L'àrea sota la corba és de 0.8540

Taula 4.3: Resultats obtinguts pel model 5 amb la partició de test corresponents a la tasca *C vs. N* amb el *dataset* PADCHEST; *precision*, *recall* i *f1-score* estan donades respecte a la classe pneumònia (N).

Model	Accuracy	Precision	Recall	F1-Score	AUC
5*a*	84.39%	0.66	0.58	0.62	0.858
5*b*	82.75%	0.72	0.54	0.62	0.853
5*c*	82.80%	0.73	0.52	0.61	0.851

Taula 4.4: Resultats obtinguts pel model 5 amb la partició de test corresponents a la tasca *C vs. N+NI* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe (N+NI).

Model	Accuracy	Precision	Recall	F1-Score	AUC
5*a*	80.25%	0.76	0.58	0.66	0.854
5*b*	83.78%	0.72	0.69	0.70	0.881
5*c*	82.91%	0.71	0.66	0.68	0.879

i *recall* de 0.58. Aquesta *accuracy* és un poc superior a la de les variants 5*b* i 5*c*, pel que podem veure que al treballar amb dues branques hem aconseguit millors resultats per a *accuracy*, encara que hem perdut un poc de *precision* com s'observa a la taula 4.3. La corba ROC mostrada a la figura 4.7 és prou similar a la corba del model 4*a*. Amb aquest model no hem millorat els resultats de l'anterior. Un *recall* de 0.50 aproximadament ens obliga a seguir intentant millorar els resultats, ja que necessitem classificar millor la classe amb pneumònia.

Per a la tasca *C vs. N+NI*, el millor resultat del model 5 ha sigut amb la variant 5*b*, amb una *accuracy* de 83.78%, una *precision* de 0.72 i un *recall* de 0.69. El *recall* ha sigut el millor fins al moment, però seguim classificant malament un 30% aproximadament de les mostres amb pneumònia.

Els resultats de les distintes variants del model 6 els podem trobar a les taules 4.5 i 4.6 per a les tasques *C vs. N* i *C vs. N+NI* respectivament. En aquest model, la millor variant ha sigut la 6*b*, amb una *accuracy* de 85.56% per a la tasca *C vs. N* i de 85.35 per a la tasca *C vs. N+NI*. Per a les dues tasques s'han millorat els resultats obtinguts fins al moment. Al model 6*b* hem obtingut la major *accuracy* en ambdues tasques. Quant a les altres mètriques, no han millorat notablement però si que hem aconseguit un *recall* de 0.72 amb el model 6c a la tasca *C vs. N+NI*, el millor valor fins al moment. També es pot evidenciar la millora observant els resultats de l'àrea sota la corba *ROC*, la qual ha augmentat lleugerament fins a valors com 0.902 al model 6*b* i que podem observar a la figura 4.8.

Quant al model 7, podem trobar els resultats a les taules 4.7 i 4.8 per a les tasques *C vs. N* i *C vs. N+NI* respectivament. En aquest cas, s'han millorat els resultats respecte als de l'anterior model. Per a la tasca *C vs. N* s'ha obtingut una *accuracy* de 86.32% amb el model 7*b*, el qual ha obtingut una *precision*, *recall* i àrea sota la corba ROC semblants a les del model 6 amb valors de 0.75, 0.70 i 0.901 respectivament.

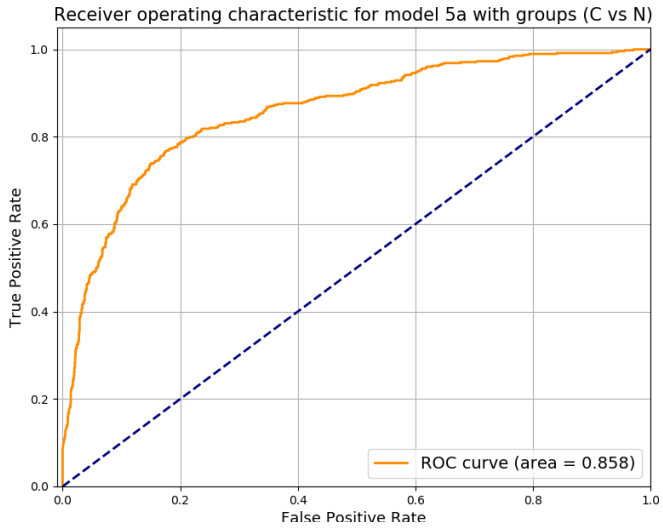


Figura 4.7: Corba ROC per al classificador binari basat en el model 5*a* per a la tasca *C vs. N*. L'àrea sota la corba és de 0.858

Taula 4.5: Resultats obtinguts per al model 6 amb la partició de test corresponents a la tasca *C vs. N* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe pneumònia (N)

Model	Accuracy	Precision	Recall	F1-Score	AUC
6*a*	84.37%	0.69	0.69	0.69	0.885
6*b*	85.56%	0.75	0.66	0.70	0.876
6*c*	85.18%	0.74	0.64	0.69	0.892

Taula 4.6: Resultats obtinguts per al model 6 amb la partició de test corresponents a la tasca *C vs. N+NI* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe (N+NI)

Model	Accuracy	Precision	Recall	F1-Score	AUC
6*a*	77.94%	0.71	0.55	0.62	0.809
6*b*	85.35%	0.76	0.70	0.73	0.902
6*c*	84.40%	0.72	0.72	0.72	0.896

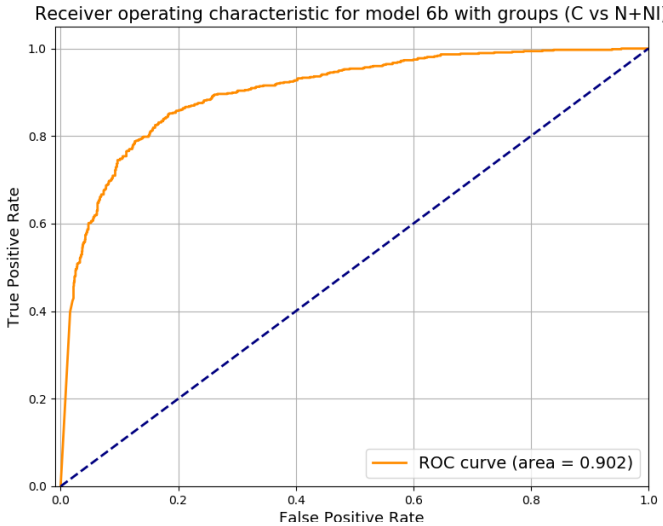


Figura 4.8: Corba ROC per al classificador binari basat en el model *6b* per a la tasca *C vs. N+NI*. L'àrea sota la corba és de 0.902

Taula 4.7: Resultats obtinguts per al model 7 amb la partició de test corresponents a la tasca *C vs. N* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe pneumònia (N)

Model	Accuracy	Precision	Recall	F1-Score	AUC
7a	84.25%	0.68	0.52	0.59	0.843
7b	86.32%	0.75	0.70	0.72	0.901
7c	84.43%	0.69	0.72	0.70	0.890

A la tasca *C vs. N+NI*, el model 7 ha aconseguit bons resultats en comparació als ja obtinguts. Les variants *7b* i *7c* han obtingut una *accuracy* de 84.26% i 84.50% respectivament. La *precision* i *recall* obtingudes per a aquests models han millorat inclús les de l'altra tasca, amb un valor de *f1-score* de 0.76 per a la variant *7c* i una *precision* de 0.79 a la variant *7b*, superant a les de tots els models provats fins al moment. Aquest model també ha obtingut el millor valor per a l'àrea sota la corba ROC, obtinguda amb la variant *7c* i amb un valor de 0.908, observable a la figura 4.9.

Com hem comentat a la secció 4.1, el model 8 està format a partir del model *7b*, amb la intenció de perfeccionar el model i millorar els resultats, que han sigut

Taula 4.8: Resultats obtinguts per al model 7 amb la partició de test corresponents a la tasca *C vs. N+NI* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe (N+NI)

Model	Accuracy	Precision	Recall	F1-Score	AUC
7a	82.20%	0.71	0.61	0.66	0.855
7b	84.26%	0.79	0.71	0.75	0.900
7c	84.50%	0.76	0.76	0.76	0.908

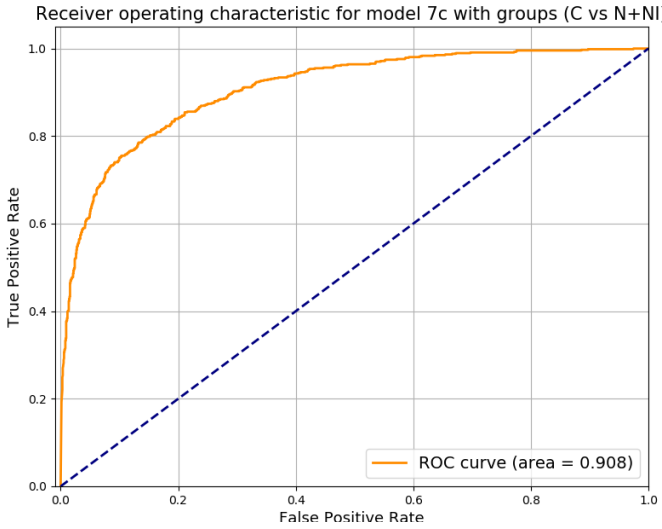


Figura 4.9: Corba ROC per al classificador binari basat en el model 7*c* per a la tasca *C vs. N+NI*. L'àrea sota la corba és de 0.908

Taula 4.9: Resultats obtinguts amb el model 8 amb la partició de test corresponents a la tasca *C vs. N* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe pneumònia (N)

Model	Accuracy	Precision	Recall	F1-Score	AUC
8*a*	87.46%	0.74	0.66	0.69	0.903
8*b*	87.63%	0.74	0.66	0.70	0.906
8*d*	83.92%	0.70	0.45	0.55	0.866

els millors fins al moment. Podem trobar els resultats a la taula 4.9 per a la tasca *C vs. N*, i a la taula 4.10 per a la tasca *C vs. N+NI*. Els resultats de la variant 8*c* s'han omés perquè no eren suficientment bons durant l'entrenament, probablement degut a que hi havia massa capes amb *dropout* i amb un valor de freqüència massa elevat, pel que s'estava limitant l'aprenentatge de la xarxa, raó per la qual el model 8*d* s'ha construït amb menys *dropout*.

A la tasca *C vs. N*, la variant 8*b* ha aconseguit els millors resultats. L'*accuracy* obtinguda és del 87.63%, la millor fins al moment. En *precision* i *recall* no s'han millorat els resultats del model 7, són lleugerament inferiors. El model ha obtingut un valor d'àrea sota la corba ROC de 0.906, el major valor obtingut per a aquesta tasca. Podem observar la corba a la figura 4.10.

Per a la tasca *C vs. N+NI*, la millor variant del model 8 ha sigut la 8*a*, amb una *accuracy* del 85.69%, *precision* de 0.77 i *recall* de 0.69. L'*accuracy* ha millorat la del model 7 i és la millor obtinguda per a aquesta tasca. En canvi, els valors per a *precision* i *recall* no han superat als obtinguts a les variants del model 7 encara que tenen valors pròxims, cosa que indica que el model classifica lleugerament pitjor la classe pneumònia.

Després d'observar els resultats obtinguts per a les mètriques als models entrenats, farem un anàlisi d'una última mètrica. Com hem comentat, calcularem la proporció de falsos negatius per tal de saber quins models minimitzen els falsos

Taula 4.10: Resultats obtinguts amb el model 8 amb la partició de test corresponents a la tasca *C vs. N+NI* amb el *dataset* Padchest; *precision*, *recall* i *f1-score* estan donats respecte a la classe (N+NI).

Model	Accuracy	Precision	Recall	F1-Score	AUC
8*a*	85.69%	0.77	0.69	0.73	0.904
8*b*	85.06%	0.73	0.74	0.73	0.904
8*d*	84.19%	0.78	0.60	0.68	0.885

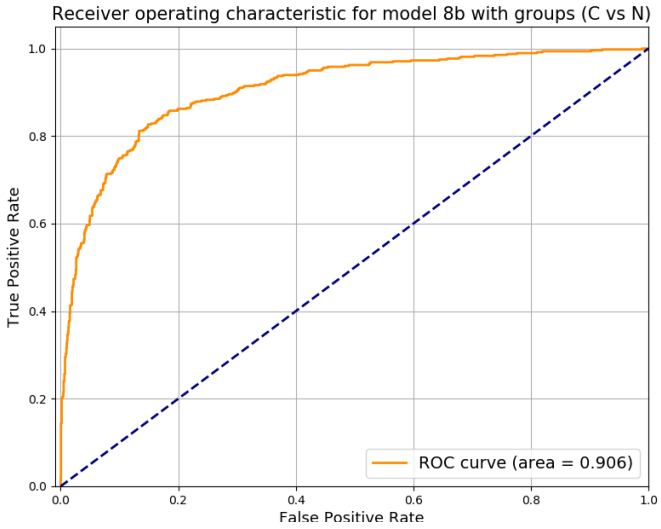


Figura 4.10: Corba ROC per al classificador binari basat en el model 8*b* per a la tasca *C vs. N*. L'àrea sota la corba és de 0.9060

negatius, és a dir, els que classifiquen menys mostres com a pacients sans quan aquests tenen pneumònia. Això és molt important perquè classificar a un pacient com a sa quan realment està malalt és molt perillós pels problemes que podrien ocórrer si no s'actua a temps. És un objectiu de vital importància reduir aquesta proporció als models si volem aconseguir l'objectiu no tècnic del projecte.

A més de calcular la proporció de falsos negatius (FNR), també analitzarem uns histogrames on es mostra el valor d'eixida del model assignat a la classe pneumònia i el nombre de mostres de cada classe a les que se li ha assignat eixe valor. Això ens pot ajudar a veure gràficament com està funcionant la xarxa, si el model classifica més o menys bé les mostres, i també ens pot ajudar a detectar errors i anomalies.

A la taula 4.11 podem trobar els valors del FNR per als millors models a la tasca *C vs. N*. Com podem observar, el valor per a la proporció està al voltant de 0.3 i 0.4 a tots els models. Per a aquesta tasca, el model que millor FNR ha aconseguit és el 7*c*. Aquest model té una *accuracy* de 84.43, un poc més baixa que el 8*b*, el qual té la màxima *accuracy* obtinguda (87.63%). La variant 7*c* ha aconseguit un FNR de 0.28, millor en comparació al valor de 0.35 obtingut pel model 8*b*. Quant a histogrames, el que millor sembla per a aquesta tasca és el del model 7*c* també. Com es pot observar a la figura 4.11, els valors d'eixida per a la classe 1 estan prou polaritzats, és a dir, per a la majoria de mostres són propers a zero o a un. Això és positiu perquè significa que el model discrimina sense massa dubte les mostres. En canvi, per a valors de pneumònia pròxims al zero, hi ha unes 100 mostres de pacients de la classe pneumònia, el que vol dir que a pacients amb pneumònia està classificant-los com a sans amb certa 'seguretat', cosa que no és gens bona per al model. En comparació a la resta d'histogrames generats amb els altres models, aquest ha sigut el millor, ja que la resta tenien més de 100 mostres de pneumònia amb un *score* proper a zero. Un histograma semblant és el del model 7*b*, el qual ha aconseguit una *accuracy* de 86.32%, major que el 7*c* encara que un FNR de 0.30 lleugerament pitjor. El podem trobar a la figura 4.12, on s'observa que és molt semblant a l'histograma de la variant 7*b*. En aquest cas, el FNR és de 0.34 i l'*accuracy* que aconseguia era de 85.56%.

Per a la tasca *C vs. N+NI*, els valors de FNR estan a la taula 4.12. En aquest cas, el model 7*c* ha obtingut un valor de 0.24, el millor per a aquesta tasca i el més baix que s'ha aconseguit. Podem observar l'histograma d'aquesta variant a la figura 4.13. En aquest cas, observem que la majoria de valors d'eixida que proporciona el model són propers a zero o a un. Es tracta d'un histograma semblant al 7*b* comentat per a la tasca *C vs. N*. Aquest model també li dona un valor prop al zero a unes 100 mostres etiquetades com a pacients amb pneumònia o infiltració. Un histograma semblant és el que genera el model 8*b*, que podem trobar a la taula 4.14. Aquest model té un FNR de 0.26. La disminució del valor de FNR a aquesta tasca pot ser deguda a la incorporació del grup de mostres d'infiltració, cosa que fa més gran el nombre de mostres de classe 1 i balanceja un poc el conjunt de mostres. Recordem que per a la tasca *C vs. N* hi havien un 70% aproximadament de mostres de Control (**C**).

Taula 4.11: Resultats de FNR obtinguts amb la partició de test corresponents a la tasca *C vs. N* amb el *dataset* PADCHEST.

Model	*5b*	*6b*	*7b*	*7c*	*8b*
FNR	0.36	0.34	0.30	0.28	0.35

Taula 4.12: Resultats de FNR obtinguts amb la partició de test corresponents a la tasca *C vs. N+NI* amb el *dataset* PADCHEST.

Model	*6b*	*7c*	*8a*	*8b*
FNR	0.29	0.24	0.31	0.26

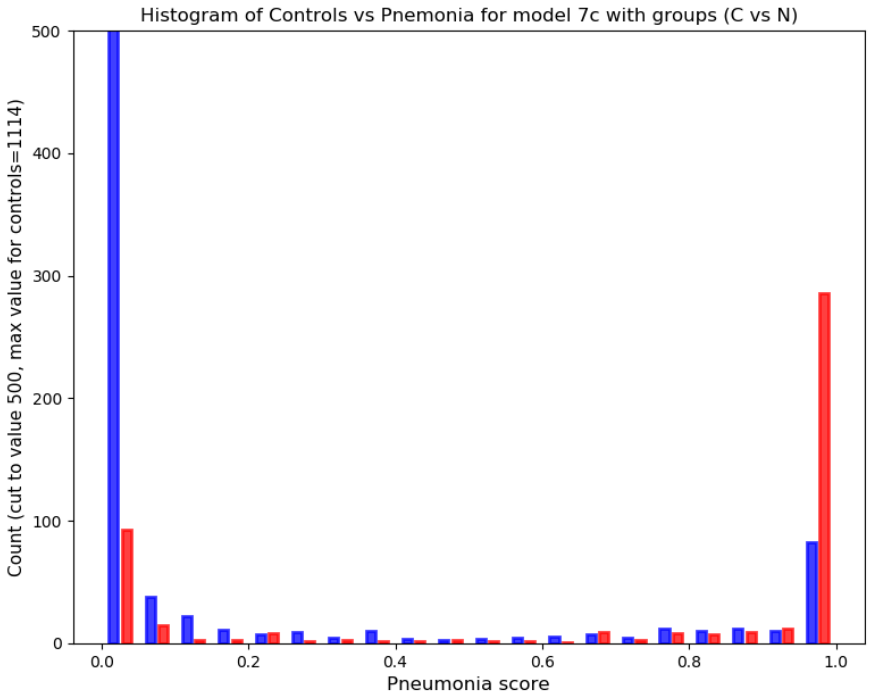


Figura 4.11: Histograma per al classificador binari basat en el model *7c* per a la tasca *C vs. N*. En blau, les mostres de Control (**C**) i en roig, les mostres de Pneumònia(**N**).

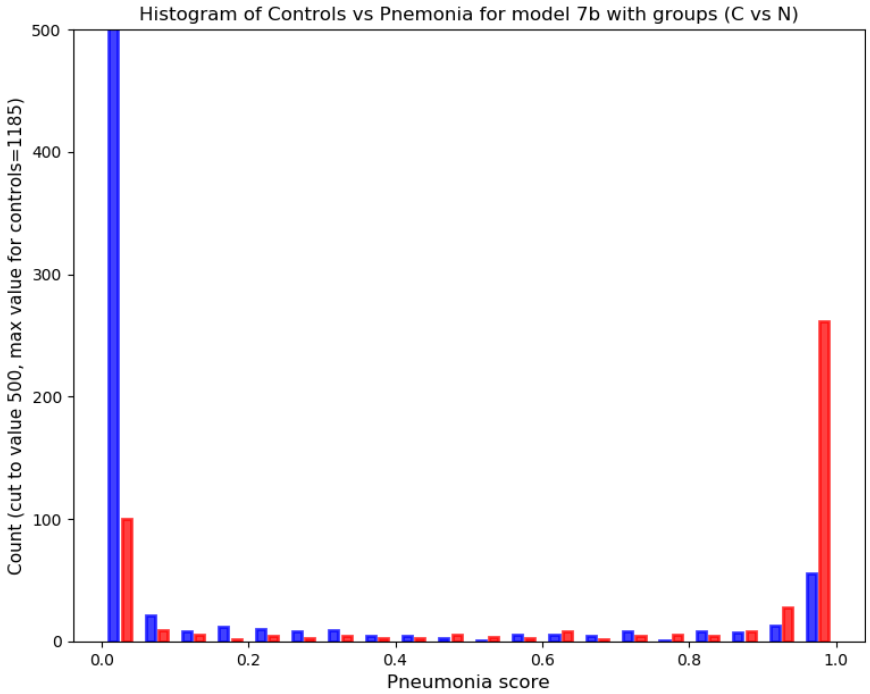


Figura 4.12: Histograma per al classificador binari basat en el model *7b* per a la tasca *C vs. N*. En blau, les mostres de Control (**C**) i en roig, les mostres de Pneumònia(**N**).

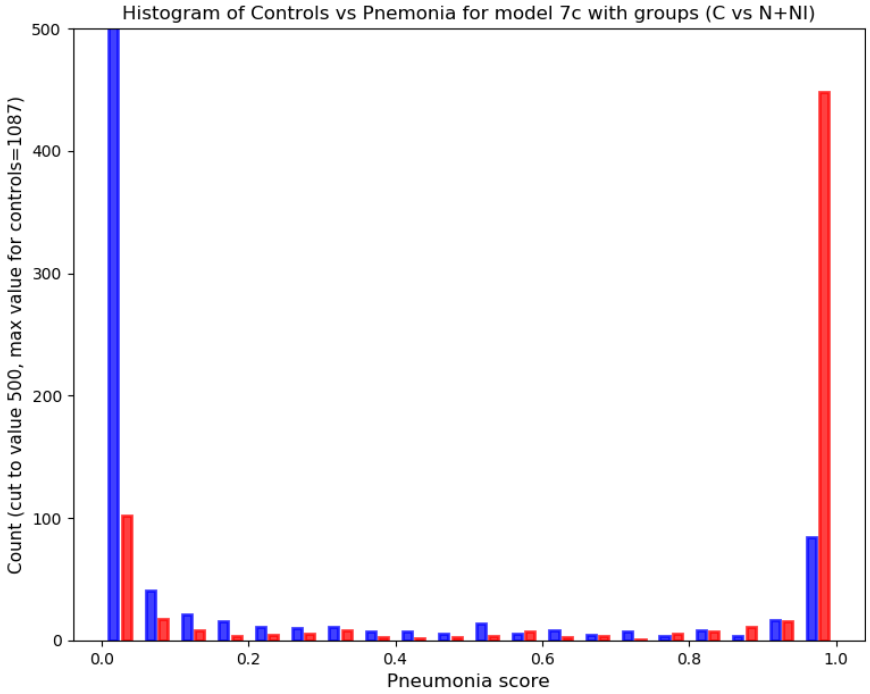


Figura 4.13: Histograma per al classificador binari basat en el model *7c* per a la tasca *C vs. N+NI*. En blau, les mostres de Control (**C**) i en roig, les mostres de Pneumònia i Pneumònia i Infiltració (**N+NI**).

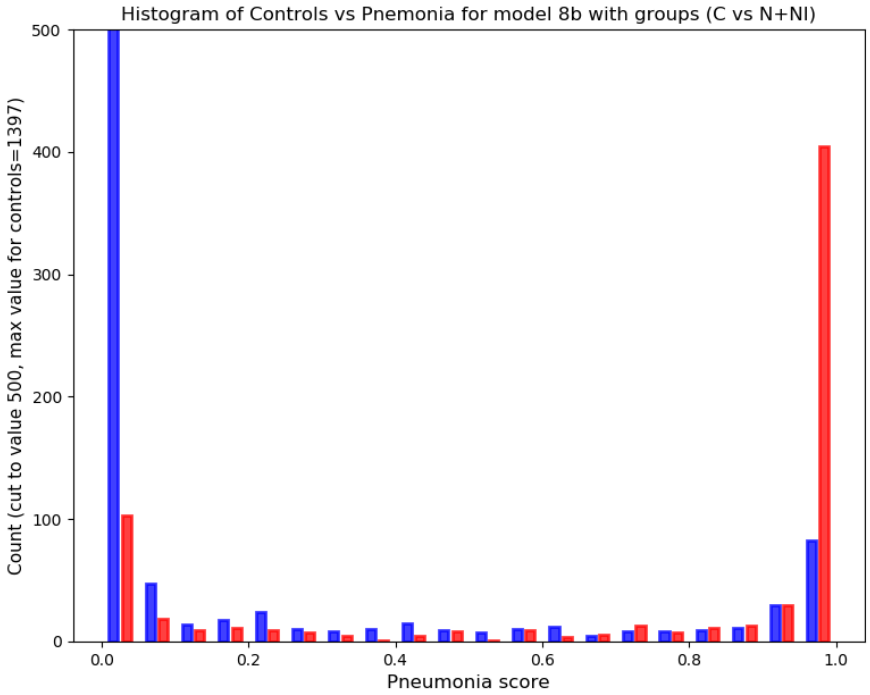


Figura 4.14: Histograma per al classificador binari basat en el model 8*b* per a la tasca *C vs. N+NI*. En blau, les mostres de Control (**C**) i en roig, les mostres de Pneumònia i Pneumònia i Infiltració (**N+NI**).

4.5 Observacions i valoracions dels experiments

Durant els experiments, s'ha utilitzat l'optimitzador Adam en alguns casos però no s'han obtingut diferències rellevants. També s'ha variat el valor inicial del *learning rate* a 10^{-4} i el factor de reducció a 0.5 en altres casos i tampoc no s'han trobat millors resultats que foren rellevants. A la majoria de processos d'entrenament ha hagut *overfitting*, malgrat l'ús del *Data Augmentation* i d'haver fixat el valor inicial de *learning rate* a 10^{-4}. Per a la majoria de models l'*accuracy* per a la partició d'entrenament ha arribat al 99% abans de l'*epoch* 200, mentre que l'*accuracy* per a la partició de validació es quedava estancada en el rang $[80, 88]$ depenent del model i la tasca.

A l'hora d'avaluar els resultats, haguérem pogut utilitzar validació encreuada, ja que garanteix que els resultats siguen independents de la partició entre dades d'entrenament i de prova, però no s'ha utilitzat per qüestió de temps i pel nombre de topologies a comprovar, ja que els entrenaments han tardat una mitjana d'entre dos i tres dies d'execució. No obstant, hem utilitzat les particions que venien definides pels creadors del *dataset*.

Comparant l'*accuracy* per a ambdues tasques, es pot observar com per a la major part dels models s'obtenen millors resultats per a la tasca *C vs. N* que per a la tasca *C vs. N+NI*. No obstant això, un s'esperaria trobar millors resultats a la tasca *C vs. N+NI*, perquè en aquesta tasca hi ha més mostres per a la classe 1 i les tres particions estan menys desequilibrades en comparació a l'altra tasca, cosa que fa pensar que el model funciona més malament quan més mostres amb pneumònia té el conjunt, pel que no estaria aprenent a distingir bé la infecció. Podria ser que

les mostres amb infiltració pulmonar fan més dificultosa la classificació. En canvi, si analitzem les mètriques que depenen d'una classe, calculades respecte a la classe 1, per a valors de *precision* i *recall* els models han obtingut valors lleugerament superiors en general a la tasca *C vs. N+NI*. De fet, si ens centrem en l'àrea sota la corba ROC, encara que les diferències entre models no són rellevants, la majoria de models obtenen major valor d'àrea per a la tasca *C vs. N+NI*. A més, a la mètrica FNR els models han obtingut millors resultats també a la tasca que inclou els infiltrats pulmonars. Això demostra que incloure els infiltrats ajuda a la detecció de la infecció. El fet de que en aquesta tasca s'obtinguen valors d'*accuracy* menors és perquè els models no encerten tant a la classe Control, pel que en terme mig s'obté menor *accuracy*, encara que com podem veure calculant les mètriques que depenen de la classe 1, el model distingeix millor la infecció. Si comparem els histogrames d'ambdues tasques també observarem que als de la tasca *C vs. N+NI* està fallant en més mostres de Control, ja que les barres blaves són en general un poc més elevades a la zona central que als histogrames de l'altra tasca.

Els millors resultats a la tasca de classificació entre Control i Pneumònia han sigut una *accuracy* del 87.63%, una *precision* de 0.74 i un *recall* de 0.66 amb el model 8*b*. Per a la tasca de classificació entre Control front a Pneumònia i Pneumònia i Infiltració, hem aconseguit una *accuracy* del 85.69%, una *precision* de 0.77 i un *recall* de 0.69 amb el model 8*a*. Aquests resultats són acceptables des del punt de vista de l'acompliment. Per contra, en comparació als resultats obtinguts amb els models de l'estat de l'art, no són suficientment bons. Quant a *accuracy*, ens aproximem a les obtingudes en alguns articles, però estem lluny dels millors resultats, com l'*accuracy* del 98% aconseguida en [11]. Quant a *precision* i *recall*, també ens quedem per darrere als models de l'estat de l'art. El fet de tindre valors com 0.74 i 0.66 de *precision* i *recall* respectivament per a la classe pneumònia a un dels millors models ens limita prou.

A l'arquitectura dels models, la tècnica d'aplicar diverses branques en paral·lel a l'entrada ens ha permés obtindre uns resultats un tant millors en alguns models. Per exemple, al model 4 la variant 4*a* és la que major *accuracy* ha obtingut per a la tasca *C vs. N* entre les variants del model i aquest tenia aplicades a l'entrada 3 capes convolucionals en paral·lel que extreien filtres de distint tamany, mentre les altres variants tenien només dues capes en paral·lel. Aquesta millora també s'evidencia al model 5, on la variant 5*a*, que tenia dues branques en paral·lel que no es connectaven fins a la fi de la part convolucional, ha obtingut el millor valor en comparació a les altres dues variants, que no aplicaven cap paral·lelisme. Al model 6 la variant 6*a* tenia paral·lelisme i no ha obtingut millors resultats que les variants germanes, les quals no tenien branques paral·leles. Això pot ser degut a que les altres variants tenien també capes de normalització de *batches*, cosa que la variant 6*a* no tenia. En general, aplicar la tècnica de paral·lelitzar la xarxa ha tingut bons resultats, sobre tot a la tasca *C vs. N*.

Quant a la fixació de llindars per a eliminar o minimitzar els falsos negatius, l'objectiu quan s'utilitzen classificadors binaris es definir el que es coneix com àrea de rebuig en teoria de la decisió. Es fixen dos llindars, un per baix (th_{low}) i un per dalt (th_{high}) de manera que no hi ha falsos negatius per al valor de pneumònia $< th_{low}$ ni falsos positius per a valors de pneumònia $> th_{high}$, referint-nos a valor de pneumònia al valor d'eixida de la xarxa per a la classe 1. Degut a la distribució de les mostres de les classes Control i Pneumònia que hem mostrat als

Taula 4.13: Resum dels resultats dels millors models per a cada tasca.

Tasca	Model	Accuracy	Precision	Recall	F1-Score	AUC	FNR
C vs. N	*8b*	87.63%	0.74	0.66	0.70	0.906	0.35
C vs. N	*7b*	86.32%	0.75	0.70	0.72	0.901	0.30
C vs. N+NI	*6b*	85.35%	0.76	0.70	0.73	0.902	0.29
C vs. N+NI	*7c*	84.50%	0.76	0.76	0.76	0.908	0.24
C vs. N+NI	*8a*	85.69%	0.77	0.69	0.73	0.904	0.31

histogrames, no es possible definir una àrea de rebuig delimitada per dos llindars. Com hem observat als histogrames, als classificadors amb millors mètriques els valors d'eixida per a la classe pneumònia estan prou polaritzats, i els models classifiquen amb molta seguretat les mostres, equivocant-se amb una gran quantitat de mostres de pneumònia, les quals classifiquen com a pacients sans amb molta certesa. Aquest és un greu problema que fa que no puguem confiar en aquests classificadors, ja que el seu ús en un entorn real seria molt perillós i podria causar problemes molt greus.

Finalment, a la taula 4.13 tenim un resum dels que considerem els millors models després de realitzar tots els experiments.

CAPÍTOL 5
Conclusions

Amb la realització d'aquest treball hem arribat a diverses conclusions, les quals anem a presentar en aquest capítol. També parlarem sobre futurs treballs i dels objectius que es van proposar per al projecte.

En primer lloc, l'objectiu principal era el disseny i l'avaluació de classificadors binaris basats en *Convolutional Neural Networks* per a distingir entre pacients sans i amb pneumònia a partir d'una radiografia pulmonar. Durant el desenvolupament del treball, s'han construït diversos models de classificadors intentant replicar tècniques de l'estat de l'art dels quals hem mostrat els resultats dels més rellevants a la memòria. Altres classificadors amb resultats insuficientment bons han estat exclosos de la memòria.

Els experiments realitzats amb el *dataset* per a les dues tasques revelen que alguns dels models dissenyats obtenen resultats que podrien considerar-se acceptables des del punt de vista de l'acompliment esperat per a classificadors binaris aplicats a tasques dificultoses com la de detectar els lleugers patrons que porten al radiòleg a etiquetar una radiologia pulmonar com a infecció de pneumònia. No obstant, els resultats observats a l'analitzar la mètrica FNR i els histogrames demostren que els models no han aprés a detectar suficientment bé la pneumònia, de manera que el nostre objectiu no tècnic, que era que els doctors pogueren adaptar aquests classificadors per a la criva de pacients, no s'ha complit. De fet, l'ús dels classificadors en escenaris reals està lluny de ser considerat per comitès mèdics.

Un dels grans problemes que s'han trobat al projecte és el *dataset*. El fet de que no estiga balancejat dificulta molt el rendiment dels classificadors. En una tasca com aquesta, es necessiten moltes mostres per poder trobar els patrons que distingeixen una radiografia pulmonar de tenir una infecció de pneumònia o no tindre-la. Encara que s'ha balancejat a nivell de *batch* i s'ha realitzat *Data Augmentation*, és evident que els models no han aprés a detectar perfectament la infecció, cosa que també s'evidenciava als articles publicats amb tasques semblants.

Com a propostes de futur, estaria la d'aplicar segmentació a les imatges del *dataset* per a eliminar la informació sorollosa al voltant dels pulmons, així com tenir en compte més característiques de les radiografies a l'hora de balancejar les particions, com l'edat, el sexe i l'equipament utilitzat, entre altres. A aquesta

conclusió ha arribat l'Institut Tecnològic d'Informàtica, publicada en un informe al repositori de GitHub del projecte[1].

Finalment, es proposa provar els classificadors amb altres conjunts de dades semblants, sobre tot amb el *dataset* de COVID-19 que va alliberar el BIMCV[2] a principis de Juny, amb el qual no hem fet proves en aquest projecte per qüestió de temps.

[1] https://github.com/BIMCV-CSUSP/BIMCV-COVID-19/blob/master/padchest-covid/balanced-one-partition/report.pdf

[2] https://bimcv.cipf.es/bimcv-projects/bimcv-covid19/

Bibliografia

[1] Alexander E. Gorbalenya, Susan C. Baker, Ralph S. Baric, Raoul J. de Groot,Christian Drosten, Anastasia A. Gulyaeva, Bart L. Haagmans, Chris Lauber,Andrey M. Leontovich, Benjamin W. Neuman, Dmitry Penzar, Stanley Perl-man, Leo L.M. Poon, Dmitry V. Samborskiy, Igor A. Sidorov, Isabel Sola, andJohn Ziebuhr. "The species Severe acute respiratory syndrome-related corona-virus: classifying 2019-nCoV and naming it SARS-CoV-2", Abril 2020

[2] World Health Organization. "WHO Director-General's opening remarks at the media briefing on COVID-19" - 11 Març 2020. Consultat a https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020.

[3] World Health Organization. Coronavirus disease (COVID-19) outbreak situation. Consultat a https://www.who.int/emergencies/diseases/novel-coronavirus-2019. Data de consulta: 4 de Maig de 2020.

[4] Europa Press, ed. "La transmisión del Covid-19 se produce 1 o 2 días antes del inicio de síntomas y podría seguir en verano". Consultat a https://www.infosalus.com/actualidad/noticia-transmision-covid-19-produce-dias-antes-inicio-sintomas-podria-seguir-verano-20200326112514.html. Data de consulta: 26 de Març de 2020.

[5] Periòdic Hosteltur "Son 156 los países con las fronteras cerradas al turismo internacional". Consultat a https://www.hosteltur.com/136596_son-156-los-paises-con-las-fronteras-cerradas-al-turismo-internacional.html. Maig 2020.

[6] J. P. Cohen, P. Morrison, and L. Dao, "COVID-19 image data collection," preprint arXiv:2003.11597, 2020. https://github.com/ieee8023/covid-chestxray-dataset

[7] D. S. Kermany, et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," Cell, 172(5), pp. 1122–1131, 2018. https://doi.org/10.1016/j.cell.2018.02.010

[8] Jianpeng Zhang, Yutong Xie, Yi Li, Chunhua Shen, and Yong Xia, "COVID-19 Screening on Chest X-ray Images Using Deep Learning based Anomaly Detection," arXiv:2003.12338v1, Març 2020.

[9] Adrian Yijie Xu, "Detecting COVID-19 induced Pneumonia from Chest X-rays with Transfer Learning: An implementation in Tensorflow and Keras," Towards Data Science, 2020.

[10] Ryan Gotesman, "Prototyping a Neural Network to diagnose Covid-19 from Chest X-ray," Towards Data Science, 2020.

[11] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," preprint arXiv:2003.10849, 2020.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[13] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, pp. 248–255, 2009.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," In Thirty-first AAAI Conference on Artificial Intelligence, Febrer 2017.

[15] O. Russakovsky, et al., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, 115(3), pp. 211–252, 2015.

[16] B. Ghoshal, and A. Tucker, "Estimating uncertainty and interpretability in deep learning for coronavirus (COVID-19) detection," preprint arXiv:2003.10769, 2020.

[17] L. Wang, and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images," preprint arXiv:2003.09871, 2020.

[18] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, Maria de la Iglesia-Vayá, "PadChest: A large chest x-ray image dataset with multi-label annotated reports," preprint arXiv:1901.07441v2, Febrer 2019.

[19] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, Maig 2015.

[20] Daphne Cornelisse, "An intuitive guide to Convolutional Neural Networks". Consultat a https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/.

[21] Arden Dertat, "Applied Deep Learning - Part 4: Convolutional Neural Networks". Towards Data Science, 2017.

[22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda

Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[23] François Chollet et al. Keras.https://keras.io, 2015

APÈNDIX A
Topologies

En aquest apèndix mostrarem les topologies dels classificadors que no hem mostrat a la memòria per qüestió d'espai i per no sobrecarregar-la d'imatges.

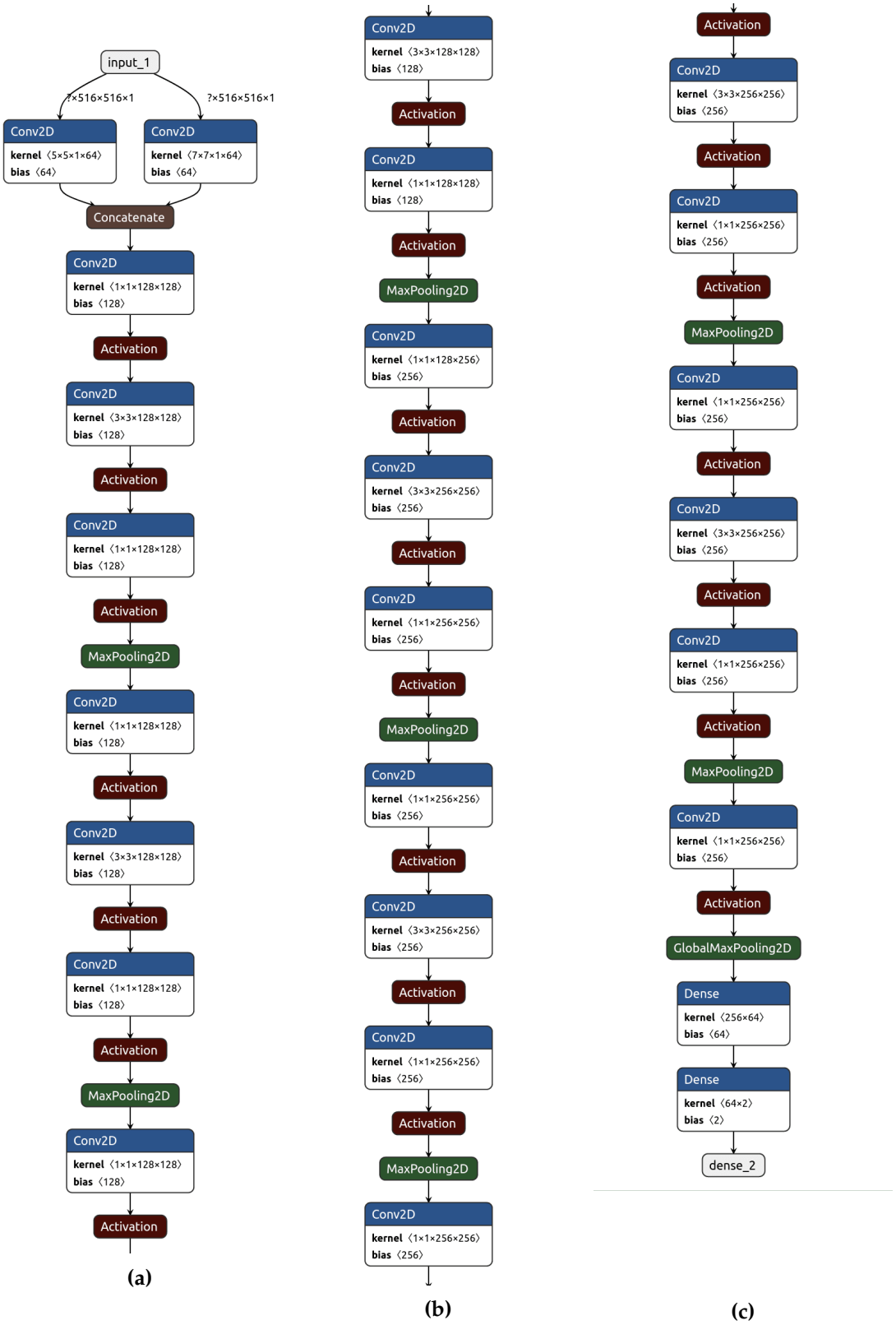


Figura A.1: Topologia del model 4b.

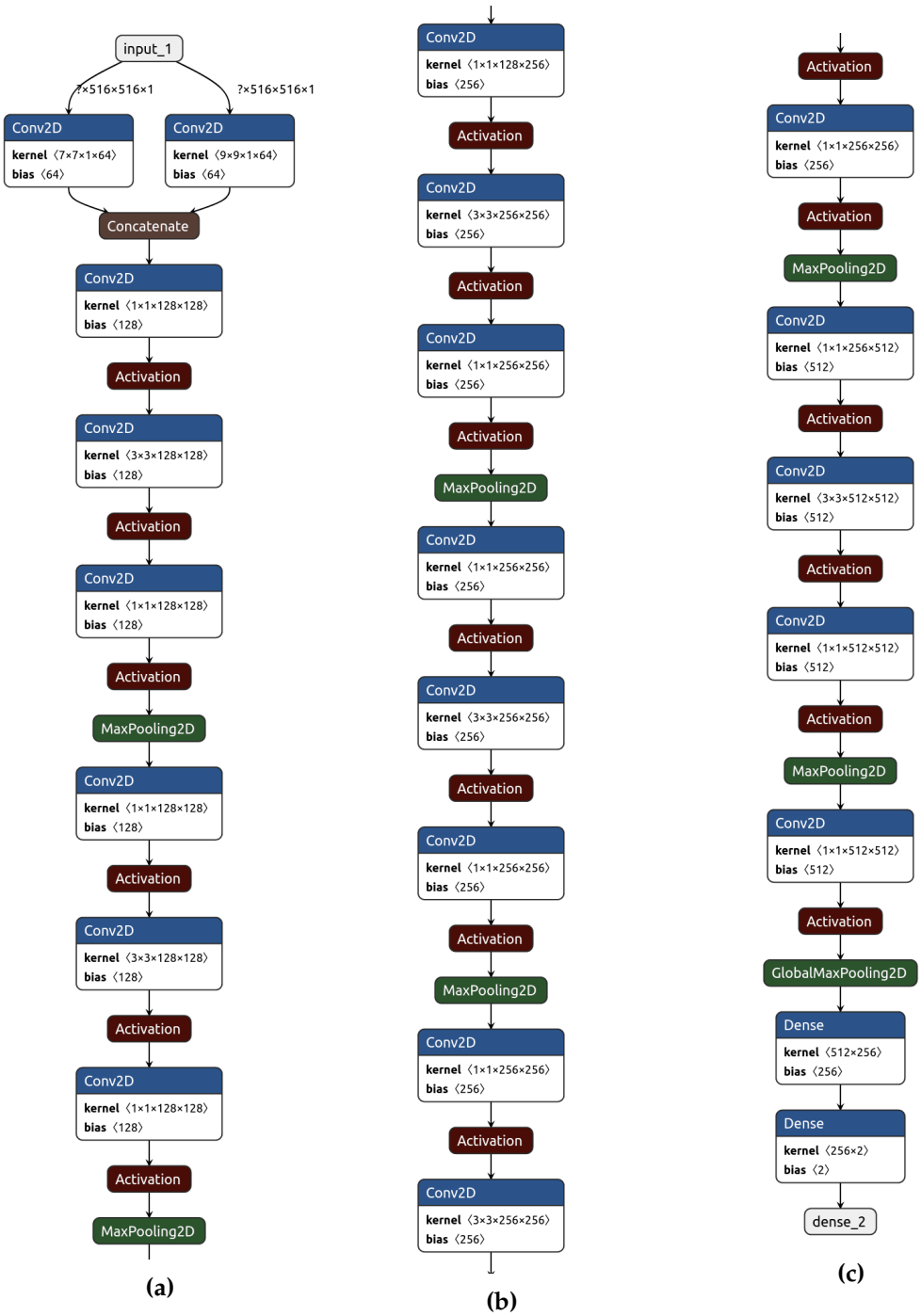


Figura A.2: Topologia del model 4c.

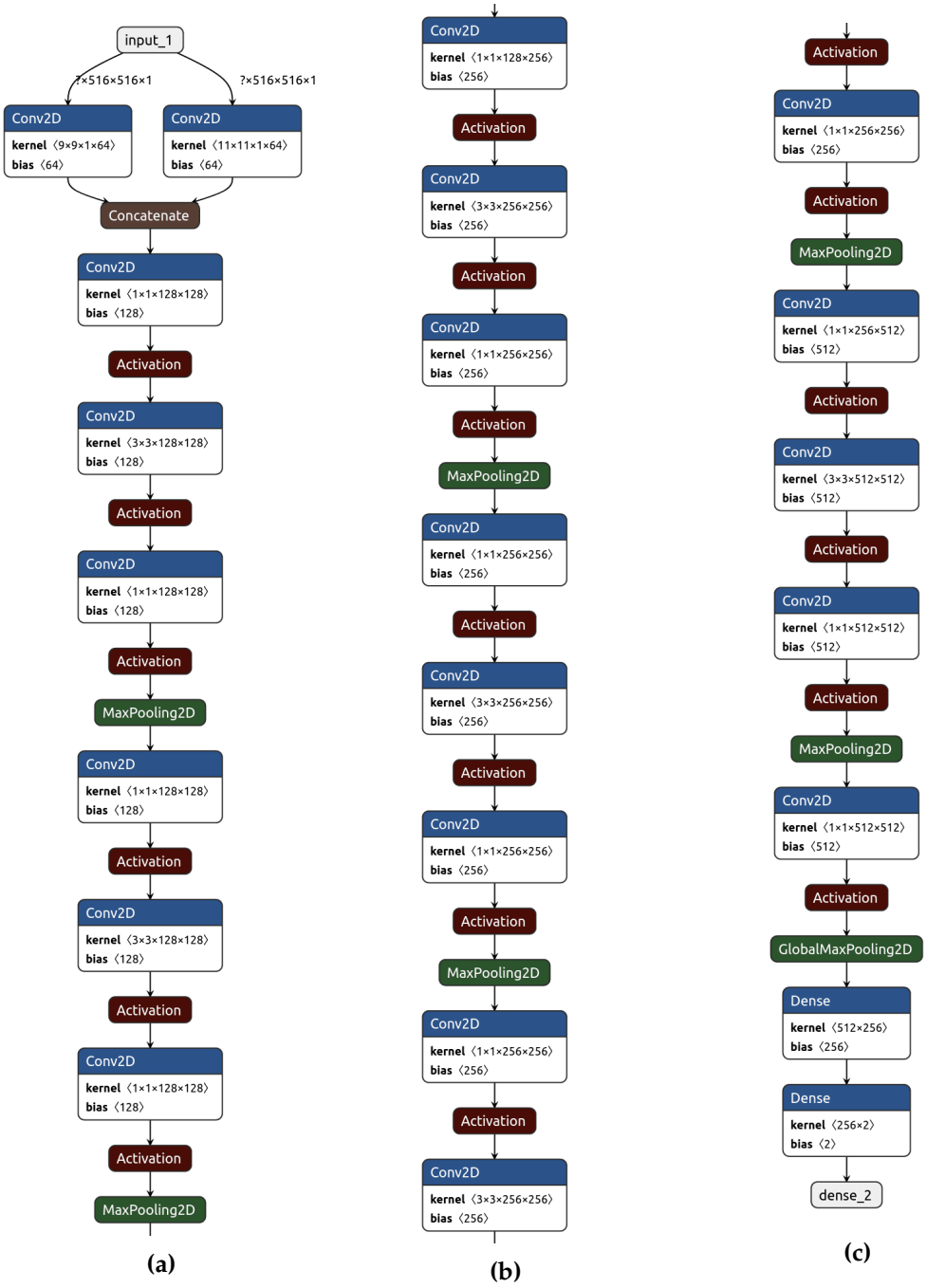


Figura A.3: Topologia del model 4d.

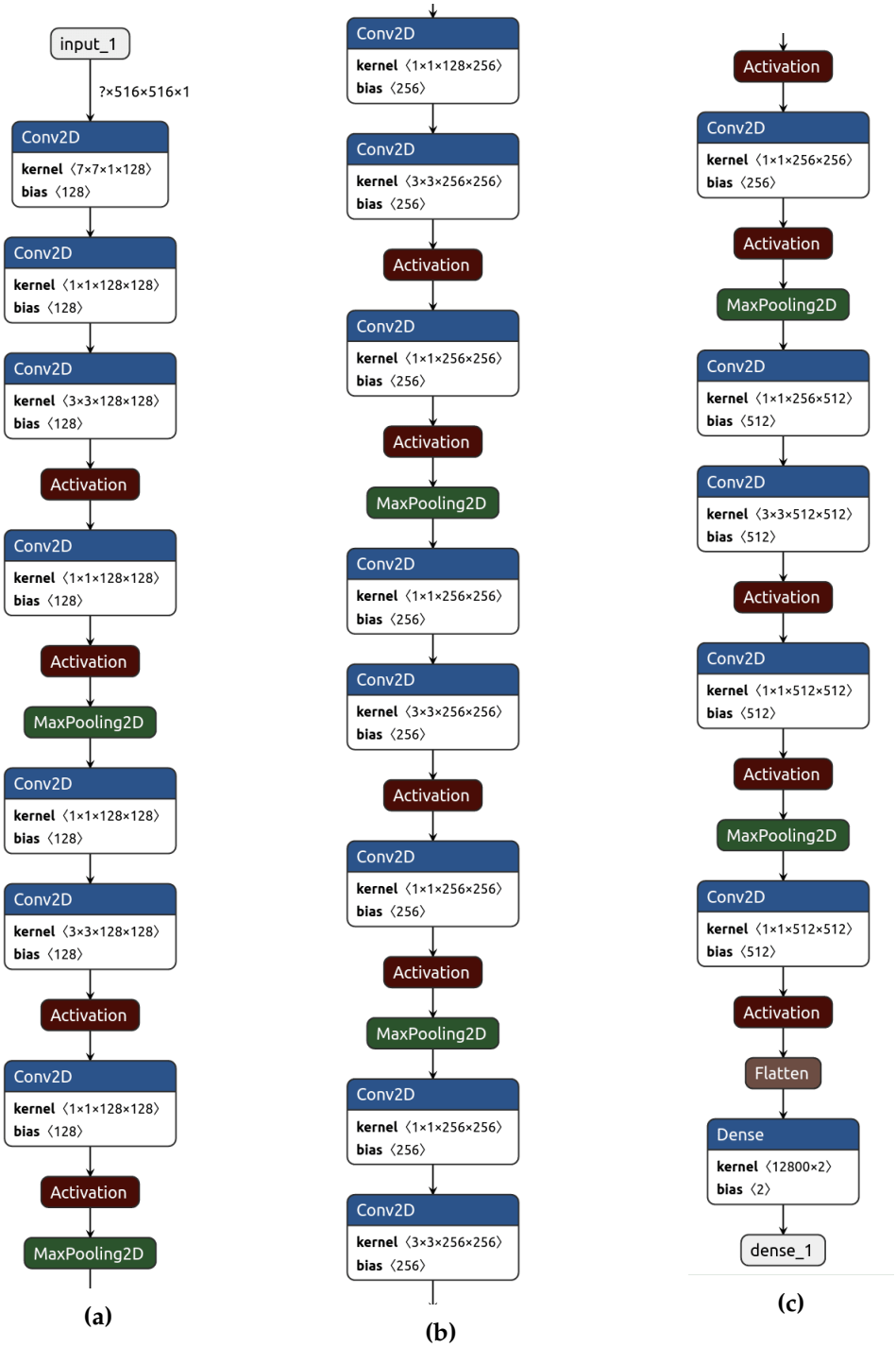


Figura A.4: Topologia del model 5b.

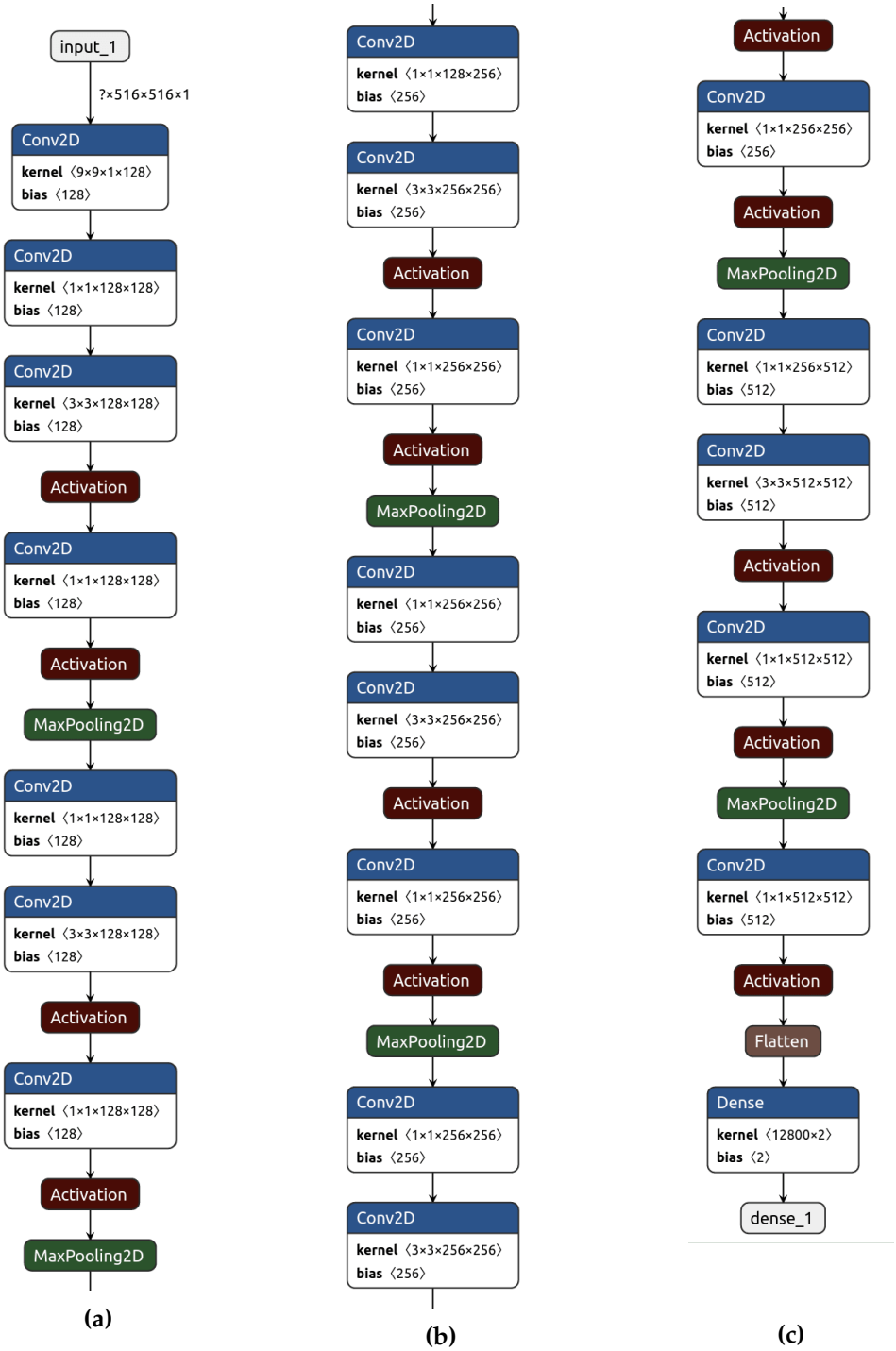


Figura A.5: Topologia del model 5c.

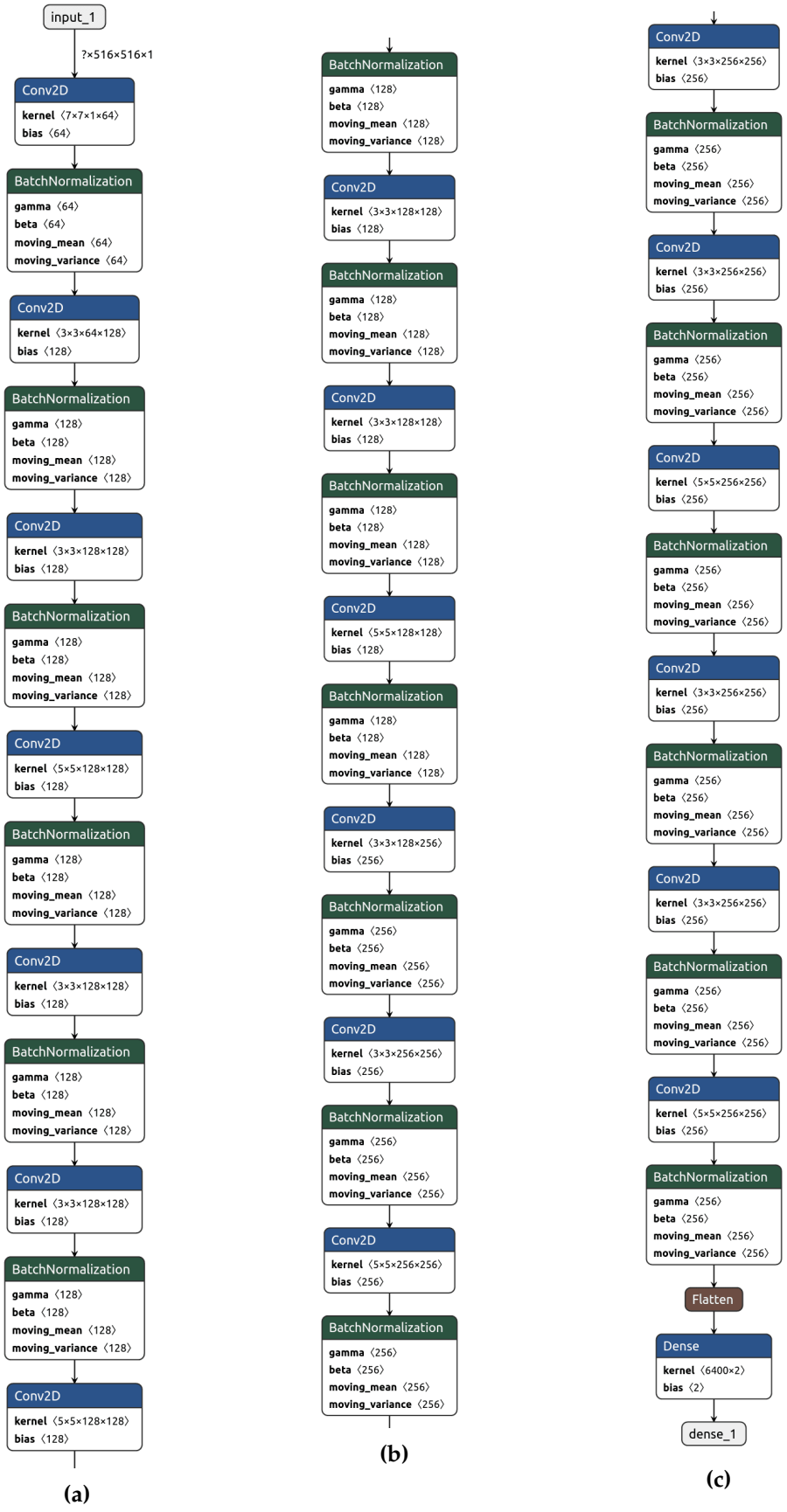


Figura A.6: Topologia del model 6b.

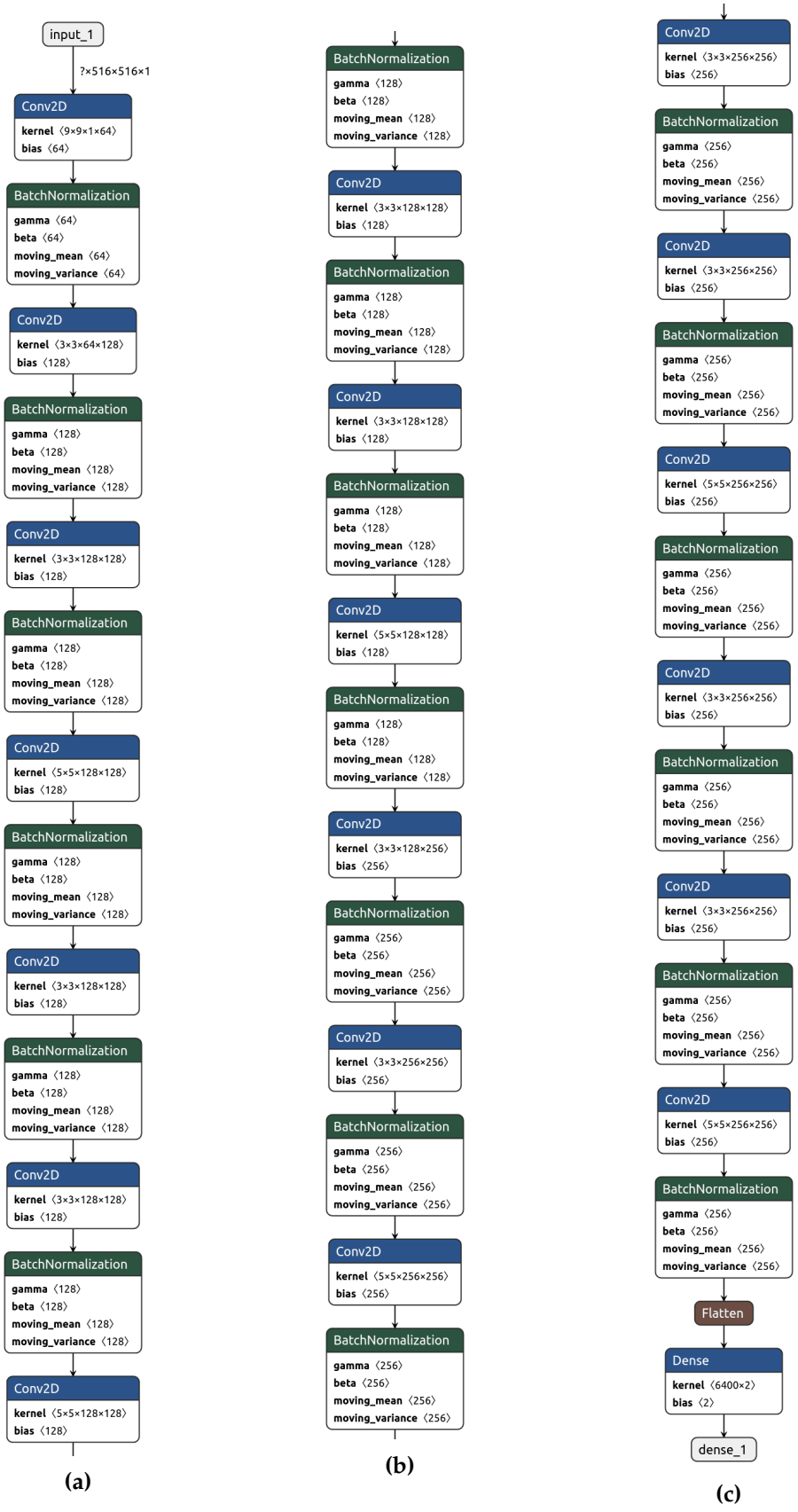


Figura A.7: Topologia del model 6c.

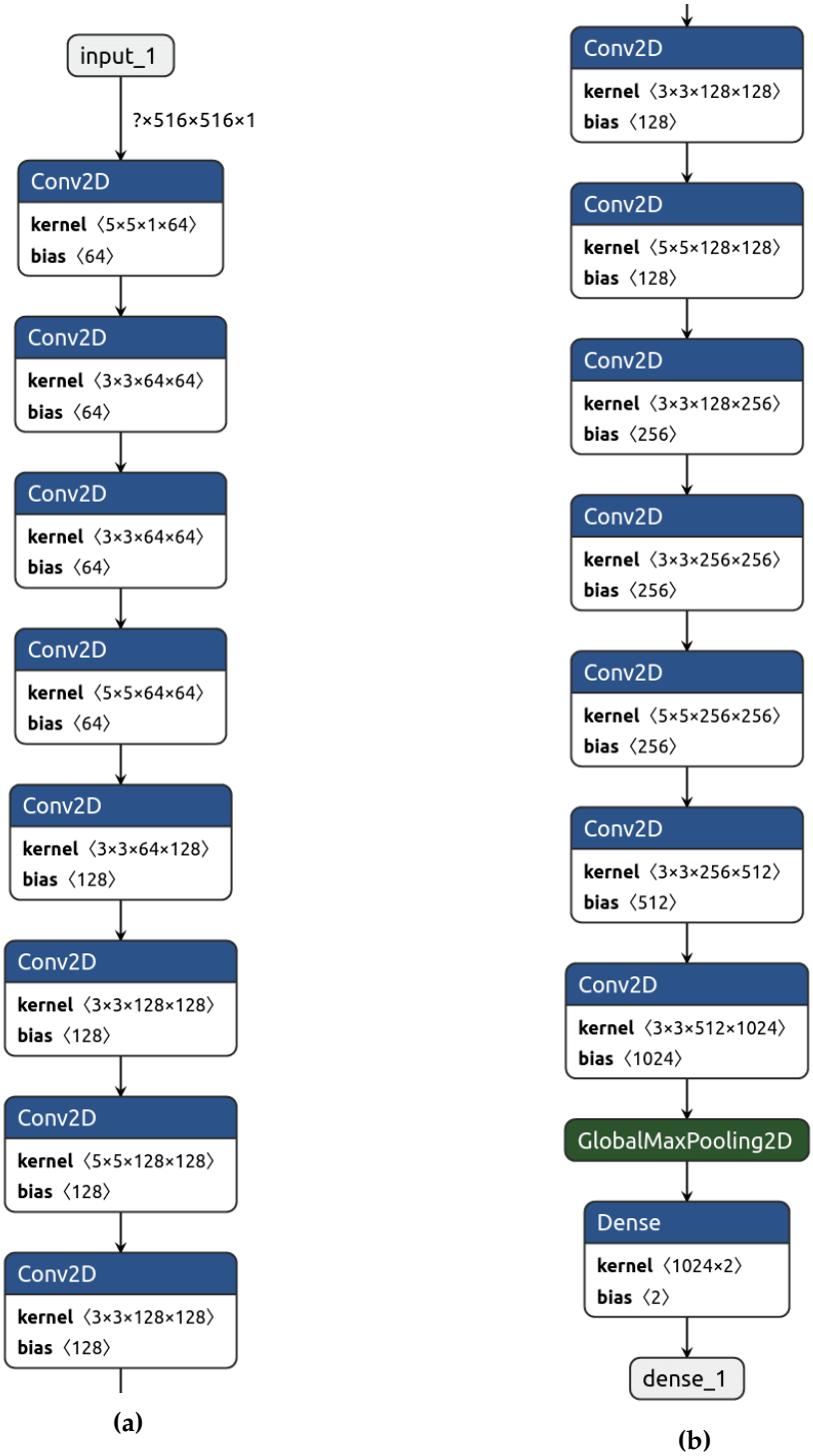


Figura A.8: Topologia del model 7a.

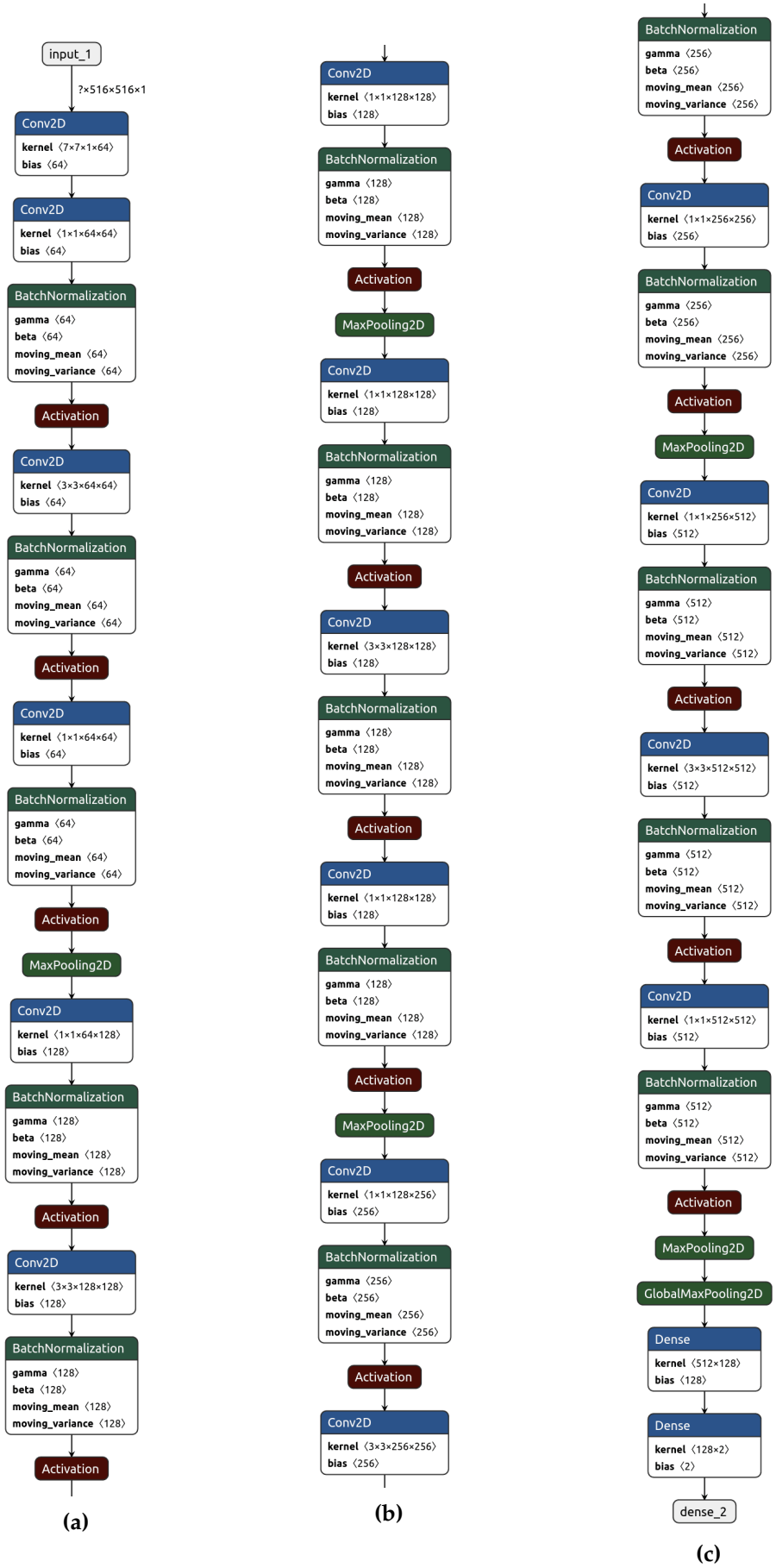


Figura A.9: Topologia del model 7b.

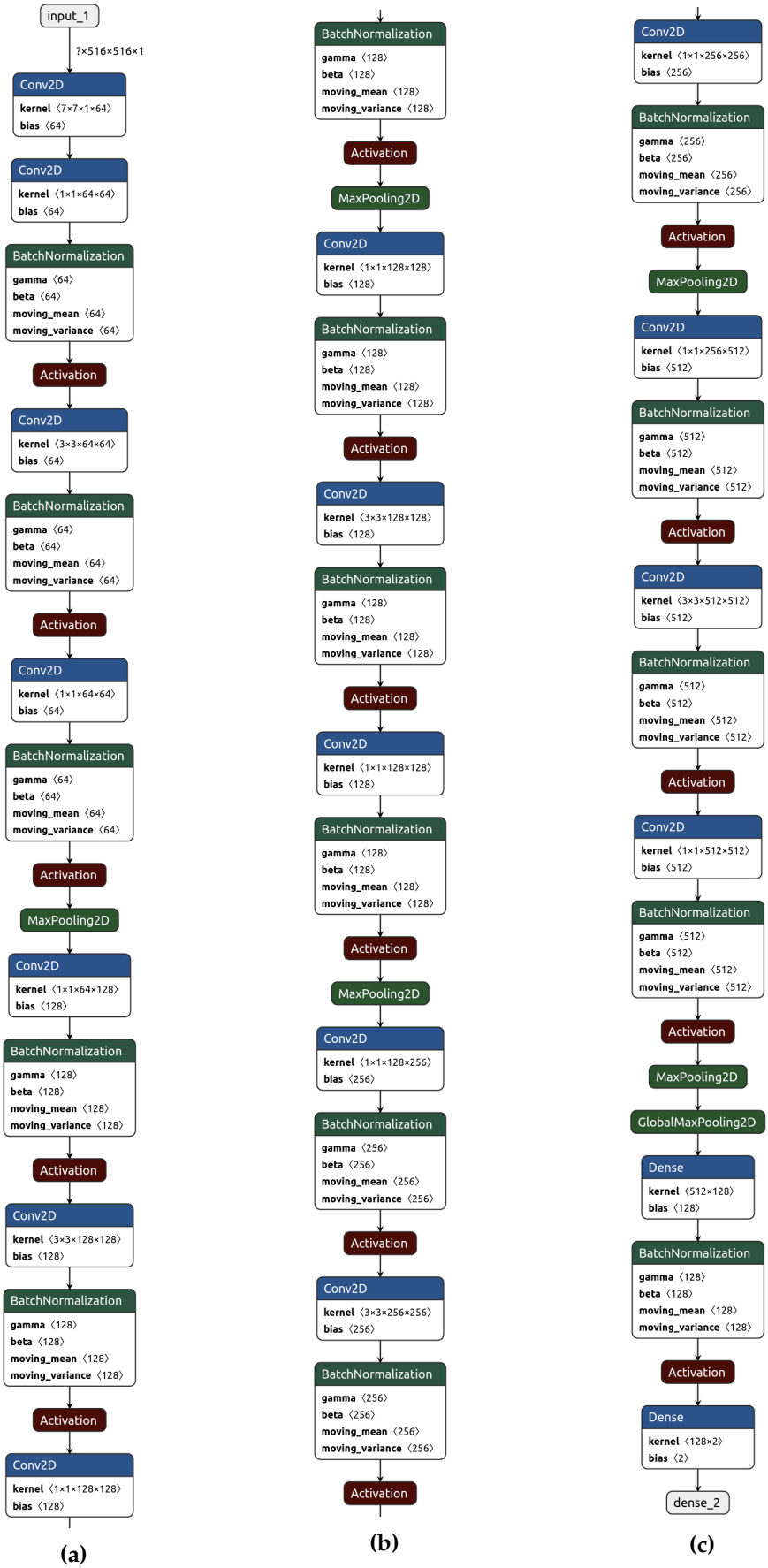


Figura A.10: Topologia del model 8a.

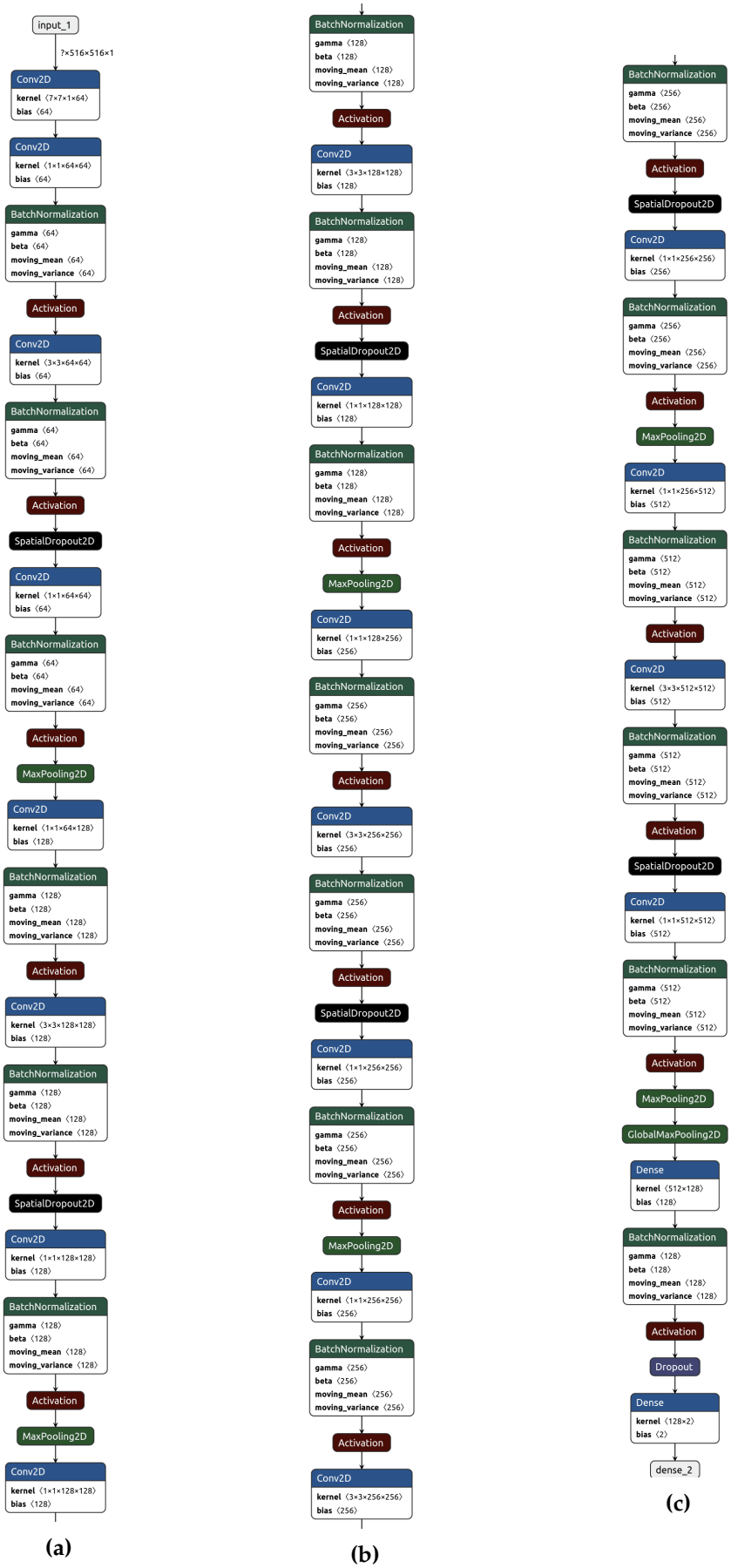


Figura A.11: Topologia del model 8d.