Aleix Vives Bosch

IDENTIFICACIÓ DE PLÀNTULES MITJANÇANT L'APRENENTATGE PROFUND

TREBALL DE FI DE GRAU

dirigit per Susana Àlvarez i Maria Ferré

Grau d'Enginyeria Informàtica/Telemàtica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona
2021

Índex

Índex de figures

Índex de taules

1 Introducció

La motivació per aquest treball sorgeix a partir d'un contacte amb una empresa que es dedica a la reforestació de boscos a la península Ibèrica. Les plantacions es fan segons uns projectes que proposen una distribució de plantes ajustada al tipus de terreny. Durant la plantació, es duen a terme controls de qualitat per revisar que la distribució de les plantes és la correcta d'acord amb el projecte. El problema és que les plantes en qüestió són molt petites, pràcticament són brots, i algunes només tenen el tronc. A més, el personal que realitza els controls de qualitat no sempre són especialistes. Això porta a errors durant la identificació de les plantes i com a conseqüència, controls de qualitat invàlids.

Aquest treball tracta de demostrar que el problema pot ser solucionat fent ús de tècniques de *deep learning*, i més concretament amb l'ús de xarxes neuronals convolucionals. El desenvolupament d'una eina d'aquestes característiques permetria als operaris de la plantació identificar l'espècie de les plantes durant els controls de forma instantània. Per això, un dels grans reptes que també afronta el treball és maximitzar la precisió de l'eina resultant, perquè encara que no es té accés al percentatge d'encert que pot arribar a tindre un operari, hauria de ser capaç de classificar la majoria d'espècimens de la plantació per tenir potencial com a alternativa.

Convé ressaltar que el deep learning tal com es coneix avui en dia, es pot considerar un concepte molt nou. Al voltant del 2010, gràcies a les últimes millores en targetes gràfiques i la publicació d'ImageNet l'any anterior, el *deep learning* va començar a agafar força al sector industrial. És un avanç fruit de molts anys d'estudis i d'investigació, que apareixen tan aviat com el 1943 amb la publicació de l'article "A Logical Calculus of the Ideas Immanent in Nervous Activity" per part de Walter Pitts i Warren McCulloch[12]. El terme es va popularitzar entre el públic uns tres anys més tard, al llarg del 2013. A partir de llavors, només va ser qüestió de temps que les grans empreses com Google comencessin a invertir en aquesta tecnologia i a aplicar-la als seus sistemes.

Figura 1: Google Trends [18]. Gràfic de popularitat del terme *deep learning* entre els anys 2005 i 2020.

Hi ha hagut grans avanços en camps on abans s'utilitzaven algoritmes més primitius d'identificació, classificació d'objectes o prediccions. Alguns exemples que podem trobar

en el nostre dia a dia són els algorismes d'optimització de recorregut de Google Maps, els assistents intel·ligents com ara l'Amazon Alexa, o simplement els filtres de correu brossa que tenen la majoria de serveis de correu electrònic avui en dia. A més, també és una peça clau per a desenvolupar les tecnologies del futur com ara els cotxes autònoms.

El *deep learning* o aprenentatge profund és una branca del *machine learning* o aprenentatge automàtic que utilitza xarxes neuronals amb diverses capes. Aquestes xarxes neuronals intenten replicar el funcionament del cervell humà, analitzant dades de manera similar a la forma en què una persona veuria un problema. En l'aprenentatge automàtic tradicional, l'algorisme rep un conjunt de característiques rellevants per analitzar, en canvi, en l'aprenentatge profund es proporcionen dades en brut a l'algorisme i aquest decideix per si mateix quines característiques són rellevants. És aquesta decisió que pren l'algorisme sense interacció de l'usuari que fa que es compari amb el pensament humà. A més, les xarxes d'aprenentatge profund sovint milloren a mesura que s'augmenta la quantitat de dades que s'utilitzen per entrenar-les. Igual que els humans aprenen de l'experiència, un algorisme d'aprenentatge profund pot dur a terme una tasca repetidament, i cada vegada modificant-la per millorar-ne el resultat. Per entendre la cadena d'esdeveniments que ocorren durant el funcionament d'una xarxa, es pot seguir la següent explicació: primerament es crea una xarxa neuronal de perceptrons, o neurones digitals, disposades en capes, amb els perceptrons de cada capa interconnectats als perceptrons de la següent capa. A continuació, es proporcionen dades d'entrenament a la xarxa neuronal, com ara imatges d'objectes, i aquesta intenta endevinar què és cada objecte. Per descomptat, les seves suposicions inicials seran atroces, però a mesura que es proporcioni retroalimentació a cada iteració, la xarxa neuronal ajustarà la manera com es connecten els seus perceptrons fins a produir conjectures molt precises. En aquest moment, s'ha aconseguit un model entrenat que pot etiquetar objectes amb una gran precisió [10].

Al camp de l'aprenentatge profund es poden trobar fins a tres tipus diferents de xarxes neuronals, però a raó d'aquest treball es basaran les explicacions entorn de les xarxes neuronals convolucions (CNN), que són les xarxes més dominants al camp del processament d'imatge.

Finalment, en aquest treball també farem ús d'un dels mètodes més coneguts dins del món de l'aprenentatge profund: el *transfer learning* o transferència d'aprenentatge. Aquest és un mètode d'aprenentatge automàtic en què un model desenvolupat i entrenat anteriorment per una tasca és reutilitzat com a model inicial per a completar una segona tasca. En l'aprenentatge profund és un procediment popular on s'utilitzen models preentrenats com a punt de partida en tasques de visió per computador i processament de llenguatge natural. Per adaptar aquests models preentrenats a la nova tasca, les últimes capes de classificació de la xarxa són reemplaçades per una capa de classificació nova amb les classes que hi ha a aquesta nova tasca. D'aquesta manera, s'obté un nou model on els primers perceptrons ja estan entrenats i només cal entrenar les capes finals de classificació. La transferència d'aprenentatge és freqüentment utilitzada per evitar invertir els recursos de càlcul i temps necessaris per a desenvolupar models de xarxes neuronals sencers. A més, també fan accessibles models molt refinats que l'usuari estàndard no tindria prou habilitat i coneixement per desenvolupar[6].

2 Objectius

Aquest treball té un únic objectiu principal:

- Aconseguir classificar espècies de planta forestal a partir d'imatges captades sobre el terreny. L'eina resultant d'aquest treball ha de ser capaç de funcionar en les mateixes condicions que els operaris que l'han d'utilitzar, i detectar i classificar plantes que han estat plantades fa poc i la seva mida és molt petita.

A part del propòsit principal del treball, es poden llistar altres objectius més específics que s'han anat seguint al llarg del treball:

- Diferenciar les imatges vàlides per la identificació. Abans d'introduir-les al sistema de classificació, s'ha de saber quines dades estan qualificades per a ser utilitzades.
- Aprendre com funciona l'aprenentatge profund i també mètodes importants com la transferència d'aprenentatge. En acabar el treball, s'ha de saber el funcionament d'aquests termes i els casos en els quals es poden utilitzar, tenint en compte que a l'inici del treball no en sabia res.

3 Estat de l'art

Un dels punts forts de l'aprenentatge profund és l'accessibilitat que ofereix; no és una tecnologia que requereixi grans inversions per fer-la funcionar ni personal especialitzat. Es tracta d'una tecnologia increïblement polivalent i amb un potencial sorprenent dins del món de la computació, aplicable a un gran nombre de camps que en poden fer ús. La botànica és un d'aquests camps que gaudeix dels avantatges de l'aprenentatge profund. Actualment podem trobar diverses aplicacions que podrien solucionar el problema que es planteja, la gran majoria d'aquestes es troben exclusivament en plataformes mòbils per garantir l'accessibilitat als usuaris. Un exemple clar és Pl@ntNet, un projecte de ciència ciutadana per la identificació automàtica de plantes a través d'imatges preses pels usuaris i utilitzant xarxes neuronals convolucionals. Fent ús de l'aprenentatge profund, aquesta aplicació és capaç de diferenciar entre més de 27.000 espècies de flora diferent i presumeix d'una base de dades amb més de 5 milions d'imatges[15]. De fet, aquests números no paren de créixer cada dia. Malauradament, la majoria d'aquests serveis no ofereixen gaires detalls sobre quins models de xarxa utilitzen ni quines tècniques de tractament de dades es fan abans de la identificació.

A part de les aplicacions com Pl@ntNet, dedicades a la identificació de les plantes normalment desenvolupades, no s'ha pogut trobar cap solució que adreci el problema que tenim abans de fer la nostra pròpia implementació. Es poden trobar solucions a problemes similars, com la que es presenta a l'article *Convolutional Neural Network Architecture for Plant Seedling Classification*[11], on s'exposa un model de xarxa neuronal convolucional

per identificar i diferenciar cultiu i males herbes en etapes inicials. El model que es proposa pot servir de referència a l'hora de fer el nostre, però s'ha de tindre en compte que les plantes amb les quals es tracta en aquest article són morfològicament diferents de les que tractem nosaltres; són plantes de cultiu, mentre que les espècies d'aquest treball són arbres.

Així doncs, primer de tot va caldre adquirir els coneixements necessaris sobre el tema per començar a abordar aquest projecte. Per sort, abans de poder començar a treballar també s'havien de demanar el conjunt de dades d'entrenament a l'organització que s'ocupa de les plantacions, ja que aquestes plantacions es troben a altres parts de la península i anar a prendre les dades personalment era una tasca massa costosa.

Tal com s'explicarà més endavant, la plataforma de desenvolupament que es va escollir va ser MatLab. A part dels avantatges d'accessibilitat, MatLab ofereix un gran ventall de cursos de formació en diversos temes, entre ells l'aprenentatge profund. A l'apartat de formació en línia que es pot trobar a la pàgina web oficial, es poden trobar dos cursos diferents sobre aprenentatge profund. El primer consisteix en una introducció al tema que té una durada de 2 hores. El segon curs té una durada de 8 hores i aprofundeix molt més en tots els àmbits de l'aprenentatge profund, combinant ensenyaments teòrics i pràctics. Més concretament, es proposen 2 projectes amb els seus respectius conjunts de dades, uns objectius a assolir i una solució. La idea és que l'usuari intenti proposar una solució i després la pugui contrastar amb la solució oficial que proposen ells.

Gràcies a aquests cursos, quan van arribar les dades necessàries per començar a treballar amb les xarxes ja coneixia els passos bàsics per construir un model i els coneixements per entrenar-lo. Alguns dels conceptes més importants sobre les xarxes els vaig aprendre en aquests cursos. Conceptes com l'anatomia de les capes convolucionals o l'algorisme de retropropagació ajuden a entendre molt millor el funcionament d'una xarxa en general. A continuació s'expliquen aquestes nocions per fer més fàcil la comprensió d'alguns apartats més endavant.

3.1 Capes d'una xarxa neuronal

En una xarxa neuronal convolucional es poden trobar fins a 4 capes diferents, i cada una té la seva funció:

- Capa convolucional
- Capa connexa
- Capa d'agrupació
- Capa de normalització

El primer tipus de capa són les convolucionals, que són les capes encarregades d'extreure característiques a les dades d'entrada de la xarxa, és a dir, podrien ser considerades les neurones digitals. El nucli central de la xarxa neuronal convolucional és la capa

convolucional que dona nom a la xarxa. Aquesta capa executa una operació anomenada "convolució", que en el fons és aplicar un filtre a l'entrada, i el resultat d'aquesta operació s'anomena activació. Repetides aplicacions d'un filtre a l'entrada resulta en un mapa d'activacions (*feature map* en anlgès), que ens mostra el lloc i la intensitat de la presència d'una característica a l'entrada, com ara una imatge. En el context d'una xarxa neuronal convolucional, una convolució és una operació lineal que implica la multiplicació d'un conjunt de pesos amb l'entrada. Tenint en compte que la tècnica va ser dissenyada per a l'entrada bidimensional, la multiplicació es realitza entre una matriu de dades d'entrada i una matriu bidimensional de pesos, anomenada filtre. El filtre és més petit que les dades d'entrada, per tant, el tipus de multiplicació aplicat és un producte escalar. L'ús d'un filtre més petit que l'entrada és intencionat, ja que permet multiplicar el mateix filtre (conjunt de pesos) per la matriu d'entrada diverses vegades en diferents punts de l'entrada. En concret, el filtre s'aplica sistemàticament a cada part superposada de les dades d'entrada de la mida del filtre, d'esquerra a dreta, de dalt a baix[7].

Ja que el filtre s'aplica diverses vegades a la matriu d'entrada, el resultat és una matriu bidimensional de valors de sortida que representen un filtratge de l'entrada. Com a tal, la matriu de sortida bidimensional d'aquesta operació s'anomena mapa de característiques, o mapa d'activacions.

Normalment, les xarxes neuronals convolucionals no aprenen només d'un sol filtre, sinó que n'apliquen diversos de forma paral·lela per a qualsevol entrada. Per exemple, és habitual que una capa convolucional aprengui de 32 a 512 filtres en paral·lel per a una entrada determinada. Això dona al model 32, o fins i tot 512, maneres diferents d'extreure característiques d'una entrada, o moltes maneres diferents "d'aprendre a veure", i després de l'entrenament, moltes maneres diferents de "veure" les dades d'entrada.

Les capes convolucionals no només s'apliquen a les dades d'entrada, sinó que també es poden aplicar a la sortida d'altres capes. Apilar capes convolucionals permet una descomposició jeràrquica de l'entrada. Si es considera que els filtres que funcionen directament sobre els valors dels píxels de l'entrada aprendran a extreure característiques de baix nivell, com ara línies, els filtres que actuen a la sortida de les primeres capes podran extreure característiques que són combinacions de característiques de nivell inferior, com ara formes compreses de diverses línies. Aquest procés continua fins que capes molt profundes extreuen cares, animals, cases, etc.

En segon lloc hi ha les capes d'agrupació o *pooling*, i n'hi ha dos tipus: agrupació màxima i agrupació mitjana. La tasca d'aquesta capa és reduir el nombre de paràmetres de l'entrada. Tenint en compte que una capa convolucional pot arribar a tindre 512 filtres, la xarxa multiplica per 512 el nombre de paràmetres a analitzar només passar per la primera capa. Utilitzant les capes d'agrupació, els models poden reduir els costos computacionals i les probabilitats de problemes com l'excés d'adequació. Aquesta capa utilitza una matriu petita (normalment de 2x2) que és desplaçada al llarg de la matriu d'entrada, i per cada posició es calcula el màxim dels valors si és una agrupació màxima, o la mitjana dels valors si és una agrupació mitjana. S'ha de tenir en compte que les operacions d'agrupació redueixen la mida de les matrius d'entrada, però el nombre de canals (habitualment canals de colors) d'aquestes entrades es manté igual[3].

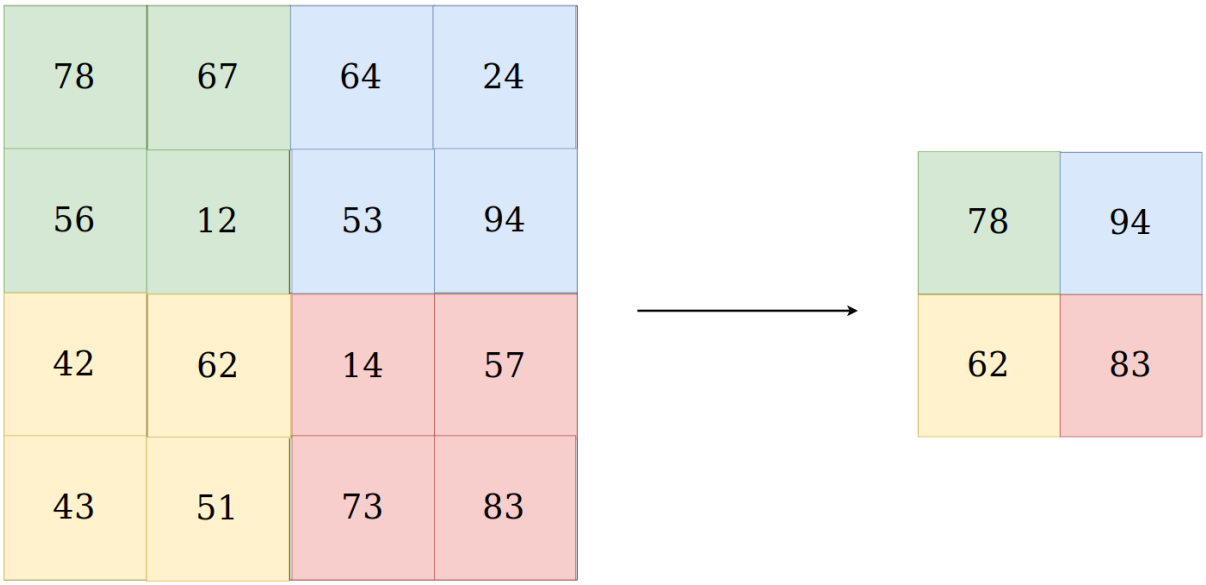


Figura 2: Il·lustració d'una capa d'agrupació màxima.

En tercer lloc trobem les capes connexes, que són les capes que normalment es troben al final de la xarxa. Cada capa connexa estarà connectada a tots els nodes de la capa anterior i de la següent, d'aquí apareix el seu nom. És a dir, l'entrada d'una capa connexa és la sortida d'una capa convolucional o d'agrupació. Aquesta entrada sempre té la forma d'una matriu tridimensional i, abans d'entrar a la capa connexa, s'aplana i es converteix en un vector de valors. Seguidament, s'aplica l'operació $g(Wx + b)$, on x és el vector d'entrada, W és una matriu de pesos, b és un biaix i g és la funció d'activació (normalment ReLU). Aquesta operació es repeteix per cada capa connexa. Un cop acabades les capes connexes, la capa final utilitza la funció d'activació de *softmax* que s'utilitza per obtenir probabilitats que l'entrada estigui en una classe particular (classificació)[3].

Per últim, la capa de normalització o *batch normalization* és una tècnica per entrenar xarxes neuronals molt profundes que normalitza les entrades a una capa per a cada mini lot. Això té l'efecte d'estabilitzar el procés d'aprenentatge i reduir dràsticament el nombre d'èpoques de formació necessàries per formar xarxes profundes. Cal recordar que la normalització es refereix al fet que les dades de redimensionament tenen una mitjana de zero i una desviació estàndard d'1, és a dir, una distribució normal (també coneguda com a distribució gaussiana)[4].

3.2 Retropropagació

L'algorisme de retropropagació o *backpropagation* és probablement un dels blocs fonamentals més importants que constitueixen l'aprenentatge profund tal com es coneix avui en dia, ja que permet als models aprendre i millorar els resultats en cada iteració. Aquest algorisme utilitza el descens de gradient per trobar l'error que hi ha entre la conjectura actual i la solució òptima. Sense detallar, el descens de gradient és un algorisme que busca el valor

més baix d'una funció. La retropropagació s'utilitza recursivament en cada perceptró per ajustar els pesos segons indica el descens de gradient, i d'aquesta manera reduir al mínim l'error. Quan s'arriba a un mínim voldrà dir que s'ha arribat a una solució òptima. Aquesta és la raó per la qual entrenar xarxes neuronals té costos temporals i computacionals tan alts. Òbviament, aquesta descripció pot semblar escassa, però tampoc cal elaborar gaire si es té el concepte anterior clar. La retropropagació és un dels aspectes més tècnics de l'aprenentatge profund, i també és un dels més discutits. De fet, alguns pensen que és una mala metodologia i hauria de ser canviada, encara que altres alternatives no existeixen i és una àrea d'investigació actualment[17].

4 Anàlisi del problema

Al començament del treball es van plantejar diverses maneres d'encarar el problema utilitzant mètodes diferents. Per una banda, aprofitant que l'empresa de reforestació disposava de les distribucions d'espècies a les plantacions, hi havia la possibilitat de desenvolupar una aplicació amb localització GPS que detectes la zona actual de la plantació i llistés les espècies que hi poden haver en aquesta. Tot seguit, el personal no especialitzat que duu a terme els controls de qualitat podria identificar els espècimens a partir d'imatges de referència i característiques que els poden diferenciar. Per altra banda, l'altra proposta consistia a utilitzar algun procediment de classificació d'imatges mitjançant tècniques de processament d'imatge, o algun mètode més modern com l'aprenentatge profund. Seguidament, es podria desplegar aquesta solució en forma d'aplicació per dispositius mòbils i d'aquesta manera el personal podria identificar qualsevol individu de la plantació. Finalment vaig escollir la segona opció perquè preferia treballar en l'àmbit de la visió de computadors, i tot i ser més difícil, la solució podia ser més eficaç.

L'objectiu general que es volia aconseguir en acabar el projecte era desenvolupar una eina que fos capaç de classificar totes les espècies de plantes de la plantació, és a dir, aproximar-se al màxim a l'estat de l'art dels classificadors de flora però amb un conjunt de classes més petit. A més, també hi havia la possibilitat de desplegar la solució obtinguda en forma d'aplicació mòbil per al seu ús en el camp, però és evident que el que es plantejava era una tasca massa difícil pel temps que es disposava i la complexitat que requereix un treball de final de grau. Així doncs, es va acordar que el treball se centraria únicament a aconseguir un sistema de classificació d'espècies.

A l'hora de decidir com afrontar el problema, es van plantejar dues direccions. Per una banda, ja que es tracta d'un problema de detecció i identificació d'objectes, es podria fer ús de tècniques de segmentació d'imatge. És un mètode més aviat senzill i que resultava més familiar perquè s'havia treballat al llarg de l'últim any del grau. Per altra banda, va sorgir la possibilitat d'utilitzar l'aprenentatge profund, una eina més potent però també totalment desconeguda. Finalment es va decidir utilitzar aquest últim, ja que la segmentació d'imatges es pot complicar si tenim en compte que la morfologia de les plantes amb les quals es tracta pot ser molt variant. L'aprenentatge profund és una eina amb més possibilitats i potencialment una solució més generalitzada que pot arribar servir per a problemes

similars en altres contextos. A més, com ja s'ha mencionat anteriorment, l'aprenentatge profund és molt accessible i només es necessitava temps d'aprenentatge i dades suficients per aconseguir bons entrenaments de la xarxa.

A partir d'aquí, aquest treball porta dues solucions diferents sobre la taula. La primera s'aconsegueix utilitzant la transferència d'aprenentatge. En el nostre cas, s'utilitzarà la xarxa AlexNet com a punt de partida, i s'entrenarà amb les nostres dades per provar de solucionar el problema. La transferència d'aprenentatge ofereix un gran avantatge per a usuaris amb menys experiència perquè permet utilitzar models molt complexos i refinats en situacions on desenvolupar-ne un seria molt costós. Utilitzant aquesta tècnica facilita molt la feina i només cal manipular els paràmetres d'entrenament de la xarxa per obtenir resultats òptims. La segona solució és un model fet des de zero i construït a partir d'iteracions de prova i error fins a arribar a una solució òptima.

Per altra banda, per fer ús de l'aprenentatge profund es necessita una gran quantitat de dades per poder arribar a un bon model que resolgui el problema. És més, en qualsevol situació on s'utilitzi alguna tècnica d'aprenentatge profund, com més dades es tinguin a l'abast, més robust serà el model i el resultat. En el nostre cas, les dades que necessitem són imatges de cada espècie que la xarxa ha de ser capaç de reconèixer. Desafortunadament, l'empresa que es dedica a la reforestació no té cap plantació situada a Catalunya i per obtenir aquestes imatges es depèn totalment de la seva disponibilitat i col·laboració. Aquest fet limita bastant el nombre d'imatges que es poden aconseguir i finalment el nombre d'espècies amb què es podrà treballar. Com a resultat, durant el treball només es van rebre dos lots d'imatges de 9 espècies.

El primer lot d'imatges que es va rebre va consistir d'uns 150 exemplars: 3 espècies diferents i 50 imatges de cada una. Les imatges es van prendre tal com les faria el personal que realitza els controls de qualitat, és a dir, amb els espècimens ja plantats i en estat natural. Aquestes primeres fotos que es van obtenir no eren massa bones. No es va establir cap mètode ni requisit per fer les fotos, i a causa d'això les imatges no eren útils per entrenar una xarxa efectivament. El primer factor a destacar de les imatges era l'enfocament amb què s'havien fet. Com que les plantes estan en etapes inicials no disposaven de fulles, només del tronc. A l'hora de fer la foto, resultava molt difícil que l'objectiu de la càmera enfoqués al tronc en comptes del fons. Tanmateix, l'angle amb el qual es van prendre la majoria de fotos era bastant alt, i per tant, totes aquestes fotos tenien com a fons el terra. Això ens porta al segon factor problemàtic: aquestes imatges només es van fer a 3 espècies diferents, i a un o dos espècimens de cada una. Per això, les imatges d'una mateixa espècie tenien el mateix fons (el terra), cosa que propicia que la xarxa detectes el mateix fons com a una característica determinant i classifiques les imatges a partir d'aquest.

Com que seria impossible aconseguir imatges útils en una situació natural, l'objectiu que s'havia acordat inicialment es va haver d'ajustar a les noves condicions. Així doncs, s'intenta conservar com a meta al final del treball el desenvolupament d'una eina d'identificació de flora eficaç, però amb la diferència que les imatges que s'utilitzaran com a referència estan agafades en condicions ideals. Anteriorment, la condició d'identificar les plantes en estat natural era primordial per la justificació de la solució, però amb el canvi de finalitats es pot dir que el problema a solucionar queda més obert i s'apunta més a la

Figura 3: Exemples d'imatges de cada espècie que es va rebre al primer lot. En ordre són *Betulas Alba, Populus Nigra i Sorbus Aria.*

creació d'una eina capaç d'identificar espècimens en etapes inicials de desenvolupament. En altres paraules, el treball es desvincula una mica del problema original d'identificació de plantes de la plantació.En altres paraules, el treball es desvincula una mica del problema original d'identificació de plantes de la plantació. Per tant, el nou objectiu principal del treball hauria de ser el següent:

- Aconseguir classificar espècies de planta forestal. L'eina resultant d'aquest treball ha de ser capaç d'identificar les espècies que han estat plantades fa poc, i que es troben a les plantacions on es duen a terme els controls de qualitat.

El segon lot va consistir en moltes més imatges: 7 espècies diferents, una cinquantena d'imatges per cada espècie excepte un parell d'elles amb un centenar. Aquest segon lot tenia molt millors imatges, ja que es van establir uns guions per fer les fotos. Tal com s'ha dit anteriorment, s'havia de prendre les imatges en condicions ideals. Per aconseguir-ho, es va fer la decisió de prendre les imatges sobre un fons fix, un paper blanc. D'aquesta manera la càmera podia enfocar molt millor a l'objectiu i el fons seria el mateix per a qualsevol exemplar. Es pot argumentar que utilitzant aquest mètode per obtenir imatges el treball perd validesa. Es considera, però, que si el model és capaç d'identificar correctament la planta en aquestes circumstàncies, també ho hauria de fer en altres de similars però amb un fons diferent. Al cap i a la fi, el fons de la imatge no ha de ser un factor determinant per identificar la planta. Altres factors com els canvis d'il·luminació i els angles de la foto es podrien fer irrellevants si s'obtinguessin suficients dades provant d'alterar aquests factors. De totes maneres, l'objectiu principal del treball queda canviat a partir d'aquest punt.

Figura 4: Exemples d'imatges de cada espècie que es va rebre al segon lot. En ordre són *Fraxinus Excelcior, Quercus Pirenaica i Sorbus Aria.*

5 Disseny i implementació

A l'hora de decidir l'entorn de desenvolupament es va escollir MatLab. És un programa que ofereix tot el necessari per treballar amb aprenentatge profund sense problemes, i també disposa de xarxes entrenades com AlexNet en format d'extensió. També permet manipular imatges fàcilment gràcies a les llibreries de processament d'imatge. A més a més, MatLab resulta més familiar perquè ja es va utilitzar durant l'últim curs del grau. Una altra eina que s'hauria pogut considerar és Keras, una llibreria de xarxes neuronals artificials de codi obert escrita en llenguatge Python i que actua com a interfície per a la llibreria TensorFlow[19]. El gran avantatge d'aquesta plataforma és que té un gran nombre d'usuaris, i com a conseqüència també disposa de molt bon suport a aquests en forma de fòrums i webs per solucionar dubtes. Es va acabar descartant perquè, inicialment, es va considerar la familiaritat que oferia MatLab aportaria un aprenentatge de l'aprenentatge profund més lleuger.

5.1 Model a partir d'AlexNet

Com ja s'ha comentat anteriorment, en aquest projecte es proposen dues solucions diferents, la primera utilitzant AlexNet i la segona elaborant un model des de zero. Per començar, s'explicarà sense entrar en detall el model que planteja AlexNet i algunes característiques que té.

AlexNet és una arquitectura de xarxa neuronal convolucional dissenyada per Alex Krizhevsky. Està composta per 8 capes, 5 capes convolucionals i 3 capes de classificació. Aquest model va resultar revolucionari en el seu moment per diverses raons[1]:

- Utilitza la funció d'activació ReLU (rectificador) en comptes de la funció tanh, que

fins al moment es considerava l'estàndard. L'avantatge que tenia aquesta funció és el temps d'entrenament; una xarxa utilitzant la ReLU pot aconseguir arribar a un 25% d'error en un conjunt de dades 6 vegades més ràpid que l'estàndard del moment.

- Utilitza múltiples GPU. Divideix les neurones de la xarxa i les reparteix la meitat en una targeta gràfica i l'altra meitat en una altra.
- Agrupació ("Pooling") superposada: tradicionalment es feia l'agrupació de neurones veïnes sense superposició. Quan es va introduir la superposició, van descobrir que l'error es reduïa un 0,5% i costava més que el model s'adeqüés en excés ("overfit").

Per evitar l'excés d'adequació, es van implementar dos mètodes diferents:

- Augment de dades: es van generar noves imatges utilitzant translacions per fer reflexos horitzontals. També es va fer ús de l'anàlisi de components principals (PCA) sobre els valors RGB dels píxels per canviar les intensitats dels canals RGB.
- Abandonament ("Dropout"): llavors una tècnica molt recent, consisteix a desactivar neurones aleatòriament. D'aquesta manera, durant cada iteració sempre hi haurà un cert nombre de neurones de les quals no es podrà dependre i això reforça la resta.

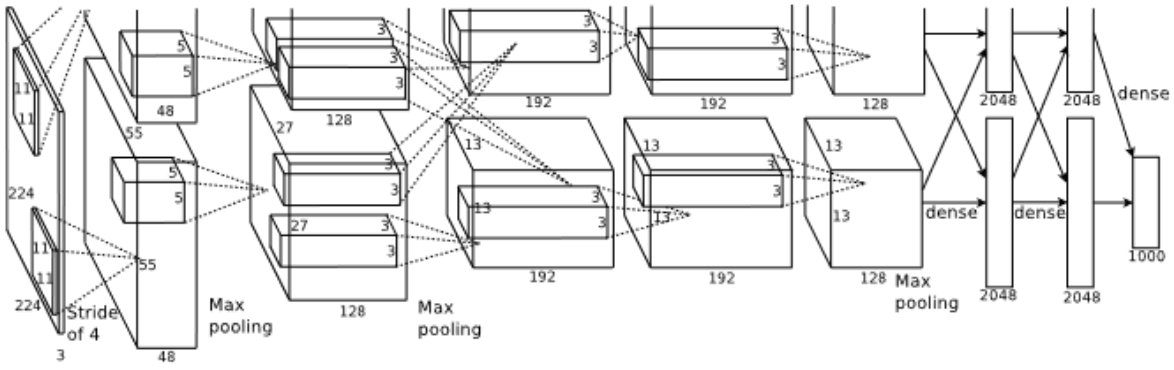


Figura 5: ImageNet Classification with Deep Convolutional Neural Networks[1]. Il·lustració de l'estructura d'AlexNet.

AlexNet va guanyar la competició anual d'ImageNet amb un percentatge d'error top-5[1] de 15,3%, comparat amb el percentatge d'error top-5 de 26,2% del segon lloc. Avui en dia encara és de les arquitectures més reconegudes.

Respecte als lots d'imatges que es reben de les plantacions, tal com s'ha dit anteriorment, el primer lot no va ser gaire útil perquè les imatges s'havien pres amb mals angles i desenfocades. De totes maneres, es va provar de fer-ne ús mentre el segon lot no arribava.

[1]Top-5 fa referència a quan es comprova que la predicció correcta d'una etiqueta es troba entre les 5 primeres prediccions (les 5 prediccions amb més percentatge).

Es va intentar desenvolupar un model utilitzant la transferència d'aprenentatge amb AlexNet, i també es va crear una arquitectura des de zero. Com que el conjunt de dades també era petit, es tenien molts problemes d'excés d'adequació. Es va provar d'augmentar les dades amb rotacions i translacions d'imatge, i també alteracions a la brillantor del color, però els resultats empitjoraven en tots els casos. Per altra banda, també es van provar els tres diferents algorismes d'optimització d'entrenament: adam, sgdm i rmsprop. Adam semblava ser l'algorisme més consistent, però el rmsprop es va obtenir el millor resultat amb un 71% de precisió en el grup de validació i prova. De totes maneres, un cop s'analitzaven les activacions de les capes de la xarxa es podia veure que els filtres obtenien poca informació de les imatges. I quan les activacions mostraven una mica de resultat, les característiques que s'obtenien mai formaven part de la planta, sinó de l'entorn en el qual es troba. Com ja s'ha explicat anteriorment, l'enfocament i l'angle de les imatges destacaven més l'entorn que la planta en si, fins al punt que la planta era indetectable en alguns dels exemplars de dades. És més, les imatges es van fer al mateix individu de cada espècie, i per tant, cada espècie tenia el mateix entorn. En definitiva, obtenim que tots els resultats que pugui arribar a obtenir la xarxa seran invàlids perquè no s'estarà entrenant per identificar la planta sinó el terra. A partir d'aquí es decideix abandonar els intents d'entrenar la xarxa amb aquest lot i se centra a aprendre més sobre l'aprenentatge profund fins que no arriba el segon lot. Un punt clau que es va poder extreure d'aquestes proves amb el primer lot és que cal que les dades siguin tan clares com sigui possible si s'està treballant amb un conjunt de dades tan petit i difícil d'aconseguir. Encara que no siguin imatges que es poden trobar en una situació normal, sempre es poden aplicar tècniques d'augment de dades posteriorment per aconseguir un conjunt més complet. D'aquí es va treure la decisió de posar pautes per obtenir les dades del segon lot, i d'aquesta manera assegurar un conjunt de dades de qualitat.

Una vegada es reben les imatges del segon lot, es pot començar a desenvolupar un model acceptable. Per començar, les imatges s'han de retallar per aconseguir que la planta ocupi el màxim de la imatge. En general, com més es pugui retallar la imatge, millor resolució tindrà un cop model l'hagi redimensionat. Aquest redimensionament es fa per motius d'optimització, perquè encara que la imatge perdi resolució, si el model utilitzes resolucions massa altes, el consum de memòria seria enorme. A més, si la imatge és en color, s'ha de tindre en compte que aquesta imatge en són realment tres d'iguals representades amb cada color del model RGB. Xarxes avançades com AlexNet o GoogLeNet utilitzen entrades de 227x227 i 224x224 respectivament.

En primer lloc, es comença a treballar amb AlexNet utilitzant la transferència de coneixement. D'aquesta manera es poden treure idees de com ha de ser l'aprenentatge correcte d'un model i de quins valors aproximats han de tindre els paràmetres d'entrenament. El primer problema que es va trobar a l'hora d'entrenar la xarxa és que li costava molt aprendre. A causa d'un problema que es descobreix més endavant, les gràfiques d'entrenament i error mostren uns resultats poc usuals: la precisió es queda estancada abans de la desena època al 21% exactament, i l'error al 2%. En alguns casos la xarxa aconseguia millorar, però els resultats que s'obtenien després de 40 èpoques eren molt irregulars i dolents. Després d'analitzar les activacions de la primera capa de la xarxa, es descobreix que quan el model redimensiona les imatges, aquestes perden moltes característiques i queden pràcticament

Figura 6: Diferència entre la primera iteració de retalls i la segona en una imatge d'un exemplar de *Crataegus Monoguina*.

irreconeixibles. Els contorns de la planta perden la forma i fa molt difícil reconèixer cada espècie per la seva morfologia. A conseqüència, les imatges s'han de tornar a retallar a una dimensió més petita que abans per evitar aquesta pèrdua de detalls a la forma de la planta.

Amb les noves imatges acabades de retallar, es torna a provar d'entrenar AlexNet i veure'n el rendiment. Però tot i els canvis, l'error que s'ha explicat abans en què la precisió i l'error queden encallats continua apareixent. Com que no s'hi troba cap explicació, es decideix provar un nou model que es presenta en un article, i que pretén fer identificacions d'entre 12 espècies de cultius i males herbes [11]. Aquest model consta d'una xarxa bastant senzilla però gran, amb 5 capes convolucionals i cada una d'elles amb un nombre de filtres que incrementa exponencialment. És tan gran que l'ordinador utilitzat es queda sense memòria i l'entrenament s'atura. Per continuar, es decideix reduir el model traient filtres a cada capa de la xarxa però mantenint l'ordre exponencial. En ser una xarxa amb tant volum, l'entrenament té una llarga duració amb temps superiors als 50 minuts d'entrenament tot i haver-ne reduït la dimensió. Això si, els resultats de l'entrenament són bastant bons, amb la precisió per sobre del 85% i l'error al voltant de l'1%. Cal destacar que es pot apreciar una mica d'excés d'adequació (overfitting) a la majoria d'èpoques de l'entrenament.

Més endavant, analitzant la matriu de confusió després de l'entrenament, es pot veure com la majoria d'imatges que s'han classificat erròniament formen part de les classes (espècies) amb menys dades disponibles. Se sospita que això és degut a un problema força comú al camp de l'aprenentatge profund anomenat desequilibri de dades (*data imbalance*) de les classes. Aquest problema fa referència típicament a la distribució pobra de dades entre les classes, és a dir, quan unes quantes classes del problema tenen moltes més dades (en aquest cas imatges) que les altres. Mirant el nostre conjunt de dades, es pot veure que les espècies *Prunus Avium* i *Sorbus Aria* tenen més d'un centenar d'imatges cada una,

mentre que les altres amb prou feines arriben a la cinquantena. Així doncs, per solucionar aquest problema s'han d'afegir imatges a les classes que en necessiten, ja sigui aconseguint noves imatges, traient-ne de les que ja en tenen o utilitzant alguna tècnica d'augment de dades. En el nostre cas, com que no es podien aconseguir noves dades, es va decidir extreure imatges a partir de les que ja es tenien. Ja que abans de fer ús de les imatges s'havien retallat a mides bastant petites, hi havia l'opció de fer nous retalls a les mateixes imatges i extreure'n parts diferents.

Una vegada s'havien retallat les noves imatges es va tornar a provar d'entrenar la xarxa importada de l'article per veure si els resultats milloraven. A partir dels resultats es conclou que tot i haver-hi una millora, no es pot considerar un gran canvi perquè la precisió de la xarxa i l'error no han canviat gaire. Les millores que es poden observar amb més claredat a la matriu de confusió, on es pot observar que les imatges mal classificades ja no estan concentrades a certes classes, sinó a imatges que són difícils de classificar en si.

Amb aquests canvis, es decideix tornar a provar d'entrenar AlexNet i veure si es pot trobar una manera de desencallar l'aprenentatge de la xarxa. Provant nous paràmetres d'entrenament de la xarxa es descobreix que el ritme inicial d'entrenament era massa alt i si en reduïm el valor, la xarxa es deix d'encallar en valors de precisió i error baixos. Inicialment, el valor del ritme d'aprenentatge inicial estava posat per defecte a 0,01 i després de varies proves, es determina que el millor ritme d'aprenentatge és de 0,0001. Aquest canvi de comportament és degut a l'algoritme d'optimització que utilitzen aquestes xarxes anomenat descens del gradient. Essencialment, el valor d'aquest ritme d'aprenentatge decideix com de ràpid el model canvia als errors estimats cada vegada que s'actualitzen els pesos. Un ritme massa baix i el procés d'entrenament es pot encallar, i un ritme massa alt i el procés d'entrenament pot ser inestable i amb resultats dolents[9]. En el cas que s'ha trobat, el ritme d'aprenentatge era tant alt que el procés semblava encallat però simplement ja no podia millorar més.

Comparat amb la xarxa importada, la precisió que s'obté és millor i la diferència de temps d'entrenament és abismal: AlexNet sempre acaba per sota dels 2 minuts mentre que la xarxa importada acaba per damunt dels cinquanta. Així mateix, també s'estava jugant amb els valors de la mida de lot amb els que s'entrena el model. A base de proves, es va poder observar que com més gran és la mida del lot, més estable és l'aprenentatge durant el procés d'entrenament. El primer pensament és posar una mida gran, però resulta que el model es pot beneficiar de dimensions de lot baixes. La causa és l'algoritme de descens de gradient un altre cop. Quan la mida de lot és alta, l'estimació de l'error compren més dades i serà més precís, però a la vegada, la freqüència d'actualització als pesos és menor perquè el model fa més prediccions abans de calcular l'estimació de l'error. Per això, quan els lots són més aviat petits, el model modifica els pesos més freqüentment encara que amb menys precisió i alhora, el procés d'entrenament serà més ràpid i sorollós, que pot resultar en un model més robust[8]. Així doncs, es decideix treballar amb lots més petits per tractar d'aconseguir millors resultats.

A la vegada, encara hi ha mals comportaments que es poden observar durant l'entrenament. Sembla que hi ha fluctuacions brusques de la precisió i l'error en etapes avançades de l'entrenament, una mica d'excés d'adequació i resultats molt diferents entre execucions.

Primer, per evitar els resultats tan volàtils, es posa una llavor estàtica generadora d'aleatoris per tenir la mínima variabilitat a les dades d'entrada de la xarxa. La idea és poder obtenir els mateixos resultats cada vegada que s'entrena, però es descobreix que és impossible, ja que durant la primera iteració d'un entrenament la xarxa assigna uns pesos aleatoris a cada capa que no van regits per la llavor d'aleatoris fixada al MatLab. Així i tot, gràcies a la llavor fixa els resultats són més estables i es pot observar millor els efectes dels canvis que es fan. D'aquesta manera, per valorar com és d'efectiu un canvi es decideix fer 10 entrenaments de la mateixa instància del model i fer la mitjana dels resultats d'aquests entrenaments.

En segon lloc, per reduir l'excés d'adequació a les dades d'entrenament es prova d'utilitzar una tècnica de regularització, l'L2. La regularització són totes aquelles tècniques que ajuden a l'algoritme a generalitzar millor el problema, en altres paraules, reduir el "overfitting". En concret, la regularització L2 o decadència de pes ("weight decay") modifica l'algoritme d'aprenentatge per fer que els pesos siguin reduïts per un factor constant cada actualització dels mateixos[14]. Més tard, després de realitzar les proves necessàries es conclou que aquest tipus de regularització no aporta millores en absolut. Cal remarcar que l'arquitectura d'AlexNet ja utilitza tècniques de regularització contra l'excés d'adequació, que són dues capes d'abandonament ("dropout") al 50%. Per acabar, cal dir que es creu que la forma ideal de contrarestar aquest excés d'adequació seria obtenint més dades.

Finalment, per adreçar les fluctuacions de la precisió i error en etapes avançades es fa ús d'alguns paràmetres que permeten reduir el ritme d'aprenentatge un cop han passat certes èpoques. Més concretament, es redueix el ritme d'aprenentatge un 50% cada 6 èpoques que passen. Gràcies a aquesta tècnica, podem utilitzar mides de lot més baixes i reduir l'impacte que tenen a l'aprenentatge. Ara, el procés d'aprenentatge és més suau i no hi ha desnivells bruscs a les etapes finals. A més, com que la xarxa arriba ràpidament a la precisió màxima, es pot reduir el nombre d'èpoques d'entrenament per aconseguir un model encara més eficient.

Després de realitzar diverses proves per acabar de determinar els millors valors per a cada paràmetre, es treu la llavor estàtica d'aleatoris per fer les últimes proves. El model és prou estable i els resultats són satisfactoris, i després de fer les proves s'obté una mitjana de precisió al grup de validació de 94% i un 96% al grup prova. La millor versió del model que es va entrenar va ser capaç de classificar correctament el 98,74% del tot el conjunt de dades.

Per acabar, es considera que la millora del model a partir d'aquest punt es fa molt mínima, ja que totes les iteracions provades han estat fallant en les mateixes classes. Aquestes classes són les que morfològicament manquen més trets identificatius i per tant, s'intueix que el problema es podria solucionar més efectivament tractant de millorar les imatges que es tenen d'aquestes classes problemàtiques. Una altra manera d'obtenir resultats millors seria utilitzant tècniques no convencionals com l'aturada prematura de la xarxa en algun punt on la precisió del grup validació sigui màxim.

5.2 Model casolà

Un cop acabat l'ajustament del model utilitzant l'arquitectura d'AlexNet es comença a treballar en l'arquitectura personalitzada i des de zero. Gràcies al desenvolupament que s'ha fet del model anterior, es podrà centrar més atenció en el disseny de l'arquitectura de la xarxa perquè es coneixen millor els paràmetres d'entrenament més importants i com utilitzar-los. Inicialment, es comença amb una xarxa simple de dues capes convolucionals i una capa de classificació, amb poques neurones i distribuïdes prioritzant la primera capa convolucional per aconseguir que la xarxa extregui més característiques a cada imatge durant l'inici. La mida de l'entrada es mantindrà constant a 200x200 al llarg del desenvolupament, ja que sembla ser una mida que funciona per altres models com GoogLeNet o AlexNet. Els paràmetres d'entrenament també es mantenen en valors per defecte per simplicitat, i a mesura que es vagin fent proves s'aniran canviant.

Després de provar el model amb aquesta configuració ja es poden veure dos problemes, el primer és un salt brusc només començar l'entrenament a la gràfica de pèrdua, i el segon és un excés d'adequació molt gran només començar el procés. Cal destacar que el model aconsegueix una precisió del 100% en el grup de dades d'entrenament, cosa que sorprèn tenint en compte que la xarxa només consta de dues capes convolucionals. Això pot significar que si s'aconsegueix millorar la generalització de les dades, es pot arribar a un model òptim ràpidament.

Per una banda, el pic a la gràfica de pèrdua és un senyal que el ritme d'aprenentatge és massa alt. A conseqüència, els canvis que es fan amb l'algorisme d'optimització són massa grans i el model mai no arriba a trobar un mínim a la funció de descens del gradient. Per solucionar aquest fenomen, podem anar fent proves reduint el ritme d'aprenentatge fins que el pic a la funció de pèrdua desapareix. En el nostre cas, es troba que el ritme mínim que compleix aquest objectiu és al voltant de 0,00005. Sembla un valor molt baix comparat amb el que s'ha estat treballant fins ara, però durant l'entrenament la precisió segueix arribant al 100%.

Per altra banda, l'excés d'adequació pot ser degut a diverses raons, però en el nostre cas es pot sospitar que és a causa del baix nombre de dades que es disposen com ja s'ha vist anteriorment. L'excés d'adequació és probablement el problema més comú que apareix quan es treballa amb aprenentatge profund, i les solucions que s'apliquen normalment sempre són les mateixes. La més ràpida és probablement fer el model més petit. Essencialment, el que significa l'excés d'adequació és que el model aprèn les dades d'entrenament tant al detall que les acaba "memoritzant" i impactant negativament el procés, i per tant, si fem el model més petit, també perdrà capacitat d'aprenentatge. Però com que el nostre model ja és petit per defecte, no és una solució que es pugui utilitzar. L'altra solució és aconseguir més dades perquè el model tingui més característiques per aprendre. Encara que ja s'ha comentat que imatges noves no se'n poden aconseguir, sí que es pot provar d'aplicar tècniques d'augment de dades a les imatges que ja es tenen. Es prova d'aplicar tècniques d'alteració de la brillantor o filtres per fer les imatges més sorolloses, però s'evita l'ús de rotacions i translacions per evitar perdre informació en les imatges, ja que tenen poques dimensions després de retallar-les. Després de fer algunes proves es determina

que el model està obtenint pitjors resultats i l'excés d'adequació no s'ha reduït, per tant, l'augment de dades està perjudicant el model. Se sospita que el model s'adapta tan de pressa al conjunt de dades que l'augment de dades no té cap efecte en absolut.

Hi ha tècniques que ja s'han vist anteriorment i que també ens poden ajudar a reduir l'accés d'adequació, com ara la regularització L2. Es pot paral·lelitzar aquest canvi amb modificacions a l'arquitectura de la xarxa, ja que la regularització L2 s'especifica als paràmetres d'entrenament i no requereix canvis estructurals. S'ha de tindre en compte que les tècniques de regularització no són efectives en tots els casos i sempre requereixen diverses iteracions fins a trobar el valor del paràmetre òptim. Després de fer les proves necessàries es pot observar que aquesta tècnica de regularització no té efectes evidents a simple vista, ja que els resultats que s'obtenen són molt similars a les proves anteriors a aquest canvi.

Depenent del model, hi poden haver tècniques de regularització més efectives que d'altres. Una tècnica molt eficaç contra l'excés d'adequació i utilitzada per AlexNet és l'abandonament ("dropout"), on un nombre especificat de neurones escollides aleatòriament són desactivades per fer la resta de neurones més robustes. El que fa tan efectiva aquesta tècnica és que els canvis que genera tenen un impacte directe en el funcionament de les neurones i la manera que té el model d'entrenar, i per conseqüència, la tècnica és menys dependent de la forma que pot tindre la xarxa en qüestió. Per aplicar l'abandonament correctament, s'han de realitzar diverses proves per determinar a quines capes de la xarxa resulta més eficaç aplicar-la, i quin percentatge de neurones es volen abandonar en cada instància. Durant les proves amb dues capes convolucionals es va poder observar que els millors resultats s'aconseguien aplicant l'abandonament només un cop a la capa final en un 50% o més. Com que només s'aplica la tècnica una vegada, el nombre de neurones per abandonar serà molt gran per compensar. Més endavant, es van fer més proves amb un nombre major de capes i amb més abandonament dins la xarxa, però com que en fer més gran el model també s'està augmentat la capacitat d'aquest, l'excés d'adequació augmenta i contraresta qualsevol millora aconseguida amb la tècnica d'abandonament. Així doncs, es conclou que els millors resultats s'obtenen utilitzant un model de dues capes convolucionals i abandonament a la capa final d'entre un 50% i un 60%. Els resultats no són gaire estables entre execucions, però es pot obtenir al voltant d'un 80% de precisió en el grup de dades d'entrenament en les millors execucions. De totes maneres, encara hi ha un gran nivell d'excés d'adequació perquè la precisió del grup de dades d'entrenament arriba al màxim i amb un valor de pèrdua nul.

A partir d'aquí, després de provar sense èxit totes les tècniques de regularització possibles, s'han de buscar mètodes alternatius menys convencionals que ens ajudin a reduir l'excés d'adequació. A banda dels canvis d'arquitectura que es van fent, també es prova de canviar la funció d'activació. Normalment s'utilitza una funció rectificadora (ReLU), però en aquest cas es provarà de canviar-la per una funció sigmoide, que és la seva predecessora i en alguns casos pot ajudar a reduir l'excés d'adequació. Aquestes funcions s'utilitzen per obtenir el resultat dels nodes, representant-los entre els valors de –1 i 1 (o 0 i 1 en alguns casos). El problema que hi pot haver amb la funció ReLU que s'ha utilitzat fins ara és que només representa els resultats dels nodes amb valors d'entre 0 i 1 i transforma

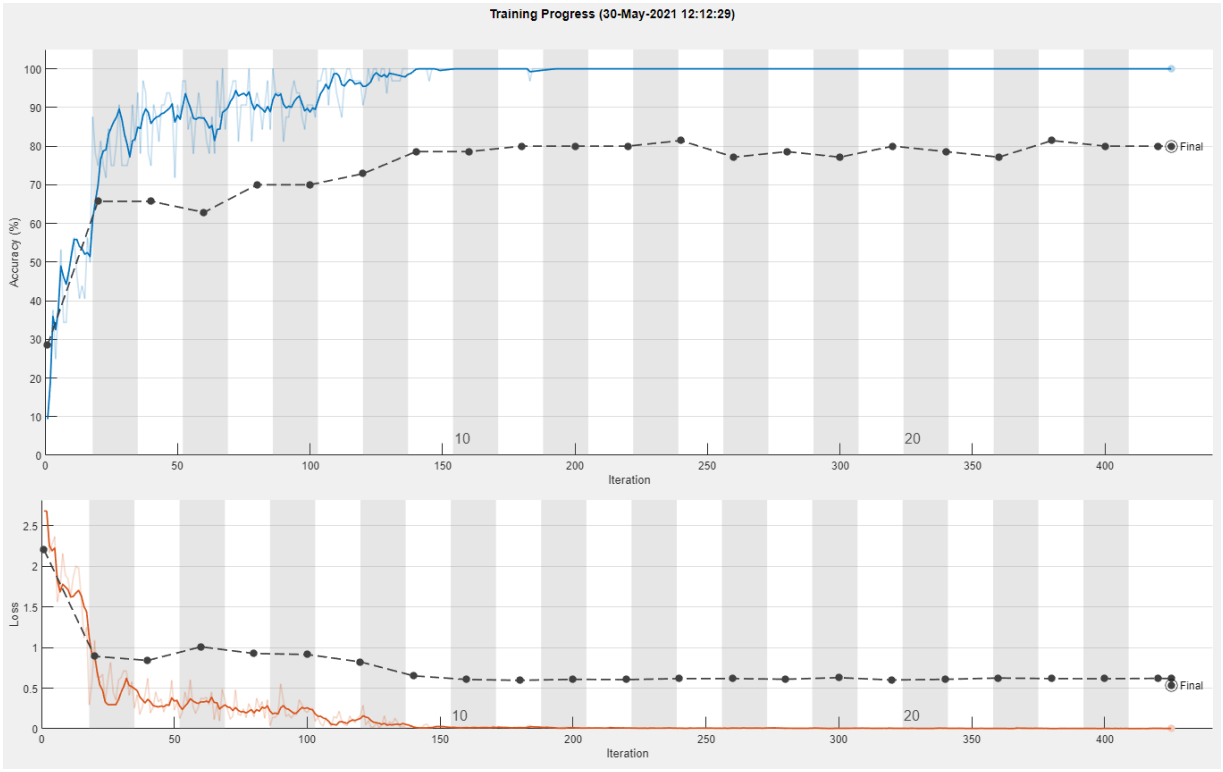


Figura 7: Gràfica d'un entrenament del model casolà on s'hi observa un gran nivell d'excés d'adequació

automàticament qualsevol resultat negatiu a 0. Aquest fet pot afectar negativament el procés d'entrenament fent que el model no s'adeqüi a les dades. Les funcions sigmoides també tenen els seus desavantatges i està demostrat que en general sempre és millor utilitzar la funció ReLU. En el nostre cas, es van provar diferents combinacions de la xarxa intercanviant les funcions d'activació per comparar i trobar la configuració òptima. Després de realitzar les proves necessàries es va poder comprovar que, tot i haver-hi diferències de precisió, les funcions d'activació sigmoide no ajudaven a reduir l'accés d'adequació suficientment per a justificar la pèrdua de precisió. D'entre les execucions que es van fer, la millor precisió s'obtenia amb dues funcions ReLU, mentre que per obtenir els resultats amb el menor excés d'adequació, la configuració que mostrava millors resultats era posant una funció ReLU després de la primera capa convolucional i una sigmoide després de la segona.

Com que les opcions ja resultaven limitades després de descartar tots els mètodes avaluats, es va provar de buscar una solució a partir d'implementacions ja funcionals i veure si se'n podia extreure alguna idea. Analitzant el codi del model mencionat anteriorment[11] es pot veure que consisteix d'una xarxa molt senzilla compresa de 5 capes convolucionals seguides de les operacions d'agrupació adients, una funció d'activació ReLU en cada cas i capes de normalització. A simple vista, la xarxa no té cap mecanisme de regularització ni augment de dades per a evitar l'excés d'adequació. És cert que els autors d'aquest model disposaven d'un gran volum de dades d'entrenament i els problemes amb l'excés d'adequació que es devien trobar a l'hora de dissenyar el model devien ser menors, però

per norma general sempre s'aplica algun tipus de mètode de regularització per reduir la probabilitat de què aparegui el problema. El que sí que es pot veure al model és una capa d'agrupació global, o "global pooling". És molt poc comuna basat en els models que s'ha vist fins ara, però és l'única manera que un model d'aquestes dimensions és capaç d'evitar l'excés d'adequació per complet. Aquesta capa executa una operació d'agrupació dissenyada per substituir les capes connexes en una CNN clàssica, o per fer reduccions agressives d'alguna característica de la imatge. Igual que una funció d'agrupació estàndard, l'operació que realitza consisteix a reduir la sortida de les capes convolucionals agafant el valor mitjà o màxima d'una secció de la matriu, però en el cas del "pooling" global l'operació s'aplica a tota la matriu resultant a la vegada[5]. D'aquesta manera s'aconsegueix destacar l'activació o presència d'una característica més forta a l'entrada de la capa. L'avantatge que ofereix aquesta operació en el nostre cas és que actua com a regularitzador estructural, i pràcticament fa desaparèixer l'excés d'adequació.

Després de diverses proves amb aquesta nova capa, es podia veure que la capacitat del model ja no havia de ser limitada, ja que l'excés d'adequació era inexistent. Es va adoptar un sistema de xarxa similar al que es presenta a l'article que s'ha comentat anteriorment, amb quatre capes convolucionals i cada una seguida de les capes de normalització i ReLU adequades. També es va poder observar que canviant el paràmetre "stride" de les capes d'agrupació, es podia aconseguir una millora substancial del temps d'entrenament sense impactar negativament els resultats del model. Només canviant aquest paràmetre en dues capes d'agrupació es va aconseguir una reducció del voltant del 30% del rendiment. A més, conservant un parell de passos d'abandonament al final del model ajudem amb la regularització dels resultats i aconseguim una petita millora més en el temps d'entrenament, ja que com a resultat s'hauran d'entrenar menys neurones. Pel que fa als paràmetres d'entrenament, s'han hagut d'ajustar lleument en comparació al model creat amb AlexNet. Entre d'altres, aquest model requereix més èpoques d'entrenament per arribar a resultats bons, al voltant d'unes 100. Sembla que en utilitzar la capa d'agrupació global, també es fa el procés d'entrenament més lent; la gràfica de la millora de la precisió ja no té un increment logarítmic exagerat i es pot observar que el model es pot beneficiar notablement d'entrenaments prolongats després d'haver passat l'etapa d'increment de precisió més elevat.

Així doncs, el model acaba oferint una precisió al voltant del 88% al grup de validació i un 85% al grup prova, amb un temps lleugerament per sobre dels 5 minuts. Es pot considerar un resultat prou bo si es compara amb l'altre model presentat i tenint en compte el nombre limitat de dades de les quals es disposava i els nombrosos problemes que ha ocasionat aquest fet.

6 Avaluació

Al llarg d'aquest treball han aparegut diversos problemes que han obligat a adaptar els objectius plantejats en començar. No obstant això, els resultats que s'han obtingut poden ser considerats satisfactoris. Pot significar un bon punt de partida per a aplicacions futures

amb el mateix objectiu o es pot continuar el desenvolupament d'aquests models resultants tenint en compte les millores que poden rebre alguns aspectes que es comentaran més endavant. És per això que en aquest apartat, a banda de discutir el resultat final de les estructures de cada model dissenyat, es vol comparar resultats amb altres aplicacions que es podrien considerar "estat de l'art" i acabar per mencionar les possibles millores que podrien rebre els models a partir d'ara.

6.1 Model a partir d'AlexNet

```
opt = trainingOptions("adam", ...
    "Plots","training-progress", ...
    "ValidationData",valIm, ...
    "MaxEpochs",40, ...
    "MiniBatchSize",32, ...
    "InitialLearnRate",0.0001, ...
    "ValidationFrequency",40, ...
    "LearnRateSchedule","piecewise", ...
    "LearnRateDropPeriod",6, ...
    "LearnRateDropFactor",0.5);
```

Figura 8: Paràmetres d'entrenament del model desenvolupat a partir d'AlexNet i transferencia d'aprenentatge.

Tal com s'ha vist a l'apartat de disseny i implementació, el model que utilitza transferència d'aprenentatge amb AlexNet ha estat el que ha donat el millor resultat. Amb un 96% de precisió mitjana al grup de prova, aquest model compleix amb els objectius proposats i amb bona nota. D'entre les 10 execucions que es van realitzar per aconseguir una mitjana fiable, la precisió del grup prova va arribar a un màxim del 98,61%, mentre que al grup de validació va arribar al 98,57% en una execució diferent. Reflexionant a partir dels apartats anteriors, es pot atribuir els bons resultats d'aquest model en part a l'entrenament previ que disposa la xarxa AlexNet. Gràcies a l'abundància i diversitat de dades de les quals disposa ImageNet, les primeres capes de la xarxa AlexNet ja tenen filtres ben entrenats i fa que la xarxa trobi i classifiqui característiques de cada imatge amb més facilitat. Si s'hagués entrenat la xarxa des de zero utilitzant el nostre conjunt de dades, el model probablement hauria donat problemes d'excés d'adequació similars al model creat des de zero. D'altra banda, si s'analitza el gràfic del procés d'entrenament que hi ha a la Figura 9, es poden extreure alguns detalls més de com funciona aquest.

Tal com es pot veure, el procés d'entrenament arriba a les 40 èpoques i arriba a les 680 iteracions. Es tracta d'un gràfic amb creixement logarítmic molt pronunciat, que es pot atribuir a la decaiguda del ritme d'aprenentatge que està especificat als paràmetres d'entrenament. A mesura que el procés avança, els canvis en la precisió i l'error es fan més suaus i atenuats. Una bona pràctica hauria estat reduir el nombre d'èpoques d'entrenament, ja que la majoria d'aquestes no tenen efecte en el resultat final i es podrien obviar. Per contra, es podrien haver mantingut les èpoques, però reduir la decaiguda del

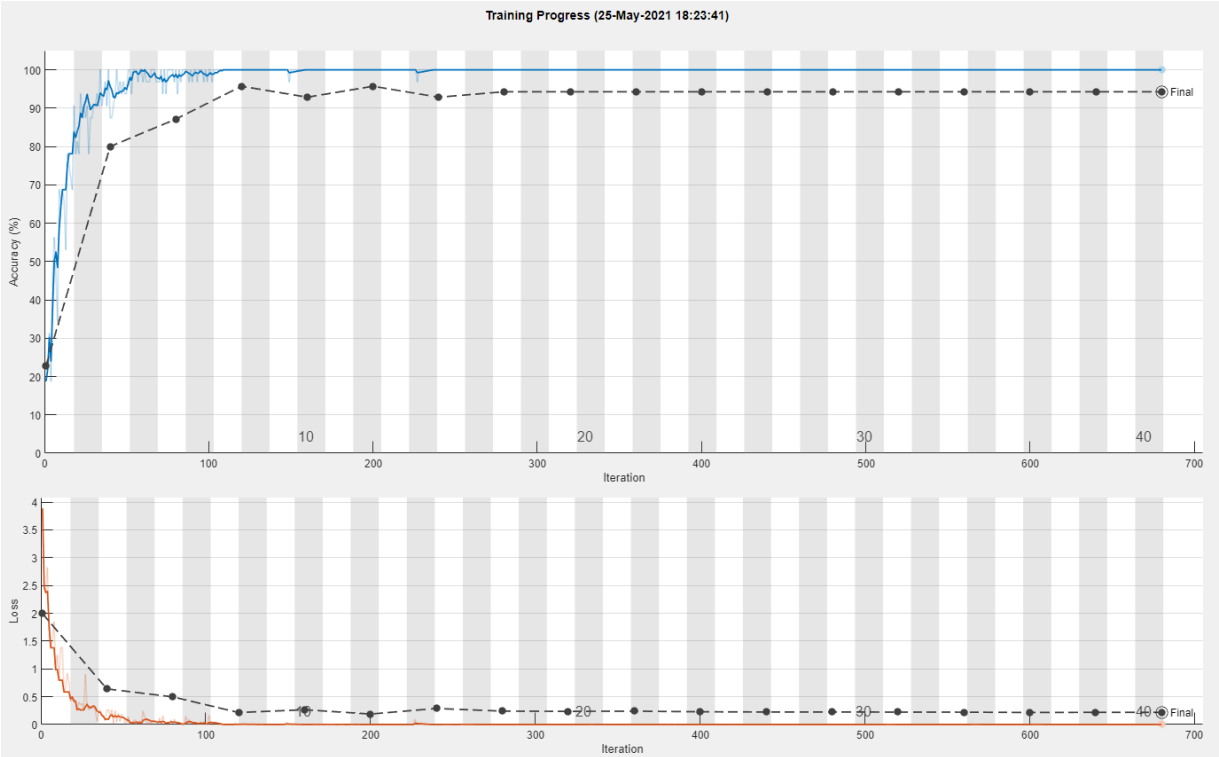


Figura 9: Gràfics representant la precisió i l'error durant l'entrenament del model amb AlexNet.

ritme d'aprenentatge per fer que aquest procés fos més sorollós però amb més possibilitats d'aconseguir màxims en els resultats. A més, tècniques com les parades prematures de l'entrenament podrien tenir més èxit amb aquestes condicions. Sense anar més lluny, si s'observa el gràfic al voltant de l'inici de la vuitena època, es pot veure com la precisió arriba a un punt màxim i l'error té el mateix valor que al final. Si el procés s'hagués aturat en aquest instant, el model tindria millor precisió al grup de validació i probablement al grup de prova, a més de ser molt més curt i eficient. De totes maneres, les aturades prematures acostumen a ser menys efectives en models on la corba és tan pronunciada i la gràfica d'error del model pot fluctuar ràpidament.

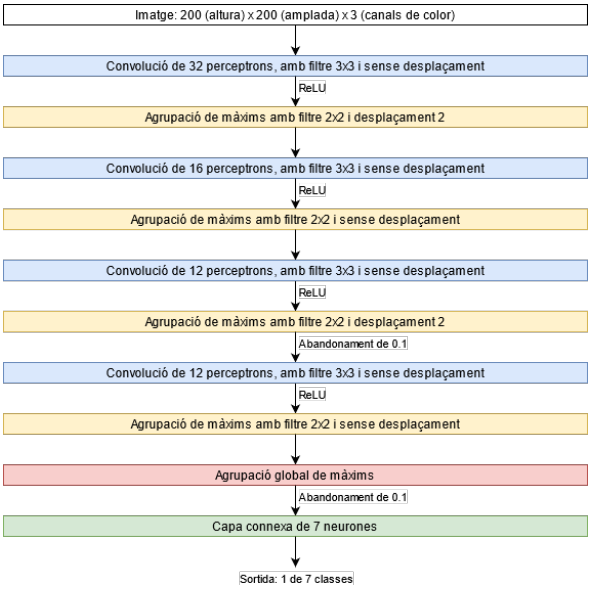
Una altra característica a destacar que es pot observar al gràfic és la presència d'excés d'adequació. Tot i que apareix amb menys intensitat que en el cas del model fet des de zero, es pot veure com la corba de precisió i d'error arriben als seus màxims ràpidament en els conjunts de dades d'entrenament, mentre que el conjunt de validació sempre queda per sota. En aquest cas és un fenomen difícil de solucionar perquè, a més de no poder aconseguir un conjunt de dades més gran, tampoc es pot modificar l'estructura de la xarxa a causa de la transferència d'aprenentatge. L'estructura d'AlexNet conté els filtres entrenats i si es canviés alguna d'aquestes capes, probablement afectaria catastròficament els resultats. Al cap i a la fi, es tracta d'una xarxa molt preparada i, si apareix algun problema com ara l'excés d'adequació, es pot intuir que la solució dependrà del conjunt de dades o els paràmetres d'entrenament. D'aquest últim es va provar d'utilitzar la tècnica de

regularització L2, però els resultats van ser més aviat negatius i per això es va deixar a banda.

6.2 Model casolà

Comparat amb el model anterior utilitzant AlexNet, és cert que el model casolà que es planteja en segon lloc pot semblar una mica mancant en quant a resultats. Durant les diferents execucions que es van dur a terme, el model va aconseguir resultats amb valors al voltant del 87.5% de precisió en el conjunt de dades de validació i gairebé del 85% al grup de proves. Tenint en compte que es tracta d'un model que s'ha entrenat únicament amb el conjunt de dades del que es disposava, es pot afirmar que hi ha satisfacció amb els resultats, i més si es té en compte les complicacions que s'han trobat fins a arribar a una solució acceptable.

En primer lloc, analitzarem l'estructura final del model i els paràmetres d'entrenament que s'han utilitzat. Aquests es poden trobar a la Figura 10.




```
opt = trainingOptions("adam", ...
    "Plots","training-progress", ...
    "ValidationData",valIm, ...
    "MaxEpochs",100, ...
    "ValidationFrequency",20, ...
    "MiniBatchSize",32, ...
    "InitialLearnRate", 0.001, ...
    "ExecutionEnvironment","gpu");
```

Figura 10: Il·lustracions representant l'estructura del model casolà i els seus paràmetres d'entrenament.

Aparentment, l'estructura és una mica similar a la d'AlexNet, en aquest cas amb una capa d'entrada d'imatges de mida similar i 4 capes convolucionals seguides de capes normalitzadores, capes ReLU i d'agrupació. Al final es poden trobar algunes capes d'abandonament i les capes connexes. En el cas d'aquest model, s'hi pot trobar una capa d'agrupació global que ajuda a restringir l'excés d'adequació, però també fa el procés d'entrenament deu vegades més llarg. Per això, s'utilitzen paràmetres que ajuden a agilitzar el procés, com ara el valor del desplaçament del filtre durant les capes d'agrupació, o el nombre baix de neurones en cada instància de capa convolucional. Sobre aquest últim detall, s'ha de comentar que el nombre decreixent de neurones en cada capa té com a

objectiu mantenir el temps d'entrenament baix, i a la vegada forçar el model a identificar més característiques en capes inicials. Realment, el model no necessita més neurones perquè en augmentar la capacitat d'aquest, també s'incentiva l'adequació en excés a les dades d'entrenament. Objectivament hi ha una infinitat de configuracions a les quals es poden arribar modificant paràmetres de l'estructura, i alguna d'elles probablement funcionarà millor que el model que es proposa en aquest treball. Però el cert és que cada iteració del model requereix diverses execucions per validar-ne el resultat. Atès que seria una tasca molt llarga i poc efectiva, hi ha alguns paràmetres com ara la mida dels filtres de les capes d'agrupació i neurones que s'han modificat poc al llarg del desenvolupament i s'han intentat mantenir sempre a valors considerats estàndards dins la comunitat de l'aprenentatge profund.

Per altra banda, als paràmetres d'entrenament s'utilitzen valors pràcticament estàndards fora del nombre d'èpoques durant l'entrenament, que s'ha de mantenir en valors alts a causa del ritme d'aprenentatge prolongat que té el model després d'afegir la capa d'agrupació global. Aquest fet es pot veure reflectit a la imatge que es troba a continuació, on s'hi ha representat el gràfic del procés d'entrenament del model.

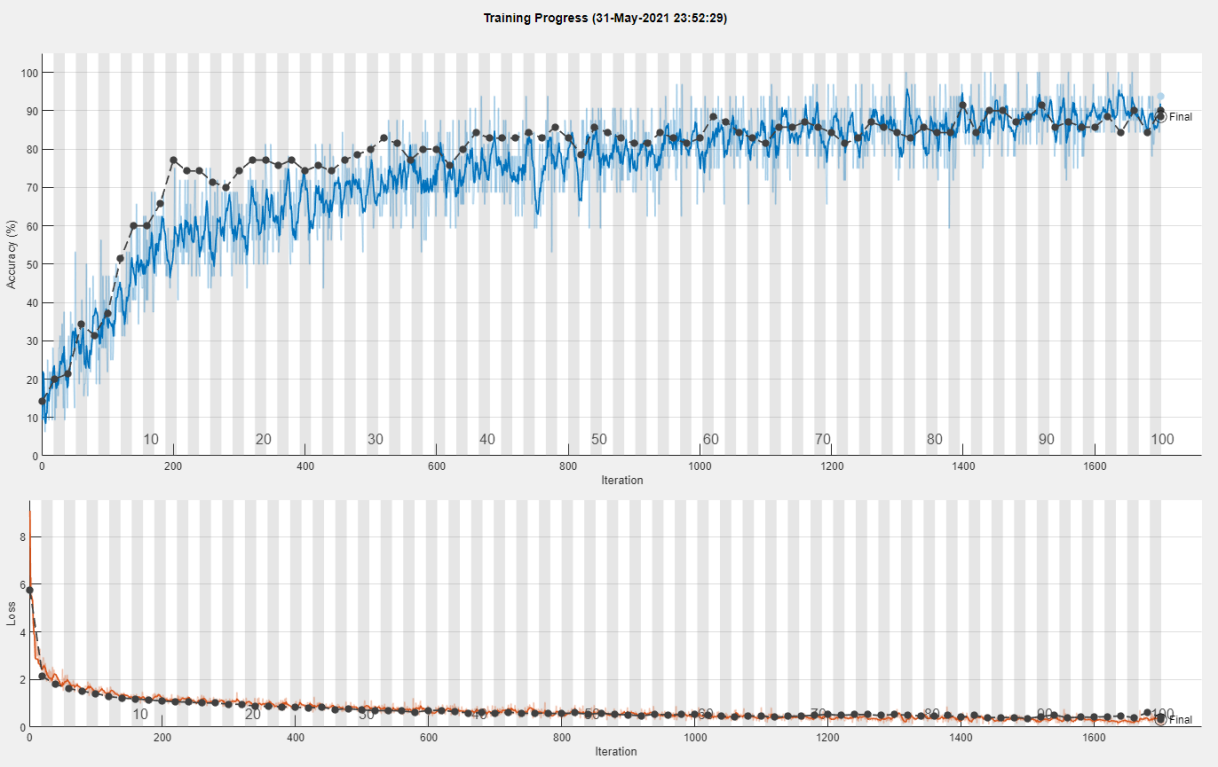


Figura 11: Gràfics representant la precisió i l'error durant l'entrenament del model casolà.

En aquest gràfic es pot observar com l'excés d'adequació pràcticament ha desaparegut del procés d'entrenament del model. En altres iteracions del mateix que es poden veure a l'apartat d'annexos, s'observa com els valors de precisió per al grup d'entrenament arribaven al seu màxim immediatament, mentre que els valors pel grup de validació amb prou feines arribava a valors propers al 70%. Seguint amb el gràfic, si es compara amb el

del model anterior, es veu com la corba d'aprenentatge no és tan pronunciada, però és molt més sorollosa. En aquesta situació, el model es beneficia bastant d'aquestes irregularitats i tècniques com l'aturada prematura, que s'ha discutit anteriorment, poden aportar millores substancials als resultats del model. En aquest mateix gràfic, al voltant de l'època 90 la corba sobrepassa el 90% de precisió. Si en aquest moment s'hagués aturat l'entrenament, els resultats potser serien millor.

Altrament, es pot observar que la corba de l'error és molt més estàndard, i arriba a valors comparables al model anterior al final del procés. En general, aquest model es beneficia bastant d'entrenaments prolongats, i no arriba als seus màxims fins al voltant de l'època 90. És per això que a partir d'aquesta època ja no és eficient seguir l'entrenament, ja que presumiblement no hi haurà més millores a la precisió.

6.3 Millores

Per començar, al llarg del treball s'ha pogut comprovar que un dels punts més importants a tenir en compte a l'hora de treballar amb aprenentatge profund és la importància que té un bon conjunt de dades. Es podria considerar imperatiu assegurar un bon conjunt de dades abans de començar un projecte en aquest camp. Per norma general, el conjunt de dades ideal és aquell que s'ajusta al màxim a la realitat i té la mida necessària d'acord amb la complexitat del problema que s'afronta (majoritàriament es busca tindre coma mínim 10 vegades més dades que classes a classificar[13][16]). En el nostre cas, durant les etapes inicials del desenvolupament dels models, el problema més gran que es va trobar va estar relacionat amb l'obtenció del conjunt de dades. En obtenir les dades d'una font aliena, va ser complicat aconseguir-ne un gran nombre i menys garantir-ne la qualitat. Més endavant es va canviar el mètode d'obtenció d'imatges per assegurar la bona qualitat d'aquestes, però a la vegada significava que el conjunt de dades no seria fidel a la realitat, és a dir, que el mètode ja no era l'utilitzat en pràctica. Per tant, una millora immediata consistiria a aconseguir un nou conjunt de dades més gran, i sobretot, trobar una manera d'aconseguir imatges més realista i ajustada a com ho faria un operari en un cas normal. Un major nombre d'imatges al conjunt de dades implica majors grups d'entrenament i validació, però també grups de proves més grans, cosa que pot portar a resultats més exactes i a una reducció dels entrenaments amb resultats aïllats (*outliers*). Tot això també depèn i està limitat per l'equipament i les circumstàncies en les quals els operaris es trobin, i s'hauria d'establir unes pautes per assegurar que les imatges es prenen en les condicions estipulades. Un exemple de tècnica per aconseguir millors imatges consistiria a utilitzar una càmera amb mode retrat, que molts dels mòbils d'avui en dia ja tenen. Aquest mode permet aconseguir fotos de l'objecte que s'està focalitzant de manera nítida, mentre que la resta d'imatge apareix difuminada. D'aquesta manera, la planta es podria distingir amb més claredat del fons.

Per altra banda, si es poguessin aconseguir imatges de qualitat de les plantes plantades, s'hauria de comprovar si el conjunt de dades actual és útil per entrenar els models. Si els models són capaços de classificar les imatges introduïdes satisfactòriament, això estalviaria molts canvis a les estructures dels mateixos models i s'evitaria haver de crear un conjunt

de dades de nou. És més, una situació similar no és impossible, ja que els models tenen els filtres de les capes entrenats per detectar les característiques de les plantes en qüestió.

En segon lloc, es podria seguir el desenvolupament dels dos models. En el cas del model utilitzant transferència d'aprenentatge, s'hauria de provar de modificar els paràmetres d'entrenament, o directament canviar la xarxa d'AlexNet per altres alternatives similars, com ara les xarxes VGG-16Net o ResNet[2]. La primera xarxa és molt similar a AlexNet, amb la diferència principal que aquesta presumeix de filtres molt més petits a les primeres capes, però més capes convolucionals en total per compensar, i d'aquesta manera manté el nombre de paràmetres del model més baix i el temps d'entrenament es redueix. La segona xarxa anomenada ResNet fa ús de connexions drecera, que permeten al model saltar-se capes de la xarxa en cas que no siguin necessàries perquè ja es pot classificar amb èxit la dada analitzada. Així s'aconsegueix reduir el temps d'entrenament i la importància d'algunes capes. Algunes d'aquestes xarxes poden ser massa pesades i amb massa capacitat per a un problema més aviat petit com el nostre, però amb la varietat de xarxes que hi ha actualment, és bastant segur poder trobar una alternativa a AlexNet que suposi una millora. D'altra banda, el model "casolà" es podria polir en més aspectes, ja que l'estructura sencera de la xarxa es podria revisar i provar d'alterar en alguns aspectes. Començant pel nombre de filtres de cada capa convolucional i la mida d'aquests, seguint per altres canvis estructurals evitant l'ús de la capa d'agrupació global, i finalment jugant amb els valors dels paràmetres d'entrenament. De totes maneres, com que es tracta d'un model entrenat des de zero, segurament el millor pas a seguir abans de res seria assegurar un millor conjunt de dades.

Per últim, un concepte que probablement no es consideraria viable en la nostra situació és el de l'execució en paral·lel. En cas que el model es tornés massa gran i el temps d'entrenament s'allargués, es podria provar de paral·lelitzar les capes de convolució de la mateixa manera que ho fa AlexNet. Un desavantatge és que per aplicar aquesta tècnica es necessitarien múltiples targetes gràfiques.

6.4 Pensaments finals

En qualsevol cas, s'ha demostrat que amb les dades que es disposaven i els models dissenyats, la transferència d'aprenentatge és el camí més viable per desenvolupar una aplicació de classificació de flora en la situació en qüestió. És un mètode que ha estat superior tant en valors de precisió com en valors de temps, i si es compara amb aplicacions de classificació de flora considerades estat de l'art, es demostra que el model que s'ha desenvolupat té resultats admirables. Per posar en perspectiva aquest fet s'utilitzarà Pl@ntNet per identificar exemplars. Una peculiaritat que tenen aplicacions similars a aquesta (i que es podria considerar una millora a fer als nostres models), és que quan s'introdueix una planta per identificar, aquesta retorna un llistat de plantes suposadament similars i acompanyades d'un percentatge que representa amb la certesa que es fa la identificació de cada una. D'aquesta manera, l'usuari també pot jutjar quina de les plantes de la llista és la correcta basant-se en els percentatges i el seu criteri. Així doncs, es provaran 10 imatges diferents escollides aleatòriament per cada espècie, i s'anotarà la precisió mitjana de la identificació,

Espècies	Precisió (%)	Primer a la llista	No és a la llista
Crataegus Monogyna	17	6	4
Fraxinus Excelsior	24	8	0
Prunus Avium	7	3	4
Prunus Spinosa	33	8	0
Quercus Faginea	5	1	7
Quercus Pyrenaica	0	0	10
Sorbus Aria	30	5	2

Taula 1: Taula per comprar la precisió de les identificacions entre Pl@antNet i el model amb AlexNet.

quantes vegades la primera planta suggerida és la correcta i quantes vegades la planta no és llistada. Els resultats es poden veure a la Taula 1.

Es pot veure que l'aplicació té resultats variants depenent de l'espècie. En el cas de la *Fraxinus Excelsior* o la *Prunus Spinosa*, l'aplicació aconsegueix resultats bons; en tots els casos és capaç d'identificar la planta i en un 80% dels casos és el suggeriment amb més certesa. Per contra, altres espècies com les dues variants de *Quercus* són molt més difícils de classificar per la xarxa i en més del 70% dels casos Pl@ntNet no aconsegueix identificar la planta correctament i no apareix a la llista de suggeriments. Aquesta diferència de resultats és atribuïble a l'aspecte i característiques que tenen les plantes en fases tan inicials dels eu creixement. Les espècies que són més fàcils de classificar per l'aplicació acostumen a tindre fulles i trets clars que les identifiquen, i que són presents quan la planta ha madurat. Les plantes com les *Quercus*, quan estan tan poc desenvolupades encara no els hi han crescut fulles i els troncs són prims i amb pocs brots. També destaca la mitjana de precisió, que és bastant baixa en tots els casos. Això informa que les identificacions que fa l'aplicació són amb poca certesa, i si l'usuari desconeix l'espècie de la planta, pot resultar en identificacions errònies.

Encara que Pl@ntNet disposi d'una base de dades tan gran, ha de diferenciar d'entre un gran nombre d'espècies amb moltes similituds entre elles. Si es compara amb el model desenvolupat en aquest treball que s'ha entrenat únicament amb les 7 espècies de la taula, les diferències són clares. Durant les últimes proves s'ha pogut comprovar aquest fet perquè en alguns casos l'aplicació identificava algunes de les espècies com a una variant molt similar de la mateixa família. A més, com ja s'ha comentat anteriorment, la base de dades de Pl@ntNet creix cada dia perquè recull totes les imatges que els usuaris introdueixen per a ser identificades. Pl@ntNet o aplicacions similars estan preparades per reconèixer plantes que es troben dins de les seves bases de dades i com que en una situació normal no es troben arbres tan poc desenvolupats com els que tenim en aquest problema, és normal que en la majoria dels casos l'aplicació tingui dificultats.

7 Conclusions

Per acabar, val la pena dir que aquest treball ha estat una mica complicat en alguns aspectes. Inicialment, aquest treball va començar com a un problema a solucionar que havia estat proposat per una de les meves tutores a partir d'un contacte amb l'organització. No hi havia una única direcció a l'hora de buscar una solució i finalment es va acabar escollint utilitzar l'aprenentatge profund. Una de les dificultats al començament era que els meus coneixements sobre l'aprenentatge profund eren nuls. Durant la carrera no vaig cursar cap assignatura que tractés d'aquest tema. D'altra banda, tampoc havia llegit ni practicat res sobre aquest tipus d'aprenentatge automàtic perquè sempre m'havia semblat un concepte fora del meu abast i interès. A base de cursos, lectures i el mateix procés de prova i error durant el desenvolupament dels models, vaig aconseguir els coneixements necessaris, i a conseqüència, els resultats del treball. De fet, aquest va ser un dels objectius secundaris que em vaig marcar en començar el treball i puc considerar que l'he pogut complir amb èxit.

L'altre objectiu secundari també el podem donar per assolit. Durant el temps que vaig treballar amb el primer lot d'imatges que vam rebre, vaig poder comprovar que les dades d'entrenament han de tindre un cert nivell de qualitat. Es pot dir el mateix del segon lot, on les imatges tenien bona qualitat però a durant l'entrenament acabaven no donant mals resultats a causa de la gran resolució que tenien. En definitiva, penso que he desenvolupat prou criteri al voltant de les dades d'entrenament d'una xarxa per identificar imatges aptes per formar part d'un conjunt.

Una altra de les dificultats que m'he trobat al llarg del treball té a veure amb l'entorn de desenvolupament que és MatLab. És una eina fàcil d'utilitzar i ja n'era una mica familiar perquè s'havia utilitzat durant la carrera. El problema és que la gran majoria de la comunitat activa que treballa amb aprenentatge profund utilitza Keras, i els fòrums i pàgines web on s'explicaven solucions a problemes sempre feien referència a aquest entorn. No va ser un factor decisiu al llarg del treball, però sí que va provocar algun entrebanc. Per altra banda, un avantatge que té MatLab per sobre de Keras és que ofereix una documentació molt detallada que ajuda a un usuari nou a entendre els paràmetres de cada funció, o el funcionament de cada capa d'una xarxa. Sobretot, va resultar molt útil a l'hora de buscar els valors òptims als paràmetres d'entrenament de les xarxes.

Per últim, tenim l'objectiu principal del projecte que es va plantejar inicialment. Tal com es comenta en apartats anteriors, aquest objectiu va haver de ser canviat perquè no es disposava de les dades necessàries per poder complir-lo, per tant, considerem que el treball no ha assolit les expectatives en aquest aspecte. De totes maneres, tenint en compte els resultats obtinguts, podem donar com a exitós l'objectiu principal nou que es va proposar.

7.1 Treballs futurs

A continuació, exposo possibles rutes que podria seguir aquest treball i els resultats obtinguts en cas que es volgués expandir. Cal remarcar que en aquestes propostes interpreto que

s'ha aconseguit un conjunt de dades nou, amb les característiques comentades a l'apartat de millores. Ho faig així perquè considero que aquesta millora és un pas primordial abans de fer cap ampliació.

- Afegir més espècies per identificar, possiblement les 21 que hi ha a les plantacions.
- Completar una aplicació que pugui ser utilitzada a les plantacions. Combinant idees inicials per al projecte com ara un sistema d'ubicació GPS dins la plantació, es podria crear una aplicació amb diverses funcions per assistir a l'usuari.
- Augmentar la complexitat de les identificacions. Aconseguint més imatges de les plantes en estacions de l'any diferents, es podria provar de crear un model capaç de detectar les espècies de la plantació independentment de l'ambient en què es trobin.

8 Bibliografia

Referències

[1] Geoffrey E. Hinton Alex Krizhevsky Ilya Sutskever. "ImageNet Classification with Deep Convolutional Neural Networks". In: (2012). DOI: https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[2] Aqeel Anwar. *Difference between AlexNet, VGGNet, ResNet, and Inception*. URL: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96. (visitat: 26.8.2021).

[3] Arc. *Convolutional Neural Network*. URL: https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05. (visitat: 25.6.2021).

[4] Jason Brownlee. *A Gentle Introduction to Batch Normalization for Deep Neural Networks*. URL: https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/. (visitat: 28.8.2021).

[5] Jason Brownlee. *A Gentle Introduction to Pooling Layers for Convolutional Neural Networks*. URL: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/. (visitat: 08.6.2021).

[6] Jason Brownlee. *A Gentle Introduction to Transfer Learning for Deep Learning*. URL: https://machinelearningmastery.com/transfer-learning-for-deep-learning/. (visitat: 25.8.2021).

[7] Jason Brownlee. *How Do Convolutional Layers Work in Deep Learning Neural Networks?* URL: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. (visitat: 20.6.2021).

[8] Jason Brownlee. *How to Control the Stability of Training Neural Networks With the Batch Size*. URL: https://machinelearningmastery.com/how-to-control-the-speed-and-stability-of-training-neural-networks-with-gradient-descent-batch-size/. (visitat: 06.6.2021).

[9] Jason Brownlee. *Understand the Impact of Learning Rate on Neural Network Performance*. URL: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/. (visitat: 04.6.2021).

[10] Kevin Casey. *How to explain deep learning in plain English*. URL: https://enterprisersproject.com/article/2019/7/deep-learning-explained-plain-english?page=0%2C1. (visitat: 25.8.2021).

[11] Heba A. Elnemr. "Convolutional Neural Network Architecture for Plant Seedling Classification". In: *International Journal of Advanced Computer Science and Applications* 10.8 (2019). DOI: https://thesai.org/Downloads/Volume10No8/Paper_41-Convolutional_Neural_Network_Architecture.pdf. (visitat: 06.5.2021).

[12] Keith D. Foote. *A Brief History of Deep Learning*. URL: https://www.dataversity.net/brief-history-deep-learning/. (visitat: 02.6.2021).

[13] Alexandre Gonfalonieri. *How to Build A Data Set For Your Machine Learning Project*. URL: https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac. (visitat: 02.6.2021).

[14] Aaron Courville Ian Goodfellow Yoshua Bengio. *Regularization for Deep Learning*. URL: https://www.deeplearningbook.org/contents/regularization.html. (visitat: 16.6.2021).

[15] Pl@ntNet. URL: https://identify.plantnet.org/. (visitat: 12.6.2021).

[16] Ryan Sevey. *How Much Data is Needed to Train a (Good) Model?* URL: https://www.datarobot.com/blog/how-much-data-is-needed-to-train-a-good-model/. (visitat: 02.6.2021).

[17] Devashish Sood. *Backpropagation concept explained in 5 levels of difficulty*. URL: https://medium.com/coinmonks/backpropagation-concept-explained-in-5-levels-of-difficulty-8b220a939db5. (visitat: 28.8.2021).

[18] Google Trends. *Deep Learning*. URL: https://trends.google.com/trends/. (visitat: 02.6.2021).

[19] Wikipedia. *Keras*. URL: https://ca.wikipedia.org/wiki/Keras. (visitat: 04.6.2021).

A Appendix

A.1 Estructura de les xarxes

A continuació es mostren els continguts de les xarxes tal i com estaven a l'entorn de desenvolupament.

1	'data'	Image Input	227×227×3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11×11×3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3×3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5×5×48 convolutions with stride [1 1] and padding [2 2 2 2]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3×3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3×3×256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3×3×192 convolutions with stride [1 1] and padding [1 1 1 1]
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3×3×192 convolutions with stride [1 1] and padding [1 1 1 1]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3×3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	'fc8'	Fully Connected	1000 fully connected layer
24	'prob'	Softmax	softmax
25	'output'	Classification Output	crossentropyex with 'tench' and 999 other classes

Figura 12: Estructura de la xarxa AlexNet a l'entorn de desenvolupament.

1	''	Image Input	200×200×3 images with 'zerocenter' normalization
2	'conv1'	Convolution	32 3×3 convolutions with stride [1 1] and padding [0 0 0 0]
3	''	Batch Normalization	Batch normalization
4	''	ReLU	ReLU
5	''	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Convolution	16 3×3 convolutions with stride [1 1] and padding [0 0 0 0]
7	''	Batch Normalization	Batch normalization
8	''	ReLU	ReLU
9	''	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
10	'conv3'	Convolution	12 3×3 convolutions with stride [1 1] and padding [0 0 0 0]
11	''	Batch Normalization	Batch normalization
12	''	ReLU	ReLU
13	''	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]
14	''	Dropout	10% dropout
15	'conv4'	Convolution	12 3×3 convolutions with stride [1 1] and padding [0 0 0 0]
16	''	Batch Normalization	Batch normalization
17	''	ReLU	ReLU
18	''	Max Pooling	2×2 max pooling with stride [1 1] and padding [0 0 0 0]
19	''	Global Max Pooling	Global max pooling
20	''	Dropout	10% dropout
21	''	Fully Connected	7 fully connected layer
22	''	Softmax	softmax
23	''	Classification Output	crossentropyex

Figura 13: Estructura de la xarxa casolana a l'entorn de desenvolupament.

A.2 Exemples d'imatges

En aquest apartat s'exposen diversos exemples d'imatges de cada espècie utilitzades durant els entrenaments dels models presentats en aquest treball. A més, les imatges que hi ha a continuació són algunes de les quals es van emprar per fer la comparació entre Pl@ntNet i el model amb AlexNet com a xarxa.

Figura 14: Crataegus Monogyna

Figura 15: Fraxinus Excelcior

Figura 16: Prunus Avium

Figura 17: Prunus Spinosa

Figura 18: Quercus Faguinea

Figura 19: Quercus Pirenaica

Figura 20: Sorbus Aria