UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

AUTOMATIC FRUIT CLASSIFICATION USING DEEP LEARNING

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Joaquim Fèlix Martínez Artigot

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATIONS SYSTEMS ENGINEERING

Advisors: Verónica Vilaplana Besler, Josep Ramon Morros Rubió

Barcelona, May 2018

Abstract

To achieve greater efficiency at harvesting tasks, the mechanization of such task is unavoidable. Apart from the mechanical aspects, the harvesting systems needs software that can locate the fruit to be harvested.

The use of machine learning and deep learning techniques to achieve such software was studied in this thesis. The results showed that an accuracy similar to other studies is feasible with a limited number of training samples using deep learning techniques.

From this thesis we conclude that the mechanization of the harvesting labour is possible, at least from the software point of view, while the crop estimation application may need some more work before being feasible.

Resum

Per assolir una major eficiència en les tasques de recol·lecció, la mecanització d'aquesta tasca és indispensable. Apart dels aspectes mecànics, el sistema de recol·lecció requereix d'un software capaç de localitzar la fruita per ser recol·lectada.

En aquesta tesi s'ha estudiat l'ús de tècniques de machine learning i deep learning per arribar a aquest software. Els resultats mostren que una precisió similar a la d'altres estudis és possible amb un nombre limitat de exemples d'entrenament fent servir tècniques de deep learning.

Podem extreure com a conclusió que la mecanització de la tasca de recol·lecció és factible, com a mínim des del punt de vista de software, mentre que l'estimació de collites necessitaria més treball previ per ser factible.

Resumen

Para conseguir una mayor eficiencia en las tareas de recolección, la mecanización de dicha tarea es indispensable. Aparte del aspecto mecánico, el sistema de recolección necesita software capaz de localizar la fruta a recolectar.

El uso de técnicas de machine learning y deep learning para lograr dicho software ha sido estudiado en esta tesis. Los resultados muestran que una precisión similar a la de otros estudios es factible con un número limitado de muestras de entrenamiento.

De esta tesis podemos concluir que la mecanización de la tarea de recolección es posible desde el punto de vida de software, mientras que la estimación de cosechas necesitaría más trabajo previo antes de ser posible.

Vull dedicar aquest treball als meus companys i amics de la feina, que sense saber-ho, m'han ajudat a passar alguns moments difícils en els últims mesos. També el vull dedicar a la meva mare, Maria Rosa; sense ella dubto que en el dia d'avui em trobés aquí, i per descomptat que aquest treball no hagués estat possible. I per últim li vull dedicar a la meva àvia; encara que ja no sigui amb nosaltres sempre em cuidarà des del cel.

Agraïments

Primer de tot agrair tant a la Verónica com al Ramón haver-me donat la possibilitat de treballar en aquest projecte, així com tota la informació que m'han facilitat de diferents maneres, ja sigui referenciant-me a articles i/o cursos o llibres per aprofundir en la matèria. També destacar la seva absoluta disponibilitat, fins i tot en horari no lectiu.

També destacar l'ajuda de l'Albert Gil Moreno, enginyer de software i sistemes del departament de teoria del senyal i comunicacions, que sempre m'ha solucionat els dubtes i errors tècnics que anava trobant a mesura que avançava amb el projecte.

Per últim reconèixer també la tasca realitzada per la Universitat de Lleida a l'hora de recopilar les imatges per poder-les fer servir en el projecte.

Història de revisió i registre d'aprovació

Revisió	Data	Objectiu
0	30/04/2018	Creació del document
1	11/05/2018	Revisió del document

LLISTA DE DISTRIBUCIÓ DE DOCUMENTS

Nom i cognoms	e-mail
Joaquim Fèlix Martínez Artigot	joaquimfelixmartinez@gmail.com
Verónica Vilaplana Besler	veronica.vilaplana@upc.edu
Josep Ramon Morros Rubió	ramon.morros@upc.edu

Elaborat per:		Revisat i aprovat per:	
Data	11/05/2018	Data	11/05/2018
Nom i cognoms	Joaquim Fèlix Martínez Artigot	Nom i cognoms	Verónica Vilaplana Besler Josep Ramon Morros Rubió
Càrrec	Autor del projecte	Càrrec	Supervisors del projecte

Índex

Llista de Figures

Llista de Taules:

1. Introducció

Tot i que el desenvolupament de varies tecnologies ha facilitat el progrés en diferents sectors, el sector de l'agricultura és un que no s'ha beneficiat d'aquest progrés. La tasca de recol·lecció de fruita és encara una feina realitzada per mà d'obra que per varies raons es veu forçada a treballar moltes hores en una feina molt exigent pel cos, el que comporta lesions físiques

És per això que la mecanització de la feina de recol·lecció en el sector agrícola és necessària. Per a dur a terme aquest canvi es requereix d'una part robòtica, és a dir, un braç mecànic preparat per a la recol·lecció i un suport mòbil per moure l'estructura d'arbre en arbre. Però també requereix d'una part de software que sigui capaç de localitzar les fruites. Aquest software permetria altres aplicacions dins del sector, com l'estimació de collites, que aportaria més capacitat de previsió al sector.

En aquesta tesi es pretén estudiar aquesta part de software, més concretament, fent ús d'algoritmes de machine learning, a partir de dades proporcionades per la Universitat de Lleida. El software utilitzat es pot trobar a la referència de Ren et al. (1). També es fa servir una eina d'anotació elaborada en Python per Bargoti i Underwood (2).

Els requeriments principals del projecte consisteixen en la detecció automàtica de pomes utilitzant tècniques de machine learning, l'ús d'una interfície visual per l'anotació de les imatges d'entrenament i per a la comprovació de les deteccions automàtiques en les imatges de validació. Per últim, la prova i avaluació dels sistemes dissenyats.

El projecte consta de tres especificacions bàsiques, fer servir Python com el llenguatge principal de programació, treballar amb les imatges proporcionades per la Universitat de Lleida (UdL) i l'ús de diferents tècniques d'entrenament.

1.1. Pla de treball

1.1.1. Estructura integrativa del treball

PT1: Proposta del projecte i pla de treball

PT2: Recerca d'informació

PT3: Desenvolupament del software

PT4: Revisió crítica

PT5: Anàlisi i resultats de l'avaluació

PT6: Memòria final

PT7: Presentació del TFG

1.1.2. Paquets, tasques i fites del treball

Paquets del treball:

<table>
<tr><td>Projecte: Proposta del projecte i pla de treball</td><td colspan="2">PT ref: (PT1)</td></tr>
<tr><td>Principal constituent: Documentació</td><td colspan="2">Full 1 de 7</td></tr>
<tr><td rowspan="2">Descripció breu:
Elaboració de la proposta del projecte i organització del projecte</td><td colspan="2">Data prevista d’inici: 07/09/2017
Data prevista de finalització: 06/10/2017</td></tr>
<tr><td colspan="2">Esdeveniment inicial: T1
Esdeveniment final: T3</td></tr>
<tr><td>Tasca interna T1: Descripció del projecte
Tasca interna T2: Pla del desenvolupament del projecte
Tasca interna T3: Revisió del document i aprovació</td><td>Lliurables:
Proposta de projecte i pla de treball.pdf</td><td>Data:06/10/2017</td></tr>
</table>

<table>
<tr><td>Projecte: Recerca d’informació</td><td colspan="2">PT ref: (PT2)</td></tr>
<tr><td>Principal constituent: Recerca</td><td colspan="2">Full 2 de 7</td></tr>
<tr><td rowspan="2">Descripció breu:
Adquisició d’experiència en el camp d’estudi</td><td colspan="2">Data prevista d’inici : 07/09/2017
Data prevista de finalització : 20/10/2017</td></tr>
<tr><td colspan="2">Esdeveniment inicial: T1
Esdeveniment final: T3</td></tr>
<tr><td>Tasca interna T1: Familiarització amb el machine learning
Tasca interna T2: Estudi de l’estat actual de les tècniques de classificació de fruites
Tasca interna T3: Recerca de software de classificació ja implementat</td><td>Lliurables:</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Projecte: Desenvolupament de software</td><td colspan="2">PT ref: (PT3)</td></tr>
<tr><td>Constituent principal: Software</td><td colspan="2">Full 3 de 7</td></tr>
<tr><td rowspan="2">Descripció breu:
Implementació i prova de diferents algorismes per a classificar fruita en imatges</td><td colspan="2">Data prevista d’inici : 20/10/2017
Data prevista de finalització : 25/04/2017</td></tr>
<tr><td colspan="2">Esdeveniment inicial: T1
Esdeveniment final: T3</td></tr>
<tr><td>Tasca interna T1: Disseny de diferents algorismes de machine learning per a localitzar i classificar fruites en imatges
Tasca interna T2: Etiquetat manual d’imatges per a entrenar els esmentats algorismes
Tasca interna T3: Implementació o ús de la interfície visual amb l’objectiu d’anotació.</td><td>Lliurables:</td><td>Dates:</td></tr>
</table>

<table>
<tr><td>Projecte: Revisió crítica</td><td colspan="2">PT ref: (PT4)</td></tr>
<tr><td>Constituent principal: Documentació</td><td colspan="2">Full 4 de 7</td></tr>
<tr><td rowspan="2">Descripció breu:
Revisió del progrés fins a aquesta data i elaboració de la revisió crítica del projecte</td><td colspan="2">Data prevista d’inici : 20/11/2017
Data prevista de finalització : 01/12/2017</td></tr>
<tr><td colspan="2">Esdeveniment inicial: T1
Esdeveniment final: T3</td></tr>
<tr><td>Tasca interna T1: Discussió del progrés
Tasca interna T2: Revisió del pla de treball
Tasca interna T3: Revisió i aprovació del document</td><td>Lliurables:
Revisió crítica</td><td>Dates:
01/12/2017</td></tr>
</table>

Projecte: Avaluació del test i dels resultats	PT ref: (PT5)	
Constituent principal: Software i simulació	Full 5 de 7	
Descripció breu: Avaluació del resultats obtinguts a l'estudi per a determinar-ne el rendiment. Comparació amb els resultats sobre l'estat actual	Data prevista d'inici : 15/12/2017 Data prevista de finalització : 30/04/2018	
	Esdeveniment inicial: T1 Esdeveniment final: T2	
Tasca interna T1: Avaluar el rendiment dels sistemes de classificació Tasca interna T2: Avaluar la millora observada a través del procés de machine learning	Lliurables:	Dates:

Projecte: Memòria final	PT ref: (PT6)	
Constituent principal: Documentació	Full 6 de 7	
Descripció breu: Documentació final on es descriu la totalitat del projecte	Data prevista d'inici : 30/04/2018 Data prevista de finalització : 11/05/2018	
	Esdeveniment inicial: T1 Esdeveniment final: T2	
Tasca interna T1: Descripció del projecte realitzat i els seus resultats Tasca interna T2: Revisió i aprovació del document	Lliurables: Revisió final	Dates: 11/05/2018

<table>
<tr><td>Projecte: Presentació del TFG</td><td colspan="2">PT ref: (PT7)</td></tr>
<tr><td>Constituent principal: (per exemple, prototipus de hardware, simulació, software)</td><td colspan="2">Full 7 de 7</td></tr>
<tr><td rowspan="2">Descripció breu:
Preparar la presentació oral del projecte.</td><td colspan="2">Data prevista d’inici : 10/05/2018
Data prevista de finalització : 22/05/2018</td></tr>
<tr><td colspan="2">Esdeveniment inicial: T1
Esdeveniment final: T3</td></tr>
<tr><td>Tasca interna T1: Disseny de diapositives
Tasca interna T2: Assaig de la presentació
Tasca interna T3: Presentació</td><td>Lliurables:
Presentació.pdf</td><td>Dates:
22/05/2018</td></tr>
</table>

Fites

PT#	Tasca#	Títol breu	Fita/lliurable	Data (setmana)
1	1	Descripció del projecte	Descripció	1
1	2	Pla de desenvolupament del projecte	Esborrany	2,3
1	3	Revisió i aprovació del document	Proposta i pla de treball del projecte.pdf	4
2	1	Familiarització amb el machine learning	Documentació	2,3,4
2	2	Estudi sobre l'estat actual	Documentació	2,3
2	3	Recerca de software de classificació	Documentació	2,3,4
3	1	Disseny d'algorismes de machine learning	Software	6-31
3	2	Etiquetat d'imatges per a l'entrenament	Entrenament	10
3	3	Implementació de la interfície visual	Software	9,10
4	1	Discussió del progrés	Esborrany	11,12
4	2	Revisió del pla de treball	Revisió	12
4	3	Revisió i aprovació del document	Revisió crítica	13
5	1	Avaluació del rendiment	Prova del software	14-32
5	2	Avaluació de la millora del learning	Prova del software	16-32
6	1	Descripció del projecte	Esborrany	32,33
6	2	Revisió i aprovació del document	Revisió final	33
7	1	Esborrany de les diapositives	Esborrany	33
7	2	Assaig de la presentació	Notes de l'assaig	34
7	3	Presentació	Presentació.pdf	35

1.1.3. Pla cronològic (diagrama de Gantt)

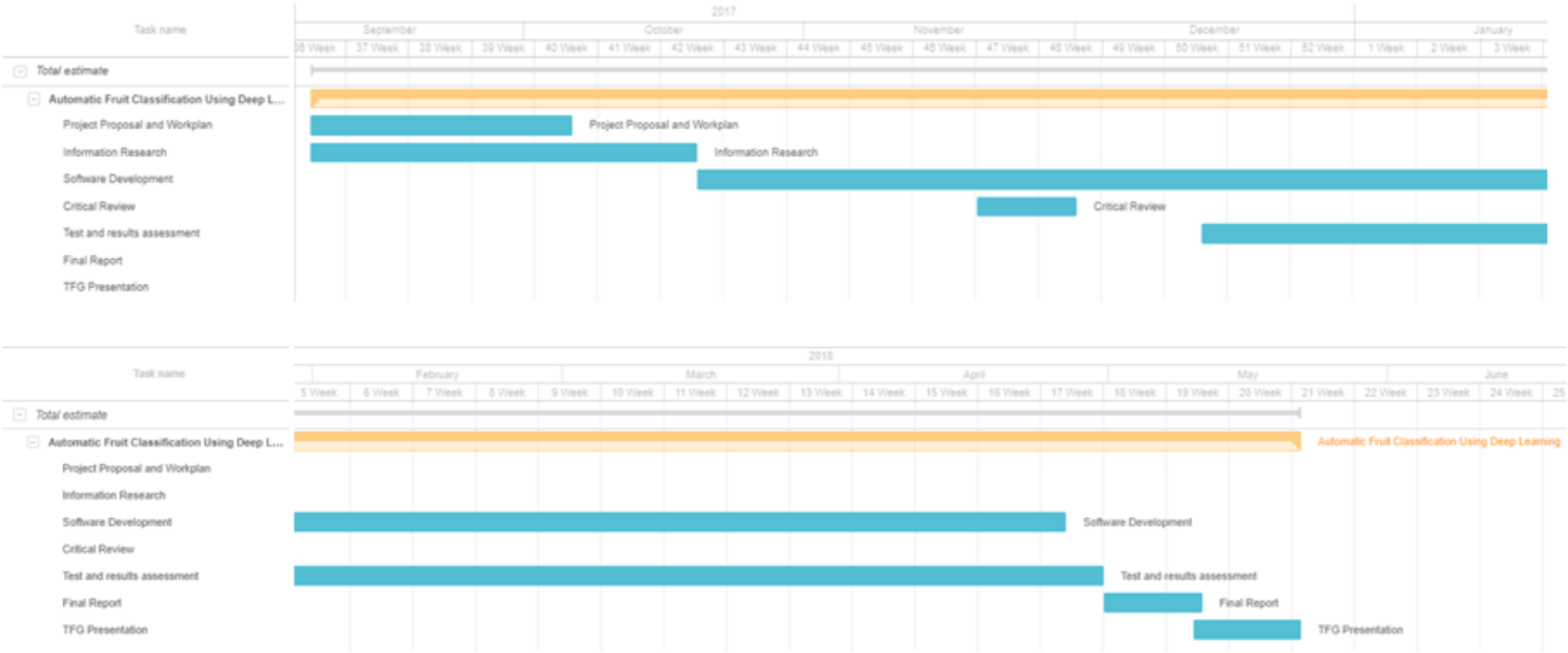

Figura 1. Diagrama de Gantt definitiu

Es denota una incidència durant el procés de Software Development i de Test and result assesment . Familiaritzar-se amb la xarxa va costar i això va alentir el progrés d'aquestes tasques.

2. Estat actual de la tecnologia emprada o aplicada en aquesta tesi

2.1. Detecció d'objectes

Els avenços més actuals en el camp de la detecció d'objectes venen de l'ús o bé de Region Proposal Methods o bé de xarxes neuronals convolucionals (CNN). Els últims desenvolupaments (Fast R-CNN) mostren que el cost computacional es pot disminuït considerablement aconseguint resultats pràcticament a temps real, si no es té en compte el temps emprat en la proposició de regions. Això es degut a que la proposició de regions es computa en CPU mentre que CNN aprofita recursos GPU. Per maximitzar l'ús de recursos, Ren et al. (1) proposen una solució en la que el cost de computació de les regions és pràcticament nul donat el cost computacional de la xarxa de detecció, que anomenen Region Proposal Networks (RPN). Afegint algunes capes convolucionals a la CNN es pot crear una RPN que permeti calcular proposicions de regió amb menor cost computacional. A aquesta nova arquitectura, els autors l'anomenen Faster R-CNN.

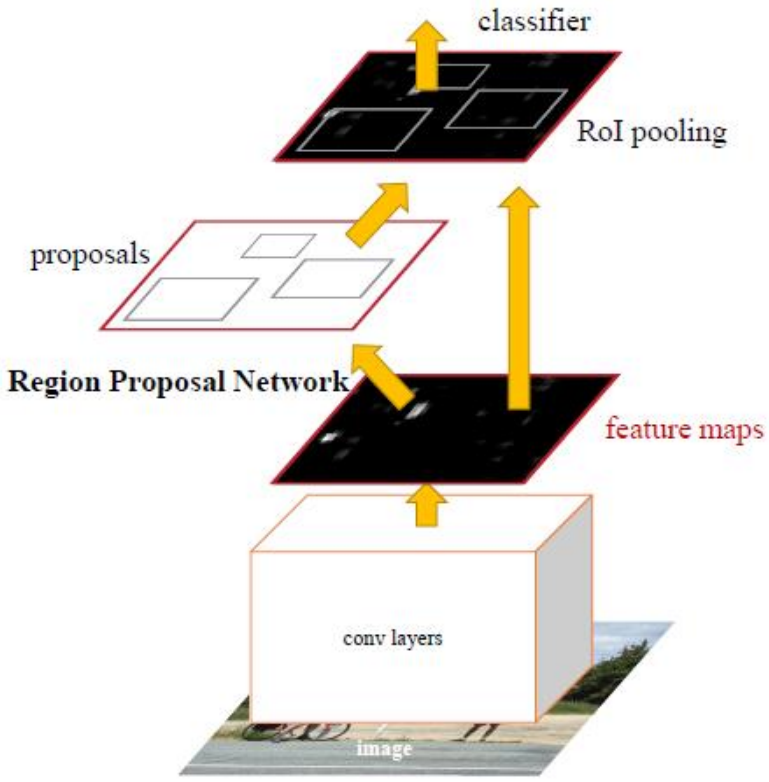

Figura 2. Faster R-CNN

Altres detectors d'objectes, que es basen en tècniques de machine learning clàssiques, són el detector Viola-Jones o les llibreries dlib[1]. Viola-Jones va ser el primer detector d'objectes amb resultats competitius en temps real, l'any 2001. Les característiques principals són robustesa, és a dir, els falsos positius són poc freqüents i alta precisió, que treballa en temps real i que permet només la detecció de rostres (en un inici). L'algorisme consta de quatre fases:

1. Selecció dels Haar features: Identificar propietats similars entre els objectes a detectar.
2. Creació d'una imatge integral a partir dels features detectats
3. Algorisme d'aprenentatge: Una variació del algorisme d'aprenentatge AdaBoost s'usa per triar els millors features i per entrenar els classificadors que faran servir aquests features. Aquest algorisme crea un detector robust a partir d'una combinació lineal de detectors no tan robustos.
4. Arquitectura en cascada: Cada capa de l'arquitectura en cascada consta de detectors amb més features. A mesura que les deteccions avancen per les diferents capes, es van descartant més falsos positius.

2.2. Detecció de fruita en particular

Gongal et al. (3) reporta diferents mètodes de classificació d'imatges per detecció de fruites. Amb mètodes d'aprenentatge no supervisat repassen dos experiments amb pomes amb resultats de 38,8% de precisió detectant pomes verdes i 80% de precisió detectant pomes vermelles.

Amb mètodes d'aprenentatge supervisat es presenten experiments amb un classificador bayesià i amb KNN clustering. Pel classificador de Bayes els autors van obtenir una precisió del 75% classificant taronges. L'ús del classificador bayesià, que necessita partir d'informació de la funció de distribució a priori, és difícil de dur a terme per aplicacions a l'exterior.
Pel que fa a KNN clustering es recullen diferents experiments realitzats per diferents autors. Es reporta els resultats per dos d'aquests experiments, amb una precisió de 85% en pomes verdes i una precisió del 90% en pomes, plàtans, llimones i maduixes, respectivament.

Amb mètodes de soft computing es repassen resultats fent servir xarxes neuronals artificials (ANN) i Support Vector Machines (SVM).
Pel mètode ANN es reporten una precisió de 87% en taronges, amb un 15% de fals positius i 5% de fals negatius. Un altre experiment utilitzant ANN obtenia una mitja de percentatge d'error de 39,6%. El darrer experiment recollit per Gongal et al. (3) sobre ANN presenta uns resultats de 66,3% de precisió amb un 33,7% de falsos negatius. Combinant dades de color amb dades tèrmiques van aconseguir millorar la precisió fins a 74,4% amb un 25,6% de falsos negatius.

Per SVM es reporten tres experiments, el primer d'ells recull una precisió de 93% en pomes. En el segon experiment es recull una precisió de 92,4% en cítrics. En el darrer experiment que es menciona, sobre imatges de cítrics verds, es reporta una precisió de 81,7% en fruites amb més del 50% de l'àrea visible, mentre que els falsos positius arriben a un 25,6%.

[1] dlib: http://dlib.net/

2.2.1. Detecció de fruita amb Faster R-CNN

Bargoti i Underwood (2) utilitzen la xarxa neuronal Faster R-CNN per la detecció de pomes, mangos i ametlles. Els autors proposen diferents mètodes d'aprenentatge amb diferents paràmetres, el que dona a un gran nombre de resultats.

Proposen l'ús de transfer learning amb els pesos de les xarxes entrenades per detectar ametlles i mangos per a la detecció de pomes. Es comprova que amb un nombre limitat de imatges d'entrenament el marge de millora respecte a un entrenament partint dels pesos de ImageNet[2] directament és pràcticament inexistent.

Tècniques de data augmentation també són considerades, essent l'escalat i el flip les tècniques que aporten un millor rendiment, el que fa pensar que la mida i la forma són paràmetres més flexibles que el color. Pel que fa la detecció de pomes, es reporta un F1-score de 0.904 en el millor dels casos, amb una precisió pròxima a 0.90 amb 1000 imatges d'entrenament.

D'una altra banda, Sa et al. (4) parteixen també de la xarxa Faster R-CNN comentada anteriorment. Utilitzant transfer learning per adaptar el model a través d'imatges RGB i NIR, i una combinació d'aquestes, assoleixen resultats al nivell de l'estat actual en quant al F1-score, que té en compte el recall i la precisió. Experimenten una millora de 0.807 a 0.838 en aquest aspecte estudiant imatges de pebrots. A part de la millora en la precisió, aquesta proposta resulta més ràpida d'implementar per noves fruites ja que la anotació bounding box és un ordre de magnitud més ràpida que les anotacions a nivell de píxel.

Aquest estudi també proporcionava un dataset amb imatges i anotacions, però després de comparar-les amb les imatges proporcionades per la UdL, vam observar que la naturalesa de les imatges utilitzades a l'article Deep Fruit Detection in Orchards (2) era molt més similar a les imatges de les que disposàvem. Per tant ens vam decantar per aquestes imatges. A continuació es mostra una imatge de l'article DeepFruits: A Fruit Detection System Using Deep Neural Networks (4).

Figura 3. Exemple d'imatge de DeepFruits

En l'article s'estudien varies fruites, incloent pomes, melons, maduixes, alvocats, mangos, taronges i pebrots. Es reporten els resultats del F1-score, amb les maduixes obtenint el valor mes alt, de 0.948, mentre que en pebrots els resultats eren els més baixos, amb un F1-score de 0.828. En pomes s'obtenia un F1-score de 0.938.

[2] ImageNet: http://www.image-net.org/

3. Metodologia/desenvolupament del projecte

3.1. Dataset utilitzat en Deep Fruit Detection in Orchards

Per comprovar que començàvem des de un punt de partida vàlid primerament vam entrenar la xarxa Faster R-CNN amb les imatges que es van utilitzar en Deep Fruit Detection in Orchards de Bargoti i Underwood (2).
Per això calia modificar les anotacions ja que les que es donaven eren anotacions circulars en format .csv i la xarxa Faster R-CNN espera rebre el fitxer de les anotacions en format .txt i en format rectangular. Per realitzar aquests canvis vam crear un script en Matlab que automatitzés aquest procés. Un cop realitzats aquests canvis les dades estaven preparades per entrenar la xarxa. Per aquest dataset es comptabilitzen 120 imatges i un total de 663 pomes. D'aquestes 120 imatges, el 90%, és a dir, 108, es van utilitzar en l'entrenament, i la resta, 12, en la validació. A continuació es mostra un exemple d'una d'aquestes imatges.

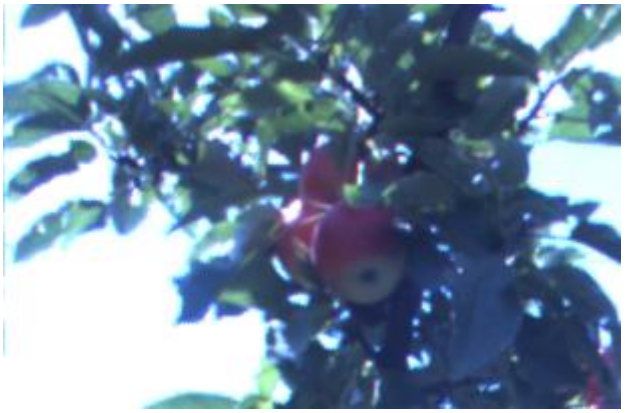

Figura 4. Imatge d'exemple del dataset utilitzat a Deep Fruit Detection in Orchards

3.2. Dataset format per imatges de la UdL

El primer pas consisteix en manualment anotar les pomes sobre les imatges de la UdL. Donat el gruix d'imatges (aproximadament 30.000) i la gran quantitat de pomes per imatge (aproximadament 67) no es van fer servir totes les imatges disponibles per l'estudi.

Les imatges van ser capturades en instants equiespaiats de temps fent servir un robot que es desplaçava a una velocitat constant i capturava les imatges aplicant cinc diferents nivells de lluminositat. Per a que la informació fos més variada les imatges a fer servir en els experiments es van triar de quatre en quatre. Aquesta selecció ve donada per dues raons. Seleccionar les imatges de quatre en quatre permet recollir imatges amb diferents lluminositats, el qual permet un entrenament més general de les pomes i que no aprengui només a detectar pomes amb una certa lluminositat. L'altre raó es basa en que les imatges capturades són molt similars entre sí degut a la velocitat del robot i el temps entre captures. Per això, agafant imatges no consecutives i deixant un espai entre imatges, podem obtenir un entrenament més general i que detecti un ventall més ample de pomes. Per aquest dataset es disposa de 120 imatges, de les quals, el 90%, és a dir, 108, es van fer servir per l'entrenament, i la resta, 12, per la validació, amb 8052 pomes en total. A continuació es mostra un exemple d'aquestes imatges.

Figura 5. Exemple de les imatges proporcionades per la UdL

3.2.1. Eina d'anotació PychetLabeller

Per a realitzar aquesta tasca es fa servir una eina d'anotació en Python anomenada PychetLabeller [3] desenvolupada per Suchet Bargoti. Aquesta eina permet afegir anotacions amb un simple clic. La mida de les esmentades anotacions es fàcilment editable per adaptar-se a objectes de diferents mides. També té uns sliders que permeten modificar el contrast i la brillantor per visualitzar millor certes parts de les imatges.

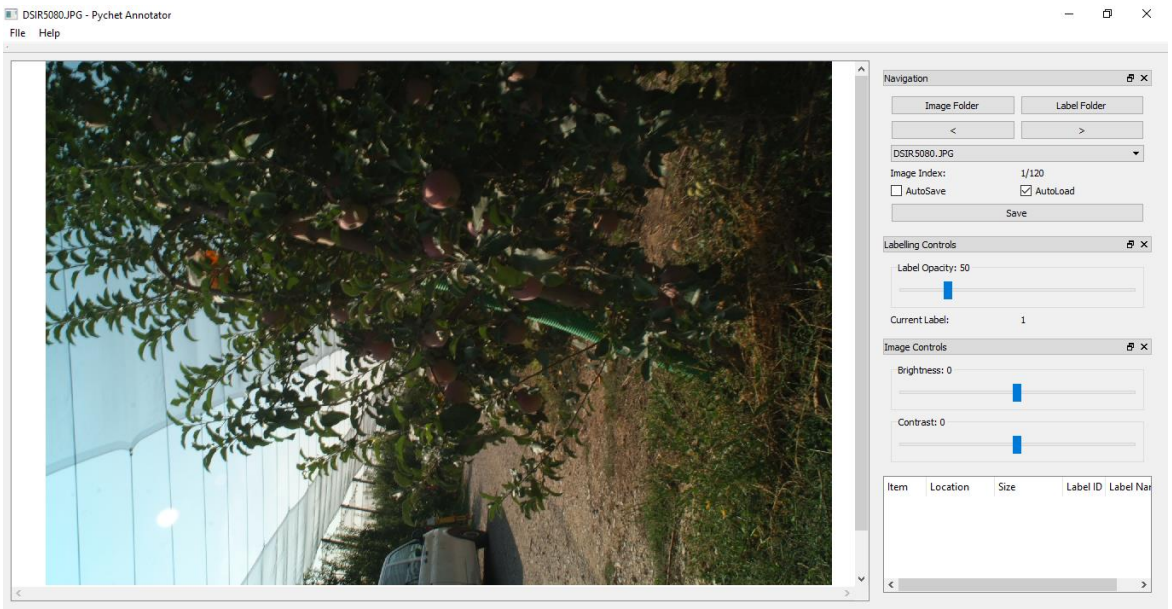

Figura 6. Eina d'annotació en Python PychetLabeller

[3] PychetLabeller: https://github.com/sbargoti/pychetlabeller

Un cop adquirides les anotacions, que es guarden en un fitxer .csv per cada imatge, cal modificar-les lleugerament degut a que la xarxa Faster R-CNN espera el fitxer de les anotacions en format .txt, en format rectangular x1, y1, x2, y2, on x1, y1 són les coordenades de la cantonada superior esquerra del bounding box i x2, y2 són les coordenades de la cantonada inferior dreta, mentre que l'eina d'anotació les guarda com x1,y1, c_x, c_y, on x1, y1 són igualment les coordenades superior esquerra i c_x, c_y corresponen a la longitud dels costats del bounding box. Per realitzar aquest canvi vam implementar un script amb Matlab ja que resultava més fàcil treballar aquests fitxers amb aquesta eina.

3.2.2. Data augmentation

En uns primers experiments vam observar que per aquestes imatges la xarxa Faster R-CNN no entrenava bé, no convergia a un valor de les pèrdues prou baix, els resultats que reportava no eren bons i les deteccions que produïa no eren correctes.

El que es va intentar després va ser, a partir de la xarxa entrenada amb les imatges del treball Deep Fruit Detection in Orchards (2), validar la xarxa amb les imatges de la UdL. En comprovar que d'aquesta manera tampoc s'obtenien els resultats desitjats es va procedir a modificar les imatges proporcionades per la UdL. Es va observar que, un cop les imatges entraven a la xarxa, es feia un resize. En aquest resize, les pomes en les imatges utilitzades a Deep Fruit Detection in Orchards (2) es redimensionaven a una mida d'uns 110 píxels en comparació a les pomes de les imatges de la UdL, que es redimensionaven a aproximadament 35 píxels. Això es deu a les dimensions originals de les imatges dels dos datasets, ja que les imatges utilitzades a Deep Fruit Detection in Orchards (2) són de 308x202 píxels mentre que les imatges de la UdL eren considerablement més grans, de 2304x1536 píxels.
A partir dels valors de resizing de la xarxa i de les imatges del dataset de Deep Fruit Detection in Orchards (2), es va calcular que les dimensions de les nostres imatges per aconseguir una mida de poma similar pels dos conjunts hauria de ser de 820x490 (afegint una rotació de la imatge per assegurar que les imatges estan en la orientació correcta). Aquestes dimensions significaven que cada imatge original s'hauria de dividir en 12 sub-imatges amb certa superposició entre elles, que a l'hora aporta més informació de la que aprendre a la xarxa. El nombre de pomes anotades augmenta de 8052 a 8991, degut al solapament entre imatges. Els fitxers de les anotacions per tant, també va caldre modificar-los, per separar els bounding boxes segons la part de la imatge a la qual pertanyin i modificar els valors de les anotacions per normalitzar-los. El nombre d'imatges total, per tant, va augmentar de 120 a 1440. Per separar les imatges entre entrenament i validació es va repetir el mateix procés de selecció aleatori, distribuint el 90% de les imatges disponibles per entrenament, és a dir, 1296, mentre que es va disposar del 10% restant, 144, per la validació A continuació es mostra un exemple. No hi ha representades totes les subimatges per claredat, ja que representar totes les particions dificultaria la comprensió del que s'està intentant explicar. En canvi es representa un exemple de com es divideix horitzontalment i verticalment la imatge original. La resta de la imatge es divideix seguint el mateix patró que es pot apreciar en la següent figura.

Figura 7. Divisó de les imatges originals

3.3. Xarxa Faster R-CNN

Com s'ha explicat anteriorment, el concepte de Faster R-CNN apareix de combinar les capes d'una fully convolutional network (FCN), la RPN, que té la funció de rebre com a input les imatges i retornar object proposals rectangulars,amb la part Fast R-CNN, que s'encarrega de la detecció d'objectes. Com la hipòtesi principal és el compartir recursos de computació, es suposa un cert nombre de capes compartibles. Es proposen dos models, el primer anomenat Zeiler i Fergus (ZF) que consta de 5 capes convolucionals compartides, i el segon, el model Simonyan i Zisserman (VGG-16) que consta de 13 capes convolucionals compartides. En els nostres experiments farem servir el model ZF.

Per generar les region proposals es llisca una petita xarxa sobre el mapa convolucional de features resultant de l'última capa convolucional compartida. Cada passada es mapeja a un feature de dimensions inferiors , i aquests features es passen com input a dues fully connected layers, una capa de box-regression (reg) i una altra de classificació (cls).

Una de les propietats d'aquesta xarxa és que és invariant en la translació. Això li atorga un gran avantatge respecte el mètode MultiBox (5). Apart de millorar el rendiment, aquesta propietat permet reduir en dos ordres de magnitud el nombre de paràmetres. Tot això ens permetrà tenir menys risc d'overfitting.

Per entrenar la capa RPN s'inicien totes les noves capes a partir dels pesos d'una distribució Gaussiana de mitja 0 amb desviació típica de 0.01. La resta de capes, és a dir, les capes compartides, s'entrenen per un model de ImageNet per a classificació. Es modifiquen totes les capes pel model ZF, mentre que pel model VGG-16 es modifiquen les capes a partir de la convolució 3_1 cap amunt. Es fa servir un learning rate de 0.001 pels primers 60.000 mini-batches i un learning rate de 0.0001 pels 20.000 següents mini-batches. S'aplica un moment de 0.9 i un decay de 0.0005. Aquesta implementació utilitza Caffe.

Un cop repassat com funciona la RPN, s'ha d'estudiar com compartir features entre la RPN i la Fast R-CNN, és a dir, com evitar que cada xarxa calculi els seus features de manera independent. Per això, els autors proposen tres mètodes (1):

- Alternate training: En aquesta algorisme, primer s'entrena la xarxa RPN i, fent servir els proposals, s'entrena la Fast R-CNN. La xarxa Fast R-CNN llavors s'utilitza per inicialitzar la RPN i s'itera aquest procés.

- Approximate joint training: En aquesta solució les dues xarxes, Fast R-CNN i RPN s'uneixen en una sola. A cada iteració els region proposals es tracten com fixos, com si es tractés d'una Fast R-CNN. Aquesta solució permet reduir el temps d'entrenament entre 25% i 50% produint bons, tot i que no òptims, resultats. Aquest algorisme és l'utilitzat en els nostres experiments.

- Non-approximate joint training: Aquesta solució pretén prendre els region proposals i calcular els gradients respecte les coordenades dels bounding boxes. Segons els autors, aquest problema no és trivial, i la solució és afegir una capa RoI warping, que queda fora de l'objecte d'estudi de l'article.

Finalment, es fa una breu menció d'un quart i últim possible algorisme, anomenat 4-step Alternate Training. En aquesta solució s'entrena la xarxa RPN com s'ha explicat anteriorment, inicialitzant-la amb un model pre-entrenat de ImageNet i es fa un fine-tuning. De igual manera, es fa un fine-tuning de la xarxa Fast R-CNN a partir de

ImageNet fent servir els region proposals calculats al primer pas. En aquest punt les dues xarxes no compten amb capes compartides. En el tercer pas s'inicialitza la xarxa RPN a partir de la xarxa Fast R-CNN, fixant les capes convolucionals compartides i només modificant les capes pròpies de la RPN. Finalment es fa un fine-tuning de les capes úniques de la xarxa Fast R-CNN. Així les dues xarxes comparteixen les mateixes capes convolucionals. Aquest mètode es pot iterar, però els autors reporten que no van detectar resultats significatius.

Per realitzar els nostres experiments aplicarem una tècnica de transfer learning anomenada fine-tuning. El transfer learning consisteix en agafar informació obtinguda en un problema per aplicar dita informació a altres problemes semblants. Més concretament, el fine-tuning consisteix en començar l'entrenament a partir d'uns pesos prèviament calculats, en comptes de començar amb uns valors aleatoris. Aquesta tècnica és molt útil quan no es disposa d'un gran nombre d'exemples d'entrenament. Per realitzar el fine-tuning dels nostres experiments partirem dels pesos de ImageNet.

Per entrenar la xarxa Faster R-CNN sobre un dataset propi cal generar certs fitxers amb una certa estructura predeterminada. Tots els fitxers que cal modificar i/o crear estan explicats a continuació.

- nom_dataset.py: Aquest fixter Python totes les configuracions específiques del dataset, en concret, les opcions més rellevants són les classes per les quals volem entrenar la xarxa, en quin format volem llegir les anotacions i defineix també la funció per reportar els resultats de validació.
- factory.py: Aquí hem d'afegir el nostre dataset amb els noms dels sets d'entrenament i validació.
- faster_rcnn_end2end.sh: És el fitxer bash que crida a les funcions d'entrenament i validació. Cal afegir el nom del dataset i el número d'iteracions per les quals volem entrenar.
- config.py: Aquí podem configurar altres paràmetres de la xarxa independents del dataset (nombre d'iteracions entre snapshots) però també cal afegir la ruta a la carpeta de models del dataset.
- models: Carpeta on es defineixen els paràmetres d'entrenament i de validació per un dataset en particular, com per exemple el learning rate, així com les convolucions i les diferents capes de la xarxa.
- Directori del dataset: Directori on s'han de guardar les imatges, les anotacions i els fitxers de traint.txt i test.txt, que llisten les imatges que seran utilitzades per entrenar i quines seran per validar.

3.4. Experiments

A continuació es presenta una descripció dels diferents experiments realitzats. Es va afegir una capa de validació durant l'entrenament ja que no venia dissenyada per defecte. Es va seleccionar com a capes d'entrada el resultat de la detecció ('cls_score') i la informació original de les etiquetes ('labels') i la capa calcula la precisió a mesura que avança l'entrenament. Per realitzar els diferents fine-tunings no es va congelar ninguna capa.

3.4.1. Experiment 1

Entrenament de la xarxa Faster R-CNN amb les imatges del dataset utilitzat a Deep Fruit Detection in Orchards (2) fent un fine-tuning a partir dels pesos de ImageNet. Es va definir el learning rate inicial a 0.001 amb una variació “step” amb gamma igual a 0.1. Es va definir el nombre d’iteracions en 40.000. Validació dels resultats amb imatges pertanyents al dataset de Deep Fruit Detection in Orchards (2). En aquest experiment ens proposàvem trobar un punt de partida, ja que podíem comparar els resultats amb els obtinguts per Bargoti i Underwood (2).

3.4.2. Experiment 2

Entrenament de la xarxa Faster R-CNN amb les imatges del dataset utilitzat a Deep Fruit Detection in Orchards (2) fent un fine-tuning a partir dels pesos de ImageNet. Es va definir el learning rate inicial a 0.001 amb una variació “step” amb gamma igual a 0.1. Després dels resultats del primer experiment, es va decidir canviar les iteracions de 40.000 a 10.000. Validació dels resultats amb imatges proporcionades per la UdL. Amb aquest experiment es pretenia observar el rendiment de la xarxa al validar-la amb imatges del mateix tipus de fruita, pomes, però amb origen diferent.

3.4.3. Experiment 3

Entrenament de la xarxa Faster R-CNN amb les imatges proporcionades per la UdL fent un fine-tuning a partir dels pesos de ImageNet. Es va definir el learning rate inicial a 0.001 amb una variació “step” amb gamma igual a 0.1. Després dels resultats del primer experiment, es va decidir canviar les iteracions de 40.000 a 10.000. Validació dels resultats amb imatges proporcionades per la UdL. Amb aquest experiment es pretén replicar l’experiment 1 amb les imatges aportades per la UdL per poder comparar resultats.

3.4.4. Experiment 4

Entrenament de la xarxa Faster R-CNN amb les imatges proporcionades per la UdL fent un fine-tuning a partir de la xarxa entrenada amb les imatges de Deep Fruit Detection in Orchards (2) (experiment 1). Es va definir el learning rate inicial a 0.0001 amb una variació “step” amb gamma igual a 0.1. S’observa que es defineix un learning rate més baix per assegurar la convergència, ja que la xarxa ja havia estat prèviament entrenada. Després dels resultats del primer experiment, es va decidir canviar les iteracions de 40.000 a 10.000.Validació dels resultats amb imatges proporcionades per la UdL. En aquest experiment l’objectiu era comprovar l’efecte del transfer learning.

A continuació s'adjunta una taula descriptiva dels diferents experiments explicats anteriorment.

Experiment	Punt de partida	Imatges d'entrenament	Imatges de validació
1	ImageNet	Deep Fruit Detection in Orchards	Deep Fruit Detection in Orchards
2	ImageNet	Deep Fruit Detection in Orchards	Imatges UdL
3	ImageNet	Imatges UdL	Imatges UdL
4	Xarxa de l'experment 1	Imatges UdL	Imatges UdL

Taula 1. Descripció dels experiments

4. Resultats

A continuació es mostren les corbes de loss d'entrenament i de precisió durant l'entrenament realitzat en els experiments 1, 3 i 4.

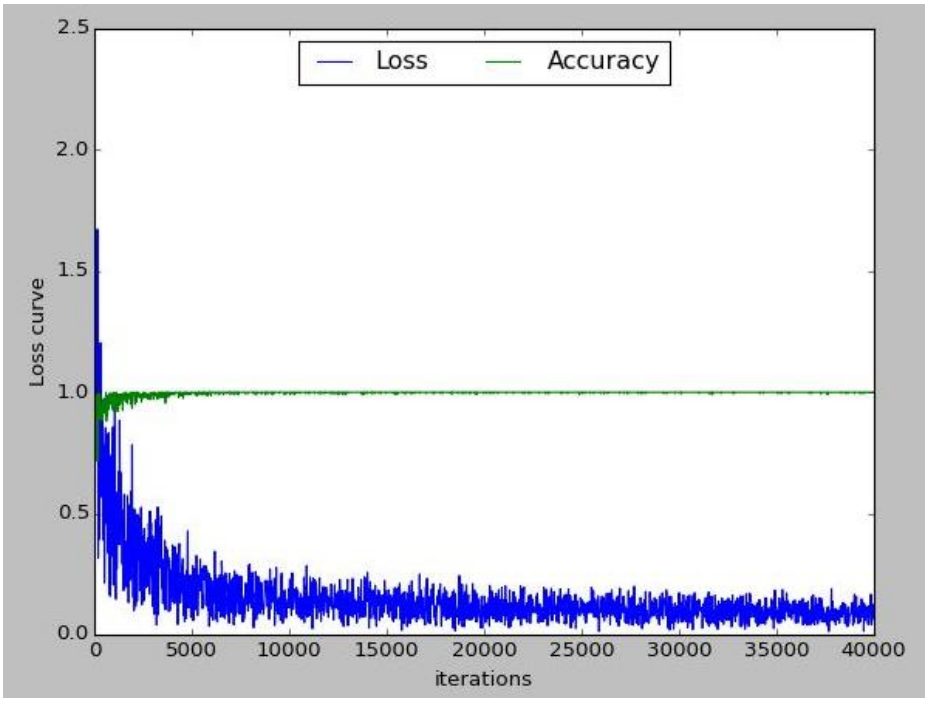

Figura 8. Corbes deep learning per l'experiment 1. Loss (blau) Precisió (verd)

Podem observar que a partir de aproximadament les 10.000 iteracions tant la precisió com les pèrdues es mantenen constants. Per això en els següents entrenaments només vam entrenar fins a les 10.000 iteracions.

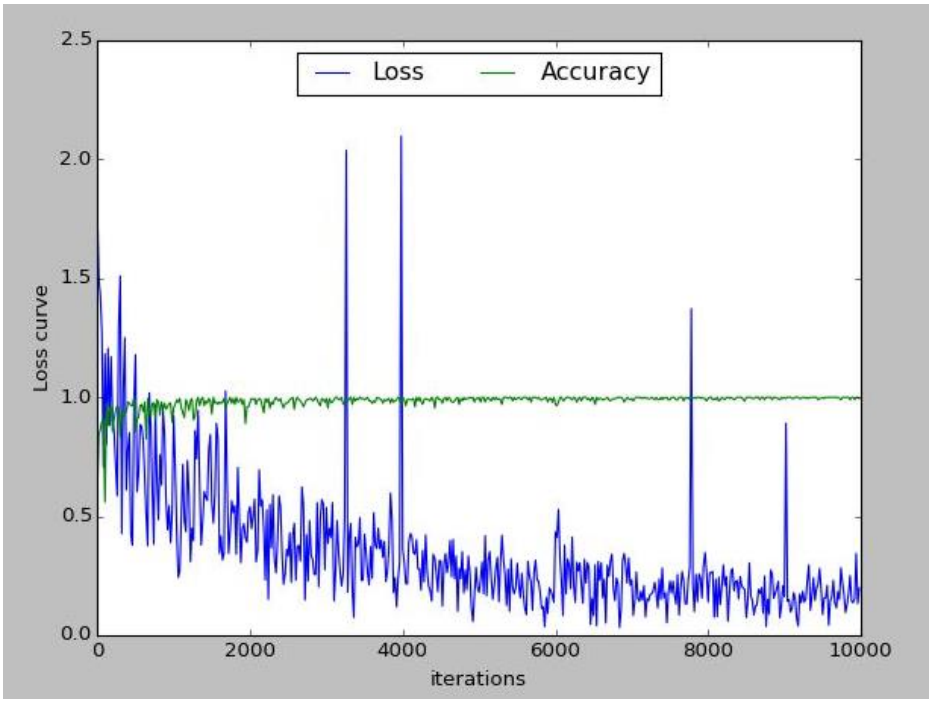

Figura 9. Corbes deep learning per l'experiment 3. Loss (blau) Precisió (verd)

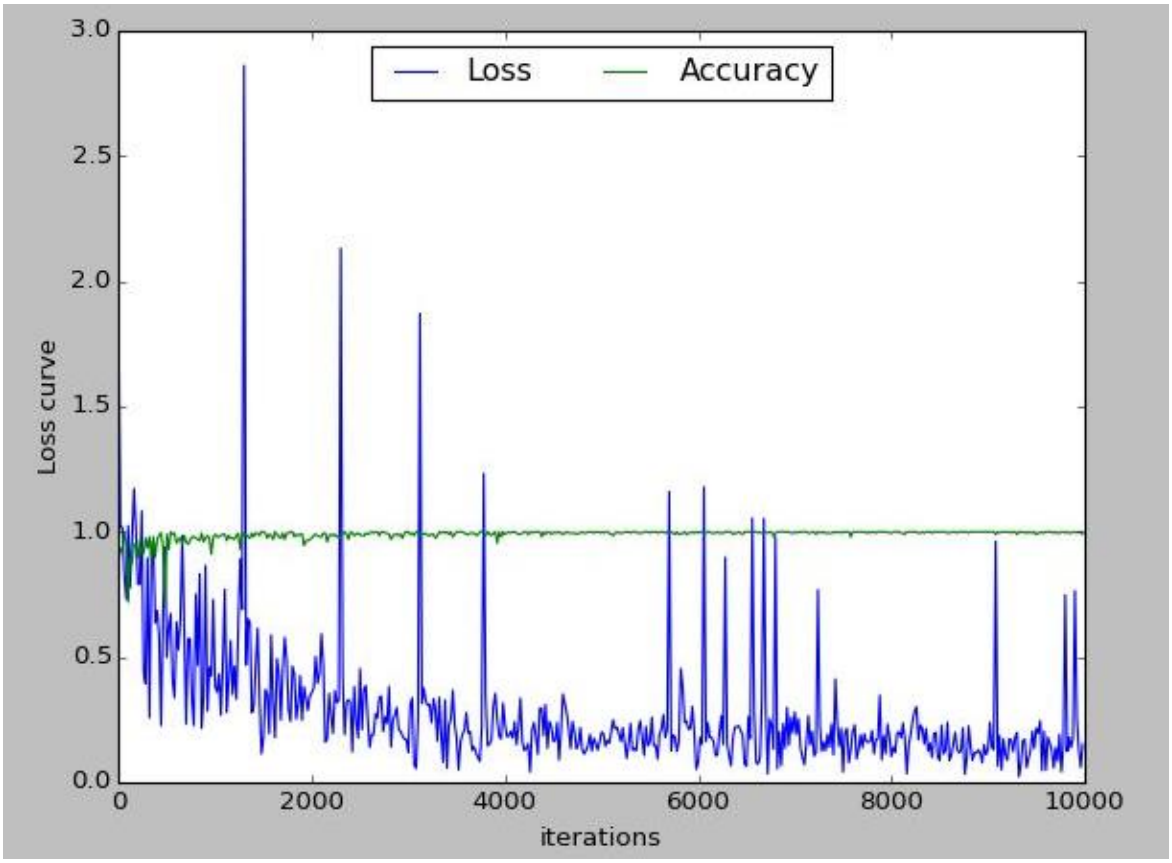

Figura 10. Corbes deep learning per l'experiment 4. Loss (blau) Precisió (verd)

En la següent taula es resumeixen els resultats obtinguts dels experiments descrits anteriorment:

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Precisió	0.7978	0.4706	0.7131	0.6800
Recall	0.8256	0.7412	0.7695	0.7729
F1-score	0.8115	0.5757	0.7402	0.7235

Taula 2. Resultats dels experiments

Quan observem els resultats dels experiments s'aprecia que per l'experiment 1, comparat amb els resultats que van obtenir Bargoti i Underwood (2) per unes 100 imatges s'obté uns resultats molt similars tot i que no arriben al mateix nivell, la qual cosa es podia esperar ja que en el nostre experiment no vam implementar les diferents tècniques de data augmentation que sí s'apliquen en l'experiment de Bargoti i Underwood (2).

D'altra banda, observem que en l'experiment 2 obtenim una baixa precisió però un recall acceptable. Això ens indica que en fer servir una xarxa amb pesos calculats a partir d'un dataset de pomes i validar-lo amb un dataset diferent el nombre de positius vertaders és alt. El problema apareix amb els falsos positius, es detecten massa i això repercuteix en la precisió general del sistema. Comparant amb l'experiment 4 observem que fer un fine-tuning a partir d'uns pesos inicials d'un altre dataset millora en un petit percentatge el recall, és a dir, es detecten una mica més de positius vertaders, mentre que la precisió creix força, és a dir, aconseguim disminuir el nombre de fals positius.

Comparant els experiments 3 i 4 observem que la semblança de resultats és considerable, fins al punt que per un experiment la precisió és més alta mentre que per l'altre el recall és millor. Així observem que l'experiment 4, que ha vist més imatges de pomes és lleugerament millor a l'hora de detectar-les, mentre que en l'experiment 3, on la xarxa s'ha entrenat amb menys pomes però més similars, realitza una millor tasca de precisió a l'hora de no detectar falsos positius, però per tant deixa de detectar més pomes.

5. Pressupost

Com la tesis no consta d'un prototip, per calcular el pressupost ens basarem en el número d'hores dedicades amb sou d'enginyer júnior.
Estimem que un enginyer júnior cobra aproximadament 25.000 euros bruts anuals en 52 setmanes, és a dir 21.000/52 = 480€/setmana, en una jornada de 40 hores setmanals són aproximadament 12€/hora. Ara, contant aproximadament 600 hores dedicades a la tesis obtenim un cost d'aproximadament 7.200 €.
També tenim en compte el sou dels codirectors de la tesis, amb un sou d'enginyer sènior de 100€/hora. Es comptabilitza una hora setmanal durant 33 setmanes, equivalent a la reunió que es produïa entre alumne i professors. El resultat final és de 3.300€ per enginyer sènior.

Concepte	Cost
Sou enginyer júnior	7.200€
Seguretat Social (30% sou enginyer júnior)	2.160€
Sou enginyers sènior	6.600€
Seguretat Social (30% sou enginyer sènior)	1.980€
Serveis i subministraments (llum, internet...)	2.500€
Lloguer lloc de treball	7.000€
Despeses indirectes	5.000€
TOTAL	32.440€

També cal tenir en compte que per algun script de tractament de dades s'ha fet servir MATLAB. La llicència anual té un preu de 800€ mentre que la llicència perpètua val 2000€. L'amortització de la llicència anual és clara, 800€, ja que un cop expirada en un any no es pot vendre per recuperar part de la inversió. En canvi la llicència perpètua pot tenir una vida molt més llarga, tant com duri el projecte per futures generacions que facin més avenços a partir d'aquí. Si aquets projecte segueix en moviment dos anys i mig més ja seria més rendible la versió perpètua que la anual.

6. Conclusions i desenvolupaments futurs

La premissa principal de la tesis era la de desenvolupar un sistema de detecció automàtica de fruites utilitzant algoritmes de deep learning. Comparant el resultats obtinguts amb els diferents estudis exposats a l'estat de l'art podem arribar a la conclusió que els resultats caben dins del que es podria esperar.

Veiem que la precisió és més susceptible de variació depenent de les dades d'entrenament mentre que el recall es manté en un rang més petit. Això indica que, per la tasca de recol·lecció automàtica, es podrien obtenir bons resultats ja que en el pitjor dels casos, un fals positiu suposarà que el sistema mecànic perdrà cert temps en recollir una poma que no hi és, però no se'n deixarà gaires, fins i tot en casos on no es tingui informació d'entrenament i es parteixi d'uns pesos calculats per unes altres imatges (experiment 2). En canvi, per la feina d'estimació de collites podria suposar un problema, ja que reportar grans números de deteccions, de les quals més de la meitat són falsos positius significaria donar una previsió de més de dos cops el que veritablement hi ha, fet que descarta l'ús de xarxes pre-entrenades per aquesta tasca particular.

També val la pena comentar que per anotar 120 imatges de les proporcionades per la UdL, a un ritme de 4 minuts per imatge, significa trigar 8 hores en anotar les imatges, mentre que la xarxa triga poc més de 45 minuts en convergir, el que significa unes anotacions més de 10 cops més veloces tot i perdre certa precisió. La solució òptima sembla ser primer passar les imatges per la xarxa per obtenir el gran gruix d'anotacions i després, manualment, refinar les anotacions, eliminant els falso positius i marcant els falsos negatius, que, combinats, com hem observat, corresponen a un nombre molt inferior respecte al total de pomes. Així es reduirà el temps emprat en aquesta tasca que tampoc obté uns resultats perfectes realitzada a mà, perquè cal tenir en compte l'error humà, sobretot en el recall (no detectar una fruita quan ja portes estona).

En el camí de continuar el treball proposaria augmentar el nombre d'imatges d'entrenament pels dos datasets i aplicar les diferents tècniques de data augmentation que proposen Bargoti i Underwood (2) sobre les imatges de la UdL. També proposaria que certs scripts per tractar les dades que s'han fet en MATLAB es redissenyessin a Python per evitar la llicència. Per últim, també afegiria primer retallar les imatges i modificar-les abans d'annotar-les. Un cop retallades, per dividir les anotacions entre les diferents imatges generades, comparàvem les coordenades del bounding box amb les coordenades de la imatge retallada. Si alguna de les coordenades del bounding box quedava fora de la imatge, es descartava aquesta anotació. Si la xarxa detecta aquestes fruites, les quals havíem descartat les anotacions, correctament, les contarà com falsos positius i això farà baixar tant la precisió com el recall. És molt probable que això ens hagi passat.

Bibliografia

1. Ren S, He K, Girshick RB, and Sun J. Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks," CoRR, vol. abs/1506.0, 2015. [Online]. Available: http://arxiv.org/pdf/1506.01497

2. Bargoti S, Underwood J. Deep Fruit Detection in Orchards.2016 [Online].Available: http://arxiv.org/pdf/1610.03677

3. Gongal A, Amatya S, Karkee M, Zhong Q, Lewus K. Sensors and systems for fruit detection and localization: A review. Computers and Electronics in Agriculture. 2015; 116:8-19.

4. Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5017387/pdf/sensors-16-01222.pdf

5. Szegedy C, Reed S, Erhan D, Anguelov D. Scalable, high-quality object detection. 2015. [Online]. Available: https://arxiv.org/pdf/1412.1441

Glossari

Faster R-CNN: Faster Region-based Convolutional Neural Network

UdL: Universitat de Lleida

CNN: based Convolutional Neural Network

RPN: Region Proposal Network

ANN: Artificial Neural Network

SVM: Support Vector Machine

FCN: Fully Convolutional Network

ZF: Zeiler i Fergus

VGG-16: Simonyan i Zisserman