

Valorització de residus amb deep learning i càmeres multispectrals

Edgar Gracia Larrosa

Resum– Actualment tots els residus que es generen en les ciutats són enviats a plantes de processament on se separen i seleccionen aquells que poden ser reutilitzats en un futur. Aquesta és una tasca complexa degut a la creixent quantitat de material que ha de ser analitzat i a la gran varietat d'elements que hi podem trobar. En aquest projecte buscarem oferir solucions a alguns dels problemes de visió presents en una planta de reciclatge, on es busca automatitzar aquest procés. S'estudiarà l'ús de diferents tecnologies i xarxes neuronals per ajudar a detectar i classificar cada tipus de residu.

Paraules clau– Visió per computador, xarxes neuronals, deep learning, machine learning, detecció d'objectes, mapes de profunditat, hiperspectral.

Abstract– Nowadays, garbage generated in cities are sent to processing plants, where different materials are separated and those that can be reused in the future are selected. This is a complex task due to the increasing amount of material that must be analysed and the variety of elements that we may encounter. In this project, we will look to provide some solutions to different computer vision problems of a recycling plant, where this process is being automated. We will study the usage of different technologies and neural network models to detect and classify several types of wastes.

Keywords– Computer vision, neural networks, deep learning, machine learning, object detection, depth map, hyperspectral.



1 INTRODUCCIÓ

LA gestió dels residus i el reciclatge s'han convertit en els últims anys en un sector de gran importància, clau per al desenvolupament d'una economia circular. En aquest projecte es tractaran diferents problemes de visió per computador presents en un Ecoparc, una instal·lació medi ambiental on es recullen i processen les deixalles generades pels municipis propers.

En aquesta planta s'utilitzen diferents tècniques i processos mecànics per a separar gran part dels residus que hi arriben, els quals són posteriorment compactats en blocs i redistribuïts a altres plantes. Tot aquell material que en un primer moment no ha pogut ser classificat, passa a una segona etapa on, per mitjà de sistemes de visió per computador, braços robòtics i personal, s'intenta separar de forma més exhaustiva. Tant en l'àmbit econòmic com medi ambiental, aquesta és una tasca important dins de l'Ecoparc, ja que tots els residus que no han pogut ser classificats correctament han de ser incinerats o enviats a abocadors.

Estudiarem tres tipus de problemes diferents amb l'ob-

jectiu de millorar aquest procés de triatge: En primer lloc, per a guiar a un braç robòtic, utilitzarem xarxes de detecció d'objectes per a localitzar i classificar els diferents tipus de residus que passen per una cinta transportadora. Entrenarem diverses arquitectures i compararem els seus resultats.

Per altra banda, s'estudiarà l'ús de xarxes de segmentació per a obtenir una reconstrucció 3D dels residus. Es realitzarà una regressió sobre imatges de color i es buscarà obtenir una nova imatge de profunditat, que ajudarà al braç robòtic a conèixer l'alçada de cada element.

Finalment s'analitzaran els beneficis que aporta l'ús de càmeres hiperspectrals a l'hora de classificar diferents materials, en comparació a les càmeres tradicionals de color.

Durant el transcurs d'aquest projecte i amb relació a l'estança de pràctiques en el Centre de Visió per Computador, també es tractaran diferents experiments que serviran de precedents al problema principal dels residus.

2 OBJECTIUS

Els objectius principals que s'han seguit durant aquest projecte són els que es mostren a continuació:

1. Aplicar diferents models de xarxes neuronals de detecció d'objectes per a comparar el seu rendiment en un problema real.

- E-mail de contacte: edgar.gracia@e-campus.uab.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Felipe Lumbreras Ruiz i Coen Antens (CVC)
- Curs 2019/2020

2. Crear i entrenar una xarxa neuronal per a la reconstrucció de mapes de profunditat a partir d'imatges RGB.
3. Analitzar la informació addicional aportada per diferents càmeres hiperspectrals a l'hora de classificar diferents tipus de materials.
4. Avaluar els resultats obtinguts de forma quantitativa en cada cas.

3 ESTAT DE L'ART

A continuació es mostraran els detalls de les diferents tecnologies i models de xarxes neuronals que utilitzarem en aquest projecte, així com el seu estat en l'actualitat i les raons per a les quals les hem escollit.

3.1 Xarxes de detecció d'objectes

Les xarxes de detecció d'objectes permeten detectar diferents elements de forma simultània en una imatge, indicant la classe a què pertanyen i la regió que ocupen. Algunes de les arquitectures més utilitzades actualment [1] per la seva precisió i velocitat són: Faster R-CNN, SSD i YOLO V3.

- **Faster R-CNN [2]:** Publicada el juny de 2015, utilitza una xarxa convolucional per extreure les característiques de les imatges. Aquestes són utilitzades com a entrada d'una xarxa de proposta de regions que genera les possibles regions on pot haver-hi un objecte. A continuació una xarxa de classificació determina l'objecte que hi ha en cada una, utilitzant d'entrada les característiques extretes anteriorment. La xarxa d'extracció de característiques utilitzada originalment és una VGG però en el nostre cas utilitzarem una Inception ResNet V2, ja que obté millors resultats [3].
- **SSD [4]:** Publicada el desembre de 2015. *Single Shot MultiBox Detector*, a diferència de la Faster R-CNN, realitza el procés de proposta de regions i classificació amb una sola xarxa convolucional. Utilitzant una xarxa VGG-16 s'extreuen les característiques de les imatges i a continuació es divideix la seva sortida en una quadrícula, on cada cel·la prediu les regions i classes de cada objecte. Aquesta arquitectura d'una sola etapa permet una velocitat de detecció superior a altres com l'anteriorment esmentada Faster R-CNN.
- **YOLO V3 [5]:** Publicada el març de 2018. Tal com fa referència el seu nom, *You only look once*, el procés de classificació i proposta de regions es realitza en una sola etapa, de forma similar a la SSD. S'utilitza una xarxa especialment dissenyada per a l'extracció de característiques i la seva sortida es divideix en una quadrícula on cada cel·la prediu les regions i classes.

Hem seleccionat la SSD i la YOLO V3, ja que són dos models d'una sola etapa que permeten predir imatges en temps real i obtenen uns nivells de precisió relativament alts. Per l'altra banda, hem seleccionat la Faster R-CNN amb l'objectiu d'obtenir una major precisió, buscant una aproximació a l'estat de l'art en detecció d'objectes, tot i obtenir una velocitat menor [1].



Fig. 1: Mostra de detecció d'objectes del dataset Ecoparc amb els perímetres i bounding boxes del ground truth

3.2 Predicció d'imatges de profunditat

Les imatges de profunditat són aquelles en les que els valors de cada píxel en lloc de descriure un color, descriuen la distància entre la càmera i el punt en l'espai representat en aquell píxel. Aquestes imatges poden ser obtingudes utilitzant tecnologies com ara la visió en estèreo, càmeres de temps de vol o l'anàlisi de certs patrons o línies entre d'altres.

En aquest projecte s'ha plantejat utilitzar una xarxa neuronal que sigui capaç d'aprendre la forma de determinats objectes i que generi imatges de profunditat utilitzant únicament imatges individuals de color, sense la utilització de cap de les anteriorment esmentades tècniques.

Actualment trobem diversos articles on es planteja aquesta mateixa idea [6]. Les xarxes utilitzades en aquests casos consisteixen en una primera etapa de contracció on s'extreuen característiques de les imatges i una segona etapa d'expansió on es construeix progressivament una nova imatge.

En el nostre cas utilitzarem una xarxa de segmentació semàntica U-Net [7], ja que té una arquitectura semblant a la descrita anteriorment. Modificarem aquesta xarxa perquè predigui les distàncies als objectes en lloc de la seva classe.

3.3 Imatges hiperspectrals

En aquest projecte utilitzarem dues càmeres hiperspectrals sensibles a l'espectre visible i a l'infraroig, amb l'objectiu de millorar la detecció i classificació de diferents residus. Les imatges hiperspectral són imatges compostes per un gran nombre de canals, que representen diferents rangs de longituds d'ona de l'espectre electromagnètic. A diferència de les imatges de color RGB convencionals, les quals estan formades per tres imatges sensibles a tres rangs amplis de longituds d'ona (corresponents a la percepció de la llum vermella, verda i blava), les imatges hiperspectrals poden posseir fins a més de 100 canals sensibles a rangs més específics de longituds d'ona [8].

Aquest tipus d'imatges s'utilitzen des de fa dècades en àrees com l'astronomia, l'agricultura de precisió, la teledetecció o en entorns industrials entre d'altres, ja que permeten analitzar les freqüències de llum reflectides per diferents elements i inferir la seva composició o propietats.

Utilitzarem les càmeres *SPECIM FX10* (rang espectral 400-1000 nm) i la *SPECIM FX17* (rang espectral 900-1700 nm).

4 METODOLOGIA

Durant el transcurs d'aquest projecte s'ha fet un seguiment del treball utilitzant Kanban. Aquesta és una metodologia Agile centrada en la finalització de les tasques i en la visualització global de l'estat del projecte.

Per organitzar el treball hem utilitzat el tauler Kanban, on s'han definit tres columnes (pendent, en progrés i finalitzat) i hem afegit tasques en cada una en funció del seu estat i prioritat. Per implementar aquest tauler s'ha utilitzat el servei web Trello. Per altra banda hem realitzat sessions de seguiment setmanals on s'han mostrat els resultats obtinguts i s'han discutit els següents passos a seguir.

Una de les principals raons a l'hora d'eleger aquesta metodologia davant d'altres, ha estat que gran part d'aquest projecte està enfocat a la recerca i a l'entrenament de xarxes neuronals. Per aquest motiu necessitem la flexibilitat que ens aporta Kanban a l'hora de definir les tasques i la seva durada. Això ens permet adaptar-nos als temps d'entrenament i a la possibilitat de modificar o crear noves tasques en funció dels resultats obtinguts.

5 DESENVOLUPAMENT

A continuació es mostra el desenvolupament de les diferents parts d'aquest projecte. S'exposaran els problemes als què ens hem enfrontat, les solucions implementades i finalment, els resultats i conclusions extretes.

D'acord amb els nostres objectius, hem dividit el treball en les següents seccions en funció de la tecnologia utilitzada en cada cas:

1. Detecció d'objectes
2. Predicció de mapes de profunditat
3. Anàlisi d'imatges hiperespectrals

5.1 Desenvolupament de xarxes neuronals de detecció d'objectes

A continuació es mostren les dades, el desenvolupament i els resultats obtinguts en dos experiments de detecció d'objectes.

5.1.1 Format de les dades

Per cada un dels experiments que realitzarem crearem un conjunt de dades o dataset, on es guardaran les imatges del problema en qüestió i les anotacions o ground truth, que descriuran el seu contingut. Utilitzarem el format definit en la PASCAL VOC [9] per a estructurar aquestes dades.

PASCAL Visual Object Classes (VOC) és una competició organitzada durant 2005 i 2012 on es comparaven diferents xarxes de detecció d'objectes i que establia un format de dades concret que s'utilitza actualment en moltes implementacions.

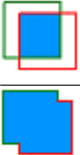
Aquest format defineix dos directoris on es guarden, de forma separada, les imatges en format JPG i les anotacions en format XML. Cada imatge té associada una anotació amb el mateix nom, dins de la qual s'especifica la mida de la imatge, la posició i el nom dels diferents objectes que hi apareixen, entre altres camps.

5.1.2 Mètriques d'avaluació

Les xarxes de detecció d'objectes ens indiquen tant la classe a què pertany cada element, com la posició que ocupen dins de la imatge, expressada en forma de coordenades màximes i mínimes (bounding box). Per aquest motiu necessitem una mètrica que ens permeti avaluar el nostre model tenint en compte ambdós factors.

La mean Average Precision (mAP) compleix aquest requisit i ens permet conèixer la precisió global del nostre model en forma de percentatge. Aquesta mètrica és la més extensa per als problemes de detecció d'objectes. El seu càlcul es pot definir en els següents passos:

1. Donades una bounding box del ground truth i una predicció, es calcula la relació entre l'àrea de la intersecció i l'àrea de la unió d'ambdues regions. Aquesta relació s'anomena Intersection over Union (IoU) i determina quant semblants són dues bounding boxes, prenent un valor de 1 quan són totalment iguals i 0 quan no coincideixen en cap punt. Si la IoU és superior a un llindar (0.5 en el cas de PASCAL VOC) i la classe assignada és la mateixa que el ground truth, es considera un positiu verdader (TP). Si la IoU és inferior a aquest llindar, es considera un fals positiu (FP). Per altra banda si existeix una bounding box que no ha estat predita o amb una classe incorrecta, serà un fals negatiu (FN).

$$IoU = \frac{\text{area}(BB_p \cap BB_{gt})}{\text{area}(BB_p \cup BB_{gt})} = \frac{\text{àrea de la intersecció}}{\text{àrea de la unió}}$$


on BB_p i BB_{gt} són les bounding box predita i ground truth respectivament.

2. Cada predicció té associat un valor de confiança que descriu quant segur està el model. Les bounding boxes són ordenades de forma descendent per aquest valor i es calculen les mètriques Precision i Recall de forma acumulativa.

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

Amb aquests dos valors es crea la corba Precision-Recall de cada classe. L'àrea existent per sota d'aquesta és l'Average Precision (AP) que indica quant bo és el model predient aquesta classe [10].

3. Finalment la mAP es calcula realitzant la mitjana de totes les AP de cada classe.

5.1.3 Experiment Smart Market

Abans d'entrar en el problema dels residus, hem buscat entrenar i avaluar una xarxa de detecció YOLO amb unes dades relativament senzilles, que servirà de precedent al problema principal.

Aquest experiment, titulat Smart Market, busca automatitzar el procés de pagament en els supermercats per mitjà d'una xarxa de detecció d'objectes. El seu objectiu és localitzar i classificar diferents productes per tal de calcular posteriorment el preu total d'una compra, sense la necessitat de llegir cap codi.

5.1.3.1 Desenvolupament

Per a poder entrenar la xarxa neuronal el primer pas a realitzar és construir el dataset en format VOC. Per això necessitem capturar imatges de cada article i indicar la seva classe i posició.

S'ha dispostat d'una cinta transportadora amb una coberta on es troben situats un sistema d'il·luminació i una càmera, que permeten obtenir imatges en un entorn controlat. Hem utilitzat recreacions a petita escala de productes reals per a simplificar el procés i s'han pres imatges de totes les seves cares, de forma individual i indicant en el nom de la imatge el nom del producte. A continuació, utilitzant la llibreria OpenCV per a Python, hem buscat la regió ocupada per cada article. S'ha utilitzat una captura de la cinta transportadora buida per calcular la diferència amb cada imatge i s'han realitzat operacions de morfologia per a obtenir una màscara dels productes (Figura 2a). Finalment, a partir d'aquesta màscara i el nom de cada imatge, s'ha determinat la bounding box i la classe de cada objecte (Figura 2b).

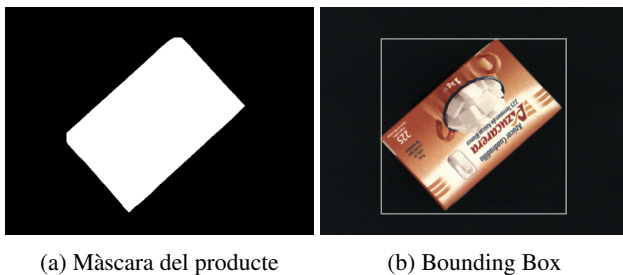


Fig. 2: Visualització del càlcul de la bounding box d'una mostra del dataset Smart Market

En total el dataset està format per 23000 imatges, amb 98 classes diferents. Cada classe correspon a una cara d'un dels 28 productes que tenim. S'ha optat per aquesta opció, en lloc de definir únicament 28 classes, ja que la majoria d'articles tenen una aparença diferent per cada cara. Per últim s'han dividit en un 75% per l'entrenament i un 25% de test per l'avaluació final.

Per aquest problema hem decidit utilitzar una YOLO V3, ja que ens aporta un temps de predicció suficientment baix per ser executada en temps real.

Durant l'entrenament hem redimensionat les imatges de 1600×1000 a 800×500 , ja que amb resolucions més altes no es disposava de memòria suficient per entrenar el model. L'entrenament ha durat un total de 4 dies i 9 hores fins a convergir.

5.1.3.2 Resultats

Hem generat les prediccions de les imatges del conjunt de test i hem calculat la mAP. El resultat ha estat un 92.95% de mAP, el qual indica un molt alt percentatge d'encert. La velocitat de predicció ha estat de 8 imatges per segon.

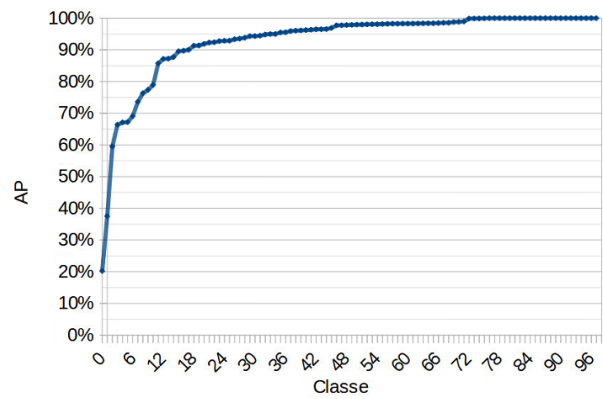


Fig. 3: AP obtinguts per la YOLO V3 en el conjunt de test de Smart Market

En la Figura 3 es pot observar l'AP obtinguda per cada classe ordenada de forma ascendent. Mentre que aproximadament la meitat de les classes obtenen una AP d'entre el 98% i el 100%, un altre grup més reduït obté uns percentatges relativament més baixos. Si analitzem les 17 classes amb pitjor AP (entre 20% i 89% d'AP), trobem que són classes corresponents a productes els quals tenen cares molt semblants entre si i resulten més difícils de diferenciar. Si realitzem l'avaluació tenint en compte únicament la classificació de productes, unificant les diferents cares, trobem que la mAP és del 96.26%.

Tot i que aquests són uns resultats relativament bons i ens permet veure la implementació d'una xarxa de detecció d'objectes, hem de tenir en compte que aquest és un problema no gaire complex, on totes les mostres són similars entre si, els objectes es troben aïllats i les seves formes, escala i il·luminació no varien. Per aquest motiu no es pot determinar si el model és capaç de generalitzar correctament o pel contrari, estem davant d'un cas d'overfitting i per això, aquest no resulta un exemple idoni amb què comparar el rendiment de la YOLO V3.

5.1.4 Experiment Ecoparc

El segon experiment que hem realitzat correspon al fil conductor d'aquest projecte, la gestió dels residus, on es busca classificar diferents tipus de deixalles que viatgen per una cinta transportadora. Per automatitzar aquest procés, en les instal·lacions de l'Ecoparc es disposa d'un braç robòtic capaç d'agafar objectes i dipositar-los en seccions diferents i d'una càmera que pren imatges zenitals de la cinta.

5.1.4.1 Desenvolupament

El dataset original que ens han proporcionat està etiquetat de forma manual i consisteix en 10000 imatges TIFF i anotacions en format JSON on s'indica la resolució, l'àrea de cada objecte i la seva classe. Trobem 4 classes diferents: llaunes, plàstics PEAD, plàstics PET i Tetra Briks.

Les àrees dels objectes estan definides per mitjà d'una llista de punts, els quals determinen el seu perímetre. En el nostre cas, per entrenar les xarxes de detecció d'objectes únicament necessitem les bounding boxes que els delimiten. Per a calcular-les hem buscat els valors màxims i mínims de tots els punts. En la Figura 1 en podem veure una mostra.

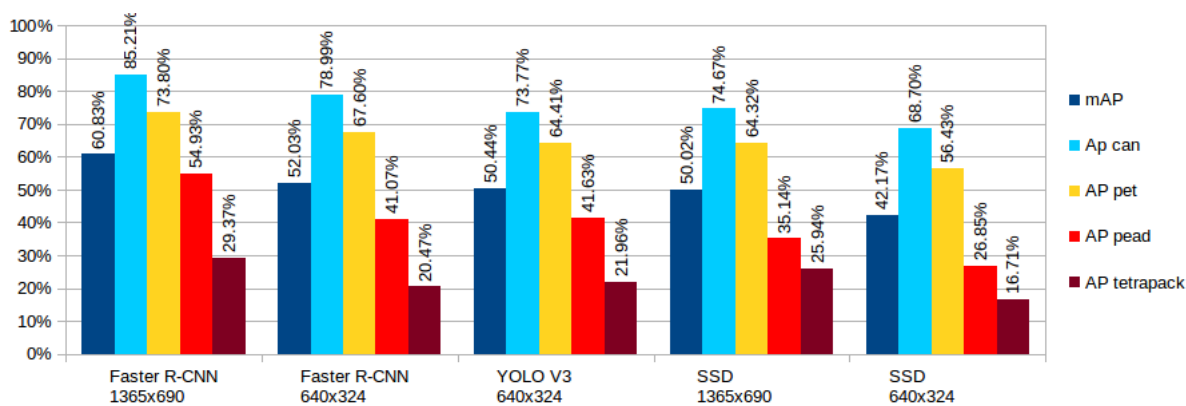


Fig. 4: AP i mAP obtingudes en els models de detecció d'objectes del dataset Ecoparc

Hem convertit aquest dataset a format PASCAL VOC, calculant totes les bounding boxes i generant les anotacions i les imatges JPG. Les dades han estat dividides de forma aleatòria en un 70% destinat a l'entrenament, un 20% per a la validació i un 10% de test. El conjunt de validació serà utilitzat per a detectar l'overfitting, avaluar el progrés de l'entrenament i modificar els hiperparàmetres. La part de test únicament serà utilitzada al final de l'entrenament per avaluar el model.

Per aquest problema hem decidit entrenar tres xarxes diferents i comparar els seus resultats. Hem utilitzat una SSD 512 i una YOLO V3, implementades en Keras i una Faster R-CNN - Inception ResNet V2, implementada en la TensorFlow Object Detection API.

Tant la SSD com la Faster R-CNN les hem entrenat amb la resolució d'imatge original de 1365×690 i amb una resolució inferior de 640×324 per determinar l'impacte que produeix la mida de les imatges en la precisió i la velocitat dels models.

Per a la YOLO V3 hem utilitzat únicament la resolució de 640×324 , ja que amb mides majors hem obtingut problemes de memòria. En la Taula 1 es poden observar els temps d'entrenament dedicats a cada model.

	1365×690	640×324
Faster R-CNN	61 h	37 h
SSD 512	105 h*	125 h*
YOLO V3	-	31 h

*L'entrenament ha estat aturat diversos cops per ajustar el learning rate

TAULA 1: Temps d'entrenament dels models Ecoparc

5.1.4.2 Resultats

Hem avaluat cada model amb el conjunt d'imatges de test. En la Figura 4 podem observar les AP i mAP obtingudes. La Faster R-CNN amb imatges de 1365×690 és el model que millor mAP obté amb un valor de 60.83%, seguida per la YOLO amb un 50.44% amb resolució 640×324 i la SSD amb un 50.02% amb resolució 1365×690 .

Trobem que totes les xarxes obtenen valors similars en cada classe, sent les llaunes l'element amb millor AP i els Tetra Briks el pitjor. Com es pot observar en la Figura 5, a l'analitzar el nombre d'imatges de què es disposa per cada classe, veiem que les imatges de Tetra Briks són les menys freqüents. Del conjunt d'entrenament en tenim únicament

1387, davant de les aproximades 5000 de PET i PEAD i les 3500 de llaunes. Aquesta falta d'imatges és una possible explicació al baix rendiment d'aquesta classe. Per a millorar-la caldria afegir més imatges reals o generar-les per mitjà de tècniques de data augmentation.

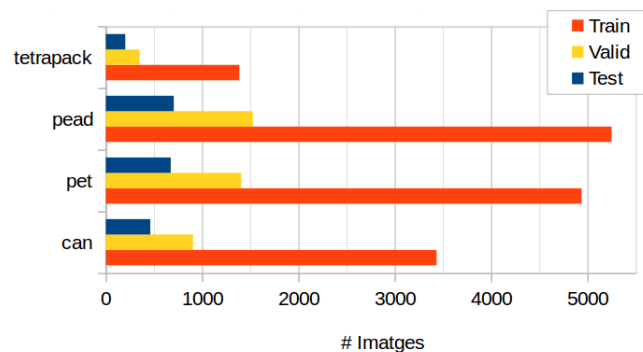


Fig. 5: Nombre d'imatges per classe del dataset Ecoparc

La segona classe amb pitjors resultats són els plàstics PEAD que, a diferència dels Tetra Briks, és la classe amb més exemples. Aquesta és possiblement la classe més complexa del problema, ja que mentre que les formes de les llaunes són bastant constants i els plàstics PET solen correspondre a ampollines de plàstic similars, en els PEAD trobem envasos plàstics de molts tipus i colors diferents que dificulten el seu aprenentatge.

Una altra mètrica que hem avaluat és la velocitat de predicció. En la Figura 6 es pot observar el nombre d'imatges processades per segon de cada model, junt amb la seva corresponent mAP. Aquests resultats s'han obtingut calculant únicament el temps dedicat a la predicció d'una imatge, múltiples vegades i realitzant la mitjana. No s'han tingut en compte altres factors com el temps de càrrega en memòria de les imatges, que no depenen de la xarxa en si.

Es pot observar com la velocitat és inversament proporcional a la precisió. La xarxa més ràpida és la SSD, que és la que pitjor mAP ha obtingut. També s'observa clarament l'impacte de la resolució de les imatges en el rendiment dels models. Una major resolució permet extraure més característiques i obtenir una millor precisió però augmenta considerablement el nombre d'operacions i el temps de processament per imatge.

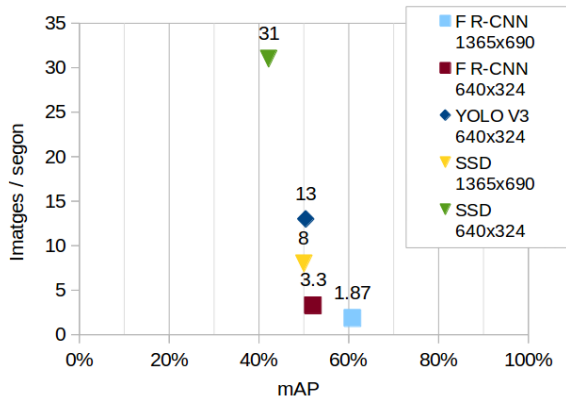


Fig. 6: Velocitat i mAP dels models de detecció d'objectes Ecomarc

5.2 Desenvolupament d'un model de predicció de mapes de profunditat

El segon objectiu d'aquest projecte és predir l'altura a la què es troben els diferents residus, utilitzant únicament imatges RGB per tal de guiar a un braç robòtic. Aquesta solució busca evitar l'ús de hardware específic com ara perfilòmetres làser o càmeres de profunditat, que requereixen una inversió i un muntatge addicional.

Per aquest fi utilitzarem una xarxa de segmentació U-Net. La seva arquitectura es divideix en dues etapes [7]. La primera, de contracció, consisteix en una sèrie de capes convolucionals que extreuen progressivament les característiques de les imatges, aportant una representació comprimida del seu contingut. La segona, d'expansió, realitza el procés invers, utilitzant el mateix nombre de capes convolucionals per a generar una imatge nova, on els valors de cada píxel aporten un nou significat.

Aquesta xarxa sol ser utilitzada per a la segmentació d'imatges, en la que els valors de sortida esperats són discrets i representen la classe a què pertany cada píxel. Per aquest motiu la funció d'activació de l'última capa sol ser una sigmoide, on la sortida es troba entre 0 i 1. Com en el nostre cas esperem obtenir valors continus positius, utilitzarem la funció d'activació ReLU [11].

5.2.1 Mètriques d'avaluació

Per avaluar els resultats obtinguts utilitzarem dues mètriques diferents:

- IoU: La Intersection over Union serà utilitzada per avaluar la segmentació dels objectes. Aquesta ens indicarà quant semblant és la forma generada dels objectes amb el ground truth, sense tindre en compte la predicció de l'altura. La funció utilitzada per a calcular-la és la següent:

$$\text{IoU} = \frac{\sum(Y_b \cap \hat{Y}_b)}{\sum(Y_b \cup \hat{Y}_b)},$$

on Y_b i \hat{Y}_b són les imatges ground truth i predita respectivament, a les quals s'ha aplicat un threshold i els seus valors són 0 o 1.

- MAE: És l'error mitjà absolut, que ens indicarà la diferència existent entre la imatge predita i el ground truth. Com en el nostre cas els valors dels píxels representaran l'altura dels objectes en mil·límetres, la MAE indicarà la distància mitjana d'error que tenen les imatges. La funció utilitzada és la següent:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|,$$

on Y_i i \hat{Y}_i són els píxels de les imatges ground truth i predita respectivament i n és el nombre de píxels que té la imatge.

5.2.2 Experiment amb imatges de porcs

Abans d'entrar en el problema dels residus hem realitzat el mateix experiment amb unes dades diferents per a determinar la viabilitat d'aquesta idea i configurar el model. Hem utilitzat imatges de porcs capturades amb una Kinect V2 en una granja, en les quals apareixen exemplars de diferents mides de perfil. L'objectiu és poder predir la forma i la distància a la que es troben de la càmera, de forma anàloga al cas dels residus.

Les captures de la Kinect generen dos tipus d'imatges diferents. Per una banda, imatges de color RGB i per l'altra, imatges de profunditat. Aquestes últimes estan codificades en sencers de 16 bits i cada píxel indica la distància en mil·límetres a la càmera.

Hem utilitzat aquestes imatges ja que les formes dels porcs són més constants que les dels residus i resulten més fàcils d'avaluar.

5.2.2.1 Desenvolupament

Per entrenar la xarxa hem necessitat preparar les dades. El primer pas que hem realitzat ha estat eliminar el soroll d'algunes de les imatges de profunditat. En aquestes únicament han d'aparèixer les distàncies dels porcs, tot i això en certs casos apareixen parts del fons. Per eliminar-lo hem utilitzat la llibreria OpenCV per a trobar els contorns dels elements de la imatge, hem buscat el de major àrea (corresponent al contorn del porc) i s'ha eliminat tot el contingut fora d'aquesta àrea.

El segon pas ha estat posar en correspondència les imatges de color i de profunditat, ja que la xarxa requereix que la seva entrada i el ground truth siguin equivalents. La Kinect utilitza dues càmeres diferents per a generar els dos tipus d'imatges, de manera que a priori aquestes dues no coincideixen. Per fer la correspondència s'ha utilitzat el SDK de la Kinect amb els paràmetres de les càmeres i s'han generat unes imatges de profunditat projectades des del punt de vista de la càmera de color. Finalment s'han destinat 1427 imatges a l'entrenament i 307 al test.

A l'hora d'entrenar el model hem realitzat fins a 23 entrenaments diferents, buscant els hiperparàmetres i la configuració de la xarxa que permetessin obtenir uns resultats vàlids. A continuació es mostren alguns dels punts més importants que hem extret d'aquestes proves.

- El learning rate òptim es troba pròxim a 1.0e-6.



Fig. 7: Mostra del dataset de porcs: (a) imatge RGB (b) ground truth (c) predicció

- Els píxels de les imatges de profunditat solen tenir valors entre 1400 i 2200. A l'entrar aquestes imatges a la xarxa hem trobat que dividir-les entre 100 produïa els millors resultats.
- En molts casos, a l'iniciar l'entrenament el model no entrenava correctament degut als valors de la inicialització dels pesos. En aquests casos ha estat necessari aturar i iniciar de nou l'entrenament múltiples vegades. Per detectar aquest problema hem implementat dues solucions. Primerament hem programat una funció que a l'acabar cada epoch, genera una predicció d'una imatge de mostra, per a poder veure de forma gràfica l'evolució de l'entrenament. Per l'altra banda s'ha afegit una mètrica a l'entrenament que calcula la mitjana de les imatges. Això ens ha permès detectar quan l'entrenament generava imatges pràcticament buides i poder aturar-lo a temps.
- Per últim, la funció de loss que vam utilitzar inicialment i que a priori ens va aportar uns bons resultats va ser la MSE (error quadràtic mitjà). Tot i això vam trobar que la funció berHu [12] produïa uns resultats considerablement millors. Aquesta funció és el cas invers a la funció de loss Huber, la qual aplica un error quadràtic o absolut en funció de l'error en cada píxel. berHu permet penalitzar millor les distàncies petites que la MSE i és més adequada per aquest tipus de problemes [6]. Hem implementat aquesta funció tal com es mostra a continuació, ja que actualment no es troba inclosa per defecte en la llibreria Keras. En el següent apartat hem volgut mostrar algunes de les diferències obtingudes utilitzant ambdues funcions loss.

$$\text{berHu}(y, \hat{y}) = \begin{cases} |y - \hat{y}| & |y - \hat{y}| \leq c \\ \frac{(y - \hat{y})^2 + c^2}{2c} & |y - \hat{y}| > c \end{cases}$$

$$c = \frac{1}{5} \max_i (|y_i - \hat{y}_i|),$$

y és la imatge ground truth, \hat{y} la imatge predita i i indexa els píxels de la imatge.

Finalment, durant l'entrenament s'ha aplicat data augmentation per mitjà de simetries horitzontals i l'entrada i sortida de la xarxa s'ha establert a 512×512 .

5.2.2.2 Resultats

En la Figura 7 es pot observar una mostra d'una imatge RGB i els seus corresponent ground truth i predicció. Els valors més clars de les imatges de profunditat corresponen

als valors més alts, i per tant més llunyans de la càmera. De forma visual es pot observar com la segmentació és bastant precisa i s'aprecia l'efecte de la profunditat, sobretot en la part posterior on trobem valors més clars que les parts frontals.

Analizant la IoU de les imatges de test, trobem un valor mitjà de 0.93 amb una desviació estàndard de 0.023, el qual indica una precisió relativament bona pel que fa a la segmentació.

L'error absolut mitjà (MAE) és de 164.48 mm amb una desviació estàndard de 43.06 mm. L'error que vam obtenir utilitzant la funció de loss MSE va resultar pitjor, 318.57 mm de mitjana i amb una desviació de 68.92 mm. En la Figura 8 es pot observar l'error obtingut en cada imatge amb ambdues funcions de loss. S'observa com la funció berHu aporta una millora significativa en aquest problema.

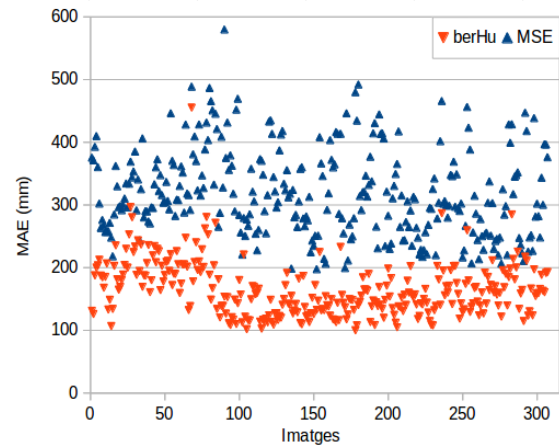


Fig. 8: Errors absoluts mitjans per imatge obtinguts amb les funcions loss MSE i berHu

Per últim també hem avaluat la capacitat del model per a predir la distància entre els porcs i la càmera. En aquest cas hem mesurat la mitjana de tots els píxels diferents de zero per a trobar la distància a la que estaven situats.

En la Figura 9 s'han representat les distàncies reals i predites, ordenades de forma ascendent per les distàncies reals. Com es pot observa, utilitzant el loss berHu s'ha aconseguit també aproximar les distàncies entre el porc i la càmera, a diferència de la funció MSE on no s'aprecia cap relació entre els valors reals i predits.

Finalment, amb els resultats obtinguts, podem concloure que tot i que existeix un cert error, aquest model permet predir tant la forma com la distància de determinats objectes coneguts, únicament amb imatges individuals de color.

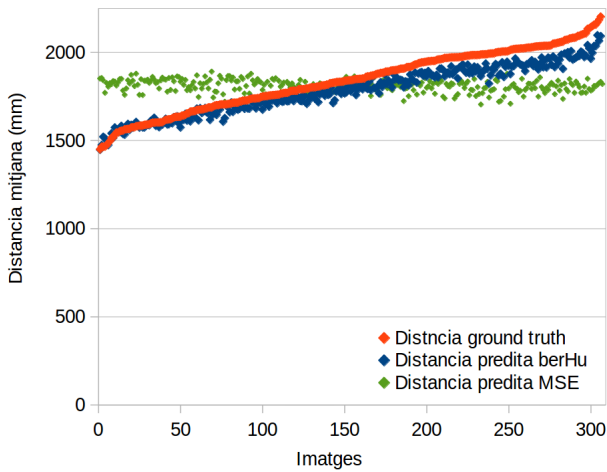


Fig. 9: Distàncies dels porcs a la càmera reals i predites

5.2.3 Experiment Ecoparc

Un cop configurada la xarxa i comprovat el seu funcionament, hem realitzat el mateix experiment amb les imatges de l'Ecoparc.

En aquest cas les imatges de profunditat han estat capturades amb un perfilòmetre làser i cada píxel indica l'altura dels elements en mil·límetres, prenent com a referència la cinta transportadora.

5.2.3.1 Desenvolupament

Per preparar les dades ha estat necessari aplicar una translocació i un escalat per tal de posar en correspondència les imatges de color i profunditat. Hem destinat 632 imatges a l'entrenament i 161 al test.

Per l'entrenament hem utilitzat uns hiperparàmetres similars a l'experiment anterior. La mida de les imatges s'ha establert a 1024×1024 i s'ha realitzat data augmentation per mitjà de simetries horitzontals. L'entrenament ha durat un total de 11 dies, a causa de l'alta resolució de les imatges i a un possible learning rate massa baix, el qual ha estat modificat durant l'entrenament.

5.2.3.2 Resultats

En la Figura 10 podem observar una imatge augmentada de la predicció generada per la xarxa i el seu corresponent ground truth.

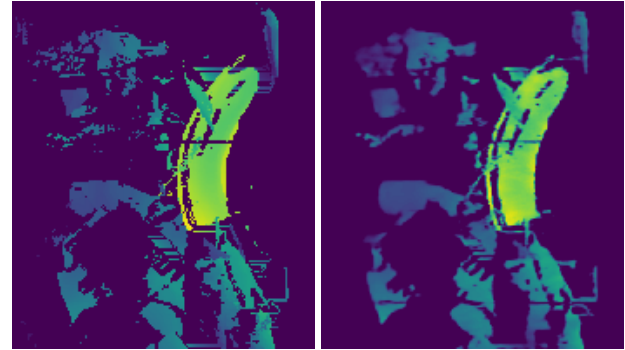
Troblem que les imatges generades tenen un cert error en els perímetres dels objectes, els quals es troben poc definits i s'aprecia un cert difuminat. Aquest error és poc significatiu, ja que únicament es troba localitzat en les fronteres, tot i això, per avaluar els resultats hem establert un límit inferior de 10 mm, per tal d'eliminar els valors excessivament baixos introduïts entre el fons i els objectes.

La Intersection over Union mitjana és de 0.71 amb una desviació de 0.089, que indica la presència d'un lleuger error en les formes predites. L'error mitjà absolut és de 11.36 mm amb una desviació de 4.25 mm. Tenint en compte que l'altura mitjana dels residus és de 45 mm, podent arribar fins als 130 mm, aquest és un error acceptable.

Finalment hem representat l'altura mitjana dels residus de les imatges ground truth i de les prediccions.



(a) RGB



(b) Ground Truth

(c) Predicció

Fig. 10: Predicció i ground truth del mapa de profunditat d'una mostra del dataset Ecoparc

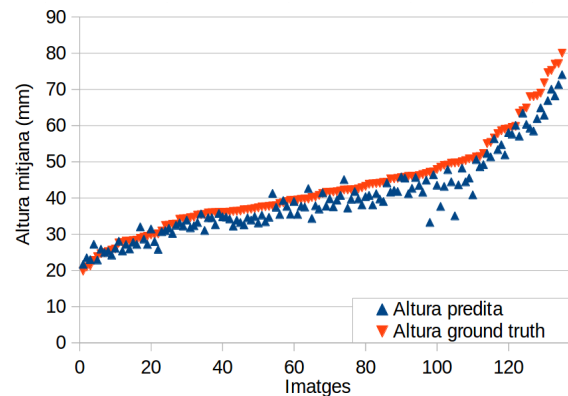


Fig. 11: Altures mitjanes dels residus per imatge

En la Figura 11 podem observar com existeix una clara relació entre ambdós valors, de manera que podem concloure que el model és capaç d'aproximar l'altura dels residus en tots els casos avaluats.

5.3 Desenvolupament d'un model de classificació d'imatges hiperespectrals

L'últim objectiu d'aquest projecte és analitzar els possibles beneficis que aporta la utilització d'imatges hiperespectrals a l'hora de detectar i classificar diferents materials. Amb aquest propòsit hem buscat construir un model capaç de classificar els residus per mitjà únicament dels valors dels píxels d'aquestes imatges.

5.3.1 Desenvolupament

A les instal·lacions de l'Ecoparc s'han pres imatges dels residus amb dues càmeres hiperespectrals. Una d'elles (FX10) captura imatges de 112 bandes, sensibles a la llum

d'entre 398 nm i 1002 nm de longitud d'ona, que compren tot l'espectre visible i part de l'infraroig proper. L'altra càmera (FX17) pren imatges de 224 bandes, entre els 935 nm i els 1720 nm, que representen part de l'infraroig. S'han seleccionat 24 tipus de residus amb composicions diferents i s'han pres imatges de forma individual per a facilitar el seu anàlisi.

El primer pas que hem realitzat ha estat extreure els valors dels píxels dels residus en forma de vectors. Hem generat una imatge d'un sol canal que pugui ser visualitzada, calculant la mitjana de totes les bandes i situant els seus valors entre 0 i 255.

A continuació hem calculat una imatge mitjana de la cinta transportadora buida, que s'ha utilitzat per a separar el fons dels residus. Utilitzant un threshold hem calculat una màscara que determina la seva àrea ocupada. Per a reduir el nombre de mostres extretes hem seleccionat de forma aleatòria diferents punts dins d'aquesta màscara.

Finalment, a partir de les coordenades d'aquests punts, hem extret els valors de cada canal de la imatge hiperespectral i hem guardat aquest vector en un fitxer CSV, junt amb el nom de la imatge i la seva classe. Aquest procés s'ha realitzat per a cada un dels dos tipus d'imatges hiperespectrals de què disposem.

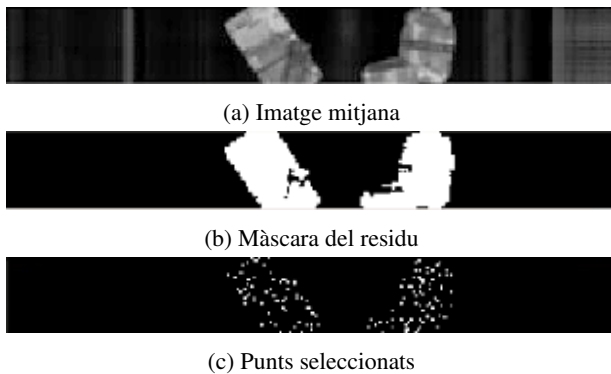


Fig. 12: Visualització del procés de selecció de mostres de les imatges hiperespectrals

Per a poder visualitzar la distribució d'aquests vectors hem utilitzat un PCA i un TSNE, dos algorismes de reducció de dimensionalitat que busquen una transformació que maximitzi la variància i que millor visualitzi la distribució de les dades en tan sols dues dimensions. Hem aplicat aquests dos algorismes als dos tipus d'imatges hiperespectrals que tenim. Per a facilitar la seva visualització hem utilitzat únicament tres classes.

Els millors resultats els hem obtingut amb l'algorisme TSNE amb imatges hiperespectrals infraroges (Figura 13), on podem veure com existeix una clara separació entre les diferents classes. Això ens indica que dins el conjunt de dades hi ha grups separables a partir dels quals podem diferenciar els diferents tipus de residus. En la secció A.4 de l'apèndix es poden trobar la resta de resultats de la reducció de dimensionalitat.

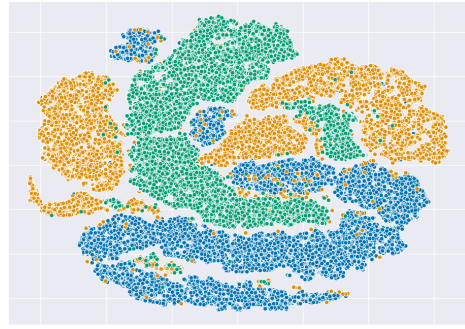


Fig. 13: TSNE aplicat als píxels d'imatges hiperespectrals infraroges. Cada color representa una classe diferent

Per a classificar aquestes dades hem creat un multilayer perceptron (MLP), una xarxa neuronal de quatre capes on totes les neurones de cada capa estan connectades amb totes les de la següent. Hem entrenat la xarxa utilitzant valors de les imatges hiperespectrals visibles, infraroges i ambdues alhora. El nombre de neurones per capa s'ha adaptat al nombre de canals de cada tipus d'imatge. La seva estructura és la següent:

Capa	# Neurones			Activació
	Visible	IR	Visible+IR	
dense_1	128	256	512	ReLU
dense_2	128	256	512	ReLU
dense_3	64	128	256	ReLU
dense_4	24	24	24	SoftMax

TAULA 2: Estructura de la xarxa de classificació hiperespectral

A l'hora d'entrenar la xarxa, l'ordre de les dades ha estat aleatoritzat, s'han estandarditzat i s'han destinat un 70% a l'entrenament (550000 visibles i 700000 infraroges), un 20% per a la validació (150000 visibles i 200000 infraroges) amb l'objectiu de controlar l'overfitting i un 10% al test (78000 visibles i 99000 infraroges) amb les que hem avaluat el resultat final. La funció de loss utilitzada ha estat l'entropia creuada categòrica [13].

5.3.2 Resultats

Per avaluar el model hem utilitzat la mètrica F1 [10], que avalua la classificació de cada classe.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Aquesta representa millor el rendiment del model que d'altres com l'Accuracy, sobretot quan el nombre de mostres per classe no està balancejat (com és el nostre cas).

La F1 mitjana obtinguda de les 24 classes de les imatges visibles és de 0.8154, la de les imatges infraroges 0.8762 i la de les visibles combinades amb les infraroges 0.9275. Dins la secció A.4 de l'apèndix podem trobar la Taula 3 amb els valors F1 obtinguts en cada classe. Observem com les imatges infraroges permet classificar millor que les visibles en 20 de les 24 classes. Al combinar ambdós tipus d'imatges els resultats han millorat generalment, sent aquesta la millor de les tres opcions provades.

Podem concloure que les imatges infraroges, a causa de les longituds d'ona que enregistra i al major nombre de ban-

des, permeten classificar millor les diferents classes de residus en aquest problema en concret. Tot i això es pot observar com existeixen diferents materials on l'espectre visible és més idoni per a classificar-los. Per aquest motiu la combinació d'ambdós tipus d'imatge aporta major informació i permet classificar-los millor.

6 CONCLUSIONS

Durant el transcurs d'aquest projecte hem analitzat l'ús de diferents tecnologies i xarxes neuronals, enfocades a millorar el procés de classificació d'un Ecoparc. Amb els resultats obtinguts, creiem que les solucions vistes anteriorment poden ser aplicades en un entorn real o bé establir una base per a un futur desenvolupament.

Amb relació al primer objectiu d'aquest projecte, hem pogut comparar satisfactòriament el rendiment de diferents xarxes neuronals de detecció d'objectes que poden ser utilitzades en funció dels requisits de precisió i velocitat de cada problema.

En ambdós casos estudiats, els models entrenats han permès localitzar i classificar els diferents elements amb un grau d'encert alt. En el cas de l'Ecoparc s'han obtingut uns resultats més baixos que en l'experiment Smart Market, degut a la gran complexitat de les imatges. Creiem que amb més dades i més temps, aquests podrien millorar en un cert grau.

Sobre el segon objectiu del projecte, on es buscava predir un mapa de profunditat, podem concloure que el model proposat ha funcionat segons l'esperat. Els resultats obtinguts han demostrat que aquest permet conèixer de forma aproximada la posició dels objectes en l'espai 3D.

El fet d'entrenar aquest model amb dos conjunts de dades diferents ens ha permès en un primer moment, crear i provar diferents configuracions amb un dataset senzill i a continuació, quan els resultats eren prou bons, entrenar un problema molt més complex.

Finalment, amb les imatges hiperespectrals hem pogut observar com aquestes aporten molta més informació que les imatges de color convencionals i com, per mitjà d'una xarxa neuronal relativament senzilla, s'han pogut classificar un gran nombre de residus diferents.

Com a treball futur creiem que seria interessant analitzar la importància de cada longitud d'ona a l'hora de detectar determinats materials, per tal d'elegir càmeres amb un nombre de bandes menor i així reduir l'alt cost d'aquest equipament. També creiem que combinar aquesta tecnologia amb la detecció d'objectes vista anteriorment permetria conèixer la posició dels residus i obtenir una major precisió en la classificació.

AGRAÏMENTS

Al meu tutor, Felipe Lumbreras i a Coen Antens (CVC) per oferir-me l'oportunitat de treballar en aquest projecte i pel suport rebut durant aquests mesos. Al CVC i tot l'equip del laboratori per l'ajuda, l'experiència i el coneixement que m'han transmès. A la meua família i amics per donar-me ànims durant el transcurs d'aquest treball i al projecte BOSS TIN2017-89723-P.

REFERÈNCIES

- [1] J. Hui, "Object detection: speed and accuracy comparison." [En línia] https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-rcnn-ssd-and-yolo-5425656ae359 [Darrer accés: 06/10/2019].
- [2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." [En línia] <https://arxiv.org/abs/1506.01497>.
- [3] S. Bianco, R. Cadene, L. Celona, and P. Napolitano, "Benchmark Analysis of Representative Deep Neural Network Architectures." [En línia] <https://arxiv.org/abs/1810.00736>.
- [4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector." [En línia] <https://arxiv.org/abs/1512.02325>.
- [5] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." [En línia] <https://arxiv.org/abs/1804.02767>.
- [6] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper Depth Prediction with Fully Convolutional Residual Networks." [En línia] <https://arxiv.org/abs/1606.00373>.
- [7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation." [En línia] <https://arxiv.org/abs/1505.04597>.
- [8] "Hyperspectral imaging." [En línia] https://en.wikipedia.org/wiki/Hyperspectral_imaging [Darrer accés: 19/01/2020].
- [9] "The pascal visual object classes homepage." [En línia] Disponible: <http://host.robots.ox.ac.uk/pascal/VOC/> [Darrer accés: 21/11/2019].
- [10] W. Koehrsen, "Beyond Accuracy: Precision and Recall." [En línia] <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c> [Darrer accés: 21/01/2020].
- [11] D. Liu, "A practical guide to relu." [En línia] <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7> [Darrer accés: 18/12/2019].
- [12] L. Zwald and S. Lambert-Lacroix, "The Ber-Hu penalty and the grouped effect." [En línia] <https://arxiv.org/abs/1207.6868>.
- [13] R. Gómez, "Understanding categorical cross-entropy loss, binary cross-entropy loss, softmax loss, logistic loss, focal loss and all those confusing names." [En línia] https://gombu.github.io/2018/05/23/cross_entropy_loss/ [Darrer accés: 29/01/2020].

APÈNDIX

A.1 Entorn d'execució

L'equip on hem entrenat i executat tots els experiments té la següent configuració:

- **CPU:** Intel Xeon E5-1620 v4 @ 3.50GHz
- **GPU:** NVIDIA GTX 1080 Ti (11 GB)
- **RAM:** 32 GB
- **Emmagatzematge:** Crucial SSD MX500 2TB
- **SO:** Ubuntu 16.04

En l'apartat de software hem utilitzat el següent entorn:

- **Python3.5:** És el principal llenguatge de programació que hem utilitzat. Aquest ens ha permès desenvolupar ràpidament eines per a processar i avaluar les dades i entrenar els models de deep learning. Algunes de les principals llibreries utilitzades han estat: OpenCV, Numpy, Scikit-learn i Matplotlib entre d'altres.
- **TensorFlow:** És un framework open source desenvolupat per Google, que hem utilitzat per a crear i entrenar xarxes neuronals utilitzant la targeta gràfica. Una de les eines que incorpora i que s'ha utilitzat per a monitoritzar en temps real diferents mètriques dels entrenaments és el TensorBoard.
- **Keras:** És una llibreria per a Python que aporta un nivell superior d'abstracció a l'hora d'utilitzar TensorFlow i que incorpora diferents utilitats i classes que permeten crear models de forma més ràpida i eficient i que hem utilitzat en la majoria de les xarxes neuronals d'aquest projecte.
- **Docker:** Permet crear entorns portables i aïllats de la resta del sistema anomenats contenidors. En el nostre cas hem entrenat les xarxes en un contenidor on hem instal·lat totes les llibreries i dependències, per tal de tenir un major control sobre el software que utilitzem. Hem usat la versió Nvidia-docker, ja que ofereix un millor suport per a targetes gràfiques.

A.2 Xarxes neuronals de detecció d'objectes

Visualització de les prediccions d'imatges del conjunt de test del dataset Smart Market i Ecoparc. Es mostra la bounding box, la classe i la confiança de cada objecte.



Fig. 14: Exemple de predicció d'una mostra del dataset Smart Market amb la YOLO V3

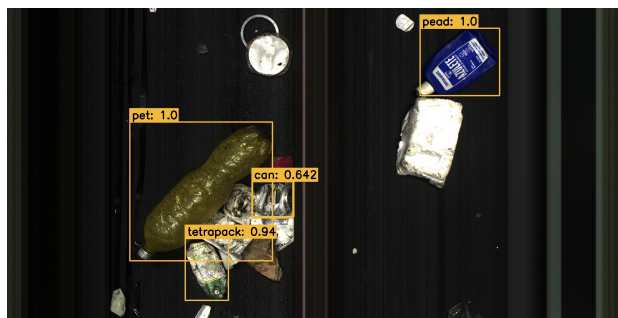


Fig. 15: Exemple de predicció d'una mostra del dataset Ecoparc amb la Faster R-CNN



Fig. 16: Exemple de predicció d'una mostra del dataset Ecoparc amb la YOLO V3



Fig. 17: Exemple de predicció d'una mostra del dataset Ecoparc amb la SSD

Evolució del loss de les xarxes de detecció d'objectes entrenades. Els pics que s'observen en les figures 19d i 19e corresponen als moments d'aturar i iniciar la xarxa per a ajustar el learning rate.

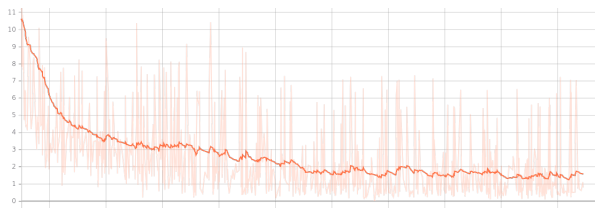


Fig. 18: Evolució del loss de la xarxa YOLO V3 amb el dataset Smart Market

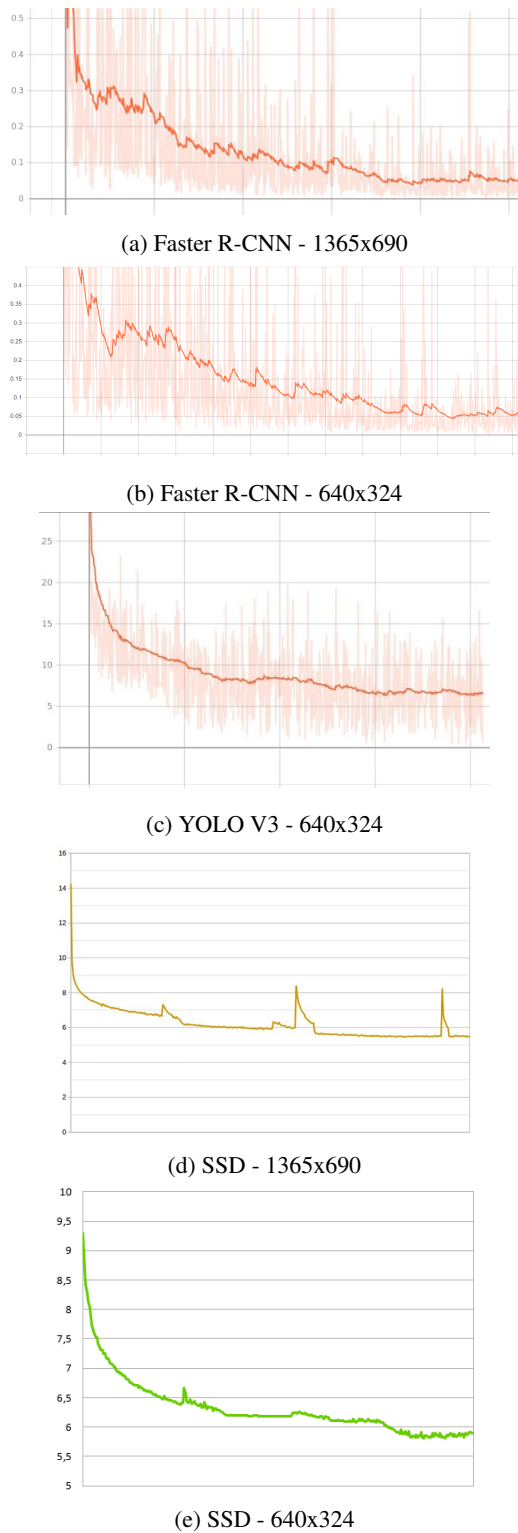


Fig. 19: Evolució del loss de les xarxes de detecció d'objectes utilitzades en el dataset Ecoparc

A.3 Predicció de mapes de profunditat

Hem generat un núvol de punts 3D amb els mapes de profunditat de les imatges dels porcs. Les components x i y corresponen a les coordenades de la imatge i la z al valor dels píxels. A continuació es mostra una representació d'aquests punts des de diferents punts de vista.

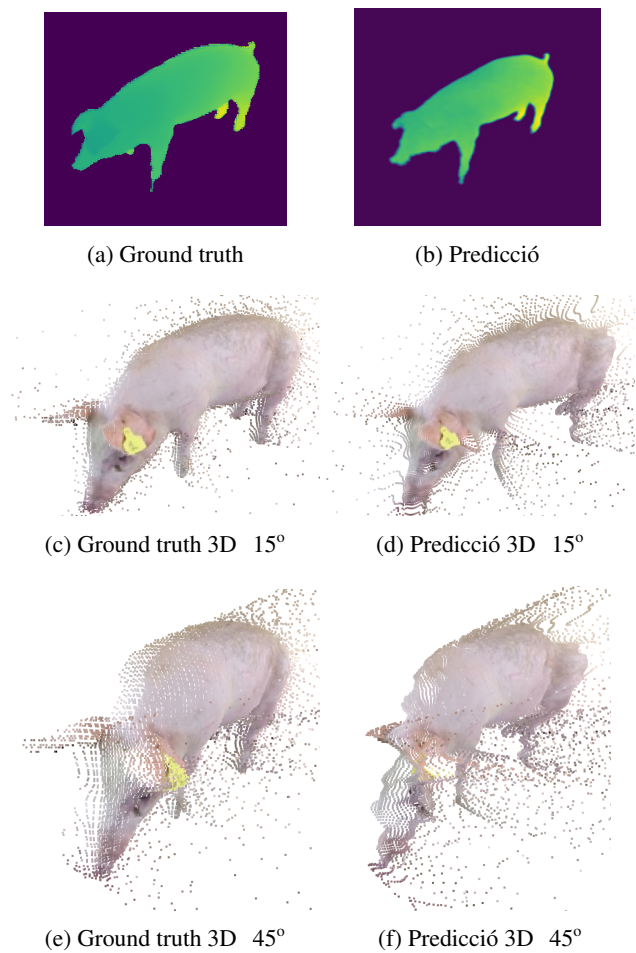


Fig. 20: Visualització 3D dels mapes de profunditat d'una mostra del dataset de porcs

A.4 Imatges hiperespectrals

Representació dels resultats obtinguts en la reducció de dimensionalitat de les imatges hiperespectrals.

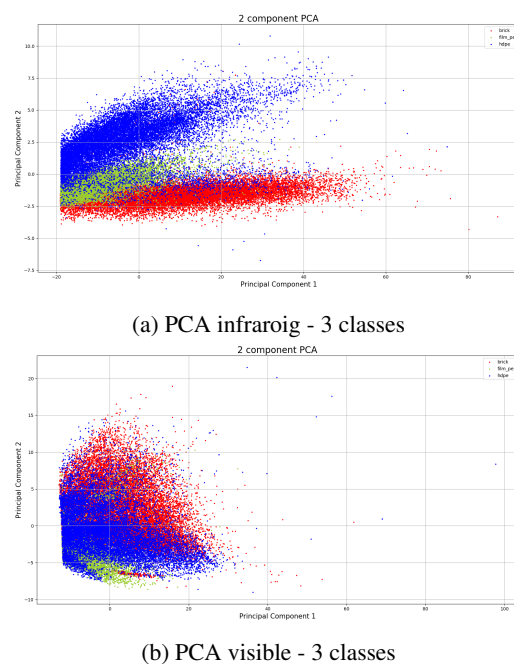
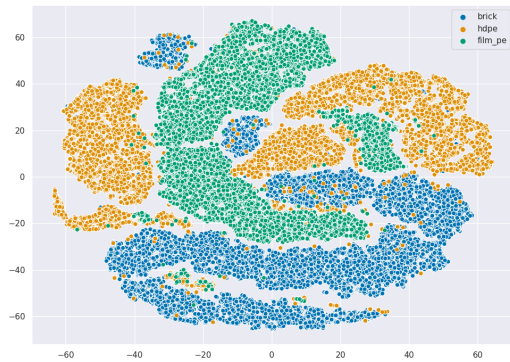
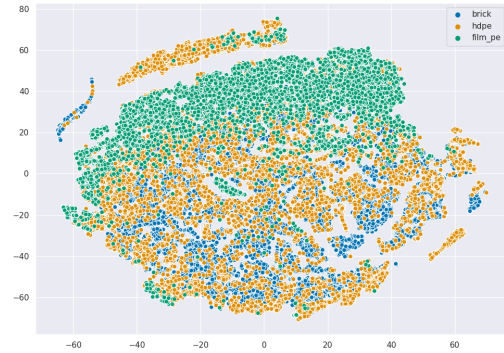


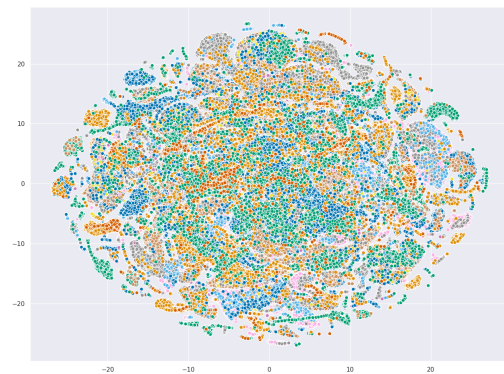
Fig. 21: PCA aplicat als valors extrets d'imatges hiperespectrals



(a) TSNE infraroig - 3 classes



(b) TSNE visible - 3 classes



(c) TSNE infraroig - 24 classes

Taula amb els resultats de la classificació d'imatges hiperespectrals amb una xarxa neuronal. Es mostra la F1 obtinguda en cada classe amb les imatges visibles (càmera FX10), infraroges (càmera FX17) i ambdues combinades. En negreta s'indiquen els valors més alts.

Classe	F1-score		
	Visible	IR	Visible+IR
brick	0,91	1	1
carton_color	0,82	0,76	0,86
carton_marron	0,94	0,86	0,95
cartoncillo	0,7	0,81	0,84
film_pe	0,88	0,98	0,98
film_pp	0,67	0,91	0,93
hdpe	0,96	0,98	0,98
hdpe_tricapa	0,86	0,95	0,97
impropio_ferrico	0,39	0,57	0,8
impropio_madera	0,81	0,97	0,95
impropio_no_ferrico	0,69	0,55	0,68
impropio_organico_hojas	0,99	1	1
impropio_textil_algodon	0,97	1	1
latas_metal_ferrico	0,83	0,88	0,96
latas_metal_no_ferrico	0,89	0,9	0,95
papel	0,87	0,9	0,93
pet_bandeja_monocapa	0,76	0,88	0,89
pet_bandeja_multi	0,86	0,96	0,95
pet_botella_azulado	0,88	0,65	0,9
pet_botella_color	0,83	0,88	0,95
pet_botella_light	0,72	0,73	0,88
pp	0,84	0,97	0,97
ps	0,78	0,98	0,98
pvc	0,72	0,96	0,96

TAULA 3: F1 obtingudes en la classificació d'imatges hiperespectrals

Fig. 22: TSNE aplicat als valors extrems d'imatges hiperespectrals

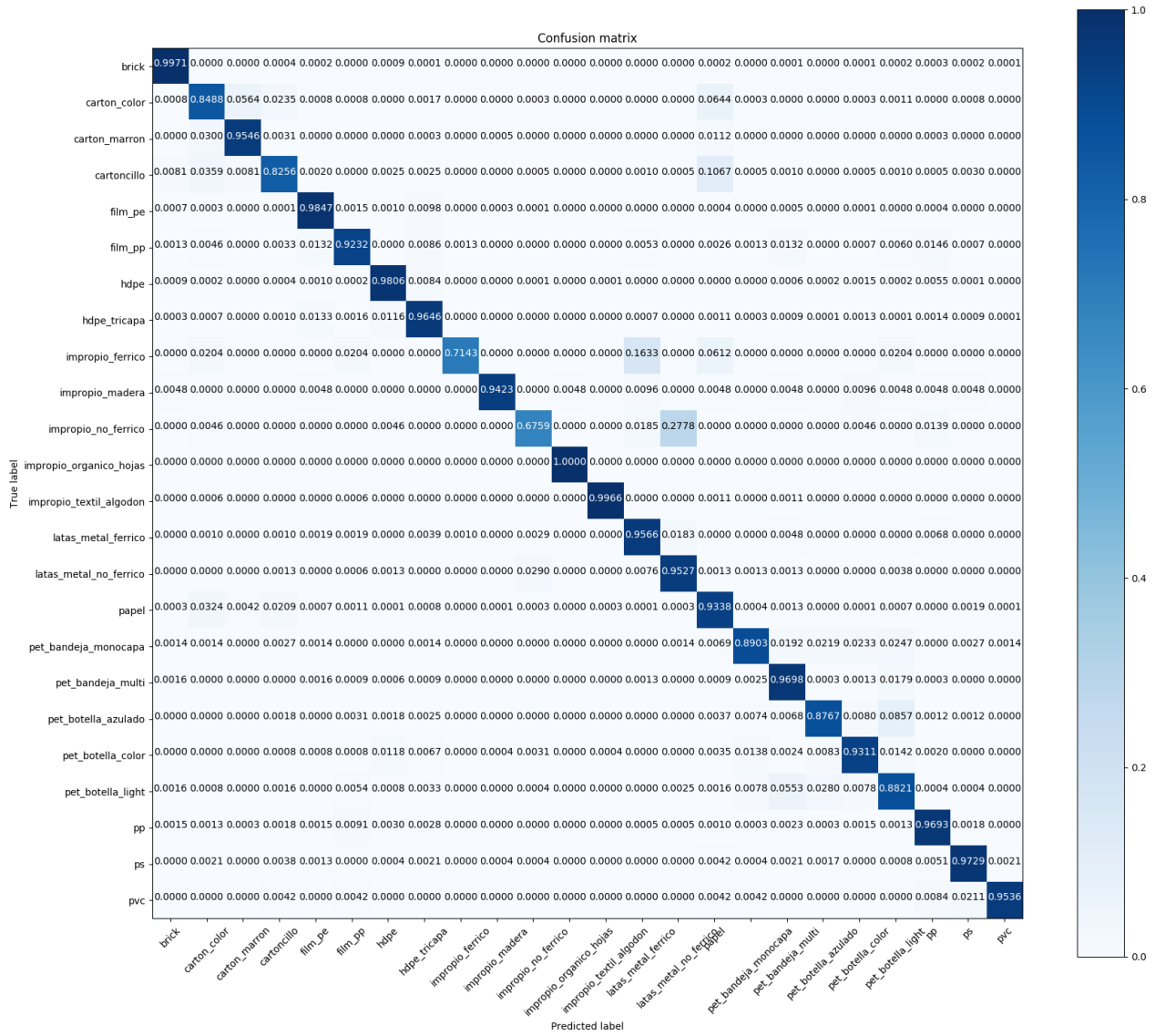


Fig. 23: Matriu de confusió normalitzada obtinguda amb el conjunt de Test amb les imatges visibles i infraroges combinades, de la xarxa neuronal de classificació d'imatges hiperspectrals