



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



LANE CONDITION ANALYSIS FOR MICRO-MOBILITY APPLICATIONS

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

By

Miquel Torrecilla Mercado

In partial fulfilment

of the requirements for the degree in

**SCIENCES AND TECHNOLOGIES OF TELECOMMUNICATIONS
ENGINEERING**

Advisor:

Josep Ramon Morros Rubió, Elisa Sayrol Clois

June 21, 2021

Abstract

Due to the increase in population in large urban areas, mobility in large cities has become increasingly problematic and difficult. With the desire to find solutions to this situation, the use of micro-mobility vehicles has become popular.

In this sense the solution presented proposes to make use of Deep Learning and Computer Vision techniques in order to identify, automatically, possible defects where micro-mobility vehicles circulate, in order to alert authorities and other users and improve user safety.

Three solutions were devised to determine which one best suited the needs of the project. The first two to test the two models separately and the third one with the intention of combining them, with ensemble learning techniques, to try to improve the results obtained.

Resum

Degut a l'increment de la població en els grans nuclis urbans, la mobilitat a les grans ciutats s'ha tornat cada cop més problemàtica i difícil. Amb l'afany de trobar solucions a aquesta situació, s'ha popularitzat l'ús de vehicles de micro-mobilitat.

En aquest sentit la solució que es presenta proposa fer ús de tècniques de *Deep Learning* i *Computer Vision* per identificar possibles defectes a les carreteres de forma automàtica i millorar la seguretat dels usuaris.

Es proposen tres solucions per determinar quin és el model que s'adapta millor a les necessitats del projecte. Els dos primers per provar dos models per separat i el tercer amb la intenció d'ajuntar-los, amb tècniques d'*ensemble learning*, per intentar millorar els resultats obtinguts.

Resumen

Debido al incremento de la población en los núcleos urbanos, la movilidad en las grandes ciudades se ha vuelto cada vez más problemática y difícil. Con el afán de encontrar soluciones a esta situación, se ha popularizado el uso de vehículos de micro-movilidad.

En este sentido la solución que se presenta propone hacer uso de técnicas de *Deep Learning* y *Computer Vision* para identificar posibles defectos en las carreteras de forma automática y mejorar la seguridad de los usuarios.

Se proponen tres soluciones para determinar cuál es el modelo que se adapta mejor a las necesidades del proyecto. Los dos primeros para probar los modelos por separado y el tercero con la intención de juntarlos, con técnicas de *ensemble learning*, para intentar mejorar los resultados obtenidos.



La feina que ha estat desenvolupada en aquest treball no hauria estat possible sense la ajuda de tres persones. En primer lloc, en Josep Ramón Morros, qui m'ha donat la oportunitat de formar part d'aquest projecte i gràcies a la seva dedicació i ajuda, s'ha pogut desenvolupar adequadament. També agrair a l'Elisa Sayrol, qui també m'ha donat la seva confiança per poder aportar els meus coneixements al projecte.

M'agradaria també agrair-li al meu company, Jaume Prats, sense la ajuda del qual hagués estat molt complicat treure el projecte endavant.

Finalment, agrair als meus companys, amics i familiars que m'han donat suport incondicional durant tota la meva etapa universitària.

Agraïments

Aquest projecte s'ha desenvolupat amb la supervisió dels professors Josep Ramon Morros Rubió i Elisa Sayrol Cloles. Amb els que hem tingut reunions de forma regular i han aportat informació biogràfica i grans idees al projecte.

Historial de revisions i registres d'aprovació

| Revision | Date | Purpose |
|----------|------------|-------------------|
| 0 | 27/05/2021 | Document creation |
| 1 | 16/06/2021 | Document revision |
| 2 | 18/06/2021 | Document Revision |
| | | |
| | | |

DOCUMENT DISTRIBUTION LIST

| Name | e-mail |
|---------------------------|-------------------------------------|
| Miquel Torrecilla Mercado | miquel.torrecilla.mercado@gmail.com |
| Josep Ramon Morros Rubió | ramon.morros@upc.edu |
| Elisa Sayrol Clols | Elisa.sayrol@upc.edu |

| Written by: | | Reviewed and approved by: | |
|-------------|---------------------------|---------------------------|--------------------------|
| Date | 27/05/2021 | Date | 21/06/2021 |
| Name | Miquel Torrecilla Mercado | Name | Josep Ramon Morros Rubió |
| Position | Project Author | Position | Project Supervisor |

Taula de continguts

| | |
|---|----|
| Abstract..... | 1 |
| Resum | 2 |
| Resumen | 3 |
| Agraïments | 5 |
| Historial de revisions i registres d'aprovació | 6 |
| Taula de continguts | 7 |
| Llistat de figures..... | 9 |
| Llistat de taules:..... | 10 |
| 1. Introducció..... | 11 |
| 1.1. Objectius del projecte..... | 11 |
| 1.2. Requeriments i especificacions..... | 11 |
| 1.3. Mètodes i Procediments | 11 |
| 1.4. Planificació..... | 12 |
| 1.5. Desviacions del pla original | 14 |
| 2. Estat de l'art de la tecnologia utilitzada o aplicada en el projecte:..... | 15 |
| 2.1. Detecció de defectes a les carreteres..... | 15 |
| 2.2. Detecció d'objectes | 15 |
| 2.2.1. Faster R-CNN ^[4] | 16 |
| 2.2.2. Cascade R-CNN ^[5] | 16 |
| 2.2.3. YOLO ^[6] | 17 |
| 2.3. Global Road Damage Detection Challenge (GRDDC) ^[9] | 17 |
| 3. Metodologia:..... | 19 |
| 3.1. Dataset | 19 |
| 3.2. Entrenament..... | 21 |
| 3.3. Non-maxima suppression..... | 22 |
| 3.4. Ensemble learning | 23 |
| 4. Resultats | 24 |
| 4.1. Faster R-CNN..... | 24 |
| 4.2. Cascade R-CNN..... | 27 |
| 4.3. Ensemble learning | 29 |
| 4.4. Comparació amb el concurs (GRDDC)..... | 31 |



| | |
|---|----|
| 5. Pressupost | 32 |
| 6. Conclusions i desenvolupaments futurs: | 34 |
| Bibliografia | 35 |

Llistat de figures

| | |
|--|----|
| Figura 1.1: Diagrama de Gantt del pla de treball..... | 14 |
| Figura 2.1 Vehicle especialitzat amb un lidar instal·lat per a la detecció de defectes a la carretera ^[3] | 15 |
| Figura 2.2 Esquema del funcionament del Faster R-CNN..... | 16 |
| Figura 2.3 Esquema del funcionament de les bounding boxes en els models YOLO..... | 17 |
| Figura 3.1: Repartició d'imatges entre els diferents països i datasets..... | 19 |
| Figura 3.2: Exemple d'imatges amb cada una de les classes. | 20 |
| Figura 3.3: Comparació de la sortida d'un model sense i amb l'algorisme NMS..... | 22 |
| Figura 3.4: Formula de l'Intersection over union. | 23 |
| Figura 3.5: Esquema del funcionament del Non-maxima suppression..... | 23 |

Llistat de taules

| | |
|--|----|
| Taula 1.1 Work package 1 | 12 |
| Taula 1.2 Work package 2 | 12 |
| Taula 1.3 Work package 3 | 12 |
| Taula 1.4 Work package 4 | 13 |
| Taula 1.5 Work package 5 | 13 |
| Taula 1.6 Work package 6 | 13 |
| Taula 1.7 Fites amb els entregables i les dates | 14 |
| Taula 2.1 Classificació dels primers 10 equips amb la seva puntuació F1Score i les solucions proposades..... | 18 |
| Taula 3.1: Llistat de classes del dataset amb la seva nomenclatura..... | 20 |
| Taula 3.2: Distribució dels tres subdatasets..... | 21 |
| Taula 4.1: Configuració del Faster R-CNN..... | 25 |
| Taula 4.2: Taula amb el mAP per classe i global del Faster R-CNN amb la millor època. | 26 |
| Taula 4.3: Recall, precision i F1Score per classes amb el model Faster R-CNN | 27 |
| Taula 4.4: Configuració del Cascade R-CNN..... | 27 |
| Taula 4.5:Taula amb el mAP per classe i global del Cascade R-CNN amb la millor època. | 29 |
| Taula 4.6: Recall, precision i F1Score per classes amb el model Cascade R-CNN | 29 |
| Taula 4.7: Recall, precision i F1Score per classes amb el NMS dels dos models | 30 |
| Taula 4.8: Comparativa amb els diferents solucions proposades i el concurs GRDDC .. | 31 |
| Taula 5.1: Sous per hora i càrrec..... | 32 |
| Taula 5.2: Costos mensual a causa dels sous. | 32 |
| Taula 5.3: Costos de les despeses generals..... | 32 |
| Taula 5.4: Costos dels materials necessaris per dur a terme el projecte. | 33 |
| Taula 5.5: Costos totals del projecte..... | 33 |

1. Introducció

Degut a l'increment de la població en els grans nuclis urbans, la mobilitat a les grans ciutats s'ha tornat cada cop més problemàtica i difícil. Amb l'afany de trobar solucions a aquesta situació, s'ha popularitzat l'ús de vehicles de micro-mobilitat. Aquest augment de vehicles d'aquesta mena fa pensar que seria del tot desitjable buscar solucions que permetin augmentar la seguretat del seu ús per part dels usuaris.

En aquest sentit, s'ha pensat que pot millorar aquesta seguretat tot el que permeti als usuaris informar de forma automatitzada incidències en la via directament als responsables del seu manteniment. D'aquesta manera, el termini entre que s'identifica una incidència a la carretera i la seva reparació podria escurçar-se. A més, es pot utilitzar per avisar a altres usuaris de les incidències, en aplicacions tipus *Waze*.

La solució que es presenta segueix aquesta idea i proposa fer ús de tècniques de *Deep Learning* i *Computer Vision* per identificar possibles defectes a les carreteres i millorar la seguretat.

El projecte s'ha realitzat a la Universitat Politècnica de Catalunya al departament de Teoria del senyal i Comunicació (TSC) a l'escola ETSTB. Aquest treball ha estat proposat pels professors Josep Ramon Morros Rubió i Elisa Sayrol Clois, i l'han dut a terme els alumnes Jaume Prats Cristià i Miquel Torrecilla Mercado. En aquesta memòria només es descriurà el treball que jo he realitzat.

1.1. Objectius del projecte

El principal objectiu del projecte és l'aplicació de tècniques de *Deep Learning* i *Computer Vision* per dissenyar i ajustar un model de *Machine Learning* que sigui capaç de detectar i identificar els possibles defectes que puguin haver a la carretera per tal de millorar la seguretat del usuaris de vehicles de micro-mobilitat.

Això es realitzarà mitjançant l'adaptació de 2 models a un dataset preestablert obtingut del concurs *Global Road Damage Detection Challenge 2020* ^[1] (GRDDC) d'unes 20.000 imatges.

1.2. Requeriments i especificacions

Per poder realitzar el projecte necessitem de:

- Coneixements de *Deep Learning* i *Computer Vision*.
- Un dataset suficientment gran per poder entrenar els diferents models de *Machine Learning*.
- Un servidor remot per poder executar els nostres scripts (Proveït pel grup de Processat d'Imatge de la UPC).

1.3. Mètodes i Procediments

La major part del projecte s'ha dut a terme a la UPC però s'ha fet ús d'algorismes ja creats com a punt de partida.

- *MMdetection*: és una llibreria basada en *python* amb diferents models proposats per diferents autors per a la detecció d'objecte d'on s'han tret els dos models que més endavant han estat adaptats.

- Els dos models mencionats abans: FasterRCNN i CascadeRCNN.
- El dataset obtingut del concurs que es va realitzar al Japó (GDRRC).

1.4. Planificació

El projecte està dividit en sis paquets de treball:

| | |
|------------|--|
| WP1 | Documentació |
| Descripció | Fer un seguiment de totes les millores realitzades durant les setmanes i emplenar tots els documents entregables durant el projecte. |
| Tasques | Tasca interna T1: Documents entregables |
| Dates | Comença: 15/02/2021. Acaba: 21/06/2021. |
| Fites | Memòria final. |

Taula 1.1 Work package 1.

| | |
|------------|---|
| WP2 | Estat de l'art |
| Descripció | Recercar sobre tècniques de <i>Deep Learning</i> i <i>Computer Vision</i> i ser capaç d'entendre-les i implementar-les. |
| Tasques | Tasca interna T1: Guanyar coneixement de les diferents tècniques que s'utilitzen actualment en aquests camps. |
| Dates | Comença: 15/02/2021. Acaba: 16/03/2021. |
| Fites | Informe de la informació més rellevant. |

Taula 1.2 Work package 2

| | |
|------------|---|
| WP3 | Adaptació del dataset del concurs GRDDC |
| Descripció | Adaptar les anotacions del dataset al format COCO i separar en tres subdatasets (entrenament, validació i test). |
| Tasques | Tasca interna T1: Crear un script que canvi el format de les anotacions de PascalVOC a COCO. Tasca interna T2: Separar el dataset en els subdatasets de entrenament, validació i test. |
| Dates | Comença: 21/02/2021. Acaba: 26/03/2021. |
| Fites | Subdatasets amb les anotacions en format COCO. |

Taula 1.3 Work package 3

| | |
|------------|--|
| WP4 | Entrenament dels models. |
| Descripció | Entrenar els diferents models per separat. |
| Tasques | Tasca interna T1: Crear un script que pugui entrenar un model de la llibreria MMdetection. Tasca interna T2: Adaptar aquest script perquè pugui utilitzar el nostre dataset i la nostra configuració per a cada model. Tasca interna T3: Entrenar els dos models |
| Dates | Comença: 20/03/2021. Acaba 15/05/2021. |
| Fites | Entrenament dels 2 models. |

Taula 1.4 Work package 4

| | |
|------------|---|
| WP5 | Model testing i avaluació. |
| Descripció | Utilitzar els models al subdataset de test i comparar els resultats obtinguts en cadascun d'ells. |
| Tasques | Tasca interna T1: Comparar els resultats obtinguts amb cada model. |
| Dates | Comença: 15/05/2021. Acaba: 31/05/2021. |
| Fites | Comparativa i selecció del millor model. |

Taula 1.5 Work package 5

| | |
|------------|--|
| WP6 | Millores als models. |
| Descripció | Utilitzar tècniques de Deep Learning tals com Non-maxima suppression per millorar els resultats obtinguts. |
| Tasques | Tasca interna T1: Crear un algoritme per poder aplicar la tècnica Non-maxima suppression als dos models. |
| Dates | Comença: 01/06/2021. Acaba: 21/06/2021. |
| Fites | Final Software |

Taula 1.6 Work package 6

Fites:

| WP | Tasca | Títol | Fita | Data |
|----|-------|---|--|------------|
| 1 | 1 | Documentació | Documentació | 21/06/2021 |
| 2 | 2 | Estat de l'art | Informe de la informació més rellevant. | 16/03/2021 |
| 3 | 3 | Adaptació del dataset del concurs GRDDC | Subdatasets amb les anotacions en format COCO. | 26/03/2021 |
| 4 | 4 | Entrenaments dels models. | Entrenament dels 2 models. | 15/05/2021 |
| 5 | 5 | Model testing i avaluació. | Comparativa i selecció del millor model. | 31/05/2021 |
| 6 | 6 | Millores | Final Software | 21/06/2021 |

Taula 1.7 Fites amb els entregables i les dates

Donat el pla de treball s'obté el següent diagrama de Gantt (veure Figura 1.1).

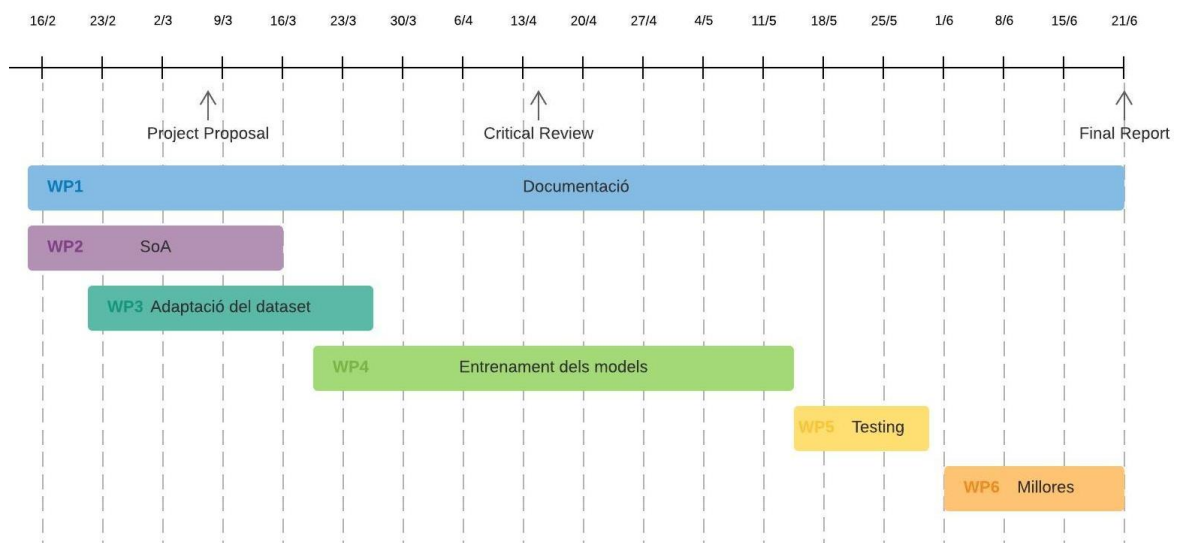


Figura 1.1: Diagrama de Gantt del pla de treball

1.5. Desviacions del pla original

En primera instància l'objectiu del projecte era crear un dataset amb imatges de Barcelona i trobar el model que millor s'adaptés a aquest. Però, durant l'estat de l'art es va trobar el concurs, Global Road Damage Detection Challenge 2020. Aquest concurs oferiria un dataset d'entrenament amb unes 20.000 imatges de diferents països que a partir d'aquell moment es va utilitzar per entrenar els models. A més es va fer servir com a comparatiu dels resultats obtinguts pels participants del concurs i els models d'aquest projecte.

2. Estat de l'art de la tecnologia utilitzada o aplicada en el projecte:

Aquest estat de l'art és el mateix que l'alumne Jaume Prats Cristià.

2.1. Detecció de defectes a les carreteres

El manteniment de les carreteres requereix d'una avaluació periòdica del seu estat. Aquest manteniment està al càrrec de diferents agències de l'estat que el duen a terme mitjançant vehicles d'inspecció vial amb diferents sensors instal·lats. Acostumen a fer ús de càmeres tridimensionals (3D) i càmeres amb làsers que van capturant la superfície del paviment amb la màxima resolució possible ^[2] (veure Figura 2.1). No obstant, aquest vehicles dedicats solen ser molt costosos i, sovint, inassequibles per a les agències locals amb pressupostos més limitats. A més, molt possiblement si estan basats en vehicles no s'adaptaran als carrils de micro-mobilitat. A causa d'això sorgeix la necessitat de buscar solucions de baix cost com l'ús de mòbils amb sistemes de *Deep Learning* i *Object Detection*.

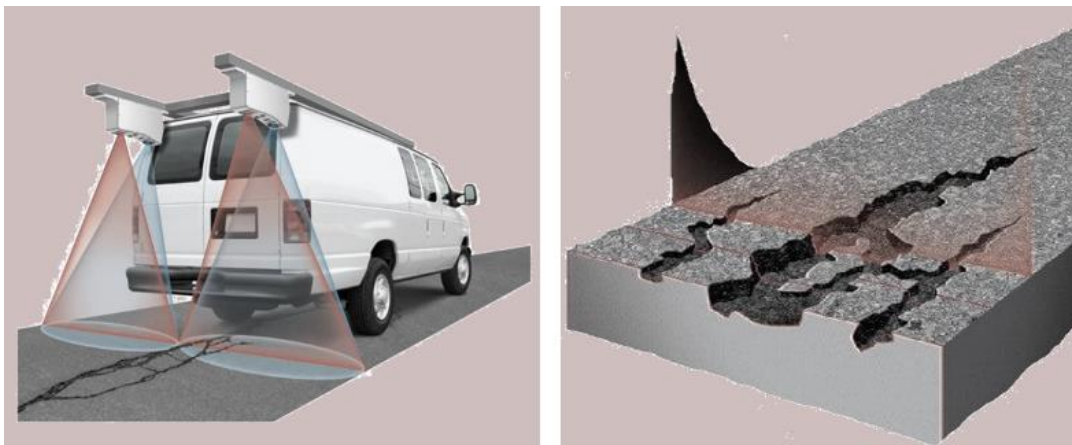


Figura 2.1 Vehicle especialitzat amb un lidar instal·lat per a la detecció de defectes a la carretera ^[3].

2.2. Detecció d'objectes

La visió per computador és un camp interdisciplinari que ha anat guanyant molt terreny en els últims anys i una part integral d'aquest és la detecció d'objectes. La detecció d'objectes és la tasca de detectar instàncies d'objectes d'una determinada classe dins d'una imatge, un repte que al seu torn es compon de dos problemes diferents, saber on es troba l'objecte dins de la imatge i saber de quina mena d'objecte es tracta.

Segons la forma en com aborden aquesta tasca, els mètodes més avançats poden classificar-se en dos tipus principals: detectors d'una etapa i detectors de dues etapes.

Els detectors de dues etapes utilitzen una *Region Proposal Network* (RPN) per a generar una sèrie de regions d'interès (ROI) i aquestes són alimentades a un segon mòdul que classifica la imatge dins de cada regió proposada. En canvi, els detectors d'una etapa intenten resoldre totes dues tasques en una sola passada.

En general, els detectors de dues etapes donen prioritat a la precisió de la detecció i els d'una etapa se centren més en la velocitat d'inferència per al seu ús en temps real.

2.2.1. Faster R-CNN^[4]

Faster R-CNN és la tercera iteració de la família R-CNN, una de les xarxes més representatives dels detectors de dues etapes. R significa regions i CNN significa convolucional neural networks. L'objectiu principal del Faster R-CNN és reduir les despeses computacionals i poder treballar en temps real.

En el Faster R-CNN la imatge es proporciona com a entrada a una xarxa neuronal convolucional que proporciona un *convolutional feature map*. A continuació, s'utilitza una altra xarxa per a predir les propostes de regió. Les propostes de regió predites es remodelen i s'alimenten a un segon mòdul que classifica la imatge en cada regió proposada (veure Figura 2.2).

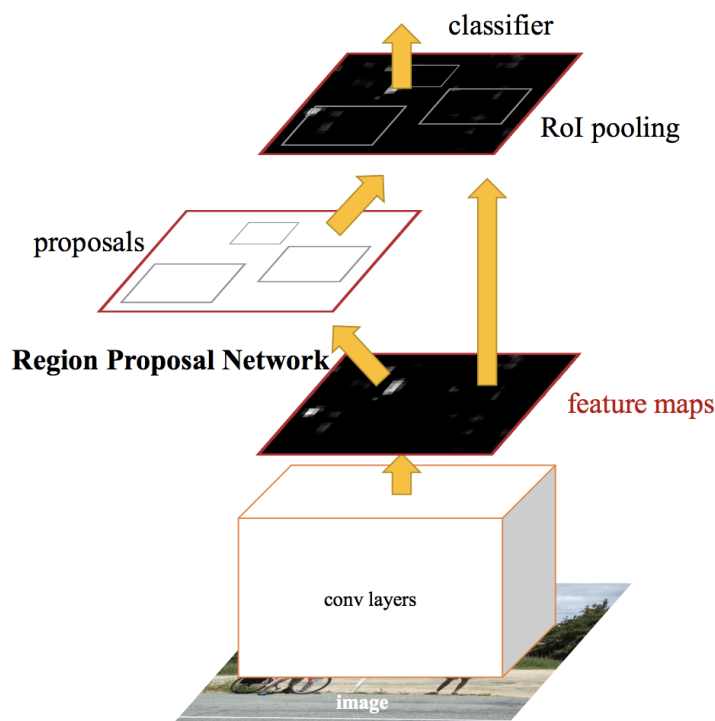


Figura 2.2 Esquema del funcionament del Faster R-CNN

2.2.2. Cascade R-CNN^[5]

Cascade R-CNN és una arquitectura de detecció d'objectes proposada per la Universitat de Califòrnia Sant Diego el 2017 per a abordar els problemes de degradació del rendiment amb l'augment dels llindars de *IoU*.

Normalment, un detector d'objectes entrenat amb un llindar de *IoU* baix produeix deteccions sorolloses. No obstant això, si intentem augmentar el llindar per a combatre el soroll, el rendiment de la detecció tendeix a degradar-se. Aquesta degradació es deu a dos factors principals, l'*overfitting* durant l'entrenament en eliminar mostres positives, i el desajustament en temps d'inferència entre els *IoU* per als quals el detector és òptim i les entrades.

Cascade R-CNN intenta resoldre aquest problema amb una extensió multi-etapa de la R-CNN, en la qual les etapes del detector més profundes de l'arquitectura són seqüencialment més selectives contra els falsos positius. Les etapes del Cascade R-CNN s'entrenen seqüencialment, utilitzant la sortida d'una etapa per a entrenar la següent.

D'aquesta manera, consisteix en una seqüència de detectors adaptats a *IoUs* cada vegada més alts.

2.2.3. YOLO^[6]

Tots els algorismes de detecció d'objectes anteriors utilitzen regions per localitzar l'objecte dins de la imatge. YOLO o *You Only Look Once* és un algorisme de detecció d'objectes molt diferent que utilitza una única xarxa convolucional per a predir els quadres delimitadors i les probabilitats de classe d'aquests quadres. Funciona prenent una imatge i dividint-la en cel·les de quadrícula, per a cada cel·la es prediuen diversos quadres delimitadors. A continuació, l'algorisme ajusta i prediu una probabilitat de classe per a cada quadre delimitador (veure Figura 2.3).

YOLO va ser proposat per primera vegada el 2016 per Joseph Redmon i va ser una fita en la recerca de la detecció d'objectes a causa de la seva capacitat de detectar objectes en temps real amb una millor precisió. Posteriorment, Redmon va millorar el seu algorisme amb dues noves versions YOLOv2 i YOLOv3. No obstant això, el 2020, quan YOLO es va fer famós, Joseph Redmond va anunciar que deixava d'investigar en el camp de visió per computador degut a diverses preocupacions sobre el possible impacte negatiu del seu treball.

Malgrat la retirada de Redmon, no va ser el final de YOLO, més tard al 2020 s'han publicat dues noves versions de YOLO en paral·lel, YOLOv4^[7] per Alexey Bochkovskiy en *Darknet*, el framework original de YOLO, i YOLOv5^[8] per Glenn Jocher una implementació en *PyTorch*.

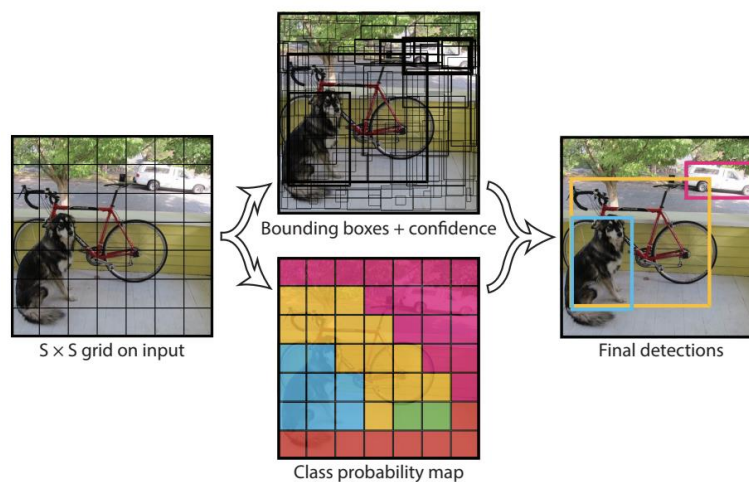


Figura 2.3 Esquema del funcionament de les bounding boxes en els models YOLO

2.3. Global Road Damage Detection Challenge (GRDDC)^[9]

Amb la intenció de millorar l'estat de l'art en detecció de defectes a les carreteres es va crear el concurs, *Global Detection Challenge* (GRDDC). Aquest, es va iniciar la Universitat de Tokio al Japó en el 2018 a partir d'un concurs anomenat *IEEE BidData Cup*. Aquest primer concurs tenia la intenció d'avaluar els mètodes contemporanis de detecció que hi havia. Això va portar a la creació del concurs actual de detecció que es va iniciar al *IEEE International Conference on Big Data'2018* realitzat a Seattle, EEUU. Aquest va tenir molt d'èxit, amb un total de 54 equips participant i amb diferents mètodes nous. Després de l'èxit d'aquest primer concurs es va continuar realitzant cada any.

Al acabar el primer concurs, diferents municipis japonesos van començar a utilitzar els mètodes proposats per a detecció de defectes. Això va ser molt beneficiós perquè es va poder obtenir feedback de qualitat del funcionament d'aquest mètodes. Les agències van arribar a la conclusió que necessitaven un sistema més robust.

Les dues premisses que componen els concurs són: un dataset d'imatges de carreteres de tres països i una competició online. Aquest dataset disposa d'unes 20.000 imatges anotades de Japó, Índia i República Checa. Els dos principals reptes són la classificació dels defectes i la detecció d'aquest.

Per avaluar els models enviats pels participants es van determinar dues condicions necessàries per distingir un encert d'un error:

- L'àrea que interseca la unió (*IoU*) entre la bounding box predita pel model i la *del ground truth* del dataset ha de ser superior al 50%.
- La classe de la bounding box predita ha de ser la mateixa que la de la *ground truth*.

Un cop establert com es van avaluar els resultats obtingut per cada equip, es va decidir els millors mètodes de detecció utilitzant la mètrica *F1Score* (veure Taula2.1).

| Classificació | Nom de l'equip | F1score test 1 | F1Score test 2 | Model |
|---------------|----------------|----------------|----------------|---|
| 1 | IMSC | 0.6748 | 0.6662 | Ensemble Learning with Ultralytics-YOLO and Test Time Augmentation |
| 2 | SIS Lab | 0.6275 | 0.6358 | Ensemble model with YOLOv4. |
| 3 | DD_VISION | 0.629 | 0.6219 | A Consistency Filtering Mechanism and model ensemble with cascade R- CNN. |
| 4 | Titan_mu | 0.5814 | 0.5751 | YOLO model trained on CSPDarknet53 backbone |
| 5 | Dongjuns | 0.5683 | 0.5710 | YOLOv5x |
| 6 | SUTPC | 0.5636 | 0.5707 | Ensemble(YOLOv4 + Faster-RCNN) |
| 7 | RICS | 0.565 | 0.547 | EfficientDet |
| 8 | AIRS-CSR | 0.554 | 0.541 | YOLOv4 |
| 9 | CS17 | 0.5413 | 0.5430 | Resnet-18 and Resnet-50 backbones based Faster- RCNN. |
| 10 | BDASL | 0.5368 | 0.5426 | Multi-stage Faster R-CNN with Resnet-50 and Resnet-101 backbones |

Taula 2.1 Classificació dels primers 10 equips amb la seva puntuació *F1Score* i les solucions proposades

3. Metodologia

Com s’ha mencionat abans, l’objectiu del projecte es trobar el model que s’adapti millor al dataset de GRDCC, utilitzant models de codi obert. Per això primer s’ha adaptat el data set per poder utilitzar-lo amb els models escollits. Després s’ha realitzat un entrenament dels dos models amb el data set. Per últim s’ha utilitzat tècniques de mecanismes de millora per intentar aconseguir una millora del resultats obtinguts amb cada un dels models.

Tots els algorismes dissenyats durant el transcurs del projecte estan a:

<https://github.com/MiquelTorrecilla/Lane-condition-analysis-for-micromobility-app>

3.1. Dataset

Una de les parts més importants per realitzar un correcte entrenament d’un model de *Deep Learning* és la qualitat i la quantitat d’imatges que formen el dataset. Aquesta qualitat del dataset vindrà definida per la variabilitat que presentin les imatges que el componen. Per això, res més començar el projecte, es va intentar crear un dataset prenent imatges dels carrers de Barcelona. Aquestes estaven fetes pels propis integrants del projecte amb l’objectiu d’obtenir en el mínim temps possible el màxim nombre d’imatges. Però ràpidament, durant la recerca de l’estat de l’art, es va trobar el dataset de 20.000 imatges del GRDDC. A partir d’aquest moment es va dividir el projecte per buscar solucions diferenciades pel dataset de GRDDC i el dataset propi de Barcelona.

Aquest dataset està compost per imatges de tres països diferents, el Japó, la Índia i la República Checa. Això va permetre obtenir un model que generalitzi millor i que es pugui utilitzar a escala global, ja que les carreteres de cadascun dels països són molt diferents unes amb les altres.

El dataset venia dividit de la següent forma (veure Figura 3.1):

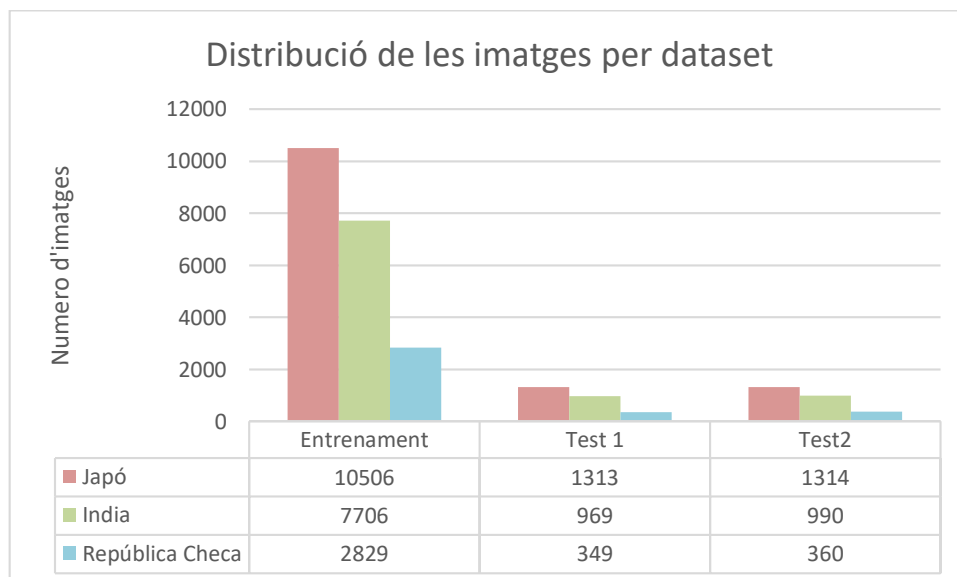


Figura 3.1: Repartició d’imatges entre els diferents països i datasets.

Gaire bé totes les imatges dintre del dataset de train estaven ja etiquetades amb unes bounding boxes que limitaven el defecte en format PascalVOC^[10], el que va fer que es pogués avançar més ràpidament. Tot i així, per poder fer-ne ús en el MMdetection es va haver de dissenyar un algorisme

que passes d'aquest format d'anotacions a format COCO^[11]. Aquestes anotacions venien dividides en 9 classes (veure Taula 3.1 i Figura 3.2):

| Tipus de defecte | | | Detallat | Nom Classe |
|------------------|--------------------|-------------------|--------------------------|------------|
| Esquerda | Esquerda lineal | Longitudinal | Marques de Roda | D00 |
| | | | Unió d'asfalt | D01 |
| | | Lateral | Interval igual | D10 |
| | | | Unió d'asfalt | D11 |
| | Esquerda de caiman | Parcial i general | D20 | |
| Altres | | | Forat | D40 |
| | | | Pas de vianants borrós | D43 |
| | | | Línies blanques borroses | D44 |
| | | | Claveguera | D50 |

Taula 3.1: Llistat de classes del dataset amb la seva nomenclatura



Figura 3.2: Exemple d'imatges amb cada una de les classes.

Per poder entrenar el sistema, i aprofitar que les imatges ja estaven anotades, es va dividir el dataset de train donat pel concurs en tres subdatasets: entrenament, validació i test (veure Taula 3.2).

| dataset | Percentatge | Número d'imatges |
|-------------|-------------|------------------|
| Entrenament | 70% | 14728 |
| Validació | 20% | 4208 |
| Test | 10% | 2105 |

Taula 3.2: Distribució dels tres subdatasets.

A partir d'aquí es va utilitzar dataset de train amb els seves respectives divisions en els subdatasets de la Taula 3.2

3.2. Entrenament

El primer pas per poder realitzar l'entrenament es decidir quin model s'utilitzarà. Com ja s'ha esmentat, l'objectiu d'aquest projecte es trobar el millor model que s'adapti al dataset. Per això es van escollir diferents models, els quals s'han entrenat per separat. Un cop entrenats s'ha realitzat una comparació amb diferents estadístiques dels dos models seleccionats.

Els dos models que s'han escollit per realitzar el nostre estudi són: FasterRCNN i CascadeRCNN. Aquest s'han extret de MMDetection^[12], una llibreria que proporciona diferents models pre-entrenats amb diferents mètodes per després poder treballar amb ells.

Un cop escollits els models es van dissenyar uns algorismes per poder adaptar les configuracions de cada model al nostre dataset. Per obtenir uns millors resultats es va implementar, dins de la configuració dels models, un parell de funcions de Data Augmentation. Aquesta és una tècnica que consisteix en augmentar la quantitat de dades afegint-hi còpies lleugerament modificades per tal de regularitzar les dades i evitar un possible overfitting. Per a cada època el que s'ha fet és un re-escalat de les imatges i un flip aleatori horitzontal amb una probabilitat del 50%.

El MMDetection el primer que fa es agafar el 50% de les imatges del subdataset d'entrenament de manera aleatòria i entrenar. Mentre va entrenant en cada iteració va modificant els pesos per reduir el màxim possible la *loss* d'entrenament. Un cop acabada la època corresponent entra en un procés de *cross-validation* per poder analitzar les estadístiques del model. Aquí utilitza el dataset de validació amb l'objectiu d'avaluar les principals mètriques del model entrenat amb un dataset independent. I torna a fer aquest dos passos anteriors fins l'última època.

Un cop entrenats els dos models es van analitzar els resultats obtinguts i es van guardar els millors pesos per un posterior estudi.

3.3. Non-maxima suppression

Un cop acabat l'entrenament de cada model, es poden aplicar diferents tècniques que ajuden a millorar els resultats. Una d'elles és el *Non-maxima suppression (NMS)*. Consisteix en quedar-te la millor predicció sobre un objecte de les diferents que genera el model (veure Figura 3.3).

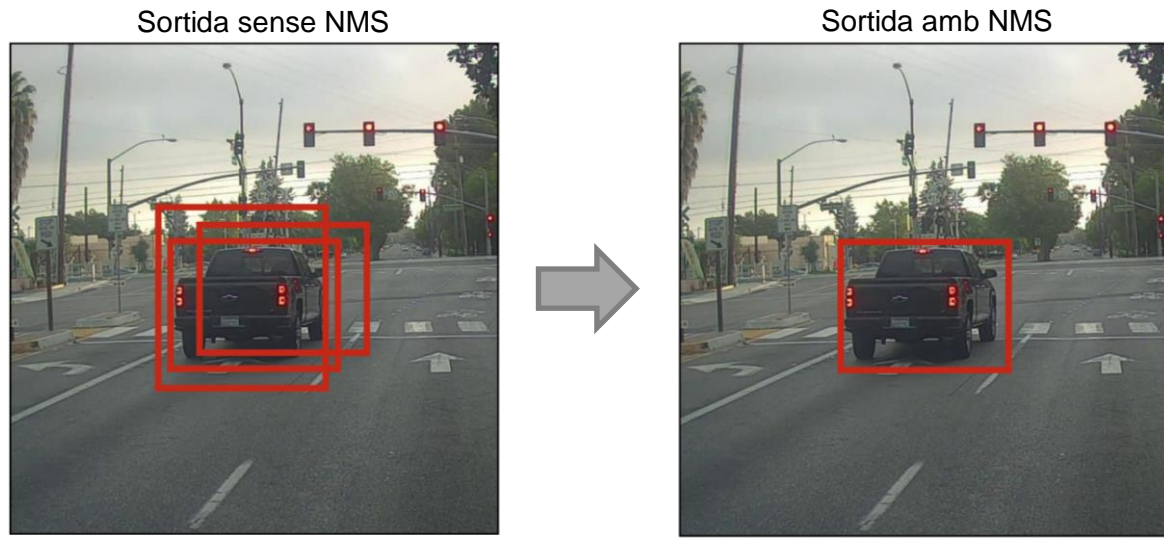


Figura 3.3: Comparació de la sortida d'un model sense i amb l'algorisme NMS

Com es pot veure a la figura, el model està generant varies prediccions del vehicle on les bounding boxes coincideixen. L'objectiu d'aplicar aquesta tècnica es quedar-se amb la bounding box que tingui una puntuació de confiança més alta. D'aquesta manera s'aconseguirà baixar el nombre de falsos negatius a costa de pujar els falsos positius. Si aquesta relació és positiva obtindrem millors resultats.

Per això es van crear dos algorismes. El primer calcula la intersecció sobre la unió (IoU), aquesta és l'àrea d'intersecció de les *bounding boxes* dividit entre l'àrea d'unió d'aquestes (veure Figura 3.4). Així es podrà saber de forma automatitzada si una *bounding box* està a la mateixa localització que una altre. També es va dissenyar un altre algorisme que elimina les *bounding boxes* que estiguin per sobre del llindar de IoU establert, que siguin de la mateixa classe i tinguin una puntuació més baixa.

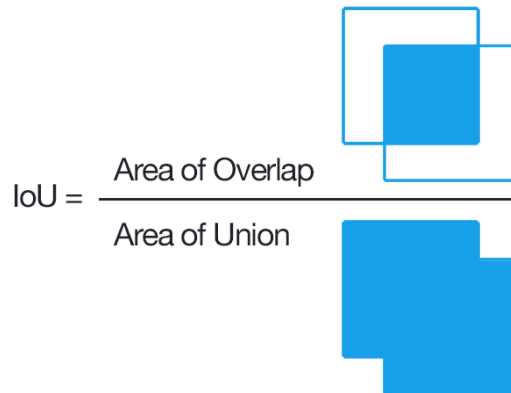


Figura 3.4: Formula de l'Intersection over union.

3.4. Ensemble learning

La tècnica de *non-maxima suppression* es pot aplicar a les *bounding boxes* que surten de les prediccions de diferents models per quedar-se amb la millor predicció de ambdós (veure Figura 3.4). D'aquesta manera el que s'aconsegueix es fer una barreja de dos models, *ensemble learning*, i quedar-se sempre amb la millor opció. A partir d'aquí el *NMS* es pot aplicar de dues maneres, només s'accepta una *bounding box* si apareix als dos models i s'aplica el *NMS* per quedar-se amb la millor opció. O s'accepten les *bounding boxes* dels dos models i s'aplica el *NMS* quan coincideixen. En el cas del projecte es va fer ús de la segona opció per intentar disminuir el màxim possible el nombre de falsos negatius, ja que el cost d'aquest és molt més alt per l'objectiu del projecte.

Per això es va dissenyar un algorisme que ajuntés les *bounding boxes* que sortien de passar les imatges pels dos models per separat i aplicar el *NMS* per obtenir un model final que extregui les millors característiques d'ambdós models (veure Figura 3.5).

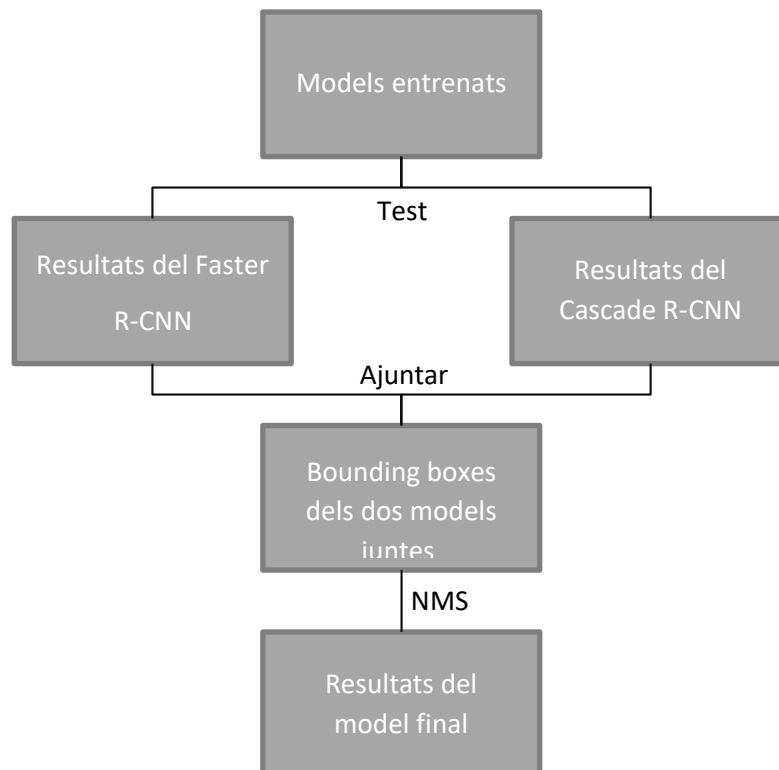


Figura 3.5: Esquema del funcionament del Non-maxima suppression

4. Resultats

Durant els mesos que ha durat el projecte, amb la intenció de trobar el millor model possible pel dataset, s'han realitzat tres experiments. Primer s'ha entrenat els dos models, el Faster R-CNN i el Cascade R-CNN, per separat i després s'han ajuntat fent ús de tècniques d'*ensemble learning*.

Primer s'ha de definir quan una predicció és encertada o no. Per això, es necessita una mètrica que ens permeti saber si dues bounding boxes, que comparteixen classe, estan al mateix lloc. D'aquesta manera, utilitzarem la mètrica D'intersecció sobre la unió (IoU).

Per tant, un encert serà si IoU de les *bounding boxes* predites pels models supera el 50% amb les del *ground truth*, a més, han de compartir la mateixa classe.

Un cop s'ha definit quines condicions s'han de complir per comptabilitzar un encert, s'ha de definir quines mètriques utilitzarem per comparar els resultats. La primera que s'ha emprat és el *mean average precisió* (mAP), aquesta correspon al valor mig del *recall* avaluada entre 0 i 1. Aquesta mesura servirà per saber com de bé es troben els positius amb cada model.

A més del *mAP* també s'ha utilitzat una mètrica que donarà la mesura de precisió que tenen els models, F1-Score. Aquesta s'ha calculat per classes i bé definida a partir del *Precision* i del *Recall*.

$$\text{Recall} = \frac{\text{Positius reals}}{\text{Positius reals} + \text{Falsos negatius}}$$

$$\text{Precision} = \frac{\text{Positius reals}}{\text{Positius reals} + \text{Falsos positius}}$$

$$\text{F1Score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Tots els resultats relatius als tres experiments del projecte s'han obtingut utilitzant el dataset de test de 2105 imatges anotades que s'ha creat a partir del dataset de train de GDRCC (veure Taula 3.2).

4.1. Faster R-CNN

El primer experiment que es va realitzar consistia en adaptar el model, Faster R-CNN, al dataset.

Per començar es va modificar la configuració del model per poder utilitzar el dataset. Es va definir formats, directoris i classes. Després es va definir els paràmetres principals per optimitzar l'entrenament amb el Faster R-CNN: *data augmentation*, número d'èpoques, velocitat d'aprenentatge, avaluació, etc.. (veure Taula 4.1)

| Paràmetre | Valor del paràmetre |
|--------------------|----------------------------------|
| Data augmentation | Flip aleatori (0.5) i re-escalat |
| Número d'èpoques | 20 |
| Format del dataset | COCO dataset |

| | |
|--------------------------------|------------|
| Número d'imatges per batch | 16 |
| Velocitat d'aprenentatge | 0.0025 |
| Número de classes | 8 |
| Interval de guardat dels pesos | Cada època |
| Avaluació | Cada època |

Taula 4.1: Configuració del Faster R-CNN

El primer que es va comprovar consistia en veure si l'entrenament funcionava correctament, és a dir, si el model va anar aprenent o si hi havia *overfitting*. Per això es va veure com va evolucionar les corbes de la loss d'entrenament (veure Figura 4.2), si a mesura que va avançant l'entrenament aquestes corbes van baixant vol dir que el model ha entrenat.

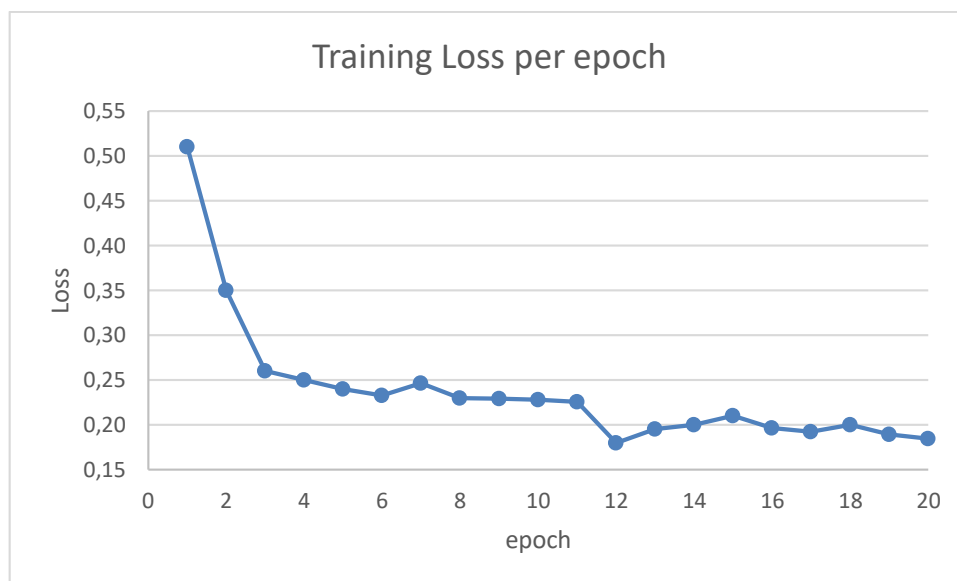


Figura 4.2: Relació Loss per època

Com es pot veure a la figura 4.2 la loss va baixant a mesura que avancen les iteracions, això vol dir que el model s'ha anat adaptant al dataset i ha anat aprenent.

A continuació s'ha de mirar si s'ha produït *overfitting*. Això es veurà si la corba de mAP, en comptes de pujar, comença a baixar a partir de certa iteració (veure Figura 4.3). No s'ha pogut utilitzar la corba de loss amb el dataset de validació per veure si hi havia *overfitting* perquè el MMDetection no la proporciona.

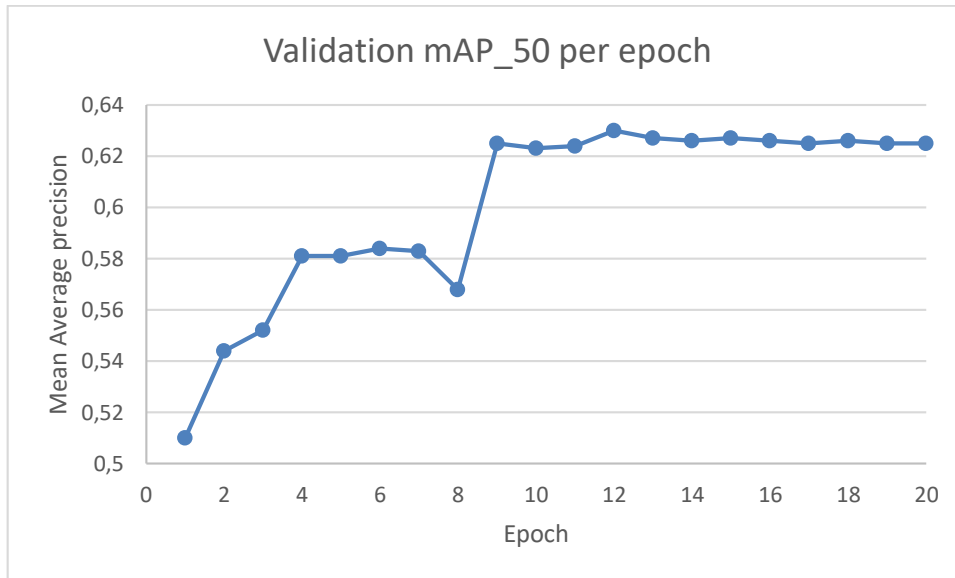


Figura 4.3: Relació mAP per època

Com es pot veure la figura 4.3 a partir de les 9 èpoques el model s'estabilitza i no aprèn a la mateixa velocitat que abans. Encara que no ha començat a baixar el valor del *mAP*, es podria dir que s'ha produït una mica d'*overfitting*.

Un cop comprovat que l'entrenament s'ha realitzat correctament es comencen a analitzar els resultats i les mètriques principals. Primer s'ha escollit la millor època dels 20 que s'han entrenat, en aquest cas el millor època és el 12 amb un *mAP_50* de 0,63.

Per analitzar els resultats s'ha aplicat el model entrenat a les imatges del subdataset de test, 2105 imatges agafades aleatòriament del dataset complet amb diferents classes. S'ha mirat el *mAP* que donen les prediccions del model global i per classes (veure Taula 4.2).

| Classe | D00 | D10 | D20 | D40 | D43 | D44 | D50 | Total |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| mAP | 0.201 | 0.164 | 0.321 | 0.209 | 0.486 | 0.384 | 0.479 | 0.321 |

Taula 4.2: Taula amb el *mAP* per classe i global del Faster R-CNN amb la millor època.

Com es pot veure a la taula 4.2 el *mAP* es bastant més baix que el *mAP_50*. Això és perquè té en compte totes les bounding box predites. El *mAP_50*, a diferència de l'altre, només té en compte les que tenen un IoU superior al 50%. Per tant el *mAP_50* ens donarà un valor que s'apropa més a les condicions que s'han establert per determinar si una predicció es encertada o no.

Com es pot veure a la taula 4.2 el model té més problemes per detectar els defectes relacionats amb les esquerdes. Això pot ser perquè les imatges no eren d'una grandíssima qualitat i de tant en tant les confon amb ombres que apareixen a la via. Es possible que el re-escalat per fer totes les imatges de la mateixa mida estigui afectant a aquestes classes.

El següent pas es calcular la F1Score del model pel dataset de test (veure Taula 4.3). Primer es van eliminar les *bounding boxes* que no tinguessin una puntuació de confiança superior a 60, així es va analitzar les prediccions eliminant les que es vegin més afectades pel soroll.

| Classe | D00 | D10 | D20 | D40 | D43 | D44 | D50 | Total |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| Recall | 0,564 | 0,724 | 0,602 | 0,591 | 0,604 | 0,801 | 0,755 | 0,758 |
| Precision | 0,728 | 0,730 | 0,592 | 0,775 | 0,753 | 0,755 | 0,765 | 0,732 |
| F1Score | 0,636 | 0,727 | 0,597 | 0,671 | 0,671 | 0,777 | 0,760 | 0,745 |

Taula 4.3: Recall, precision i F1Score per classes amb el model Faster R-CNN

Com es pot veure el model funciona prou bé. S'obtenen uns resultats molt semblants en *recall* i *precision*, això vol dir que el model està bastant ben balancejat. Com ja s'ha vist al *mAP*, el model pateix amb les classes relacionades amb esquerdes. Això si, funciona molt bé amb les altres classes per això acabem obtenint tant bons resultats en el model en general.

4.2. Cascade R-CNN

A continuació es va realitzar el segon experiment, aquest consistia en adaptar el model Cascade R-CNN al dataset. Com en el experiment anterior, el primer pas consisteix en modificar la configuració del model perquè es pugui utilitzar amb el mateix dataset del experiment anterior. Els principals paràmetres de la configuració es van deixar igual que en el experiment anterior, ja que van funcionar prou bé (veure Taula 4.4). A més, es podran comparar els resultats amb els obtinguts en el experiment anterior i determinar quin model funciona millor.

| Paràmetre | Valor del paràmetre |
|--------------------------------|----------------------------|
| Data augmentation | Random flip (0.5) i resize |
| Número d'epochs | 20 |
| Format del dataset | COCO dataset |
| Número d'imatges per batch | 16 |
| Learning rate | 0.0025 |
| Número de classes | 8 |
| Interval de guardat dels pesos | Cada epoch |
| Avaluació | Cada epoch |

Taula 4.4: Configuració del Cascade R-CNN

Primer s'ha comprovat que el model ha entrenat i que ha après a mesura que han avançat les èpoques. Per això es va analitzar la corba de Loss per època amb el mateix procediment de l'experiment anterior(veure Figura 4.4).

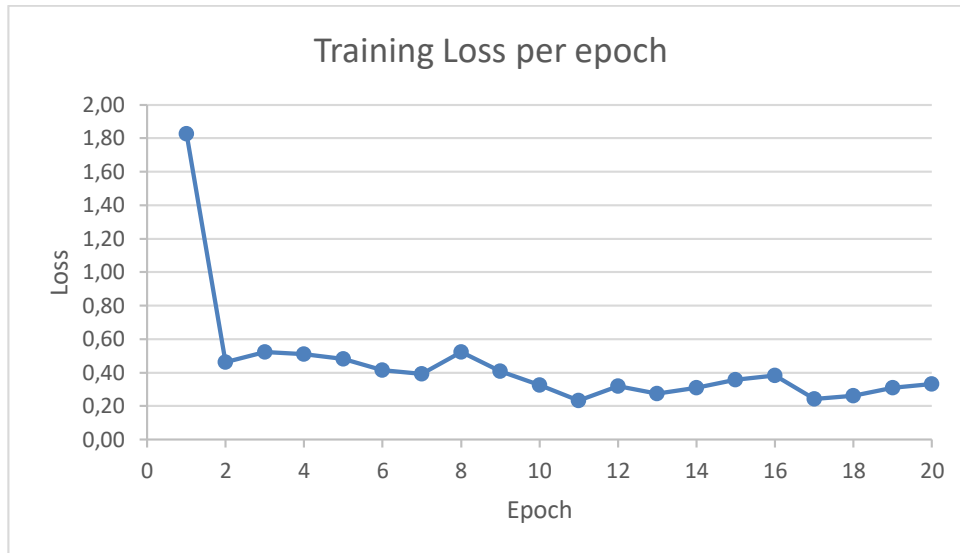


Figura 4.4: Relació Loss i èpoques pel model Cascade R-CNN

Com es pot veure les corbes evolucionen correctament, a mesura que han avançat les èpoques la corba de Loss ha anat baixant. Això vol dir que el model ha anat aprenent al llarg de les diferents èpoques. Després, per comprovar si hi ha *overfitting* s'ha representat la corba de *mAP* en relació amb les èpoques (veure Figura 4.5).

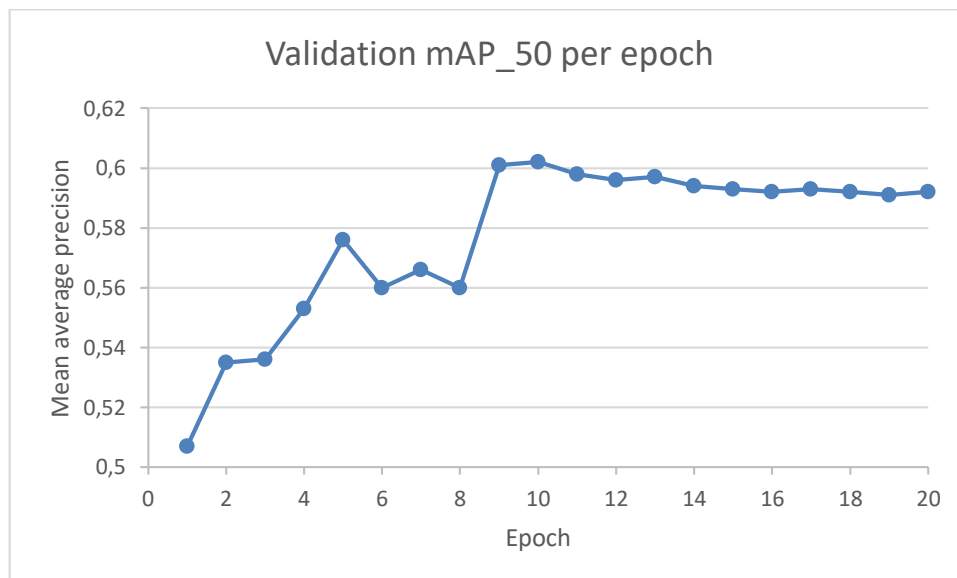


Figura 4.5: Relació *mAP_50* per època del model Cascade R-CNN

En aquest cas es pot veure que la corba evoluciona correctament fins a les 10 èpoques on troba el màxim rendiment. A partir d'aquí baixa lleugerament, es podria dir que existeix una mica d'*overfitting*.

Un cop comprovat que l'entrenament s'ha realitzat correctament es comencen a analitzar els resultats i les mètriques principals. Primer s'escull el millor època dels 20 que s'han entrenat, en aquest cas el millor època és el 10 amb un *mAP_50* de 0,602.

Per obtenir el rendiment del model s'ha analitzat amb el subdataset de test i s'ha calculat el *mAP* global i per classes (veure Taula 4.5).

| Classe | D00 | D10 | D20 | D40 | D43 | D44 | D50 | Total |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| mAP | 0,189 | 0,156 | 0,314 | 0,202 | 0,494 | 0,386 | 0,483 | 0,290 |

Taula 4.5: Taula amb el mAP per classe i global del Cascade R-CNN amb la millor època.

Es pot veure que tenim uns resultats molt similars als obtinguts amb l'anterior model. Tenim problemes amb les prediccions de les classes relacionades amb esquerdes però amb les altres obtenim bons resultats.

Com s'ha fet al primer experiment el següent pas es calcular la *F1Score* del model pel dataset de test (veure Taula 4.6). També s'han eliminat les *bounding boxes* que no tinguessin una puntuació de confiança superior a 60 per eliminar les possibles prediccions que apareixen pel soroll.

| Classe | D00 | D10 | D20 | D40 | D43 | D44 | D50 | Total |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| Recall | 0,574 | 0,753 | 0,636 | 0,596 | 0,629 | 0,75 | 0,75 | 0,775 |
| Precision | 0,854 | 0,781 | 0,667 | 0,817 | 0,867 | 0,941 | 0,777 | 0,823 |
| F1Score | 0,686 | 0,767 | 0,651 | 0,688 | 0,729 | 0,835 | 0,773 | 0,798 |

Taula 4.6: Recall, precision i F1Score per classes amb el model Cascade R-CNN

S'obté un molt bon resultat amb aquest segon model. Com en el cas del primer experiment les mesures de *recall* i *precision* són molt semblant entre elles, això és perquè el model està ben balancejat. A més, com es pot veure encara que teníem un *mAP_50* més petit que al Faster R-CNN hem obtingut un resultat de *F1Score* més alt amb el Cascade R-CNN.

4.3. Ensemble learning

El tercer experiment consisteix en introduir l'algorisme de *non-maxima supression* als models per unir-los i intentar obtenir millors resultats. Com s'ha esmentat prèviament un cop es tenen les *bounding boxes* dels dos models el que es fa és quedar-nos amb la que millor representi la predicció.

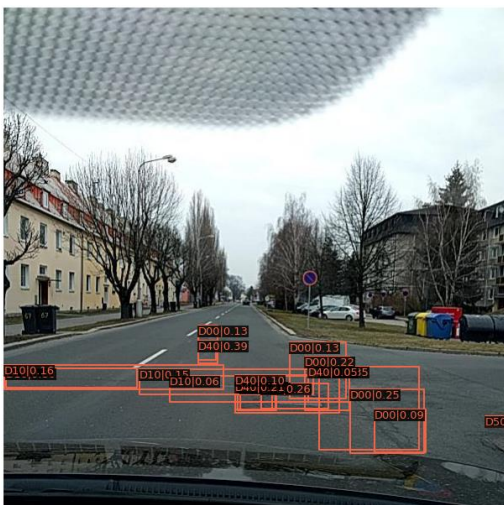


Figura 4.6: Predicció amb el Faster R-CNN

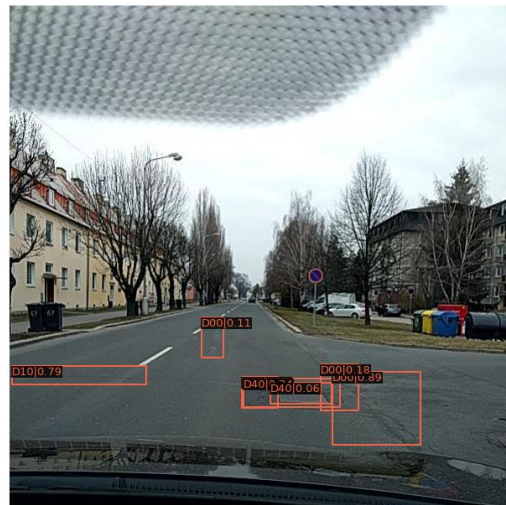


Figura 4.7: Predicció amb el Cascade R-CNN

Com es pot veure a les Figures 4.6 i 4.7, els dos models proposen unes *bounding boxes* que a vegades estan solapades entre elles. A més, hi ha alguns defectes que un model els ha detectat i l'altre model no, per exemple, la claveguera que està a la dreta de la imatge, el Cascade R-CNN no l'ha detectat. Per tant amb el NMS el que es vol aconseguir es quedar-se sempre amb la millor predicció dels dos models.

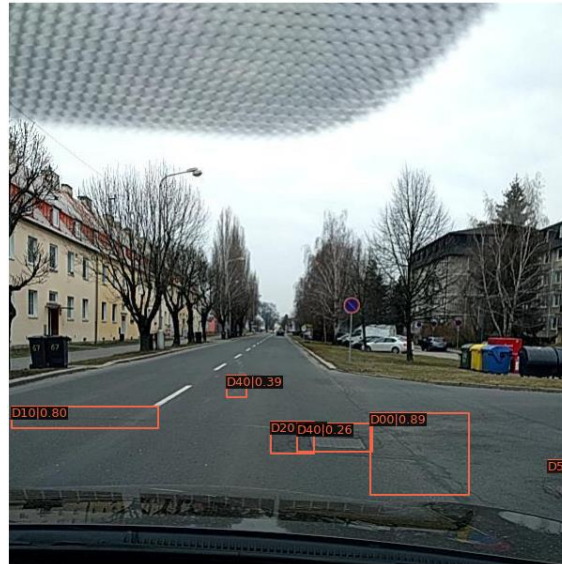


Figura 4.8: Predicció amb l'algorisme NMS implementat per ajuntar els dos models.

Un cop aplicat l'algorisme de *non-maxima supression* es pot veure a la Figura 4.8 que ens hem quedat amb la millor predicció de les *bounding boxes* que es solapaven i a més, ens han aparegut noves deteccions que no s'havien detectat amb un dels models.

D'aquesta manera s'ha calculat la *F1Score* per classes dels resultats obtinguts amb el *ensemble learning* dels dos models per poder comparar els resultats posteriorment (veure Figura 4.7).

| Classe | D00 | D10 | D20 | D40 | D43 | D44 | D50 | Total |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|
| Recall | 0,598 | 0,723 | 0,279 | 0,611 | 0,642 | 0,262 | 0,758 | 0,793 |
| Precision | 0,727 | 0,741 | 0,588 | 0,745 | 0,741 | 0,739 | 0,729 | 0,724 |
| F1Score | 0,656 | 0,732 | 0,379 | 0,672 | 0,688 | 0,386 | 0,744 | 0,757 |

Taula 4.7: Recall, precision i F1Score per classes amb el NMS dels dos models

Com es pot veure a la Taula 4.7 els resultats són prou bons però amb el Cascade R-CNN s'obtenen millor resultats. Això és perquè al realitzar l'algorisme de *non-maxima supression* el que s'aconsegueix és reduir el número de falsos negatius a canvi d'augmentar el nombre de falsos positius. D'aquesta manera s'obtindran millors resultats de *F1score* si el numero de falsos negatius reduïts és més gran que el número de falsos positius augmentats. En el aquest cas es pot veure, el *recall* ha augmentat però no lo suficient per compensar la caiguda en la *precision*. Per tant, els millors resultats obtinguts són amb el Cascade R-CNN.

4.4. Comparació amb el concurs (GRDDC)

Per últim s'ha fet una comparació de les solucions proposades amb els resultats del concurs, GRDDC.

Com s'ha mencionat abans els resultats obtinguts del projecte són a partir d'un subdataset de test creat del dataset de train del GDRCC. En canvi, els resultats dels models del concurs són del dataset test1 que dona el concurs. Això s'ha fet així perquè no es tenia a l'abast les anotacions del *ground truth* de les imatges de test1 del GDRCC. Per això encara que es faci una comparació dels resultats, veure Taula 4.8, no es poden extreure conclusions ben fonamentades perquè els resultats provenen de dos datasets diferents.

| Models | Faster R-CNN | Cascade R-CNN | Models amb NMS | Millor model concurs |
|----------------|--------------|---------------|----------------|----------------------|
| F1score | 0,745 | 0,798 | 0,757 | 0.6748 |

Taula 4.8: Comparativa amb els diferents solucions proposades i el concurs GRDDC

Com es pot veure els models del projecte superen els resultats dels models del concurs. Com ja s'ha comentat aquesta comparació no es gaire significativa, encara així, els resultats obtinguts amb les solucions proposades són molt bons.

5. Pressupost

Aquest projecte s'ha dut a terme, en la seva totalitat, a casa a causa de la pandèmia de la COVID-19. Encara així s'han afegit algunes despeses bàsiques, tals com el lloguer, per realitzar una aproximació més real del cost del projecte.

El pressupost del projecte està dividit en tres parts: el costos del personal, les despeses generals i les amortitzacions del material emprat.

| Càrrec | Salari per hora |
|-----------------------|-----------------|
| Desenvolupador junior | 11 € |
| Supervisor | 70 € |

Taula 5.1: Sous per hora i càrrec.

| Càrrec | Salari | Despeses a la SS (33,4%) | Número | Total |
|-----------------------|--------|-----------------------------|--------------|-------------------------|
| Desenvolupador junior | 880 € | 293,92 € | 1 | 1173,92 € |
| Supervisor | 560€ | 187,04 € | 2 | 747,04 € |
| | | | Total | 1920,96€/mensual |

Taula 5.2: Costos mensual a causa dels sous.

| Despeses Generals | Mensuals |
|-----------------------|----------|
| Lloguer | 500 € |
| Costos d'amortització | 125 € |
| Llum | 92 € |
| Aigua | 30 € |
| Internet | 40 € |

Taula 5.3: Costos de les despeses generals.

| Materials | Preu per unitat | Cost d'amortització | Longevitat | Amortització anual | Amortització mensual |
|------------------|-----------------|---------------------|------------|--------------------|----------------------|
| Ordinador | 825 € | 8940 € | 5 | 1788 € | 149 € |
| Pantalla | 150 € | 1620 € | 5 | 324 € | 27 € |
| Escriptori | 175 € | 1882 € | 8 | 235 € | 20 € |
| Cadira d'oficina | 160 € | 1720 € | 8 | 215 € | 18 € |
| | | | | Total | 214 € |

Taula 5.4: Costos dels materials necessaris per dur a terme el projecte.

En aquesta simulació s'ha tingut en compte que el projecte s'ha dut a terme durant quatre mesos.

| Tipus de Despesa | Cost |
|-------------------|------------------|
| Salaris | 7683,84 € |
| Despeses Generals | 3148 € |
| Material | 0€ |
| Total | 10831,84€ |

Taula 5.5: Costos totals del projecte.

6. Conclusions i desenvolupaments futurs:

S'ha aconseguit l'objectiu principal del projecte que consistia en trobar el model que millor s'adaptés al dataset del concurs i que després es pogués utilitzar a escala global. S'han presentat fins a tres possibles solucions a implementar on la millor és utilitzar el Cascade R-CNN. A més no tan sols s'ha aconseguit aquest objectiu si no que s'han obtingut millors resultats que qualsevol model del concurs, veure Taula 4.8. Encara així la comparació entre les solucions proposades i els models del concurs no és del tot significativa perquè s'han utilitzat dataset diferents. Això pot fer que el dataset del concurs sigui més difícil o amb una variabilitat que el creat en el projecte no pugui aconseguir.

Amb els tres experiments duts a terme durant la duració del projecte s'ha arribat a la conclusió que el millor model és el CascadeRCNN. Encara que es va intentar millorar els resultats utilitzant tècniques d'*ensemble learning* no van ser del tot bons perquè al reduir el nombre de falsos negatius van augmentar molt el de falsos positius. Es podria dir que això no es del tot negatiu ja que en l'entorn del projecte el cost dels falsos positius no es tan elevat com els dels falsos negatius, ja que interessa detectar tots els defectes possibles sense deixar-se cap.

Els propers passos per continuar la feina d'aquest projecte podrien ser els de dissenyar un algorisme més complet de *data augmentation* per millorar la variabilitat i quantitat de les imatges i així intentar trobar un model que generalitzi més. També es podria dissenyar un algorisme de *ensemble learning* més complet que tingui en consideració més aspectes a l'hora de decidir les *bounding boxes*.

Una altre solució que es podria plantejar es agrupar les classes que tinguin el mateix manteniment. Per exemple, totes les classes que estan referides a esquerdes s'arreglaran de la mateixa manera per tant, a l'hora de calcular la *precision* i el *recall* del model si aquest s'equivoca en alguna d'aquestes classes posant una de la mateixa família no es comptaria com a un error. Per veure si això augmentaria els resultats obtinguts s'hauria de fer una matriu de confusió per analitzar quants cops el model s'equivoca amb classes de la mateixa família.

Per últim, per poder avisar de forma automatitzada dels defectes de la carretera es podria crear una aplicació de mòbil que implementes el model i que permeti als usuaris gravar la via. D'aquest manera amb relativa velocitat les autoritats encarregades del manteniment de les carreteres estarien avisades.

Bibliografia

- [1] Deeksha Arya, Hiroya Maeda, Wakana Nogami. "Road Damage Detector Global Challenge". Online: <https://github.com/sekilab/RoadDamageDetector>
- [2] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, Yoshihide Sekimoto "Global Road Damage Detection: State-of-the-art Solutions". Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020. DOI: 10.1109/BigData50022.2020.9377790
- [3] Faizaan Naveed. "Application of Deep Learning in identifying road cracks". Oct 2019. Online: <https://towardsdatascience.com/application-of-deep-learning-in-identifying-road-cracks-8153e50ce9e2>
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". Cornell University. Jan 2016. Online: <https://arxiv.org/abs/1506.01497v3>
- [5] Zhaowei Cai, Nuno Vasconcelos. "Cascade R-CNN: Delving into High Quality Object Detection". Dec 2017. Cornell University. Online: <https://arxiv.org/abs/1712.00726>
- [6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. "You Only Look Once: Unified, real-Time Object Detection". Cornell University Jun 2015. Online: <https://arxiv.org/abs/1506.02640>
- [7] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection2. Cornell University. Apr 2020. Online: <https://arxiv.org/abs/2004.10934>
- [8] Glenn Jocher, Ultralytics. "Yolov5". Online: <https://docs.ultralytics.com/>
- [9] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, Yoshihide Sekimoto "Global Road Damage Detection: State-of-the-art Solutions". Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020. DOI: 10.1109/BigData50022.2020.9377790
- [10] Renu Khandelwal, "COCO and Pascal VOC data format for Object detection". Dec, 2019. Online: <https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5>
- [11] Renu Khandelwal, "COCO and Pascal VOC data format for Object detection". Dec, 2019. Online: <https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5>
- [12] . Different Contributors, more than 220. "Mmedetection",
Online github: <https://github.com/open-mmlab/mmdetection>
Online Documentation: <https://mmdetection.readthedocs.io/en/latest/>