FlutterCV: Flutter App development with Computer Vision modules

Juan Carlos Soriano Valle

Resum—Les tecnologies referents al Machine Learning i la Visió per Computació avancen molt ràpidament en temps, però no només en temes de precisió sinó d'eficiència computacional, on cada vegada es necessiten menys recursos de còmput per executar tasques igual o més complexes. És per això que dispositius tecnològics lluny de superordinadors o fins i tot ordinadors personals estan cada vegada més a prop de tenir compatibilitat nativa amb aquest tipus de mòduls. Juntament amb la poca implementació d'interfícies d'usuari avançades que hi ha a la menció de Computació, aquest projecte es centrarà en unir aquests dos mons en una aplicació mòbil desenvolupada en Flutter incorporant funcions aplicades amb xarxes neuronals com YOLOv5 aplicat a Object Recognition o ML-Kit per funcions d'OCR. Es desenvoluparan dues versions de l'aplicació per comparar una implementació local i una Cloud dels models de ML. El cas d'ús serà desenvolupar una app "Companion" pel videojoc Animal Crossing: New Horizons que serveixi al jugador com a suport per tenir un catàleg de tot el seu inventari i pugui ajudar-se de funcionalitats que no s'han posat al mercat d'aquest tipus d'aplicacions fins ara.

Paraules clau—Machine Learning, Visió per Computació, Animal Crossing: New Horizons, Flutter, Dart, YOLOv5, ML-Kit, SSD-MobileNetV2, Cloud, local

Abstract—Machine Learning and Computer Vision technologies are growing fast nowadays, not only on precision aspects but on computation efficiency where less computing resources are required in order to run the same tasks or more complex ones. This is the reason behind devices far from supercomputers or domestic computers are going to adquire native compatibility sooner or later. In conjunction with the very low presence of advanced GUIs in the career branch, this project will be focused on getting together those two worlds inside a Mobile phone app developed with Flutter and integrating neural networks funcions such as YOLOv5 applied to Object Recognition or ML-Kit for OCR funcions. Two versions of the app will be developed due to having a comparison between a local and Cloud implementation of the ML models. The use case will be to develop a "Companion" app for the Animal Crossing: New Horizons videogame that helps the player recording his inventory and game progress and having the opportunity to use functionalities that are unknown in the corrent "Companion" apps market.

Index Terms—Machine Learning, Computer Vision, Animal Crossing: New Horizons, Flutter, Dart, YOLOv5, ML-Kit, SSD-MobileNetV2, Cloud, local

1. INTRODUCCIÓ - CONTEXT DEL TREBALL

En els darrers anys, el sector de l'oci té un component que creix de forma exponencial: el videojoc. Aquest fet té moltes causes: van ser un salvavides per la gent durant la pandèmia, cada vegada apareixen més en tot tipus d'anuncis, està creixent la capacitat de fer videojocs més reals o més òptims per poder jugar en dispositius de baix-mig nivell, etc.

Un dels grans videojocs que van ser claus durant la pandèmia va ser l'*Animal Crossing: New Horizons* [1]. Aquest és un videojoc que va sortir just va començar la quarantena global, el 20 de Març del 2020, i gràcies a aquesta situació juntament amb l'estil de videojoc i el temps que portaven els jugadors esperant una nova entrega d'aquesta saga, va acabant sent un exit en vendes [2]. Tant és així que va aconseguir vendre un total de 11,7 milions d'unitats durant els primers 10 dies al mercat a tot el mon. A data de febrer de 2022, es situa en les 37,62 milions d'unitats, havent venut 32,63 milions només en el primer any (fins març de 2021). A més de totes les vendes, va ser nominat a tots els premis de millor joc de l'any 2020 i 2021 [3], guanyant premis com "Japan Game Awards – Game of the Year", "The Game Awards 2020 – Best Family Game", entre d'altres.

Com que aquest és un videojoc que tracta de tenir el teu personatge que va fent la seva vida dia a dia amb diferents tasques, objectius i col·leccionables, arriba un moment en el progrés del joc que el jugador no pot tenir memoritzat l'estat de totes les tasques i catàleg del seu inventari. És per aquest fet que ja fa un temps s'han posat de moda unes aplicacions o programes que t'ajuden a complir aquest objectiu afegint al jugador eines concretes per a cada videojoc.

Aquestes eines reben el nom de "Companion". En el cas de l'*Animal Crossing*, una app *Companion* ajudaria l'usuari a mantenir un estat actualitzat de la seva col·lecció

- *E-mail de contacte: juancarlos.soriano@autonoma.cat*

- *Menció realitzada: Enginyeria de Computació*
- *Treball tutoritzat per: Felipe Lumbreras Ruiz (departament de Ciències de la Computació)*
- *Curs 2021/22*

d'animals i obres d'art que pot recopilar al museu de la seva ciutat, tenir informació detallada de cada animal i quan es poden obtenir (meteorològicament, estació de l'any, hora del dia, etc.).

Aquestes tasques ja les fan aplicacions que s'han publicat, però des d'una perspectiva d'un jugador "veterà", trobo que hi ha moltes tasques que no s'han fet encara i que millorarien molt l'experiència d'usuari del videojoc. Entre elles, ajudaria a l'usuari a reconèixer si l'obra d'art que un personatge determinat del videojoc intenta vendre és verdadera o és una imitació quasi perfecta. Aquest personatge té un gran nombre d'obres d'art per vendre, però només apareix 1 vegada cada una o dues setmanes, i de les 5 peces que et ven, normalment només 1 és verdadera.

Aquest projecte doncs implementarà funcions que no s'han implementat fins ara en les apps Companion d'aquest joc amb l'ajuda de mòduls de visió per computació dintre de l'app mòbil.

Per donar una mica de context a les accions que s'implementaran cal explicar amb detall les mecàniques del videojoc.

En primer lloc, les obres d'art són un element important ja que per completar el museu s'han de donar totes les peces reals, les imitacions no s'accepten. El jugador farà una fotografia a una pantalla de la consola Nintendo Switch [4] similar a la que es mostra a la figura 1.

Fig. 1. Vista prèvia abans de comprar l'obra "Dama amb un ermini"

Una altra mecànica és la d'atrapar insectes, peixos o criatures marines. Quan això passa el personatge del videojoc ensenya per pantalla l'animal que ha atrapat juntament amb una frase característica i diferent per a cada espècie. Aquesta reacció es pot veure en la figura 2.

Els animals també es poden trobar documentats en la

Fig. 2. Frase de pesca d'una tonyina

enciclopèdia que incorpora el videojoc. En aquest apartat es poden veure elements com el nom de l'animal, quan es pot atrapar tant en hores com en mesos i si s'ha donat al museu. A més, hi ha una fotografia gran de l'animal en qüestió. Aquesta és una eina molt important pels jugadors amb un progrés més avançat, ja que des d'aquí es pot comprovar que falta per completar cada secció o si hi ha animals que surten en mesos o hores llunyanes, pot servir per no haver d'esperar-se mesos per comprovar qualsevol aspecte. La enciclopèdia canvia segons l'especie que recull i és similar al que es mostra a la figura 3.

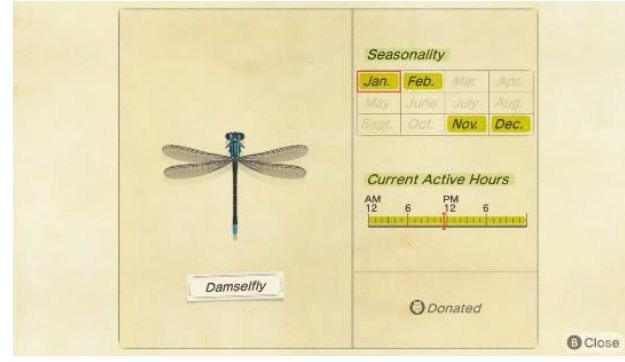

Fig. 3. Mostra de l'entrada de l'enciclopedia d'un insecte

2. OBJECTIUS DEL PROJECTE

Tal com s'ha comentat breument en l'anterior punt, aquest projecte té com a objectiu aprofitar funcions de la Visió per Computació per ajudar a l'experiència de joc de l'"Animal Crossing". El desenvolupament tindrà una forma seqüencial, a on a cada iteració s'implementarà una funcionalitat diferent, i no es començarà la implementació de les següents funcions fins que l'actual no arribi a un estat de *Minimum viable product*. Aquests subobjectius són:

- **Preparar un dataset personalitzat** per a cada model que s'hagi d'entrenar.
- **Reconeixement de l'obra d'art**. La pantalla a la que el jugador farà la fotografia és de 6,2" amb tecnologia LCD, per tant s'ha de tenir molt en compte que el programa que es desenvolupi tingui en compte els detalls a una pantalla tan petita i s'haurà de tenir en compte aspectes com la il·luminació, la qualitat del sensor, estabilització, etc.
- **Reconeixement de l'animal en pantalla de captura (OCR)**. L'objectiu és utilitzar un model OCR per esbrinar de quin tipus d'animal es tracta i que posteriorment l'aplicació et doni tota la informació necessària.
- **Reconeixement de l'animal en l'enciclopedia**. D'igual manera que l'objectiu anterior, però en cas que sigui un exemplar molt difícil de trobar, no s'estigui en la hora o mes correcte o estiguis en un punt molt avançat del joc.
- **Aplicar els mòduls desenvolupats a un entorn de producció** i comparar les implementacions local i Cloud.

3. ESTAT DE L'ART

Els diferents mòduls de Visió per Computació que con-

formen el projecte són de temàtiques diferents, d'aquesta manera, en aquesta secció es tractaran diferents temes en comptes de centrar-se en un de sol.

3.1 Object Detection

Aquesta tasca es basa en detectar instàncies d'objectes d'una classe determinada en una imatge o un vídeo. Dintre dels mètodes més coneguts i utilitzats es poden trobar 2 tipus:

- **Mètodes "One-Stage":** Aquests mètodes es basen en prioritzar la velocitat de la inferència per tal de tenir resultats més ràpids, sacrificant la precisió del mètode. Entre els mètodes que entren dintre d'aquest rang es troben el YOLO [5], SSD [6] i RetinaNet [7].
- **Mètodes "Two-Stage":** Aquests mètodes es basen en donar més passos ja sigui de transformacions a les dades o passades extres dels models, el que fa que augmentin la precisió de la detecció per tenir uns millors resultats finals. Exemples de mètodes d'aquest tipus són: Faster R-CNN [8], Mask R-CNN [9] i Cascade R-CNN [10].

A part d'aquests mètodes més coneguts, n'hi ha un que recentment s'està consolidant com el mètode que millors resultats aconsegueix en aquest camp. Aquest és el SwinV2-G [11], que es basa en escalar la capacitat i la resolució màxima possible amb la que es pot entrenar el model Swin Transformer. Les puntuacions es fan sobre els datasets ADE20K [12], COCO minival i COCO test-dev [13], ImageNet [14], entre d'altres.

3.2 OCR

En aquest camp hi ha diferents aproximacions que intenten interpretar el text que conté la imatge a processar. Hi ha dues fases per tal de llegir un text en una imatge. El primer pas és detectar el text en la imatge. Entre aquestes tècniques es troben la "sliding window" que es basa en anar desplaçant una finestra per tota la imatge. Un exemple de mètode que utilitza aquesta tècnica és una CNN [15]. D'altres tècniques que existeixen són les "single shot" com la YOLO [16] a la qual es passa només una vegada la imatge i detecta el text en una regió determinada i per últim l'"EAST" [17], un mètode molt avançat i precís que fins i tot pot detectar text en un vídeo a 13fps en qualitat 720p.

L'altre pas de l'OCR és reconèixer el text que hi ha a la regió. Per això es fan servir Xarxes Neuronals tals com la CRNN [18], una combinació de xarxa convunacional i xarxa recurrent.

3.3 Aplicacions Mòbils

Aquest és un sector que evoluciona molt ràpid, on surten noves tecnologies cada poc temps i les que existeixen s'actualitzen de forma molt freqüent si no volen estancar-se i desaparèixer. Entre aquestes plataformes [19] s'ha de tenir en compte el factor més important: el SO. Les que més s'utilitzen actualment en el món d'Android són:

- **JavaScript**: Aquest llenguatge de programació va ser un dels més utilitzats gràcies al gran nombre de frameworks que l'utilitzen com poden ser Angular, React, etc. Destaca la seva facilitat per crear una estructura de full-stack molt còmodament. En el cas de React, també ens permetrà programar apps per iOS.
- **Java**: D'igual manera que JavaScript, aquest és un llenguatge molt polivalent que també permet fer una estructura full-stack.
- **Kotlin**: Al tenir suport natiu d'Android, Kotlin és la millor opció si es vol desenvolupar una aplicació en Android i es volen aprofitar totes les característiques tant del sistema operatiu com del telèfon mòbil.

En canvi, si es vol desenvolupar una app en iOS, les millors opcions són:

- **Swift**: És el llenguatge oficial d'iOS. Per aquest fet, té suport directe d'Apple, una gran avantatge al ser un sistema operatiu tan tancat. Al ser natiu, utilitza totes les funcions possibles del telèfon mòbil.

Fora de l'àmbit natiu, el framework que està creixent d'una forma exponencial és Flutter [20]. Aquest framework que ha creat Google fa servir el llenguatge Dart. La principal avantatge d'aquest sistema és que d'un mateix codi es pot crear una aplicació d'Android, d'iOS, d'escriptori de Windows i fins i tot una pàgina web.

4. Metodologia i Eines

En aquest projecte s'utilitzaran diferents mètodes i eines per tal de portar una correcta gestió i desenvolupament de tot el conjunt de fases del projecte.

Pel que fa a metodologies, es farà servir una metodologia àgil tal com és SCRUM [21], amb una configuració d'sprints. La seva durada estarà determinada per les diferents entregues que hi hagi en el transcurs del TFG, amb l'excepció d'alguna funcionalitat gran que tingui moltes subtasques o períodes molt grans entre lliuraments. Això fa que els sprints tinguin una durada d'entre 2-3 setmanes, una durada òptima per tal de treballar amb aquesta metodologia.

Pel que fa a les eines que s'utilitzaran, aquestes seran: Jira [22] per la gestió i els seguiments dels sprints, Notion [23] per la documentació interactiva del progrés i experiments que es duran a terme en el transcurs del projecte, Github [24] per la creació del repositori del projecte i Toggl [25], una eina de time tracking.

Pel que fa a eines de desenvolupament, s'utilitzaran Roboflow [26] per la creació i gestió de datasets a més de l'etiquetatge de les classes per tal d'entrenar els models. Una altra eina serà WandB [27] que ens ajudarà a tenir una millor perspectiva del progrés dels entrenaments i experiments.

Es trobarà informació més detallada sobre les eines i la seva utilització en el dossier (pàgina de Notion) del projecte.

La majoria d'aquestes eines i metodologies estàn pensades per aplicar-les en projectes col·laboratius amb un equip de persones. En aquest cas, el projecte només es desenvoluparà per una persona, però és una molt bona oportunitat per tenir un primer contacte en un àmbit de projecte real.

5. PLANIFICACIÓ INICIAL DEL PROJECTE

Les diferents tasques del projecte es basen en la redacció de l'informe del treball, la documentació dels diferents temes que es tractaran en el seu transcurs i les tasques de desenvolupament de les diferents funcionalitats, tant de l'aplicació com dels mòduls de Visió per Computació.

Per tant, s'haurà de tenir en compte que en algunes etapes del projecte es faran algunes d'aquestes tasques de manera simultània.

D'una manera molt preliminar i sense tenir cap referència del temps de les tasques la planificació inicial seria aquesta:

Data	Output desitjat
06-03-22	Primer Informe i documentació tècniques de VC inicials
20-03-22	Primera versió mòdul image recognition
27-03-22	Millores mòdul Img Rec i primera versió de l'App
10-04-22	Informe progrés I i millores mòdul Img Rec
24-04-22	Img Rec acabat, primera versió OCR i millora de l'aplicació
08-05-22	OCR acabat
22-05-22	Informe progrés II i primera versió Img Rec animals
05-06-22	Desenvolupament app i Img Rec animals acabat
12-06-22	Proposta Informe Final
26-06-22	Proposta Presentació
27-06-22	Lliurament Dossier

Taula 1. Planificació inicial del projecte

Per aprofundir més en aquesta planificació, consultar al dossier del projecte el Diagrama de Gantt corresponent.
Cal aclarir que aquesta planificació és aproximada i pot canviar segons el transcurs del projecte i del seu progrés.
També existeix el fet que la part *backend* de l'aplicació es desenvoluparà tant d'una forma "local" com Cloud.

6. EXPERIMENTS I RESULTATS

6.1 Object Recognition (Obres d'Art)

La primera tasca que s'ha fet en aquest mòdul és la creació del dataset personalitzat amb imatges extretes directament del videojoc. En total, s'han aconseguit les 70 obres entre quadres i estàtues que existeixen i s'han fet múltiples fotografies per a cadascuna, canviant aspectes com la direcció de la foto, la proximitat de l'obra i la càmera, etc. La majoria de les imatges s'han capturat amb un telèfon mòbil en format horitzontal amb unes dimensions de 4000x3000 píxels. Un 2% de les imatges totals són captures de pantalla d'extractes de vídeos d'internet o una comparació directa entre 2 obres, l'original i la falsa.

Un cop construït el dataset, s'han etiquetat amb boxes i classes a través de l'eina Roboflow. Un dels avantatges que té aquesta eina és la possibilitat de fer un data augmentation sobre el dataset. Fent servir aquesta funció, s'ha afegit variació de brillantor, exposició i desenfocament intentant emular escenaris que es patiran en l'ús real de l'aplicació (càmeres de diferent qualitat, diferent enquadrament de les fotos, diferent llum, etc.).

El dataset consta d'unes 70 classes diferents i un mínim de 6 imatges per classe. En total, hi ha unes 446 imatges on hi ha mínim una classe representada i aproximadament 10 fotos on no hi ha cap classe. El total del dataset puja fins a les 982 imatges tenint en compte el data augmentation que s'aplica sobre el set de training.

Model	YOLOv5m	YOLOv5l	YOLOv5s6
Dimensions	640x640	640x640	1280x1280
Epochs	1000	800	1000
Temps	13 m 45 s	48 m 51 s	12 h 49 m
mAP_0.5	0.75	0.87	0.95
mAP_0.5:0.95	0.56	0.75	0.87
Recall	0.64	0.72	0.82
Precisió	0.87	0.82	0.91

Taula 2. Resultats de les configuracions de YOLOv5. La fila de "Temps" correspon al temps d'entrenament del model

Un cop creat un dataset robust i precís per entrenar un model d'Object Recognition, s'estableix YOLOv5 com l'arquitectura a entrenar i que es farà servir.
Gràcies a l'eina WandB que recomanen els propis autors del YOLOv5, es poden aconseguir i guardar tota la informació dels experiments fets amb la xarxa d'una manera eficaç i interactiva.
Als primers entrenaments, es va utilitzar la configuració de xarxa YOLOv5l amb imatges de 640x640 píxels, l'estàndard en aquesta xarxa. Es va escollir la configuració 'l' perquè tenia una precisió molt bona, però també era gran els requeriments de còmput. Els resultats no van ser molt satisfactoris, ja que les diferències entre les peces d'art amb alt grau de detall no tenien un bon resultat. Després de provar diferents solucions, l que millors resultats va donar va ser la d'augmentar la resolució de les imatges a 1280x1280.

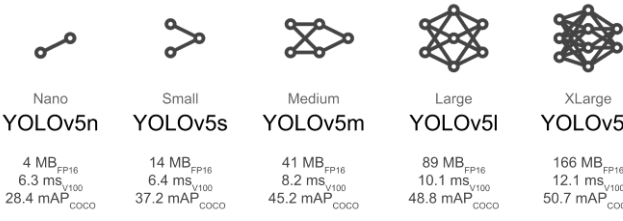

Fig. 4. Diferents models de YOLOv5

Per aquest motiu es va baixar la complexitat de la xarxa escollint la configuració YOLOv5s6 (el 6 del final denota una arquitectura especial per a imatges d'input amb dimensió 1280x1280). D'aquesta manera i al tenir imatges amb més definició, les característiques més subtils en les diferents peces d'art es podien identificar millor.

A més, el YOLO té una funcionalitat anomenada "*Hyperparameter Evolution*" que serveix per optimitzar els hiperparàmetres del model fent execucions curtes amb petites modificacions automàtiques i aleatòries als hiperparàmetres del model per trobar la millor configuració pel problema específic.

En la taula 2 es poden observar els resultats més rellevants amb les diferents configuracions.
Per la tasca d'integrar el model YOLO entrenat a l'entorn de l'aplicació Flutter es van fer servir els plugins *tflite* [28] i *tflite_flutter* [29]. La idea inicial era exportar el model entrenat a un arxiu .tflite per tal d'implementar-lo directament amb els plugins. Cap dels dos va donar resultat. Aquest fet es deu a que els plugins tenen una compatibilitat molt limitada amb els tipus d'extensions i de configuracions de xarxes. En el cas d'object detection, el factor que creava la incompatibilitat era el procés d'extracció del model YOLO a un arxiu ".tflite".

6.2 Image Classification (Obres d'Art)

A conseqüència de la incompatibilitat del model exportat de YOLOv5 amb l'entorn de Flutter, es van plantejar les alternatives d'entrenar un model YOLO anterior que fos compatible amb els plugins o canviar el tipus de model a un d'Image Classification. A més, es van entrenar altres xarxes d'Object Detection com MobileNetV2 o EfficientDet_Lite sense els resultats esperats.
Com que l'usuari final només veurà de quina obra d'art es tracta, la decisió final va ser canviar el model només per la part de desenvolupament local. El desenvolupament en cloud continuarà tenint com a model un YOLOv5.
El dataset consta de les mateixes imatges que s'utilitzen en l'anterior model, però en comptes d'etiquetar les classes amb caixes, se li assigna una classe a la imatge sencera. La manera d'etiquetar-les és crear un directori per cada classe. D'aquesta manera s'acabarà tenint un directori arrel que contindrà un directori per a cada classe de la que consta el dataset.

El model entrenat és una xarxa MobileNetV2 per a Image Classification, que rep una imatge i retorna la classe amb més confiança juntament amb el seu grau de confiança.

Els resultats obtinguts superen les expectatives (cal recordar que aquest model farà les inferències utilitzant els recursos d'un telèfon mòbil) tal i com es pot comprovar en la taula 3, comparant el model anterior d'Object Detection amb el d'Image Classification.

Model	YOLOv5m	YOLOv5l	YOLOv5s6	MobNetV2
Dimensions	640x640	640x640	1280x1280	224x224
Epochs	1000	800	1000	35
Temps	13 m 45 s	48 m 51 s	12 h 49 m	5 m
Precisió	0.87	0.82	0.91	0.87

Taula 3. Comparacions de resultats de YOLOv5 amb MobileNetV2. Cal recordar que com que un classificador només prediu la classe de la imatge no hi pot haver falsos negatius, llavors la recall no té sentit en aquesta comparació

L'explicació a la diferència en la relació temps d'entrenament – precisió es troba al tipus de model. Mentre que a un Object Detection es troba la ubicació dels diferents objectes en la imatge, apart de que es poden detectar més d'un, en un model d'Image Recognition només es detecta la classe de la imatge, sense localitzacions ni multiclasses, tal i com es veu en la figura 5. Per això, es poden entrenar amb imatges més petites i el temps d'entrenament és tant baix en comparació amb el model previ.
Gràcies al canvi de model, s'ha pogut canviar també el

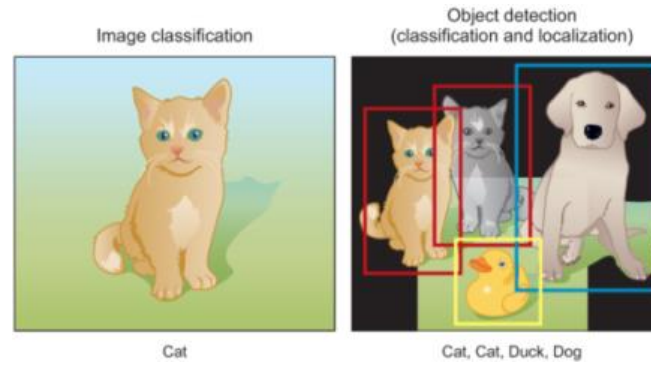

Fig. 5. Comparació entre Image Classification i Object Detection

plugin amb el que s'incorpora el mòdul de Visió per Computació a l'app. El nou plugin és el *tflite_flutter_helper* [30]. Aquest incorpora les opcions del *tflite_flutter* i millora la seva gestió i la facilitat de treballar amb ell.

A l'Apèndix 1 es poden trobar exemples de resultats de la fase de validació dels models, tant del YOLO com del MobileNet. Per a més detall en els experiments i els resultats, consultar el dossier del projecte.

6.3 OCR Local

Per la implementació d'aquest mòdul es van fer tests amb 3 tecnologies diferents: Tesseract [31], EasyOCR [32] i Google ML-Kit [33]. També estava l'opció de Keras-OCR [34], però es va descartar perquè no hi havia un mètode viable per in-cloure'l directament a Flutter, com si que hi ha amb els altres 2.
En l'entorn del projecte, les proves van afavorir Tesseract quan aquest es processava amb els recursos del telèfon. Aquest mètode funciona millor quan el text és desordenat i amb un reconeixement de text, a més de treure molt més rendiment de la CPU quan aquesta és limitada. En canvi, EasyOCR té millors resultats amb texts ordenats i estructurats com pot ser una factura i extreu millors resultats reconeixent números. Es pot veure aquesta comparació en profunditat en aquest article [35].
Però quan es van fer les proves amb el ML-Kit de Google es va veure que tant el rendiment com la precisió en el resultat eren molt satisfactoris. A més, és el que millor detecta l'estructura i forma del text apart de donar menys falsos negatius. Mentre que a Tesseract i EasyOCR es generaven molts caràcters en la sortida que no constaven a la pantalla, el ML-Kit donava uns resultats més reals. Tant a la figura 6 com a la taula 4 es pot veure una comparació entre els diferents motors d'OCR implementats en la solució del projecte.

El programa implementat directament a Flutter és capaç d'identificar tots els texts que apareixen en una imatge determinada. D'aquest text processat, s'extreu la categoria d'animal que hi ha.

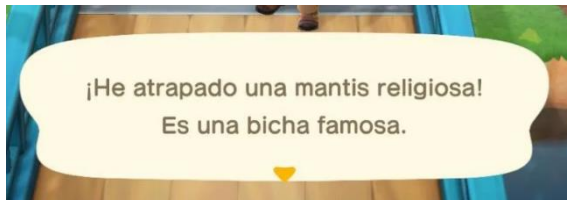

Fig.6. Imatge de captura d'insecte

Tesseract	(¡He atrapado una mantis, relogiosa!, Es una bicha famosa., V)
EasyOCR	(#84 , 69, 89, 40, '¡He atrapado una mantis religi famosa.)
ML-Kit	(¡He atrapado una mantis religiosa!, Es una bicha famosa.)

Taula 4. Comparació dels resultats dels diferents models d'OCR

6.4 OCR Cloud

L'objectiu d'aquest mòdul és crear des de 0 el motor d'OCR passant tant per la part de detecció com reconeixement de text. Per tant, la primera tasca ha sigut crear el dataset per detectar les zones de text.
En aquest problema en particular, les zones de text que es volen identificar sempre surten en la mateixa zona de bafarada en forma de núvol tal i com es veu en la figura 6. El dataset s'ha generat amb les mateixes eines i tècniques que el d'Object Recognitionm però en aquest cas s'han fet fotos a pantalles amb diferents diàlegs i s'han etiquetat les boxes com a text_bubble.

Fig. 7. Detecció de bombolles de text

S'ha entrenat un altre YOLOv5 per tal de fer la detecció de les zones de text.
Quan la zona on hi ha text ha estat reconeguda, la part d'OCR s'aplica sobre ella. En aquest cas s'ha escollit EasyOCR pel fet que té un rendiment superior en cas d'utilitzar la potencia d'una GPU i al tenir més capacitat de còmput els resultats són molt millors que el model directament implementat en Flutter. La figura 7 mostra un exemple del resultat d'una inferència del model sobre una imatge amb diàleg.

6.5 API Cloud

L'objectiu inicial d'implementar les funcions en un entorn Cloud comportava fer el desplegament en serveis com Google Cloud (o Firebase), AWS o similars, tot buscant les opcions gratuïtes i/o versions de prova. La tasca a fer és molt senzilla: rebre peticions a l'API, fer inferències amb el model prèviament carregat i retornar el resultat.

L'aproximació més propera a aquesta idea era utilitzar la plataforma de Google Cloud juntament amb Firebase. Com que Flutter i Firebase són de Google, existeix molta documentació i el procés és relativament senzill. El principal problema és que en aquest servei el model es descarrega en el telèfon mòbil i es allà on es fan les inferències. L'aspecte Cloud només ajuda en no haver-se de descarregar una versió nova de l'aplicació a qualsevol canvi al model, si no que es descarrega en local la nova versió.
Com que aquest no era l'objectiu del projecte, el que es va decidir va ser emular una experiència Cloud creant una API amb l'ajuda del paquet Flask [36] de Python. El funcionament és molt simple: Un script de Python funciona com a endpoint de l'API la qual rep peticions HTTP des del telèfon mòbil amb una fotografia que pot sortir tant de la càmera com de la galeria. El programa a l'ordinador fa la inferència sobre la imatge i retorna el resultat (classe predita i confidence) en format JSON que després l'app de Flutter el descodifica i interpreta la informació rebuda.

L'API està preparada per rebre peticions a una adreça IP de la xarxa de l'ordinador que executa l'API determinada pel programa i a un port determinat (en aquest cas s'ha configurat el port 5000). Depenent del procés que es vol executar, la petició acaba en /predictart si es vol interpretar una peça d'art o /predictocr si es vol interpretar un text de captura. El programa s'alimenta dels models exportats tant de l'Object Recognition en format .pt i l'OCR del checkpoint del model i el label map per la part de reconèixer la zona de text. Després utilitza EasyOCR per interpretar el text dintre de les boxes que ha generat el pas anterior.

Aquest procés és més lent que qualsevol mòdul localment executat ja que al fet que els procesos requereixen molta més capacitat de còmput i són més complexes, s'ha d'afegir la latència entre enviar i rebre la petició a l'API.

Un aspecte positiu d'aquest mètode és que l'execució a l'ordinador serveix també com a debug tant de l'Object Recognition com OCR, ja que mostra tant les coordenades de les boxes identificades com la confidence i el resultat de l'execució.
Una altra avantatge que te aquesta alternativa en front les implementacions locals dels mòduls és que se li aplica un preprocessament a les imatges abans de passar-les pels diferents algorismes. Entre d'altres, aquests preprocessaments són la correcció d'inclinació de la fotografia, una correcta transformació de mida de la imatge perquè sigui optima per cada model, etc.
La limitació més important que té aquesta implementació és que el telèfon i l'ordinador que està executant el programa han d'estar connectats a la mateixa xarxa. Aquest fet es deu a que la IP pertany a la connexió entre l'ordinador i la xarxa d'Internet, per això per poder accedir-hi s'ha de connectar al mateix Internet que l'ordinador.

6.6 Detecció d'espècie des de l'enciclopèdia

L'OCR segueix sent el mètode principal en aquesta funcionalitat. La primera aproximació va ser crear 2 versions com es porta fent en tot el projecte.
L'inconvenient que es va trobar va ser que el detector de bombolles de text no funcionava en aquesta pantalla, ja que la forma i el color del requadre no era el mateix tal i com es pot veure a la figura 3.
Depenent de la llum i la fotografia, era capaç de reconèixer la zona en un 40-50% de confidence, però en molts casos no ho detectava. Això és un problema perquè si es baixa el llindar de la confidence al model els resultats empitjoraran.
Per aquesta raó i per la falta de temps es va optar per només fer servir 1 proposta en aquest mòdul. S'utilitza el mateix paquet que es fa servir per l'OCR local, EasyOCR. El funcionament és molt senzill: el model busca en el text que surt per pantalla l'espècie d'animal i si surt el segell del museu. Amb tota aquesta informació, l'aplicació busca i mostra si l'usuari té marcat com a vist aquest animal en concret i si l'ha donat al museu. A més, l'aplicació canvia de pàgina per mostrar la informació sobre l'animal que s'ha processat.

6.7 Comparació models aplicacions

Després de veure totes les funcions implementades a les dues aplicacions, es poden observar petites diferencies entre elles. En la taula 5 es poden veure les diferencies més importants entre l'aplicació local i l'aplicació Cloud. Totes les característiques que no figuren en la taula estan implementades d'igual manera en ambdues aplicacions.

	Local	Cloud
Reconeixement d'art	✓	✓
Localitza l'art i pot reconèixer més d'1		✓
Predicció d'art com a categoria	✓	
OCR en frases de captura	✓	✓
OCR en pàgina d'enciclopèdia	✓	✓
Detecta a la pàgina d'enciclopèdia si l'espècie esta al museu	✓	
Requereix connexió a internet		✓
Guarda les dades d'usuari localment	✓	✓

Taula 5. Comparacions de funcionalitats per versió d'aplicació

6.7 Disseny de l'Aplicació Mòbil

Com ja s'ha parlat en els objectius i en plantejament del projecte, aquesta secció es separa en 2 versions: una aplicació amb models que s'executen localment al dispositiu mòbil i una versió Cloud, on les funcions de Visió per Computació s'executen de manera remota a través de peticions a una API que s'executa a un ordinador.
A nivell de disseny, totes dues són estèticament les mateixes i en termes de funcionalitats, només hi ha una diferencia entre elles.
El disseny recorda molt a la paleta de colors i els tons que s'utilitzen al videojoc, juntament amb els colors de la versió especial de la consola Nintendo Switch d'Animal Crossing: New Horizons.
Per navegar entre les diferents pantalles de l'aplicació, la barra de navegació inferior i el títol de la pàgina a la barra superior indiquen en quina pantalla ens trobem i si polsem en qualsevol icona de la barra es desplaçarà fins la pantalla del tema seleccionat.

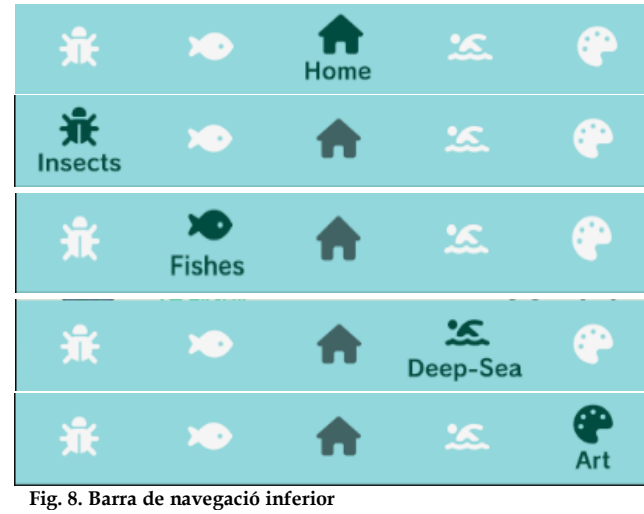

Fig. 8. Barra de navegació inferior

Les dades introduïdes es guarden localment gràcies a la característica de Flutter anomenada *shared preferences*, que permet guardar petites dades dintre de la memòria local del telèfon associada a l'aplicació. Per tant, cada vegada que s'inicia l'aplicació l'usuari veu les seves dades prèvies. Les dades entre l'aplicació local i Cloud són independents.

6.7.1 Pantalla Principal

Quan l'usuari entra a l'aplicació, aquesta és la pantalla que apareixerà. Aquí, es mostren els elements més importants i generals de l'aplicació. Entre aquests elements trobem:

- Dia actual
- Progrés de les diferents llistes d'items del videojoc: Insectes, peixos, criatures marines i peces d'art
- Especies d'animals que són novetat aquest mes diferenciant insectes, peixos i criatures marines.

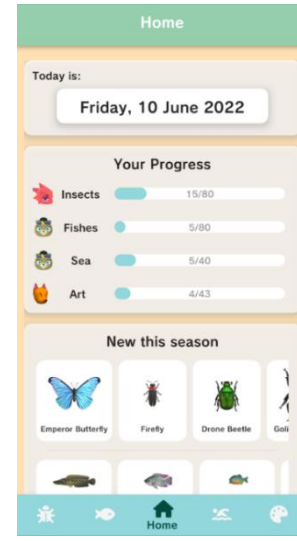

Fig. 9. Pantalla d'inici

Aquestes són les opcions que es troben a la pàgina principal i que són les més generals i habituals que el jugador utilitzarà. Si es pressiona sobre algun animal en la secció de novetats, s'obrirà la pantalla específica de l'animal.

6.7.2 Pantalles de llistes

Per a cada secció de l'aplicació hi ha una pàgina amb un llistat de tots els elements que es poden aconseguir de cada tipus. Hi ha 2 checks, un per seleccionar que has aconseguit aquest element i un altre per dir que l'has cedit al museu.

La informació que apareix a cada element de la llista varia segons l'estat del check que indica si has aconseguit l'item o no. En cas de tenir el check no seleccionat, la imatge apareix amb un to fosc, apareix la informació de com i quan obtenir-lo en el cas dels animals i si té falsificacions en el cas de les obres d'art. Quan es selecciona l'opció d'obtingut d'algun item, la seva icona es posa a color i la descripció canvia a la frase de captura en el cas dels animals. És en aquest moment quan l'usuari pot polsar sobre l'objecte per poder entrar en la seva pantalla d'informació.

En totes les pantalles de llistes figura a la dreta de la barra superior de l'aplicació una barra de progrés que indica l'estat de la col·lecció del tipus de la pantalla.

També hi ha 2 botons a l'espai inferior dret que serveixen per activar les funcions de visió per computació. Un serveix per buscar una imatge de la galeria del telèfon i una altra per obrir la càmera. Les dues opcions fan inferència amb el model corresponent i la imatge seleccionada.

6.7.3 Pantalla d'informació d'insectes

Aquesta pantalla mostra tota la informació corresponent a l'insecte que s'ha seleccionat.

A la barra superior es mostra el nom de l'insecte juntament amb la seva icona.

Tots els elements que es troben a la pàgina es poden veure a la figura 10. Aquests elements són:

- Imatge gran de l'insecte en qüestió. Si es pressiona a sobre s'obre en pantalla completa.
- Frase de captura de l'insecte.
- Preu comú i especial de venta de l'insecte.
- Estat d'obtenció i donació al museu.
- Mes on apareix i mes actual.
- Localització i raresa.
- Hora d'aparició i hora actual.
- Informació sobre l'insecte de Blathers, el conserge del museu.

6.7.4 Pantalla d'informació de peixos

Aquesta pantalla mostra la informació del peix seleccionat. D'igual manera que als insectes, a la barra superior figura tant el nom del peix com la seva icona.

A la figura 10 es pot veure un exemple d'aquesta pantalla. Els seus elements són:

- Imatge gran del peix en qüestió. Si es pressiona a sobre s'obre en pantalla completa.
- Frase de captura del peix.
- Preu comú i especial del peix.
- Estat d'obtenció i donació al museu.
- Mes on apareix i mes actual.
- Localització, raresa i forma de l'ombra a l'aigua.
- Hora d'aparició i hora actual.
- Informació sobre l'insecte de Blathers, el conserge del museu.

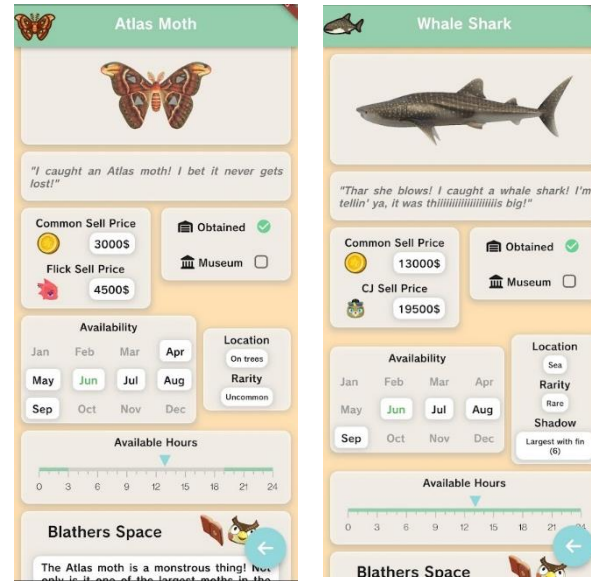

Fig. 10. Pantalles d'insectes i peixos, respectivament

6.7.5 Pantalla d'informació de criatures submarines

D'igual manera que la pantalla dels insectes i els peixos, en el cas de les criatures marines la pantalla inclou els següents elements, a part de la barra superior amb la icona i el nom de l'animal:

- Imatge gran de la criatura marina en qüestió. Si es pressiona a sobre s'obre en pantalla completa.
- Frase de captura de l'animal.
- Preu comú de l'animal.
- Estat d'obtenció i donació al museu.
- Mes on apareix i mes actual.
- Velocitat de moviment i forma de l'ombra a l'aigua.
- Hora d'aparició i hora actual.
- Informació sobre l'animal de Blathers, el conserge del museu.

A la figura 11 es pot veure un exemple d'aquesta pantalla amb tots els seus elements.

6.7.6 Pantalla d'informació d'obres d'art

Aquesta és la pantalla més diferent de totes ja que no es tracta de cap animal.

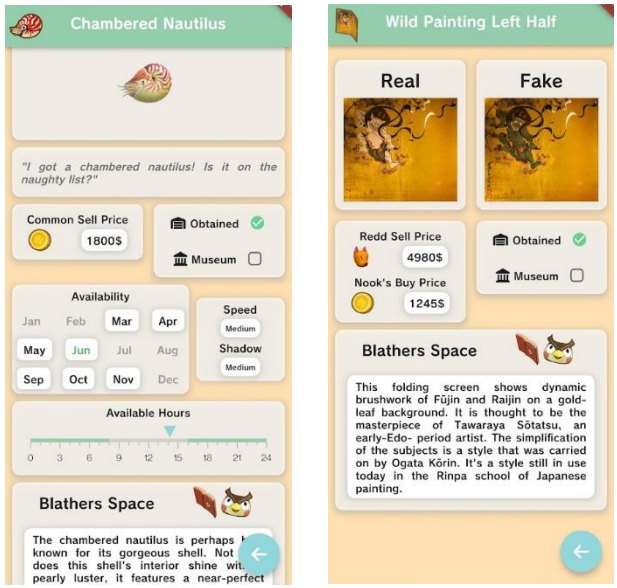

Fig. 11. Pantalles d'informació de criatures marines i obres d'art, respectivament

La barra superior si que té el mateix aspecte que les anteriors pantalles, tenint la icona del quadre o la estàtua i el seu nom. A partir d'aquí, els elements que figuren a la pantalla són:

- Comparació entre real i falsa, mostrant les imatges una al costat de l'altra, tenint la possibilitat de tornar-les en pantalla completa fent click a sobre. En cas que l'obra en concret no tingui falsificació, a la secció on hauria d'haver l'obra falsa hi haurà un text indicant que aquesta obra no té falsificació.
- El preu de compra i venta de l'obra. És el mateix tant per les obres reals com les falses.
- Estat d'obtenció i donació al museu.
- Informació sobre l'obra de Blathers, el conserge del museu.

Es pot veure un exemple d'aquesta pantalla a la figura 11.

6.7.7 Procés d'utilització de l'aplicació

El procés d'utilització dels mòduls de visió per computació canvia segons sobre quin tipus d'item es vol fer la inferència. L'usuari haurà de navegar fins la pàgina del tipus que es vulgui interpretar i polsar sobre el botó de càmera o galeria. El procés complet es pot veure apèndix 2.

Si l'usuari accepta el resultat de la predicció, l'aplicació obre la pantalla d'informació de l'objecte predit, on l'usuari pot mirar la seva informació i seleccionar els checks que corresponguin.

6.7.8 Feedback de l'aplicació

Per tal d'avaluar el rendiment i l'eficàcia de l'aplicació no hi ha mètriques com la precisió, recall per exemple com si que hi ha per avaluar els models. És per això que s'han fet user tests per tal de rebre feedback de persones externes al projecte.

El procediment s'ha basat en escollir persones que no han tingut cap participació en el projecte i demanar que utilitzin l'aplicació sense explicar el seu funcionament. El que si que s'ha intentat és que aquestes persones tinguin contacte previ amb el videojoc, ja que el target principal d'aquesta app és un jugador d'aquest videojoc.

En general, el feedback general és que l'aplicació té molta similitud a l'estètica original del videojoc. La usabilitat també es veu com un punt molt positiu ja que també recorda la usabilitat dels menús del videojoc.

Troben molt útil les funcions de reconèixer animals i obres d'art i s'apunta que és suficientment característic i amb personalitat que pot reemplaçar qualsevol aplicació Companion que han provat.

Pel que fa als aspectes negatius, sense cap explicació prèvia no s'entén quin check és d'obtenció i quin de museu sense, però a l'obrir la pàgina d'informació es veu cadascuna. També es troba a faltar una opció per restablir els checks i una opció de cerca.

7. CONCLUSIONS

Els resultats obtinguts en l'apartat dels mòduls de Machine Learning són molt satisfactoris. S'ha aconseguit cobrir els objectius inicials i s'han superat les expectatives prèvies en temes de rendiment i precisió. Ha hagut un canvi en la funció que identifica l'animal des de la pantalla d'enciclopèdia, però funciona correctament amb la implementació que s'ha aplicat.

També s'ha pogut desenvolupar una aplicació molt funcional i amb un bon disseny, preparada per ser publicada en qualsevol botiga d'aplicacions.

En general, s'ha pogut desenvolupar una aplicació amb funcions de Machine Learning tant en local com en Cloud gràcies a APIs per a un problema en particular, però aquest només és un cas d'ús. Aquesta aproximació es podria portar a una aplicació a empreses per tasques de control o distribució, per a una aplicació per gent que viatja i vol veure informació sobre qualsevol monument o edifici que té davant, etc. Les aplicacions són infinites, però l'important és que gràcies a aquest projecte s'ha pogut comprovar que és possible.

Pel que fa entre escollir entre la versió Cloud i la local, segons els resultats que s'han aconseguit en aquest projecte es faria un mix entre les dues versions, agafant els mòduls Cloud referent a interpretació d'imatges i la versió local per OCR.

8. SEGÜENTS PASSOS

Tot i que els resultats són satisfactoris, hi ha aspectes que es poden millorar, tant per falta de temps o perquè fa falta més investigació per poder implementar-ho. Aquests serien els punts importants a millorar en cas que es seguís el desenvolupament:

- Aplicar un Test-Time Data Augmentation per millorar les prediccions de les obres d'art.

- Més idiomes a l'aplicació.
- Millorar el disseny i desenvolupar més funcions a l'aplicació de gestió del videojoc escoltant el feedback rebut.

9. AGRAÏMENTS

Primer de tot m'agradaria agrair al meu tutor Felipe Lumbreras per tota la implicació durant tot el curs, tenint una reunió de seguiment cada setmana menys en situacions especials, per tal de seguir el progrés del projecte i donar feedback per les següents entregues. Després agrair a la meva parella Culara per tota l'ajuda tant per comentaris i feedback de jugador experimentat com per el suport anímic donat. Per últim, als meus amics i la meva família per el suport incondicional i desinteressat.

BIBLIOGRAFIA

[1] "Animal Crossing: New Horizons - Wikipedia". Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Animal_Crossing:_New_Horizons (accedit el 17 de febrer de 2022).

[2] "Unit Sales of Animal Crossing: New Horizons worldwide as of December 2021". Statista. https://www.statista.com/statistics/1112631/animal-crossing-new-horizons-sales/ (accedit el 4 de març de 2022).

[3] "Animal Crossing awards and nominations". IMDb https://www.imdb.com/title/tt10476972/awards/ (accedit el 4 de març de 2022).

[4] "Technical Specs - Nintendo Switch™ - System hardware, console specs - Nintendo - Official Site". Nintendo. https://www.nintendo.com/switch/tech-specs/#switch-section (accedit el 18 de febrer de 2022).

[5] Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao, "YOLOv4: : Optimal speed and accuracy of object detection" [Online]. Disponible: https://arxiv.org/abs/2004.10934

[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector" [Online]. Disponible: https://arxiv.org/abs/1512.02325

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, "Focal Loss for Dense Object Detection" [Online]. Disponible: https://arxiv.org/abs/1708.02002

[8] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" [Online]. Disponible: https://arxiv.org/abs/1506.01497

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, "Mask R-CNN" [Online]. Disponible: https://arxiv.org/abs/1703.06870

[10] Zhaowei Cai, Nuno Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection" [Online]. Disponible: https://arxiv.org/abs/1712.00726

[11] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, Baining Guo, "Swin Transformer V2: Scaling Up Capacity and Resolution" [Online]. Disponible: https://arxiv.org/abs/2111.09883

[12] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba, "ADE20K Dataset". https://groups.csail.mit.edu/vision/datasets/ADE20K/ (accedit el 25 de febrer de 2022).

[13] Tsung-Yi Lin, Genevieve Patterson, Matteo R. Ronchi, Yin Cui, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, Larry Zitnick, Piotr Dollár, "COCO Dataset". https://cocodataset.org/#home (accedit el 25 de febrer de 2022).

[14] Li Fei-Fei, Jia Deng, Olga Russakovsky, Alex Berg, Kai Li, "ImageNet Dataset". https://www.image-net.org/index.php (accedit el 25 de febrer de 2022).

[15] Mikey Taylor, "Computer Vision with Convolutional Neural Networks". https://medium.com/swlh/computer-vision-with-convolutional-neural-networks-22f06360cac9 (accedit el 27 de febrer de 2022).

[16] Grace Karimi, "Introduction to YOLO Algorithm for Object Detection", https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/ (accedit el 27 de febrer de 2022).

[17] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang, "EAST: An Efficient and Accurate Scene Text Detector" [Online]. Disponible: https://arxiv.org/abs/1704.03155v2

[18] Chandra Churh Chatterjee, "An Approach Towards Convolutional Recurrent Neural Network", https://towardsdatascience.com/an-approach-towards-convolutional-recurrent-neural-networks-a2e6ce722b19 (accedit el 27 de febrer de 2022).

[19] Javinpaul, "Top 5 Programming languages for Mobile App Development in 2022", Medium. https://medium.com/javarevisited/top-5-programming-languages-for-mobile-app-development-in-2021-19a1778195b8 (accedit el 27 de febrer de 2022).

[20] "Flutter Documentation". Flutter - Build apps for any screen. https://flutter.dev/learn (accedit el 17 de febrer de 2022).

[21] "What is Scrum?" Scrum.org. https://www.scrum.org/resources/what-is-scrum (accedit el 20 de febrer de 2022).

[22] "Jira | Issue & Project Tracking Software | Atlassian". Atlassian. https://www.atlassian.com/software/jira (accedit el 20 de febrer de 2022).

[23] "Notion - One workspace. Every team". Notion. https://www.notion.so/product (accedit el 20 de febrer de 2022).

[24] "Build software better, together". GitHub. https://github.com/about (accedit el 20 de febrer de 2022).

[25] "Meet Toggl. Three producs; One mission". https://toggl.com (accedit el 20 de febrer de 2022).

[26] "Overview – Roboflow". Roboflow. https://docs.roboflow.com (accedit el 11 de març de 2022).

[27] "Weights & Biases". WandB. https://docs.wandb.ai (accedit el 14 de març de 2022).

[28] Tflite Package. https://pub.dev/packages/tflite (accedit el 14 de març de 2022).

[29] Tflite_flutter Package. https://pub.dev/packages/tflite_flutter (accedit el 17 de març).

[30] Tflite_flutter_helper Package. https://pub.dev/packages/tflite_flutter_helper (accedit el 27 de març de 2022).

[31] "Tesseract OCR: Text localization and detection". Tesseract. https://pyimagesearch.com/2020/05/25/tesseract-ocr-text-localization-and-detection/ (accedit el 20 d'abril de 2022).

[32] EasyOCR. https://github.com/JaidedAI/EasyOCR (accedit el 20 d'abril de 2022).

[33] ML-Kit. https://developers.google.com/ml-kit (accedit el 21 d'abril de 2022).

[34] Keras-OCR. https://github.com/faustomorales/keras-ocr (accedit el 21 d'abril de 2022).

[35] "OCR engine Comparison – Tesseract vs EasyOCR". https://medium.com/swlh/ocr-engine-comparison-tesseract-vs-easyocr-729be893d3ae (accedit el 21 d'abnril de 2022).

APÈNDIX 1. VALIDACIONS DELS MODELS

Ground truth dataset Object Detection

Predicció model YOLO

Predicció model MobileNet

APÈNDIX 2. PROCÉS D'ÚS DE L'APLICACIÓ