Detecció de piscines en imatges aèries utilitzant tècniques de deep learning

Rubén Gasque Todolí

12/03/2023

Resum– L'objectiu d'aquest projecte és trobar el millor mètode de detecció d'objectes per tal de detectar piscines a les zones costeres catalanes a partir d'ortoimatges.
Per aconseguir-ho s'entrenaran i posaran a prova diverses xarxes neuronals i algoritmes classificadors amb imatges com a dades d'entrada per tal de determinar les imatges que contenen piscines. Un cop detectades totes les piscines compararem amb els resultats obtinguts per mitjà de mètriques d'avaluació per tal de decidir quin model s'adapta millor al problema plantejat.
Altrament es vol demostrar que per aquests problemes de detecció els algorismes de deep learning milloren considerablement als models clàssics de processament d'imatges.
Paraules clau– Aprenentatge profund, xarxa neuronal, detecció d'objectes, detecció de piscines, algoritmes classificadors

Abstract– The aim of this project is to find the best method of object detection in order to detect pools in Catalan coastal areas from orthoimages.
To achieve this, various neural networks and classifier algorithms with images as input data will be trained and tested in order to determine the images containing pools. Once all pools are detected, we will compare the results obtained by means of evaluation metrics to decide which model best fits the problem raised.
Otherwise, it is shown that by these detection problems deep learning algorithms improve considerably in classical image processing models.

Keywords– Deep learning, neural networking, object detection, pool detection, classifier algorithms

1 Introducció - Context del treball

Durant els darrers anys, hi ha hagut un increment considerable d'ús de mètodes de deep learning (xarxes neuronals) per afrontar problemes de detecció d'objectes. En aquest projecte ens volem centrar en la zona de Catalunya per tal de detectar piscines amb mètodes de deep learning i comparar els resultats amb mètodes de classificació clàssics per tal de veure la millora que els nous mètodes aporten al camp de la detecció d'objectes.

Anteriorment, s'han fet altres intents de detecció de piscines, però amb algoritmes de detecció per píxels[1], per aquest motiu ha sorgit la motivació de trobar el mètode més adient per fer aquesta detecció i encertar el nombre més gran de piscines possibles.

Com aquests treballs fets no eren els òptims alhora de la detecció, s'ha decidit aplicar tècniques d'aprenentatge profund a través de xarxes neuronals per tal d'aconseguir un model que pot ser més lent, però que classifiqui molt millor. Això és un avanç, ja que en aquest cas no tenim cap necessitat d'immediatesa sinó de qualitat a l'hora de classificar.

Considerem clau, per tant, que la primera fase de detecció sigui fiable per tal de tenir en compte el màxim nombre de piscines i que el filtre que fem sigui el més acurat possible.

Per aquest motiu el principal objectiu del treball serà trobar el mètode que millor detecti les piscines, centrant l'interès en les mètriques resultants de cada aprenentatge.

Les imatges utilitzades seran ortoimatges extretes de la pàgina de l'Institut Cartogràfic i Geològic de Catalunya. [2].

• E-mail de contacte: 1563725@uab.cat
• Menció realitzada:Computació
• Treball tutoritzat per:Robert Benavente (Ciències de la computació)
• Curs 2022/23

2 OBJECTIUS

Primer de tot ens hem plantejat una sèrie d'objectius que volem complir al llarg de la realització del projecte. Aquests objectius els hem classificat en objectius principals (objectius que s'han de complir necessàriament durant el projecte) i objectius secundaris (interessants de complir però no imprescindibles).

La prioritat del projecte és l'anàlisi de les xarxes neuronals YOLO i els resultats obtinguts de la detecció de piscines. Per tant, valorem que com a objectiu principal del projecte hem de trobar el model de detecció que millor funciona per al nostre cas i tractar de maximitzar la detecció per trobar un model el més bo possible.

Per això la part de l'anàlisi de mètriques que avaluen els models passa a ser molt prioritari de cada a poder fer aquesta comparació.

També és considera com a objectiu l'entrenament de mètodes clàssics com KNN o SVM per tal de comprovar que són menys eficients que els anteriorment mencionats.

2.1 Objectius Principals

- Entrenar tres diferents versions YOLO de Machine Learning de detecció d'objectes (xarxes neuronals de tipus CNN) amb ortoimatges de la costa de Catalunya per tal de detectar les piscines que s'hi troben en elles, i aprofitar les imatges per entrenar models de classificació més tradicionals i antics per comprovar la seva qualitat de classificació.

- Fer una anàlisi detallat dels resultats per mitjà de diverses mètriques d'avaluació de models per tal de concloure quin model és el que més ens convé en el nostre cas i extreure informació útil sobre l'estudi fet per tal d'optimitzar els resultats.

2.2 Objectius Secundaris

- Estimar els paràmetres òptims dels algorismes classificadors clàssics utilitzats per tal d'aconseguir una optimització en els models i que classifiquin les piscines simulant una detecció.

- Millorar els paràmetres que s'han obtingut en treballs similars anteriors on s'han implementat xarxes YOLO per tal de detectar piscines.

3 ESTAT DE L'ART

La detecció de piscines ha sigut un tema estudiat per la visió per computador i el processament d'imatges. Pel seu estudi s'han realitzat diversos mètodes que exposo a continuació.

3.1 Detecció d'objectes

- Ús de xarxes neuronals convencionals: com explicarem més endavant, les xarxes neuronals convencionals són una de les opcions més utilitzades per la detecció d'objectes. S'han realitzat una sèrie d'estudis amb aquestes xarxes, ja que tenen la capacitat d'aprendre característiques úniques presents a les imatges per tal de trobar en noves imatges patrons similars i així poder detectar les piscines noves. [3] [4] Amb aquestes tècniques s'han obtingut uns resultats molt millors que amb altres mètodes tradicionals i, per tant, ha sigut un avanç considerable de cara a la detecció.

- Detecció de textures: Ús de la textura que té la piscina per la seva detecció. Per fer aquests treballs s'analitzen algunes característiques de la textura com el coeficient de correlació o l'homogeneïtat. S'han realitzat treballs en altres camps relacionats amb aquest tipus de detecció. [5] [6] Aquests estudis han resultat positius per a la detecció, però no són fiables depenent del context perquè segons les imatges seleccionades les textures de la piscina poden ser confoses amb altres objectes.

- Detecció de vores i contorns: Ús d'algorismes de detecció de contorns (com Sobel o Canny) per tal de limitar els objectes a través del seu contorn. Amb aquests resultats i aplicant tècniques de segmentació d'imatges es podrà separar la piscina del fons. Altres treballs realitzats sobre la detecció de contorns per detectar objectes han obtingut bons resultats de cara a la detecció. [7] [8] Són mètodes molt utilitzats, però tampoc acaben de ser fiables, ja que són sensibles a canvis d'il·luminació o textures i pot donar lloc a confusions, reduint, per tant, la precisió del model.

- Anàlisi de formes geomètriques: Mètodes que fan servir la forma o la mida de les piscines per tal de detectar-les. Alguns estudis han fet servir tècniques com la transformada de Hough per detectar cercles i el·lipses, formes característiques de les piscines. En aquest cas no s'han trobat treballs relacionats de forma directa amb les piscines però si en detecció mitjançant deep learning. [9] Els resultats obtinguts per aquests mètodes han sigut prometedors, però igual que els anteriors són sensibles a trobar formes semblants que poden fer que la seva precisió no sigui tan elevada.

3.2 Detecció de piscines

Com hem explicat anteriorment, al llarg dels anys s'han fet molts estudis sobre la detecció d'objectes en molts camps, i un camp molt estudiat ha estat la detecció de piscines. A continuació expliquem una sèrie de treballs relacionats.

- El primer treball és un treball realitzat als Estats Units fet amb la motivació de controlar un virus anomenat "Virus del Nil"que es transmet pels mosquits. Per això es vol controlar les zones amb més piscines i s'implementa un mètode HPS (high pass filter) i mètodes de classificació de píxels per tal de detectar les piscines. [10] És un treball basat en tècniques de preprocessat d'imatges alhora de realitzar deteccions de piscines.

- Aquest treball realitzat a Espanya tracta sobre la detecció de piscines a partir d'un mètode creat pels mateixos autors basat en la teoria Dempster–Shafer, una generalització de la teoria de Bayes (probabilística). En aquest treball comparen els resultats obtinguts amb un anàlisi SVM i obtenen molt bons resultats i similars en ambdós casos.[11]

- Com a treball realitzat en detecció de piscines amb xarxes neuronals s'ha triat un treball realitzat a Portugal. L'objectiu d'aquest treball és implementar xarxes neuronals amb arquitectura ResNet per tal de detectar piscines il·legals. [12] En aquest cas obtenen uns resultats pràcticament perfectes ja que classifiquen la gran majoria de les piscines de forma correcta.

3.3 Punt de partida

S'han tingut en compte dos projectes semblants al que es vol realitzar, dos treballs que fan ús de xarxes neuronals CNN per tal de detectar piscines.

El primer treball és un treball de fi de màster dut a terme pel Biel Stela Ballester, de la Universitat Autònoma de Barcelona, i tracta de detectar piscines il·legals a través de xarxes neuronals YOLO i ortoimatges de l'illa de Mallorca.[13] Aquest projecte ha sigut la motivació del nostre treball i hem volgut ampliar-ho per tractar de testejar altres models d'aprenentatge profund no vistos anteriorment i algorismes de classificació més eficients per tal de comprovar que el model escollit és el més adequat, i en cas negatiu, justificar per què.

L'altre treball escollit consta d'un informe teòric que exposa els resultats d'uns models de classificació (KNN) i xarxes neuronals (CNN) a l'hora de detectar piscines a Uruguai. L'estudi s'anomena "Detección de piscinas utilizando Convolutional Neural Networks (CNN)ï ha sigut portat a cap per l'equip Saturno. [14] En aquest cas han obtingut uns resultats millor en xarxes neuronals, però volem comprovar que aquesta hipòtesi és certa i aplicar més models diferents.

4 Metodologia

En aquest apartat explicarem les eines utilitzades per a la realització del projecte, tal com els models classificadors, les xarxes neuronals escollides per realitzar en el projecte i els paràmetres que calcularem.

4.1 Algoritmes de Classificació

4.1.1 SVM

Aquest algorisme d'aprenentatge s'utilitza per problemes de classificació i regressió, i consisteix a representar els punts estudiats de la mostra l'espai, separats en N classes mitjançant un hiperplà, que és un vector entre dos dels punts, els més propers. [15]

4.1.2 KNN

L'algorisme KNN (k-nearest neighbors) és un mètode de classificació supervisada que estima la probabilitat que té un element de pertànyer a una classe determinada per tal de classificar-lo correctament.

Com el seu nom indica, aquest algorisme es basa en els punts més propers a un element per tal de classificar l'element en si, i classificarà segons la tipologia dels veïns més pròxims.

En aquest projecte tractarem d'estudiar com varia l'encert de piscines segons el paràmetre K, que defineix el nombre de veïns a explorar. [16]

4.2 Xarxes neuronals, CNN

El sistema CNN (Convolutional neural network) és una xarxa neuronal que utilitza algoritmes per classificar elements visuals. Per aconseguir-ho fa servir principis d'àlgebra lineal com multiplicació de matrius per a detectar patrons que es repeteixin en les imatges. El seu mètode de connectivitat entre neurones està inspirat en el còrtex visual dels animals i fa servir operacions de convolució (superposició). [17]

Tot i ser un sistema més antic (la primera CNN data de 1990) nosaltres farem servir DenseNet, la versió més recent de 2016. [18][19]

En el nostre projecte voliem aprofundir en un tipus de xarxa neuronal concreta i vàrem optar en estudiar tres opcions per escollir la millor.

En el nostre cas vàrem estudiar YOLO, Faster R-CNN i SSD (podeu veure la taula comparativa dels tres models a l'apèndix **A.2**).

Davant les similituds entre Faster R-CNN i SSD es va decidir implementar Faster R-CNN i YOLO, tot i que considerant els resultats obtinguts la xarxa en la que es va fer l'estudi final va ser YOLO i les seves respectives versions.

A continuació mostrem una taula resum de l'esmentada a l'apèndix, on es comparen les xarxes YOLO i Faster R-CNN.

TAULA 1: Comparació de YOLO, Faster R-CNN

Model	**YOLO**	**Faster R-CNN**
Arquitectura	Xarxa única	Dues etapes
Precisió	Mitja	Alta
Velocitat	Alta	Mitja
Arquitectura	Darknet	VGG/ResNet
Mida dades	Molt flexible	Mida especifica
Robustesa	Baixa	Alta
Facilitat	Mitja	Alta

4.2.1 YOLO

També conegut com a "You Only Look Once"és un algoritme de detecció d'objecte en temps real que com bé indica el seu nom només realitza una única passada per cada imatge. [20] Pel seu funcionament només fa ús d'una xarxa neuronal amb la qual divideix la imatge per zones per tal de calcular les probabilitats de trobar objectius a cada regió. És un sistema ràpid i força precís i per aquest motiu s'ha decidit fer ús d'ell. [21]

YOLO és un sistema força recent tenint en compte que la primera versió anomenada YOLO data de 2015.

En aquest projecte utilitzarem la versió de YOLOv5 que és l'última versió del sistema de detecció creada el 2018.

Per implementar aquest apartat hem utilitzat l'entorn PyTorch, que ens hem instal·lat prèviament a l'ordinador i hem utilitzat el repositori oficial de YOLOv5 de Github per tal d'entrenar la xarxa neuronal.

A continuació creem un entorn virtual on instal·lem les dependències necessàries per poder executar els codis amb els quals treballarem i treballar de forma correcta amb la llibreria descarregada de GitHub.

Com ja tenim el conjunt d'entrenament, creem un fitxer de configuració per definir les propietats de la xarxa, com el

nombre d'etiquetes d'objectes i la grandària d'imatge d'entrada.

Entrenem la xarxa neuronal amb el codi d'entrenament, però editant el fitxer per tal d'adaptar el model a les imatges d'entrada que li estem proporcionant, perquè així s'adapti correctament al conjunt d'entrenament seleccionat.

Finalment, avaluem la xarxa neuronal fent servir les dades de validació per veure el seu rendiment i guardem els pesos entrenats per utilitzar-los en el següent informe per a la detecció d'imatges noves.

En aquest cas trobem tres diferents versions de YOLO que hem considerat les més importants de cara al treball que estem realitzant, YOLOv3, YOLOv5 i YOLOv8.

A continuació podeu trobar una taula comparativa entre els tres models esmentats.

TAULA 2: COMPARACIÓ DE CARACTERÍSTIQUES ENTRE YOLOV3, YOLOV5 I YOLOV8

Característiques	**YOLOv3**	**YOLOv5**	**YOLOv8**
Arquitectura	Darknet	CSPDarknet	Darknet
Mida de xarxa	Gran	Mitjana	Gran
Precisió	Bona	Bona	Millorada
Velocitat	Moderada	Ràpida	Ràpida
Mida del model	Gran	Petit	Gran
Versió actualitzada	No	Sí	No
Número de classes	80	80	Configurable
Millores escenes denses	No	Sí	Sí

4.3 Mètriques d'avaluació

Per últim, simplement mencionar les mètriques que utilitzarem per fer la comparativa entre els models entrenats i els resultats obtinguts. [22]

En aquestes fòrmules observarem 4 paràmetres que expliquem a continuació:

- **TP:** True Positive, casos de detecció en el model detecta de forma correcta la classe (detecció de piscina correcta).
- **TN:** True Negative, casos de detecció en el model detecta de forma correcta la no pertinença a la classe. (detecció de no piscina correcta).
- **FP:** False Positive, casos de detecció en el model detecta de forma incorrecta la classe. (detecció de piscina no correcta)
- **FN:** False Negative, casos de detecció en el model detecta de forma incorrecta la no pertinença a la classe. (detecció de no piscina no correcta)

4.3.1 Precision

És el paràmetre que s'utilitza per mesurar la qualitat del model. És una mesura molt fiable per saber si un model classifica correctament o no, ja que té en compte els positius verdaders i falsos.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

4.3.2 Recall

Mesura la quantitat d'elements que el model és capaç d'identificar, per tant, té en compte els falsos negatius i positius encertats.

$$recall = \frac{TP}{TP + FN} \tag{2}$$

4.3.3 F1

En aquest cas F1 combina el *precision* i *recall* (mètriques esmentades anteriorment) per tal de facilitar la comparativa tenint en compte els dos paràmetres units en un de sol.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{3}$$

5 DESENVOLUPAMENT

Al llarg d'aquesta secció s'explicarà les parts en que s'ha dividit el projecte, detalls de l'implementació realitzada, i explicació dels entorns utilitzats.

En total el projecte es desenvoluparà al llarg de 100 dies, equivalent a una mica més de 3 mesos i començarà el 6 de febrer per finalitzar el 27 de maig.

Seguint aquesta distribució, es planteja treballar un total de 3 hores diaries, que multiplicades pels 100 dies de treball equivalen a les 300 hores de duració del treball de fi de grau.

Per visualitzar aquesta planificació s'ha decidit dissenyar un diagrama de Gantt. La imatge d'aquest diagrama es pot consultar a l'Apèndix **A.1** del document.

5.1 Ortoimatges

Per començar el nostre projecte, necessitem algunes imatges amb les quals crear el conjunt d'entrenament i de test que usaran els nostres models per detectar les piscines.

En aquest cas, hem decidit fer servir ortoimatges, ja que són imatges que han estat corregides per a la deformació de la perspectiva i altres distorsions per representar amb precisió l'espai terrestre. Això les fa útils per a la detecció d'objectes, perquè proporcionen una vista clara i precisa de les característiques de l'objecte.

Les imatges ortogràfiques, per tant, són una eina útil per a la detecció d'objectes amb xarxes neuronals convolucionals, pel fet que proporcionen una vista precisa i plana dels objectes i són riques en característiques que poden ser útils per als algorismes de detecció d'objectes.

5.1.1 Selecció imatges

En el nostre cas hem decidit agafar les imatges de la línia costera de Catalunya, ja que fent una anàlisi de les imatges és on més piscines hem trobat i això ens resultarà útil per fer el nostre conjunt d'entrenament i test.

El format de les imatges ha sigut JPGE, però d'alta resolució perquè es puguin veure les piscines amb una qualitat alta que ajudi als models per a la seva adequada detecció.

Les imatges les hem obtingut de la pàgina web oficial de l' (Institut Cartogràfic i Geològic de Catalunya.

La selecció s'ha fet a mà, seleccionant les zones costaneres interessant pel projecte per així reduir el possible soroll que podem trobar a les imatges (com zones de bosc o mar).

En total hem descarregat 12 imatges de 15K x 13k píxels aproximadament, imatges molt grans, però que pesaven força menys que les imatges en TIFF i de qualitat semblant al mida que s'han descarregat.

5.1.2 Tractament de les imatges

Pel tractament d'imatges, el primer que hem realitzat és la divisió de les imatges en subimatges de 448 x 448 píxels amb Python, ja que és una mida recomanada per entrenar models de tipus YOLO i són les mides amb les quals s'entrenaven els models originals. La mida de les imatges perquè YOLO pugui entrenar amb major facilitat i precisió ha de ser múltiple de 32. [14]

Això es deu al fet que YOLO fa ús d'una arquitectura de xarxa neuronal que processa la imatge a través de múltiples capes de convolució i pooling. Això redueix la mida de la imatge en cada capa. Usar imatges de mida 448x448 assegura que la imatge es redueixi a una mida prou petita per processar-la eficientment, mentre es manté un nivell raonable de resolució per a la detecció de piscines.

Un cop finalitzat aquest procés, amb un nou programa Python hem observat totes les imatges i amb un mètode de detecció de color per píxels hem eliminat les imatges que tenien molt soroll i no eren interessant de cara al nostre grup d'entrenament. Aquestes imatges eren sobretot les que detectàvem colors verd, blau o marró a la major part de la imatge (fent servir el codi hexadecimal del color objectiu visualitzat a les imatges) per eliminar imatges amb moltes zones de bosc, mar o cultius, que eren poc interessants.

En total hem utilitzat un conjunt d'entrenament i validació de 1000 imatges (800 per test y 200 per validació). A més s'ha reservat unes 500 imatges per realitzar la part de test alhora d'extreure els resultats.

5.1.3 Etiquetatge de les imatges

L'etiquetatge d'imatges és el procés de marcar regions específiques d'una imatge amb etiquetes o categories que representen objectes, característiques o atributs.

Aquest procés es realitza amb les imatges destinades a entrenament perquè el model pugui visualitzar i aprendre dels objectes que nosaltres li especifiquem i el tipus que són. En el nostre cas aquest procés s'ha fet de forma manual i hem hagut de marcar a totes les imatges les piscines delimitant el seu contorn amb un programa anomenat labelImg.

LabelImg és un programa de codi lliure gratuït que facilita a l'usuari una interfície on poder treballar amb imatges i el seu etiquetatge. L'usuari només ha de delimitar els objectes de forma manual i indicar de quin tipus són i el programa de forma automàtica genera les etiquetes en el format adequat, o bé format YOLO o bé PascalVOC (ambdues útil en aquest cas per les xarxes que volem entrenar).

Fig. 1: Imatge etiquetada amb LamelImg

Cal destacar que hem dividit les imatges perquè no hi hagi superposició amb elles, per tant, entre imatges poden quedar piscines divididas. Per compensar-ho hem tractat d'etiquetar no només les piscines senceres sinó també les piscines que es veien de forma parcial per tal d'aconseguir que els models aconsegueixin detectar que allà hi ha piscina tot i no ser-hi al cent per cent visible.

El conjunt d'entrenament final seleccionat consta de 1001 imatges de 448x448 píxels cadascuna i etiquetades amb les piscines detectades en cada una d'elles en format YOLO i PascalVOC, segons la xarxa entrenada.

Amb LabelImg hem generat les etiquetes en format YOLO i després per mitjà d'un codi a Python hem transformat les etiquetes a format PascalVOC.

En total s'han classificat 3717 piscines (etiquetes), divides en 3029 de train i 688 de test.

5.2 Entrenament amb xarxa YOLO

Un cop tenim les imatges classificades dividim el nostre dataset etiquetat en conjunt d'entrenament i conjunt de validació. En el nostre cas hem decidit que un 80% de les imatges siguin de train (800 imatges) i un 20% siguin per validació (200 imatges).

L'entorn de treball que s'ha utilitzat per implementar els models de detecció ha estat Google Colab, ja que és un entorn de programació i execució que accepta Python i ofereix a l'usuari GPU per poder entrenar els models, cosa que facilita molt la seva execució a nivell de hardware ja que minimitza les limitacions que l'usuari pot tenir de memòria o processament.

Alhora d'entrenar els models s'ha optat per utilitzar els mateixos paràmetres ja que així la comparació seria equitetiva a la realitat i no hi hauria variables que puguin alterar els resultats. Els paràmetres a destacar de cara a l'entrenament del model han sigut els següents:

- Epochs: Una època fa referencia a una passada completa a totes les dades d'entrenament, per tant a cada època es reajusten els pesos del model per tal de utilitzar els nous a la següent iteració. En el nostre cas hem decidit utilitzar 70 èpoques ja que és un valor suficientment alt perquè el model classifiqui correctament però alhora no es generi overfitting (el model s'acostuma a les dades).

- Batch size: Un lot (batch) defineix una porció de dades d'entrenament que s'utilitza per cada pas de l'entrenament, és a dir, dividim totes les dades en lots més petits. En el nostre cas escollim un valor de 16, un valor mig perquè el model classifiqui ràpidament i sense fer molt ús de memòria, però alhora no perdi informació rellevant per la detecció.
- Image size: Aquest paràmetre indica al model de quina mida són les dades que ha d'estudiar, en el nostre cas 448x448 píxels.

Per últim, caldrà editar un fitxer de tipus YAML perquè el nostre model sàpiga les classes que ha de detectar i les rutes a les imatges d'entrenament i validació corresponents. En el nostre cas només busquem classificar piscines, per tant serà la nostra classe única.

5.3 Classificadors tradicionals

5.3.1 KNN i SVM

Hem utilitzat dos algorismes de classificació diferents sobre les imatges per tal d'analitzar la capacitat classificatòria d'aquests mètodes en vers als més actuals.

A partir de les imatges s'ha fet una classificació a nivell de píxel utilitzant KNN i SVM per diferenciar entre dues classes de píxels: píxel que forma part d'una piscina i píxel que forma part del fons.

Amb aquest sistema aconseguim obtenir fitxers binàris amb 1s a la zona de piscines i 0s a tota la resta de l'imatge.

Fig. 2: Màscara generada d'imatge de test

5.3.2 Visió per computador

Un cop tenim ambdúes imatges s'ha seguit un procés de processat d'imatges amb tècniques de visió per computador per tal d'aconseguir minimitzar el soroll que hi pot haver en l'informació obtinguda. A continuació s'explica els passos que hem seguit:

- Etiquetatge zones connexes: Primerament hem utilitzat un algorisme d'etiquetatge de zones connexes per diferenciar de forma única cada zona connexa en la imatge de píxels amb valor de 1, que serien els espais de la imatge on trobarem piscina.
- Filtratge per mida: A continuació hem aplicat un filtre per descartar imatges excessivament petites o grans per tal de descartar zones que ens puguin haver generat soroll (zones semblants a piscines que poden confondre segons el color). Aquest pas ens ajudarà a descartar possibles falsos positius.
- Erode/Dilate: mètodes utilitzats per millorar la continuitat i forma de les regions de píxels detectades. Amb aquests mètodes eliminarem soroll i acabarem de tancar les regions de zones que no hagin sigut detectades de forma clara.
- Trobar centre: Finalment, trobarem el centre de la piscina per tal de saber en comparació amb les etiquetes realitzades en format YOLO (que contenen la zona on trobem la piscina) si hem detectat de forma correcta la piscina o no.

6 RESULTATS

En aquesta secció es mostraràn els resultats obtinguts en el nostre projecte a través de gràfiques i mètriques d'evaluació generats pels nostres models entrenats. A més es generaràn taules comparatives per visualitzar quin és el model i mètode que millor detecta piscines.

Cal destacar que els resultats que vàrem obtenir de Faster R-CNN eren pitjors que els de la primera versió de YOLO i per tant es va decidir implementar directament YOLO i les seves versions per així aconseguir trobar el millor model pel nostre cas.

Faster R-CNN tenia una precisió inferior a YOLO a més de ser més lenta la seva execució i entrenament. Això sumat al fet de tenir major dificultat d'implementació a partir de la documentació oficial van fer que el projecte es fés al voltant de YOLO.

6.1 Versions de YOLO

S'ha entrenat tres versions diferents de YOLO (Yolov3, Yolov5 i Yolov8). A continuació s'exposen el resultats obtinguts i un anàlisi dels mateixos per tal de decidir quina versió del model és la més adequada per al nostre problema de detecció de piscines.

6.1.1 Mètriques d'avaluació

Primer de tot hem obtingut una sèrie de mètriques amb el mòdul sklearn com les matrius de confusió o les gràfiques precision-recall curve per veure com de bé han detectat els nostres models les piscines.

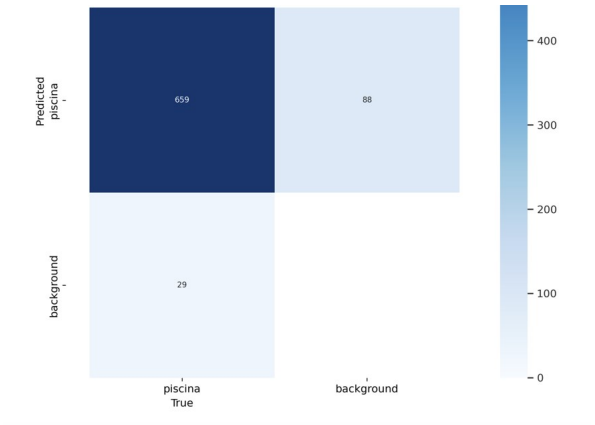

Fig. 3: Matriu de confusió Yolov5

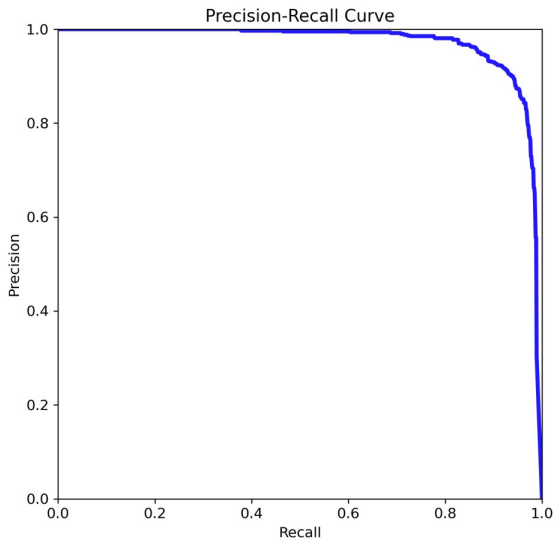

Fig. 4: Precision-recall curve

A la figura 4 podem observar la gràfica precision-recall que ens indica com varien aquests dos paràmetres al llarg de l'execució del model. En aquest cas l'àrea per sota de la recta ens detalla un recall i precisió alts (equivalent a pocs falsos positius i negatius). Per tant la gràfica ens està indicant que el nostre model està detectant de forma precisa i detecta la majoria de resultats positius.

A partir de la matriu de confusió (figura 3) i els paràmetres que conté, podem fer un anàlisi extens de les mètriques que volem estudiar.

TAULA 3: COMPARATIVA MÈTRIQUES YOLO

Métricas	YOLOv3	YOLOv5	YOLOv8
TP	659	691	694
FP	88	51	48
FN	29	24	23
Precisió	0.88	0.93	0.94
Recall	0.96	0.97	0.97
F1	0.92	0.95	0.95

En aquest cas si ens fixem en els resultats, podem observar que a nivell d'encert tots tres models obtenen resultats molt semblants i precisos. Tot i que Yolov8 millora Yolov5, i Yolov5 millora Yolov3, la diferència dels resultats ha estat mínima ja que amb la versió Yolov5 ja hem obtingut una taxa d'encerts sobre la classe positiva del 93%, percentatge molt difícil de millorar.

Els resultats que observem a la taula mostren una igualtat molt gran si comparem les mètriques d'avaluació. Tots tres models classifiquen de forma quasi perfecta i podem afirmar que tots tres són molt vàlids alhora de realitzar una detecció de piscines amb imatges ortogràfiques.

Cal destacar també que l'obtenció d'uns resultats tan bons es deu a la qualitat del dataset, ja que s'ha etiquetat de forma manual i s'han utilitzat moltes imatges diferents perquè el model obtingui totes les característiques possibles de cara a la detecció.

A continuació es mostra una imatge d'exemple per observar com ha classificat el model Yolov5 una imatge de test (els outputs generats com imatges són iguals a les tres versions).

Fig. 5: Imatge amb la detecció generada per YOLOv5

També, com s'ha explicat prèviament a la part d'etiquetatge, s'ha tingut en compte les piscines que no s'observen de forma completa en una fotografia, és a dir, que surten parcialment. Aquestes piscines també han sigut etiquetades com a piscina perquè el model les aconsegueixi detectar correctament, i observant els resultats generats podem concluir que el model detecta aquests casos particular de forma correcta.

Fig. 6: Imatge on model detecta piscina vista parcialment

6.1.2 Velocitat d'execució

Una part molt important per la comparativa dels models, sobretot per part del programador, és el temps d'execució dels models, el temps que triguen en entrenar i aprendre a partir de les imatges. Aquest procés pot ser molt costós a nivell de hardware i per tant, com més ràpid aconseguim finalitzar el procés d'entrenament, més òptim serà el model en aquest aspecte.

Els tres models han trigat relativament poc en entrenar, però si que hi hem trobat diferències significatives analitzant els resultats.

Cal puntualitzar que aquests resultats han sigut obtinguts en igualtat de condicions per a poder realitzar la compara-

ció, és a dir, amb paràmetres d'entrenament equivalents a cada versió per tal de no obtenir resultats alterats.

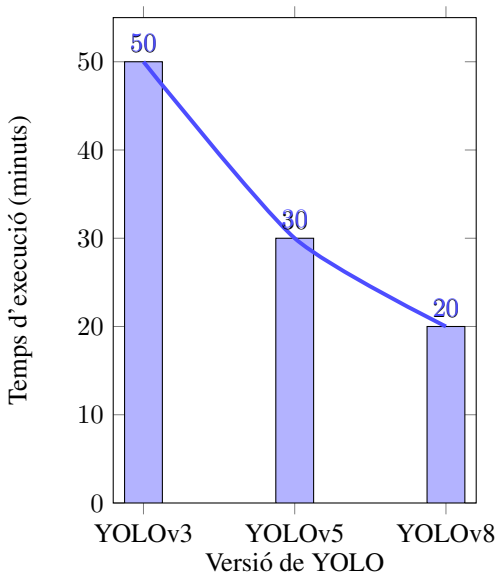

Fig. 7: Comparativa de temps d'execució en diferents versions de YOLO

En el nostre cas veiem una diferència notòria de la versió més antiga utilitada (Yolov3) respecte la nova versió (Yolov8), ja que aquesta última finalitza l'entrenament més del doble de ràpid que la primera. En canvi Yolov5, és una versió intermitja, més propera a la versió 8 que a la 3.

Per tant, la versió que millor ens ha funcionat en aquest cas és Yolov8, tot hi que la diferència amb Yolov5 no és tant gran com per ser un aspecte decisiu per descartar Yolov5, en canvi, Yolov3 si que és significativament més lenta i si que afecta alhora d'entrenar el model.

6.1.3 Documentació Oficial

Un dels punts més importants de cara a l'implementació d'un model és la documentació oficial que pots trobar del model i que l'empresa creadora del mateix ofereix.

En aquest cas, depenent de la versió de Yolo s'han tingut problemes per trobar informació sobre els models o sobre com implementar certes millores alhora d'ajustar els paràmetres.

Per aquest motiu hem trobat important també per decidir el millor model, considerar la dificultat d'obtenir informació oficial sobre el model.

- Yolov3 consta de documentació oficial completa i amb gran detall sobre la xarxa i la seva implementació. Per altra banda, no consta d'un repositori on expliqui la implementació amb Yolov3 ja que al seu repositori ja utilitzen Yolov5, per tant no es pot trobar fàcilment el su repositori i s'ha d'utilitzar altres fonts per la seva execució.
- Yolov5 consta d'informació molt detallada sobre el model implementat i al seu repositori oficial explica de forma detallada com implementar el seu model per detectar objectes. En aquest aspecte considerem que Yolov5 ha sigut la versió en que més facilitats hem trobat per part de pàgines i repositoris oficials de cara a la implementació fent el treball menys costós a nivell de temps.
- Yolov8 conté un repositori oficial en el que és molt intuïtiu treballar, i és el model més fàcil d'entrenar pel programador seguint l'experiència obtinguda en aquest procés. Tot i això, com és un model molt novedós, no hi ha informació detallada sobre el model i les millores implementades respecte v5, per tant no és un model molt conegut encara.

6.1.4 Comparativa de versions Yolo

Per realitzar la comparativa dels models i decidir quin és el més adequat pel nostre problema, farem servir els resultats obtinguts en l'apartat anterior per realitzar una taula de decisió. Utilitzarem la taula de decisió ponderada ja que considerem que hi han entrades de la taula a considerar més importants que d'altres i, per tant, que s'han de valorar més.

Els pesos escollits a la taula els hem escollit de forma manual segons els criteris que hem considerat més importants per al nostre anàlisi i es detallen a continuació:

- **Velocitat d'entrenament:** Es considera el temps que triga el model a entrenar i extreure resultats a partir del dataset utilitzat. En aquest cas no ho considerem un factor clau ja que un projecte d'aquest tipus no requereix immediatesa.
- **Precisió del model:** El factor més important amb diferència dels estudiats ja que representa el percentatge d'encert alhora de detectar les piscines. En el cas del treball estudiat es prioritza la maximització de piscines detectades correctament, per tant, el ponderem amb un valor alt.
- **Suport oficial:** Referit a la documentació oficial que ofereix Ultralytics (empresa que ofereix el model YOLO) per tal de facilitar informació sobre cada versió en concret. Es considera important però de cara al problema general no afecta en gran mesura.
- **Facilitat d'implementació:** Nivell de dificultat alhora de l'implementació del model, és a dir, la dificultat alhora de generar els fitxers necessaris, trobar els pesos adequats, llibreries oficials, etc.
- **Accés a recursos de suport:** Entenem accés a recursos de suport a tota la informació que pots trobar a partir de recerca propia sobre treballs d'altres persones. Contemplem repositoris de github, tutorials, webs de consultes, etc. Aquest apartat el considerem important ja que facilita molt l'implementació dels models i possibles millores.
- **Hardware necessari:** Hardware que el nostre entorn necessita per poder executar els codis. Tampoc el considerem tant important ja que el projecte ha sigut executat amb Google Colab, entorn que ofereix GPU extra a la del computador personal.

TAULA 4: COMPARACIÓ DE MODELS YOLO

Criteri	Pes	Yolov3	Yolov5	Yolov8
Velocitat entrenament	0.10	3	6	8
Precisió model	0.5	8	8.5	9
Suport oficial	0.10	5	9	6
Facilitat d'implementació	0.10	3	9	7
Accés a recursos de recolzament	0.15	7	10	5
Hardware necessari	0.05	4	8	6
Puntuació total		6.35	8.55	7.65

6.2 Algorismes tradicionals

Com s'ha explicat a la metodologia s'han aplicat models tradicionals de detecció per mitjà de processat d'imatges per tal de detectar piscines a nivell de píxel. Els resultats no han estat tan bons com els models de Machine Learning que hem vist anteriorment. A continuació mostrem els resultats obtinguts que constaten que en el nostre cas la opció tradicional queda lluny de ser la més òptima a nivell de detecció. Per realitzar-ho hem escollit el model YOLOv5, que és el que millor s'adapta al problema i els dos algorismes classificadors.

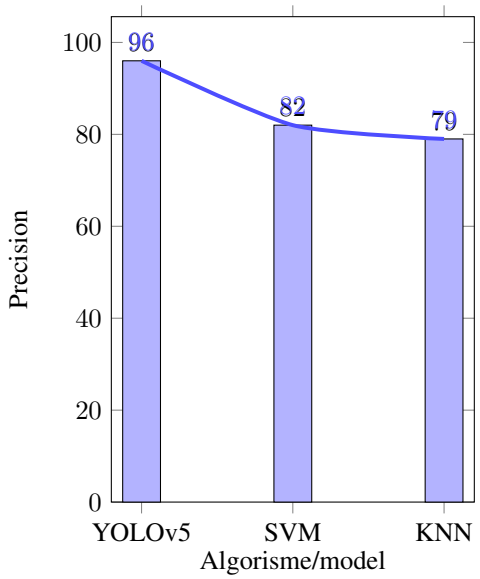

Fig. 8: Comparativa entre models tradicionals i YOLOv5.

A més de la precisió (valor que ja ens fa descartar els mètodes tradicionals) també cal destacar que realitzar un model tradicional ha resultat molt més tediós que un model d'entrenament modern.

A més d'aplicar l'algorisme s'ha de realitzar un treball de processament d'imatges costós en quant a temps i s'ha de tenir en compte molts aspectes que de l'altre forma queden ocults davant la xarxa neuronal (com la gestió del soroll o el filtratge de dades).

7 CONCLUSIONS

Després de fer un anàlisi dels resultats, s'ha extret la informació suficient per extreure conclusions rellevants sobre el funcionament dels nostres models i la comparativa dels mateixos.

Com podem observar a la taula 4, el model YOLOv5 és el que més ens convé implementar per realitzar un treball d'aquestes característiques.

A priori la lògica en diu que la versió YOLOv8 es la més millorada i recent i hauria de ser la que millor funciona. En el nostre cas no ha sigut així ja que els resultats de tots tres models han sigut molt semblants a nivell d eprecisió, però tots els altres aspectes esmentats anteriorment fan que YOLOv5 sigui la millor opció. YOLOv3, és una versió anterior i ja no s'utilitza per realitzar entrenaments ja que es pitjor que la següent versió.

En canvi, YOLOv8, és una versió millorada respecte YOLOv5, més precisa i ràpida, però al ser una versió tan recent no té pràcticament documentació ni suport tècnic.

A més, practicament no hi han treballs realitzats amb aquesta versió, i en canvi, amb la versió v5 hi han molta documentació i projectes complets sobre com millorar les deteccions.

Per tant, la falta de documentació i d'ús per part dels usuaris penalitza molt el ús de YOLOv3 i YOLOv8 i, per tant, fa que la seva implementació i millora sigui més difícil que la de la versió 5.

Per tots aquests motius podem concloure afirmant que tot i que YOLOv8 pot ser una opció potencialment millor que la seva competència, a falta de més documentació afirmem que YOLOv5 és la millor versió de YOLO per tal d'entrenar i detectar piscines amb el nostre dataset.

A més a més, al treballar amb algorismes tradicionals com KNN o SVM hem pogut observar la millora d'aplicar algorismes de Machine Learning respecte aquests anteriors. En el nostre cas els resultats reflecteixen que el model d'IA s'adapta molt millor a les nostres dades i és molt més adequat per realitzar un treball de detecció d'objectes.

8 MILLORES A FUTUR

A continuació exposem unes propostes de millores a futur de cara a seguir ampliant el projecte prenent els resultats obtinguts com a punt de partida i motivació.

- Ampliació del dataset: el primer punt que hem considerat és ampliar el dataset de forma considerable per tal de realitzar un anàlisi més extens i permetre als models més marge d'entrenament per veure com millora o empitjora la classificació.
- Reentrenament dels models: una bona aportació de millora seria reentrenar els models amb les imatges que el model ha classificat de forma errònia o no ha sàpigut identificar correctament la classe piscina.
- Entrenar nous models: en aquest treball ens hem centrat en analitzat com YOLO s'adapta al nostre projecte, però hi han molts més models de xarxa que podrien adaptar-se correctament. Es podria fer un estudi profund de les diverses versions d'altres models com podrien ser Faster-RCNN o SSD entre d'altres.

REFERÈNCIES

[1] L. Méndez Martínez, "Detección automática de Piscinas y Balsas mediante técnicas de teledetección orientada a píxeles," UPV, 16-Sep-2014. [Online]. Available: https://riunet.upv.es/handle/10251/50682.

[2] Ïnstituto Geográfico Nacional". Geoportal oficial del Instituto Geográfico Nacional de España. https://www.ign.es/web/sc-cnig-fototeca

[3] C. Galindo, P. Moreno, J. Gonzalez and V. Arévalo, "Swimming pools localization in colour high-resolution satellite images,"2009 IEEE International Geoscience and Remote Sensing Symposium, Cape Town, South Africa, 2009, pp. IV-510-IV-513, doi: 10.1109/IGARSS.2009.5417425.

[4] Jonas1312, "Swimming Pool Detection,"Repositorio de GitHub. [En línea]. Disponible: https://github.com/Jonas1312/swimming-pool-detection

[5] Desconocido, LLBP Cascade Classification Swimming Pool Detection,"OpenCV QA Forum. [En línea]. Disponible: https://answers.opencv.org/question/41424/lbp-cascade-classification-swimming-pool-detection/

[6] Zhang, B., Wu, X., You, J., Li, Q., Karray, F. (2010). Detection of microaneurysms using multi-scale coefficients. Pattern Recognition, 43(6), 2237-2248. https://doi.org/10.1016/j.patcog.2009.12.017

[7] W. V. Nicholson, "Object detection by correlation coefficients using azimuthally averaged reference projections,ïn IEEE Transactions on Biomedical Engineering, vol. 51, no. 11, pp. 2006-2012, Nov. 2004, doi: 10.1109/TBME.2004.834271. Disponible en:https://ieeexplore.ieee.org/abstract/document/1344203

[8] W. Jiang, H. Zhou, Y. Shen, B. Liu y Z. Fu, Ïmage segmentation with pulse-coupled neural network and Canny operators,Çomputers & Electrical Engineering, vol. 46, pp. 528-538, 2015.
Disponible en:
https://www.sciencedirect.com/science/article/abs/pii/S0045790615001214

[9] P. Maya and C. Tharini, "Performance Analysis of Lane Detection Algorithm using Partial Hough Transform,"2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 2020, pp. 1-4, doi: 10.1109/ACIT50332.2020.9300083.

[10] M. Kim, J. B. Holt, R. J. Eisen, K. Padgett, W. K. Reisen y J. B. Croft, "Detection of Swimming Pools by Geographic Object-based Image Analysis to Support West Nile Virus Control Efforts", Photogrammetric Eng. Remote Sens., vol. 77, n.º 11, p. 1169–1179, noviembre de 2011. Accedido el 1 de julio de 2023. [En línea]. Disponible: https://doi.org/10.14358/pers.77.11.1169

[11] B. Rodríguez-Cuenca and M. Alonso, "Semi-Automatic Detection of Swimming Pools from Aerial High-Resolution Images and LIDAR Data," Remote Sensing, vol. 6, no. 4, pp. 2628–2646, Mar. 2014, doi: 10.3390/rs6042628.

[12] C. Coelho, M. Fernanda P. Costa, L.L. Ferrás, A.J. Soares, Development of a Machine Learning Model and a User Interface to Detect Illegal Swimming Pools, SYMCOMP 2021, 5th International Conference on Numerical and Symbolic Computation: Developments and Applications, Évora, Portugal, 25-26 March 2021

[13] B.S.Ballester. *Supervised detection of swimming pools from orthoimages in Mallorca.* Barcelona: Universitat Autònoma de Barcelona, Espanya, 2021.

[14] A.Barón, M.V.Landaberry, J.M.Sancho, L.Vidal, M.Vidal. *Detección de piscinas utilizando Convolutional Neural Networks (CNN).* Uruguay, 2021.

[15] J. L. Martínez, "Support vector machines,Üniversidad de Zaragoza, 2001. [En línea]. Disponible en: http://www.alumnes.udl.cat/EPS/enginyeria-informatica/tfg/0506/docs/memoria/mem1039285.pdf

[16] S. Raschka, "k-Nearest Neighbors,Üniversity of Wisconsin-Madison, Madison, Wisconsin, 2018. [En línea]. Disponible en: https://sebastianraschka.com/Articles/2018knnintro.html

[17] "Red neuronal convolucional," Wikipedia, 20-Jan-2023. [Online]. Available: https://es.wikipedia.org/wiki/Red_neuronal_convolucional

[18] "Papers with code - densenet explained," Explained — Papers With Code. [Online]. Available: https://paperswithcode.com/method/densenet

[19] S.-H. Tsang, "Review: DenseNet - dense convolutional network (image classification)," Medium, 20-Mar-2019. [Online]. Available: https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803

[20] M. Chablani, "Yolo - you only look once, real time object detection explained," Medium, 31-Aug-2017. [Online]. Available: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006

[21] J. Bernal, "YOLO: You Only Look Once - Una aproximación algoritmo de detección de objetos en tiempo real,Üniversidad de los Andes, Bogotá, Colombia, 2018.

[22] J. M. Heras, Lita, Gabriel, Jose, Manuel, Ruben, and Juan, "Precision, recall, F1, accuracy en clasificación," IArtificial.net, 09-Oct-2020. [Online]. Available: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

APÈNDIX

A.1 Planificació projecte

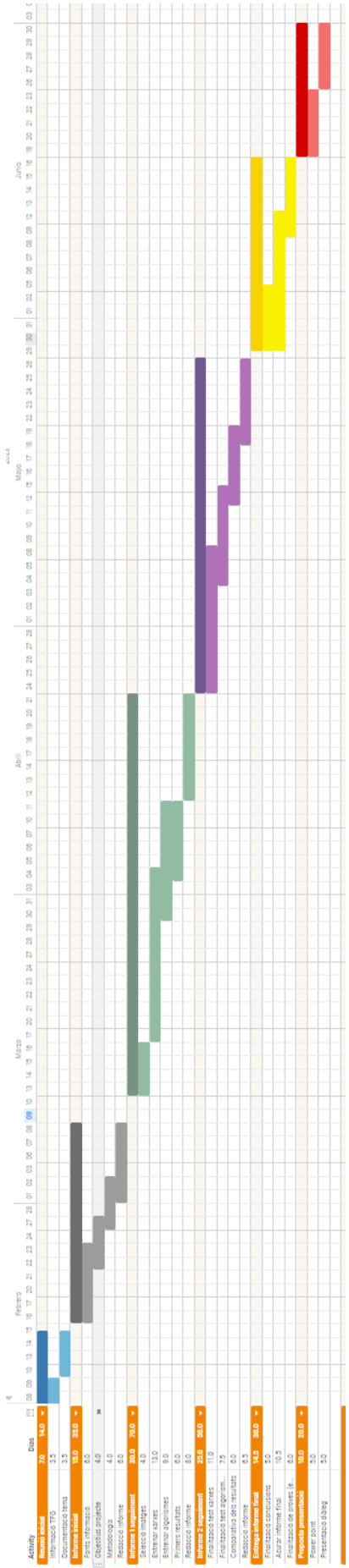

Fig. 9: Diagrama de Gantt del projecte

A.2 Taula comparativa entre models CNN

TAULA 5: COMPARACIÓ DE YOLO, FASTER R-CNN I SSD

Model	**YOLO**	**Faster R-CNN**	**SSD**
Arquitectura	Detecció per xarxa única	Detecció per dues etapes	Detecció per xarxa única
Precisió de detecció	Mitja	Alta	Alta
Velocitat de detecció	Alta	Mitja	Mitja
Arquitectura base	Darknet	VGG/ResNet	VGG/ResNet
Mida dades entrada	Molt flexible	Mida especifica	Mida especifica
Robustesa d'escala	Baixa	Alta	Mitja
Facilitat d'entrenament	Mitja	Alta	Alta