Vpliv velikosti nabora podatkov na zaznavanje objekta z modelom YOLO na izbrani podatkovni zbirki

Jaka Bezovšek, Urban Burnik

Univerza v Ljubljani, Fakulteta za elektrotehniko
Tržaška cesta 25, 1000 Ljubljana

E-pošta: jb6837@student.uni-lj.si

Influence of dataset size on object detection with YOLO model

Abstract. *This paper presents basic concepts of object detection technique based on several different images as dataset. The research shows how the number of training images can affect object detection. To start things off, we present some theoretical concepts of object detection. Second, we detail the procedures of YOLO model training and its performance evaluation based on different images. Finally, we discuss and compare the results of trained YOLO models for both output after training, and actual results.*

1 Uvod

Zaznavanje objektov je tehnika, katere cilj je locirati določene predmete na slikah ali videoposnetkih. Gre za široko področje, ki je v praksi zelo uporabno. Tehnika zaznavanja objektov je dandanes vključena v velik del aplikacij in je eden izmed ključnih elementov pri razvoju varnostnih kamer, robotov in avtonomnih vozil.

Pristopi zaznavanja objektov so različni. Danes pogosto uporabljamo algoritme, ki temeljijo na strojnem ali globokem učenju, pri katerih je ključni element učenje modela [1]. Poleg omenjenih, obstajajo tudi druge tehnike zaznavanja objektov, na primer s segmentacijo slike, ki temeljijo na preprostih lastnostih predmetov, kot so velikost, oblika ali barva, ter z zaznavanjem na podlagi značilk, ki uporabljajo značilne parametre predmetov na sliki, kot so robovi, koti ipd. V tem članku se osredotočamo na zaznavanje objekta na podlagi nabora podatkov, s katerim naučimo model.

Ko govorimo o modelu, imamo v mislih algoritem, ki je naučen, da na slikah in videoposnetkih zaznava objekte. Model za zaznani objekt tipično vrne položaj, kje na sliki ali videu se objekt nahaja, kateremu razredu oz. skupini podatkov, ki jih določimo pri učenju modela, pripada ta objekt in s kakšno gotovostjo je zaznani razred pravilen [2].

Opravljenih je bilo več raziskav o slikah, potrebnih za učenje modelov, vendar se mnenja o velikosti nabora teh slik razlikujejo, od takih, ki trdijo, da je za učinkovito delovanje detekcije objektov potrebno nekaj 100 slik na razred [3], do predlogov o uporabi vsaj 1000 slik na razred [4].

Cilj te raziskave je naučiti nekaj modelov z različno velikim naborom podatkov oz. različnim številom slik, ki jih uporabimo pri učenju modela, ter preveriti, kako to vpliva na zaznavanje objekta v praksi.

2 Orodja in postopki

Sistem je sestavljen iz dveh faz in sicer iz učenja modelov in testiranja naučenih modelov. Model, ki smo ga uporabili je YOLOv8, ki je ena izmed zadnjih verzij iz skupine modelov YOLO in poleg učenja na lastnem naboru podatkov omogoča uporabo že naučenega modela. Objekti, ki smo jih tekom te raziskave učili in zaznavali so športne žoge, pri čemer jih nismo ločili glede na šport, temveč smo vse priredili istemu razredu.

2.1 Pridobivanje nabora podatkov

Najprej je bilo potrebno izbrati ustrezen nabor podatkov. Priprava nabora za učenje modela deluje tako, da na vsaki sliki iz nabora ustrezno označimo objekt, ki ga želimo identificirati. Ta postopek lahko opravimo s katerim izmed temu namenjenih orodij, s katerimi objekte na sliki označimo in oznake prenesemo v ustreznem formatu, ki običajno vsebujejo pozicijo, višino in širino ter razred objekta na sliki. Ti podatki se v nahajajo v besedilnih (txt) datotekah, katerih imena se morajo ujemati z imeni slikovnih datotek, da algoritem za učenje lahko ustrezno združi slike z oznakami.

Ker bi bilo ročno pridobivanje in označevanje fotografij časovno preveč potratno, smo se odločili za uporabo vnaprej označenega nabora podatkov. Slike in oznake smo prenesli s spletne strani Roboflow, ki poleg številnih različnih naborov omogoča tudi testiranje modelov, naučenih s temi nabori, v realnem času. Osnovni podatki vsebujejo 1377 slik za učenje, ter za vsako sliko pripadajočo besedilno datoteko. V nadaljevanju bomo število slik zmanjševali, ter spremljali rezultate. V osnovi smo model učili z vsemi slikami, ki so bile na voljo, v nadaljnjih korakih pa smo dvakrat odstranili tretjino ostalih slik in nato dvakrat polovico ostalih slik.

2.2 Učenje modela

V tej raziskavi smo naučili 5 modelov, vsakega z različnim številom slik in oznak. Začeli smo s celotnim naborom slik, nabor pa smo nato z vsakim modelom nekoliko zmanjšali tako, da smo nekaj slik enostavno izbrisali. Modele smo učili s 1377, 918, 459, 229 in 115 slikami, na katerih so bile označene športne žoge, v vsak

nabor pa so bile vključene tudi slike, na katerih športnih žog ni bilo.

Skripta za učenje YOLO modela je skladna z dokumentacijo in napisana v jeziku Python. Gre za enostaven postopek, pri katerem je ključna datoteka *yaml*, v kateri so navedene poti do direktorijev, v katerih se nahajajo slike in oznake.

Učenje YOLO modela omogoča tudi upravljanje s številom ponovitev učenja na celotnem naboru. To storimo prek parametra »epochs« v skripti Python. Načelno več ponovitev vodi do boljših rezultatov, vendar pri prevelikem številu ponovitev lahko pride do prevelikega prilagajanja (overfitting). Priporočena je uporaba okoli 100 ponovitev, v našem primeru pa smo zaradi časovnih omejitev uporabili 30 ponovitev, pri katerih smo dosegali zadovoljive rezultate. Vsebina skripte za učenje YOLO modela je vidna na sliki 1.

```python
def trainer():
    from ultralytics import YOLO

    # Load a model
    model = YOLO("yolov8n.yaml")  # build a new model from scratch

    # Use the model
    model.train(data="data.yaml", epochs=30)  # train the model

if __name__ == "__main__":
    trainer()
```

Slika 1: Skripta Python za učenje YOLO modela

2.2.1 Rezultati učenja

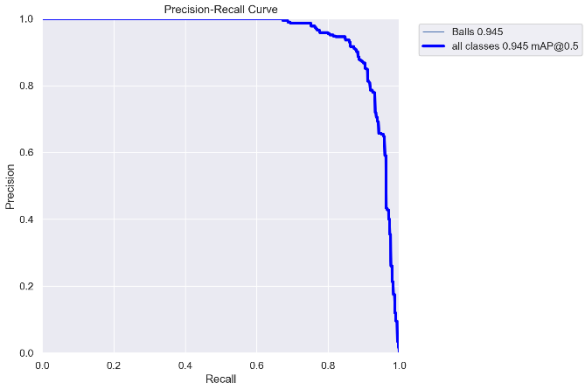
Po zaključenem učenju rezultate prikažemo v obliki slik in grafov, da uporabnik lahko oceni uspešnost naučenega modela. Dve metriki, ki se pogosto uporabljata pri interpretaciji podobnih rezultatov, sta natančnost in priklic. Natančnost meri delež pozitivnih identifikacij med zbirko, medtem ko priklic predstavlja združitev ustreznih in pridobljenih elementov nad skupnim številom ustreznih rezultatov. Natančnost in priklic se lahko izračunata, kot je prikazano v enačbah (1) in (2) v katerih spremenljivke predstavljajo naslednje:

- Razred je prepoznan in je dejanski razred (True positive - TP)
- Razred je prepoznan in ni dejanski razred (False positive - FP)
- Razred ni prepoznan in je dejanski razred (False negative - FN)
- Razred ni prepoznan in ni dejanski razred (True negative - TN)

$$natančnost = \frac{TP}{TP + FP} \quad (1)$$

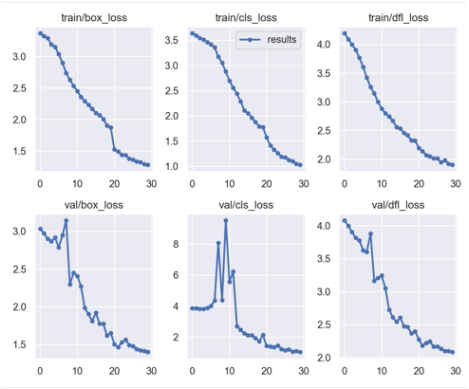
$$priklic = \frac{TP}{TP + FN} \quad (2)$$

Za lažjo in hitrejšo interpretacijo, sta natančnost in priklic prikazana s krivuljami v grafični obliki. Omembe vredna je krivulja natančnost v odvisnosti od priklica (ang. »PR curve«), pri čemer večja površina pod krivuljo pomeni boljše rezultate, krivulja pa naj bi se čim bolj približevala zgornjemu desnemu kotu grafa. Primer krivulje je prikazan na sliki 2.



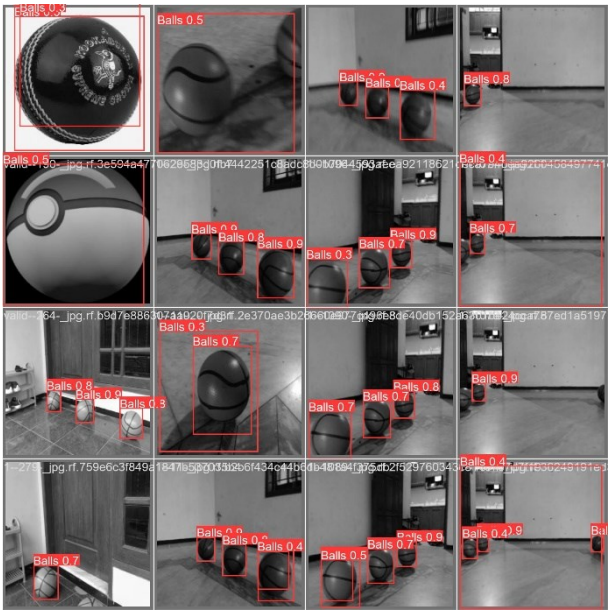
Slika 2: Krivulja natančnost v odvisnosti od priklica

Eden izmed uporabnejših rezultatov pri učenju modela je sklop grafov, ki lahko sproti podajajo koristne informacije. Krivulje na grafih kažejo količino napak med samim procesom učenja. V splošnem velja, da se morajo napake manjšati, vendar vodoravna krivulja ne pomeni nujno slabega rezultata, saj to lahko pomeni, da je model optimalno naučen. Naraščajoča krivulja nedvomno pomeni slab rezultat, v tem primeru pa takoj vidimo, da učenje modela ni bilo uspešno, še preden poženemo detekcijo z naučenim modelom. Grafe napak prikazuje slika 3.



Slika 3: Grafi izgub po zaključenem učenju modela

V zaključku je podanih še nekaj fotografij, ki prikazujejo zaznavo na testnih primerih. Na ta način lahko vidimo, kako zaznava s tem modelom deluje v praksi, ter ocenimo, ali prihaja do napak pri zaznavi. Primer takega rezultata je prikazan na sliki 8.



Slika 4: Prikaz rezultatov detekcije na praktičnih primerih

2.3 Testiranje naučenih modelov

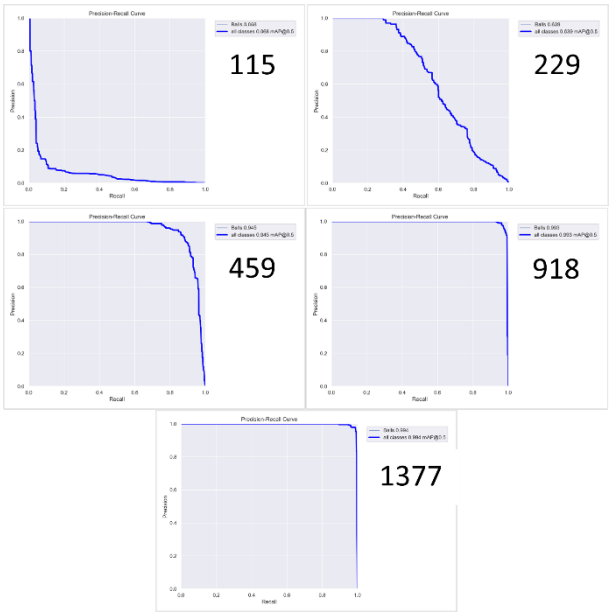
Na osnovi interpretacije rezultatov, predstavljenih v prejšnjem razdelku, smo modele testirali z lastnimi praktičnimi primeri. To smo storili na 5 različnih naključnih slikah športnih žog in 5 različnih naključnih slikah, na katerih ni športnih žog. Za testiranje smo uporabili preprosto skripto Python, v kateri definiramo pot do modela in sliko, na kateri želimo narediti detekcijo. Rezultat detekcije je nova slika, ki vsebuje pravokotnik okoli predmeta, ki je na podani sliki zaznan. Vsebina skripte je prikazana na sliki 5.

```
from ultralytics import YOLO;

model = YOLO("../runs/detect/train/weights/best.pt");

results = model("testImg.jpg", save=True);

```

Slika 5: Skripta za testiranje naučenega modela

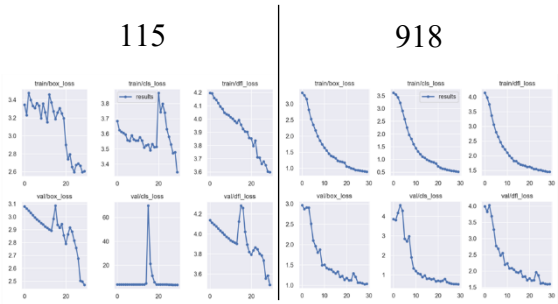
3 Rezultati

Že iz rezultatov krivulj natančnosti v odvisnosti od priklica je jasno, da model naučen na 115 slikah ne bo deloval dobro. Krivulja je praktično povsem pri spodnjem levem kotu, kar je ravno nasprotno od idealnega. Pri modelu, učenemu na 229 slikah je krivulja boljša, a vseeno ni idealna, saj kaže, z večanjem števila priklicanih primerov, nismo dolgo natančni. Spremembe se prikažejo pri 459, 819 in 1377 slikah, kjer je krivulja zelo blizu zgornjem desnem kotu grafa, kar pomeni, da smo z večjim številom priklicanih primerov natančni dlje časa. Krivulje so prikazane na sliki 6.



Slika 6: Krivulje natančnosti v odvisnosti od priklica

Količina napak se pri vseh naučenih modelih v povprečju manjša, je pa pri modelih z manjšim številom slik možno opaziti artefakte v obliki špic na krivuljah. Zanimiv je pogled na rezultate pri modelih z 918 in 1377 slikami, kjer se krivulje v zaključku počasi uravnavajo, kar bi lahko kazalo na to, da se je model na primerih naučil toliko, da nadaljnje interacije model izboljšujejo počasneje, saj se število napak približuje optimalni vrednosti 0. Grafe z artefakti za primer modela s 115 slikami in grafe za model z 918 slikami prikazuje slika 7.

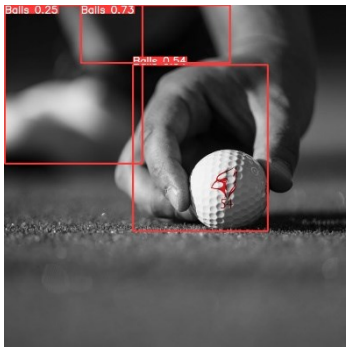


Slika 7: Krivulje napak za modela z 115 in 918 slikami

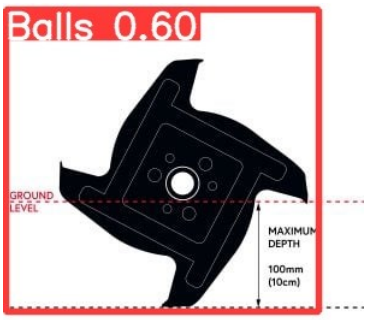
Testne slike, ki so podane po koncu učenja modela kažejo podobne rezultate. Medtem ko model, naučen s 115 slikami ne prepozna nobene izmed žog na testni sliki, je modela z 918 in 1377 slikami pravilno prepoznata vse žoge. Če za primerjavo pogledamo še model, učen z 459 slikami, se razlike pojavijo poleg nekaj napačnih zaznav predvsem v tem, koliko model v zaznave zaupa. Model s 459 slikami v povprečju v zaznavo zaupa s 70%, med tem ko je model s 918 slikami v rezultate zaznave zaupa s povprečno 90%.

Približno podobne rezultate so pokazali tudi testi z naključno izbranimi slikami. Model, naučen s 115 slikami ni prepoznal nobene žoge, model z 229 slikami je prepoznal 2 žogi od 10ih, model s 459 je prepoznal 3 žoge, model s 918 slikami 6 žoge, med tem ko je model, naučen s 1377 žogami prepoznal vseh 10 žog. Z večanjem števila slik se veča tudi verjetnost, koliko je

model prepričan, da res gre za zaznani objekt, saj model, ki se je učil na 229 slikah prepoznane žoge zazna s povprečno 52,5% verjetnostjo, med tem ko model z največjim naborom slik žoge prepozna s povprečno verjetnostjo 90,75%. Pri skoraj vseh modelih prihaja tudi do nekaj napačnih zaznav na slikah, kjer se žoga nahaja in slikah na katerih žoge sploh ni. Te se pojavljajo predvsem na slikah, na katerih se pojavljajo okrogle oblike, kar modele v določenih primerih zmede. Število napačnih zaznav se z večanjem števila testnih slik manjša. Primera napačnih zaznav prikazujeta sliki 8 in 9.



Slika 8: Primer napačne zaznave na sliki, ki vsebuje žogo



Slika 9: Primer napačne zaznave na sliki, ki ne vsebuje žoge

Rezultati testiranja z naključno izbranimi slikami so prikazani v tabeli 1. Stolpec uspešne zaznave prikazuje število žog, ki so bile na slikah in jih je model za določeno količino učnih slik pravilno prepoznal, poleg tega pa so prikazani povprečni deleži, v katerih je bil model prepričan v pravilno zaznavo. Stolpec napačne zaznave prikazuje število, v kolikšnem številu primerov je model prepoznal žogo, čeprav je tam ni bilo in kolikokrat žoge, ki je bila na sliki, ni prepoznal. Za primer s 115 slikami torej navajamo vrednosti 0/10, saj model ni prepoznal nobene izmed 10 žog, ki so se pojavljale na slikah. Vidimo, da se število napačnih zaznav zmanjšuje, a je vseeno prisotno tudi v modelu z največjim številom slik za učne podatke.

Tabela 1: Rezultati zaznav na praktičnih primerih

Število slik uporabljenih pri učenju	Uspešne zaznave	Napačne zaznave
115	0	10
229	2 (52,5%)	14
459	3 (75,7%)	10
918	6 (87%)	7
1377	10 (90,75%)	2

4 Zaključek

Iz dobljenih rezultatov je razvidno, da z večanjem števila slik pri učenju modelov lahko precej izboljšamo detekcijo objektov. Iz naših raziskav bi za zelo zanesljivo delovanje detekcije objektov ob uporabi podobnih parametrov potrebovali nekaj več kot 1000 slik. Glede na natančnost in priklic, velikih razlik med modelom, naučenim z 918 slikami in modelom, naučenim z 1377 slikami ni, pregled na praktičnih primerih pa je pokazal, da je model z 1377 slikami precej boljši predvsem v primerih, ko se na sliki pojavlja več objektov.

Projekt dopušča tudi možnosti izboljšav, s katerimi bi raziskave nekoliko nadgradili. V prvi vrsti bi bilo zanimivo preveriti, če lahko z večanjem števila ciklov, izvedenih med učenjem, dosežemo dobre rezultate tudi ob uporabi manjšega števila slik. V našem primeru smo za testiranje v praksi uporabili le 5 naključno izbranih slik športnih žog, zato bi bilo smiselno to izboljšati tako, da v testiranje vključimo 5 slik, ki bi vsebovale več t.i. motečih faktorjev. Ena izmed možnih izboljšav pa je nedvomno tudi testiranje uporabe na zaznavanju v realnem času.

Z rezultati smo v splošnem zadovoljni, saj smo dosegli pričakovane cilje in dokazali, da spreminjanje števila slik pri učenju vpliva na zaznavanje objektov v praksi.

Literatura

[1] X. Zou, „A Review of Object Detection Techniques,“ v *2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, 2019.

[2] D. Mesquita, „Introduction to Object Detection Model Evaluation,“ Towards Data Science, 2021.

[3] J. Rosenbacher, „How many images do you need to train a model?,“ Roboflow, 2022.

[4] M. S. Swetha, „Survey of Object Detection using Deep Neural Networks,“ International Journal of Advanced Research in Computer and Communication Engineering, 2018.