**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**Prepoznavanje objekata pomoću dubokih neuronskih mreža**

Projektni zadatak iz kolegija Obrada slike i računalni vid

**Ivan Kaučić**

**Osijek, 2022.**

SADRŽAJ

1. UVOD ......................................................................................................................................... 3

2. Opis projektnog zadatka ............................................................................................................ 3

3. PREGLED PODRUČJA TEME ........................................................................................................ 3

4. KORIŠTENI ALATI .........................................................................................................................4

5. DATASET .....................................................................................................................................5

6. RETINANET .................................................................................................................................6

7. PROGRAMSKO RJEŠENJE ........................................................................................................... 7

8. OSTVARENI REZULTATI .............................................................................................................. 8

9. ZAKLJUČAK ............................................................................................................................... 10

10.LITERATURA ............................................................................................................................ 11

1. **Uvod**

Trend u računarstvu sve više postaje detekcija objekata, iako koncept postoji i koristi se već dugi niz godina, uvijek se rezultati pokušavaju usavršiti. U robotici je pogotovo bitna detekcija objekata za razne zadatke za koje su roboti programirani.

Postoji puno modela za detekciju objekata, jedan od najpouzdanijih, najpopularnijih ali i najboljih su modeli koji koriste duboko učenje.

1. **Opis projektnog zadatka**

Ovaj projekt će se pozabaviti testiranjem već postojeće istrenirane duboke mreže (npr. ResNet, VGG16, Inception) za detekciju objekata, opisivanjem korištenog dataset-a, ali i pričanjem o samoj strukturi izabrane istrenirane duboke mreže.

1. **Pregled područja teme**

Detekcija objekata je široka i istražena tema. Često ide ruku pod ruku s robotima, jer oni koriste razne vrste detekcije objekata za poslove koje su programirani. U novije vrijeme detekcija objekata je ključna kada pričamo o autonomnim vozilima. Moraju prepoznati, identificirati i klasificirati objekte u vožnji u realnim brzinama. U industrijama i industrijskim pogonima roboti moraju prepoznavati objekte na kojima rade, npr. u tvornicama čokolade, moraju moći prepoznati cijeli komad upakirane čokolade kako bi ga podigli i prebacili u kutiju koju će, kada napune, poslati za izvoz. Detekcija objekata događa se i na satelitskim slikama ili pomoću dronova gdje se s velikih visina traže i detektiraju objekti od interesa.

Neke od najpoznatijih dubokih neuronskih mreža za detekciju objekata su ResNet, RetinaNet, VGG16 i Inception. Svaki od njih se razlikuje u algoritmu pomoću kojega detektiraju objekte, a važno je da mogu što više objekata detektirati sa što većom točnošću. Nekada je potrebno imati veću brzinu detekcije, za malo manju cijenu točnosti, dok u osjetljivim poslovima puno je važnije imati što veću preciznost i žrtvovati vremenske zahtjeve.

1. **Korišteni alati**

Za razvoj ovog projekta sam u potpunosti koristio programski jezik Python, a radio sam u okolišu razvojnog okruženja Spyder koji se nalazi unutar platforme Anaconda.

Također, unutar samog Pythona, koristio sam biblioteke tensorflow, matplotlib, numpy, keras i biblioteke za rad s datotekama.

Python[1] je programski jezik opće namjene, interpretiran i visoke razine kojeg je stvorio Guido van Rossum 1990. godine (prva javna inačica objavljena je u veljači 1991. godine), ime dobiva po televizijskoj seriji Monty Python's Flying Circus. Po automatskoj memorijskoj alokaciji, Python je sličan programskim jezicima kao što su Perl, Ruby, Smalltalk itd. Python dopušta programerima korištenje nekoliko stilova programiranja. Objektno orijentirano, strukturno i aspektno orijentirano programiranje stilovi su dopušteni korištenjem Pythona te ova fleksibilnost čini Python programski jezik sve popularnijim. Python se najviše koristi na Linuxu, no postoje i inačice za druge operacijske sustave.

Anaconda[2] je distribucija programskih jezika Python i R za znanstveno računarstvo (informacijske znanosti, aplikacije na bazi strojnog učenja, obrada velike količine podataka, prediktivna analitika i sl.) kojoj je cilj pojednostaviti upravljanje i implementaciju paketa. Distribucija uključuje pakete pogodne za Windows, Linux i macOS. Osnovana je i još uvijek održavana od strane Anaconda Inc., koja su stvorili Peter Wang i Travis Oliphant 2012. godine. Također, postoji i manja, bootstrap verzija koja se zove Miniconda.

Spyder[3] je integrirano višeplatformsko razvojno okruženje otvorenog koda za znanstveno programiranje bazirano na jeziku Python. Spyder sadrži niz istaknutih integriranih paketa u znanstvenom paketu Python, uključujući NumPy, SciPy, Matplotlib, 4pande, IPython, SymPy i Cython, kao i druge softvere otvorenog koda. Pušteno je na tržište pod licencom MIT-a. Prvotno je kreirano od strane Pierre Raybauta u 2009. godini, a od tad je svakodnevno nadzirano i poboljšavano uz pomoć Python programera i Python zajednice.

Spyder je također proširiv i putem vanjskih dodataka, uključuje podršku interaktivnih alata za proučavanje podataka te ima ugrađene instrumente za osiguravanje kvalitete. Njegova višeplatformska podrška je još jedan od velikih prednosti, putem Anaconde na Windowsima, MacPorts-a na macOS-u te raznim inačicama na Linuxu poput Arch Linux, Debian, Fedora, Ubuntu i sl.

1. **Dataset**

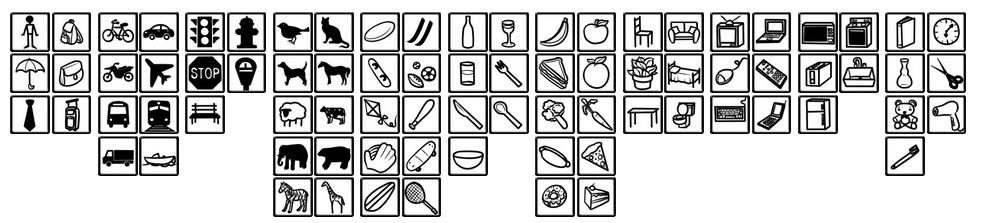
Skup podataka MS COCO[4] (Microsoft Common Objects in Context) skup je slika na kojima su detektirani, označeni i segmentirani objekti od interesa (Slika1.).



**Slika 2.** Prikaz jede slike iz COCO dataset-a.

Prva verzija skupa podataka MS COCO objavljena je 2014. Sadrži 164K slika podijeljenih u skupove za obuku (83K), validaciju (41K) i test (41K). U 2015. objavljen je dodatni testni skup od 81K slika, uključujući sve prethodne testne slike i 40K novih slika.

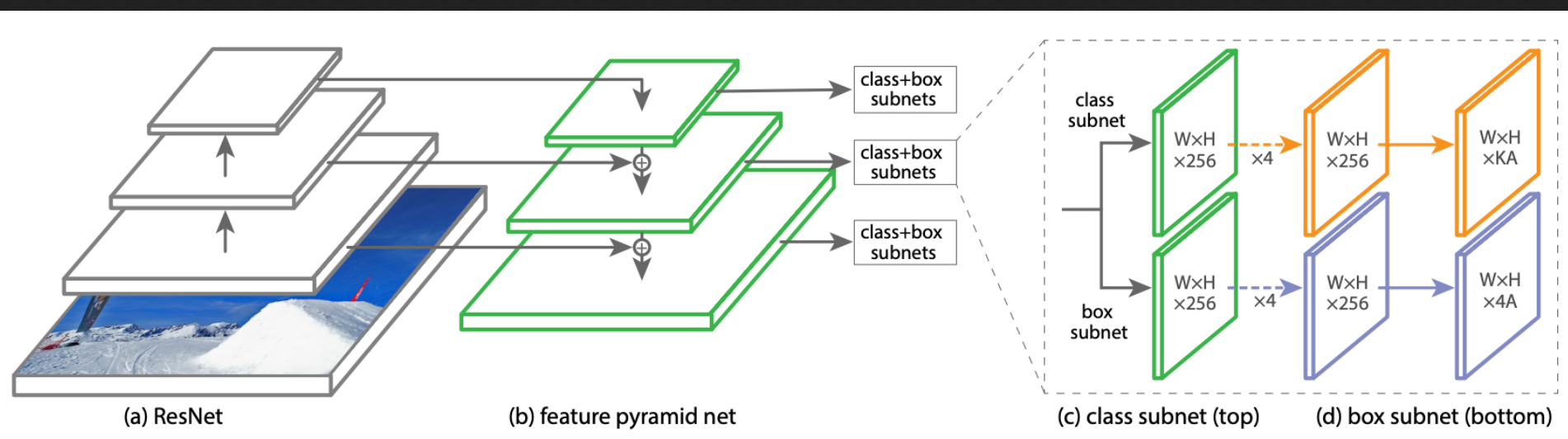
Dataset ima preko 80 kategorija objekata (Slika 2.).



**Slika 2.** Popis kategorija objekata COCO dataseta.

1. **RETINANET**

RetinaNet[5] je jednostupanjski model detekcije objekta koji koristi funkciju gubitka fokusa za rješavanje neravnoteže klasa tijekom treninga. Fokalni gubitak primjenjuje modulirajući termin na gubitak unakrsne entropije kako bi se fokusiralo učenje na teške negativne primjere. RetinaNet je jedinstvena, objedinjena mreža sastavljena od okosnice mreže i dvije podmreže specifične za zadatke. Okosnica je odgovorna za izračunavanje karte konvolucijskih značajki preko cijele ulazne slike i predstavlja zasebnu konvolucijsku mrežu. Prva podmreža izvodi konvolucionu klasifikaciju objekata na izlazu okosnice; druga podmreža izvodi regresiju konvolucijskog graničnog okvira (Slika 3.). Dvije podmreže imaju jednostavan dizajn koji autori predlažu posebno za jednostupanjsko, gusto otkrivanje.



**Slika 3.** Arhitektura RetinaNet modela.

1. **PROGRAMSKO RJEŠENJE**

Najvažniji dio programskog rješenja ovog zadatka nalazi se u korištenju već postojećeg i treniranog modela detekcije objekata duboke neuronske mreže(Slika).

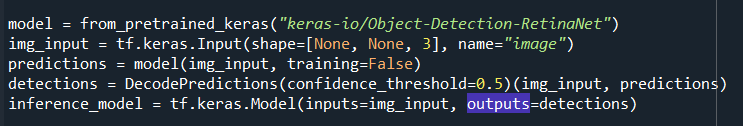
U keras biblioteci postoji veliki izbor već treniranih modela, stoga je potrebno odabrati „Object-Detection-RetinaNet“ model.

Kako bi uopće mogli koristiti model potrebno je imati i ulazni podatak, što su u ovom slučaju slike. Potrebno je učitati slike koje želimo testirati na modelu.

Model zatim treba zaposliti da obavi svoju zadaću, pokuša pronaći objekte koje je treniran pronaći na slici i klasificirati ih.

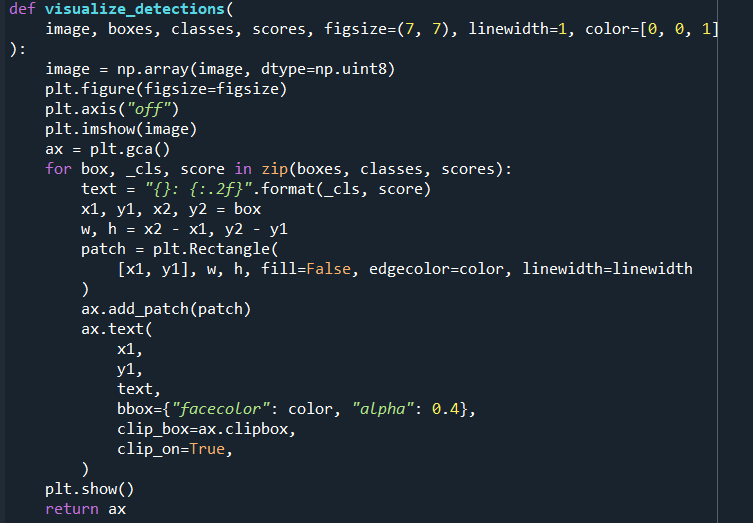
Međutim, prilikom procesa predikcije, model će detektirati razne objekte koje ne bi ni trebao. Stoga, mora se u program ubaciti granična vrijednost. Ako predikcija nije veća od granične vrijednosti (npr. predikcija je 0.44 a granična vrijednost je 0.6) taj se objekt neće uzeti u obzir kao ni njegova vjerojatnost da se nalazi na slici.

Važno je znati korake operacije tijekom korištenja istreniranog modela koji se bavi detekcijom objekata na slici.



**Slika 4.** Korištenje treniranog modela pomoću keras-a.

Također, radi bolje vizualizacije krajnjeg rješenja testiranja trenirane mreže implementirano je i prikazivanje detektiranog objekta unutar boudry-boxa zajedno s njegovom klasom i vjerojatnošću da se radi o istoj klasi(**Slika 5.**).

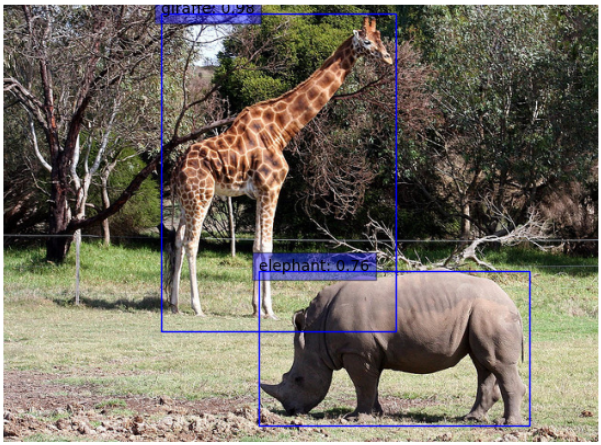


**Slika 5.** Implementacija vizualizacije rezultata na output slici.

1. **OSTVARENI REZULTAT**

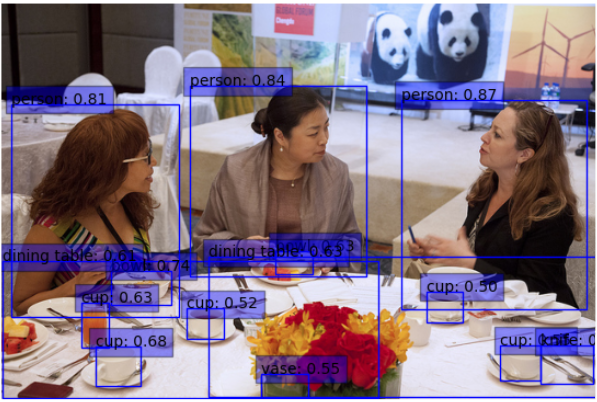
Rezultati korištenja ovog modela su zadovoljavajući, model nije uvijek uspio prepoznati sve objekte koje je treniran za prepoznavanje, ali je većinu bez problema prepoznavao. Zbog korištenja manjeg broja podataka iz dataseta za treniranje modela, često vrijednosti predikcije nisu prelazile 0.75 score. Za dodatno testiranje modelu bi dao objekte koje ne bi smio moći prepoznati, ali bi ih u krajnjem rezultatu prepoznao kao neki od modela koje je imao moć prepoznati, tako bi npr. nosoroga prepoznao kao slona (Slika 6.).

Odlično radi i u scenama gdje ima više objekta od interesa, većinu ih prepozna, međutim nekada se muči s detekcijom objekata koji su sakriveni iza drugih objekata.



**Slika 6.** Model prepoznaje nosoroga kao slona.

Kao rezultate prilažem i još nekoliko zanimljivih slika koje je model izbacio kao rješenje:



**Slika 7.** Model s velikom vjerojatnošću prepoznaje scenu s puno objekata.



**Slika 8.** Model može prepoznati pribor za jelo, ali u ovom primjeru nije ništa detektirao.

1. **ZAKLJUČAK**

Trenirani model duboke neuronske mreže koji sam koristio za testiranje dao je odlične rezultate u detektiranju objekata na kojima je treniran. Prepoznavao je i detektirao objekte s velikom preciznošću. U nekim krajnjim slučajevima davao je iznenađujuće rezultate gdje na cijeloj slici nije uspio prepoznati objekte koje je bio treniran prepoznati (Slika 8.).

Detekcija objekata pomoću dubokih neuronskih mreža je stalno razvijajuća disciplina i u budućnosti možemo očekivati još nevjerojatnije rezultate. Primjene su ograničene samo na ljudsku maštu.

1. **LITERATURA**

[1] <https://www.python.org/>

[2] <https://www.anaconda.com/products/distribution>

[4] [https://cocodataset.org/#home](https://cocodataset.org/%23home)

[3] <https://www.spyder-ide.org/>

[5] <https://paperswithcode.com/method/retinanet>