Detekcija objekata nad otvorenim rječnikom

Ćorić, Bruno

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 605

DETEKCIJA OBJEKATA NAD OTVORENIM RJEČNIKOM

Bruno Ćorić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 605

DETEKCIJA OBJEKATA NAD OTVORENIM RJEČNIKOM

Bruno Ćorić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 605

Pristupnik:	**Bruno Ćorić (0036517561)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Detekcija objekata nad otvorenim rječnikom**

Opis zadatka:

Detekcija objekata važan je zadatak računalnog vida s mnogim zanimljivim primjenama. Međutim, učenje na unaprijed definiranim taksonomijama otežava primjene u otvorenom svijetu. Ovaj problem možemo ublažiti primjenom modela koji ulazne slike ugrađuju u latentni jezični prostor velikih jezičnih modela. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće arhitekture za gustu predikciju. Odabrati slobodno dostupne skupove slika te oblikovati podskupove za učenje, validaciju i testiranje. Oblikovati model za detekciju objekata nad otvorenim rječnikom te uhodati postupke učenja i validiranja hiperparametara. Primijeniti naučene modele, prikazati eksperimente na javnim podatcima te usporediti generalizacijsku izvedbu sa stanjem tehnike. Komentirati učinkovitost učenja i zaključivanja. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

Uvod

Detekcija objekta jedna je od primjena umjetne inteligencije u računalnom vidu te je cilj detekcije objekata na slici detektirati točno gdje se nalazi objekt i što je taj objekt. U većini slučajeva se za detekciju objekata koriste konvolucijske mreže te se detektiraju objekti iz određenog skupa kategorija. To znači da se istraži problem, odrede vrste objekata koje bi model umjetne inteligencije trebao detektirati te se nauči model kako bi detektirao te objekte.

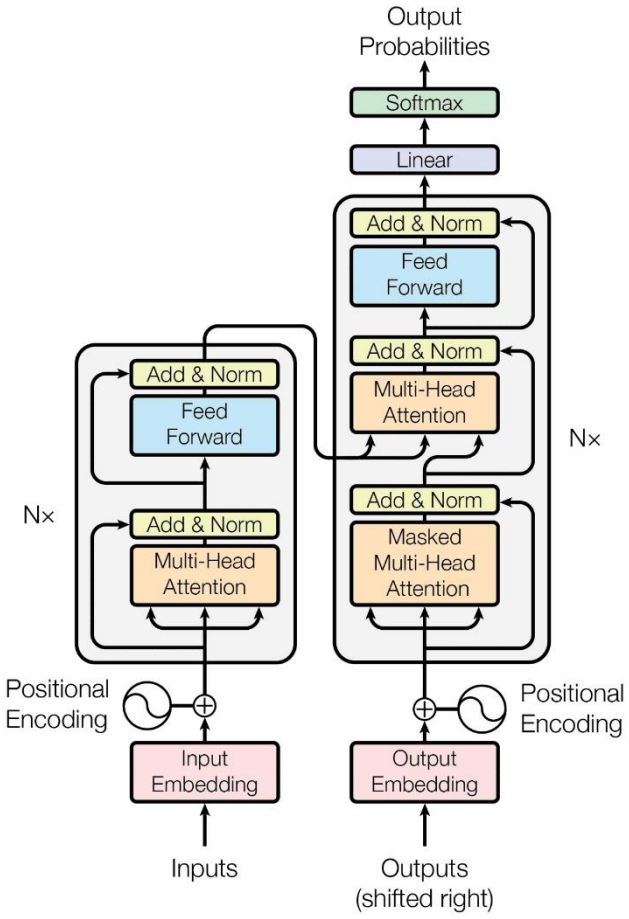
Jedan od nedostataka takvog pristupa je baš ta zatvorenost na određen skup kategorija. U svijetu postoji bezbroj mogućih vrsta objekata koje bi model mogao detektirati te je nemoguće zamisliti trenirati model za primjenu na nekom jako velikom skupu kategorija.

U ovom diplomskom radu opisati ćemo kako detektirati proizvoljan broj objekata sa slike. Koje su prednosti, a koji nedostaci takvog pristupa te zašto takav pristup još uvijek nije u široj uporabi.

Rezultat ovoga diplomskog rada će biti model umjetne inteligencije specijaliziran za detekciju neodređenog broja objekata sa slika dobivenih s bespilotne letjelice te trenutno prema istraživanju autora ovoga rada ne postoji dostupan ni jedan model specifično treniran da radi u takvim uvjetima.

1. *Transformer*

Transformer [1] je arhitektura dubokih modela na kojoj će se temeljiti naš model za detekciju objekata s otvorenim vokabularom. Ova arhitektura je prvo revolucionirala područje obrade prirodnog jezika, a zatim i područje računalnog vida. Najvažnija karakteristika ove arhitekture je sloj pažnje, zbog čega je *Transformer* postao popularan u mnogim granama umjetne inteligencije. Ključna prednost *Transformer* arhitekture u odnosu na konvolucijske mreže je globalno receptivno polje [46]. Globalno receptivno polje omogućava svakom dijelu ulaza da vidi ostale dijelove ulaza. To se najlakše može predočiti na primjeru rečenice. *Transformer* dijeli rečenicu u određene dijelove, bilo to slova, riječi ili nešto treće, te svaki taj dio rečenice pretvara u vektor. Svaki taj vektor ima utjecaj na druge vektore dok konvolucijske mreže imaju veću induktivnu sklonost, gdje dijelovi utječi samo na susjedne dijelove [46]. Opisati ćemo sloj pažnje, detaljnije predstaviti prednosti i nedostatke *Transformer* arhitekture objasniti zašto se model koji je rezultat ovoga rada temelji na *Transformer* arhitekturi.



Slika 1. Arhitektura transformer [1]. Lijevo se nalazi koder, a desno dekoder. Ulaz se prvo pretvara u niz vektora pogodan za *Transformer*. Tim vektorima se dodaje informacija o poziciji u nizu. Zatim ulaz prolazi kroz koder dio *Transformera* te će se izlazi iz kodera koristiti u dekoder dijelu modela kako bi se generirali novi izlazi.

1.1. Prilagodba podataka za ulaz u *Transformer*

U uvodu u poglavlje opisan je primjer pretvaranja rečenice u dijelove koji se potom obrađuju u modelu. S obzirom na to da se ovaj rad temelji na primjeni *Transformera* u računalnom vidu i obradi prirodnog jezika, objasnit ćemo najčešći način pretvorbe ulaznih podataka u valjane podatke za unos u model.

Glavni rezultat prilagodbe podataka trebao bi biti vektor koji predstavlja te podatke u vektorskom prostoru.

1.1.1. Prilagodba podataka u prirodnom jeziku

Prirodni jezik se sastoji od skupa znakova koji tvore neku cjelinu. Najjednostavniji način za podjelu jezika na dijelove bio bi podjela teksta na znakove ili riječi. Takav pristup se često koristio na početku primjene T*ransformera* u obradi prirodnog, pri čemu bi se stvarao vokabular. Vokabular je popis svih podržanih znakova ili skupova znakova koje model može prepoznati, a svaki element tog skupa ima svoj broj.

Kao što je navedeno u uvodu u poglavlje, svaki element koji ulazi u model treba pretvoriti u vektor. Da bismo to postigli, koristimo tablicu za pretraživanje koja za svaki element u vokabularu sadrži njegov vektor. Ta tablica nije statična, već je promjenjiva i uči se zajedno s modelom. Na taj način možemo podijeliti tekst na elemente koji se nalaze u vokabularu i te elemente jednostavno pretvoriti u njihove vektorske reprezentacije. Kako se model poboljšava, tako se poboljšavaju i te vektorske reprezentacije [46].

Metoda za prilagodbu teksta koja će se koristiti u ovom radu je *Wordpiece tokenizer* [47]. Ova metoda prilagodbe teksta korištena je u modelu BERT [4]], koji će se također koristiti za obradu teksta u ovom radu. *WordPiece tokenizer* dijeli tekst u određene skupove znakova koji se često pojavljuju zajedno i imaju određeno značenje. To znači da se znakovi često dijele prema riječima, ali to nije nužno uvijek slučaj. Ponovno se stvara vokabular od mogućih elemenata te se svakom elementu pridjeljuje njegova vektorska reprezentacija.

1.1.2. Prilagodba podataka u računalnom vidu

Prilagodba slika predstavljala je velike izazove u ranim pokušajima korištenja *Transformera* u računalnom vidu. Tekst je relativno jednostavno i logično podijeliti na dijelove, a tekst često ne sadrži veliki broj dijelova. Primjerice, ova rečenica podijeljena na riječi sadrži 31 riječ. S druge strane, sliku nije tako jednostavno i logično podijeliti na dijelove; najjednostavniji pristup bio bi uzeti svaki piksel kao jedan element. To bi značilo da čak i mala slika, recimo 64x64 piksela, sadrži 4096 elemenata [46].

Veliki problem *Transformera* je kvadratna vremenska i prostorna složenost sloja pažnje s obzirom na dužinu ulaza. Ovo predstavlja veliki problem kod slika jer one u prosjeku sadrže puno više elemenata nego tekst. Slike visoke rezolucije, koje su predmet istraživanja ovog rada, često imaju preko milijun piksela [46].

Vision Transformer [3], u daljenjem radu nazivan pod poznatom kraticom *ViT,* uveo je novi način primjene *Transformera* u obradi slika. *ViT* prvo dijeli sliku na nepreklapajuća kvadratna okna te se ta kvadratna okna pretvaraju u vektore koji se zatim prosljeđuju u *Transformer* [3]. U originalnom radu [3] *ViT* koristi kvadratna okna veličine 16x16 piksela, čime smanjuje broj ulaznih elemenata u model.

1.2. Sloj pažnje

Sloj pažnje je glavna karakteristika *Transformera*, te je upravo ovaj sloj omogućio ovoj arhitekturi da postigne velike prednosti u odnosu na druge, ali i određene nedostatke. Sloj pažnje se temelji na tri elementa: upit, ključ i vrijednost. Svaki element u ulazu imat će svoj vektor upita, vektor ključa i vektor vrijednosti. Ti vektori se dobivaju množenjem ulaznog vektora s parametrima sloja pažnje [1].

Pažnja se zasniva na funkciji *Scaled dot-product attention* [1].

$$Pažnja(q, k, v) = s\left(\frac{qk^T}{\sqrt{d}}\right)v$$

U ovoj funkciji **q**, **k** i **v** predstavljaju tenzore upita, ključa i vrijednosti, dok **s** označava funkciju *softmax*, a **d** veličinu vektora. To znači da se tenzor **q** sastoji od svih vektora upita za sve ulaze. Svaki vektor upita množi se sa svakim drugim vektorom ključa, a dobiveni rezultati zatim se množe sa svakim vektorom vrijednosti [46].

Jedna važna stvar koju treba primijetiti je da pozicija određenog vektora ne igra ulogu u računanju pažnje. To predstavlja problem jer želimo modelu pružiti informaciju o poziciji svakog elementa u odnosu na druge elemente. Stoga se u svaku vektorsku reprezentaciju ugrađuje i informacija o poziciji elementa u nizu.

2. *Transformer* u prirodnom jeziku

Iako je *Transformer* ista arhitektura i za računalni vid i za prirodni jezik, način na koji se primjenjuje i kako se uči različit je. Kod obrade prirodnog jezika važno je naglasiti da se originalni *Transformer* sastoji od dva dijela: kodera i dekodera. Koder pretvara podatke u vektorski oblik, dok dekoder postupno generira izlaze. Najlakši primjer za opisivanje bio bi prevođenje teksta iz jednog jezika u drugi. Koder predstavlja originalni tekst u vektorskom obliku, a dekoder zatim prevodi taj tekst u drugi jezik. Budući da duljina teksta uglavnom nije ista u oba jezika, ne možemo samo svaki token prevesti u token u drugom jeziku, već moramo postupno prevoditi dok ne završimo prijevod.

Transformer u području prirodnog jezika je zamijenio neke ranije popularne arhitekture poput *LSTMa* [48] i ostalih mreža tog tipa iz nekoliko razloga.

Jedan od glavnih razloga je način treniranja *Transformera*. Za razliku od tadašnjih popularnih mreža, *Transformer* se mogao trenirati paralelno, a ne postupno korak po korak. Na primjer, kod prevođenja jezika, u korištenju bismo prvo preveli jedan dio, a zatim bismo, s obzirom na taj prevedeni dio, morali prevesti još jedan dio i tako postupno dok ne prevedemo cijeli tekst. Kada bi model morao tako postupno raditi i tijekom treniranja, to bi trajalo jako dugo. Zbog toga je uvedena maska koja određuje koji token može utjecati na koji token. Tako bismo modelu istovremeno mogli dati sve željene izlaze, ali mu odrediti da prvi token ulaze ne može biti pod utjecajem budućih tokena, dok budući tokeni mogu biti pod utjecajem prvog tokena, ali ne i tokena poslije njih, i tako dalje [46].

Druga velika prednost je sposobnost pamćenja *Transformera* kod dužih sekvenci. To znači da možemo imati vrlo dugu sekvencu, ali prvi token će jednako jako utjecati na zadnji token kao i ostali tokeni. Međutim, ovdje dolazimo do problema *Transformera* koji je već bio spomenut, a to je složenost. *Transformer* ima kvadratnu vremensku i prostornu složenost s obzirom na duljinu sekvence, tako da ne možemo imati predugačke sekvence bez trošenja puno resursa.

2.1. BERT

U ovom radu ćemo koristiti model *BERT* [4]. Za razliku od većine trenutno popularnih *Transformera* kao što su GPT [50] ili Gemini [49], BERT je samo koder, a ne i dekoder ili oboje kako je to slučaj u originalnom radu. To znači da je rezultat ovog modela vektorska reprezentacija svakog ulaznog elementa. Osim vektora za svaki element ulaza, BERT kao izlaz daje i skupni token. Taj skupni token (označen kao [CLS]) je vektor koji reprezentira cijeli ulazni tekst te se često koristi u klasifikacijskim zadacima na tekstu kao što su predviđanje emocija u tekstu, predviđanje neželjenog sadržaja i slično [4].

BERT će poslužiti kako bismo mogli reprezentirati željene objekte za detekciju kao što su tekstualni opisi poput automobila ili osobe, te ćemo *BERT* koristiti kako bismo te tekstualne opise pretvorili u vektorski oblik. Kasnije u radu ćemo opisati kako ovi vektori interaktiraju s vektorima dobivenim iz slike.

3. *Transformer u računalnom vidu*

Glavni izazov primjene *Transformera* u računalnom, kako je opisano u prethodnim poglavljima, jest vremenska i prostorna složenost s obzirom na duljinu ulazaKako bismo djelomično prevladali ovaj izazov, predložili smo upotrebu kvadratnih prozora poput onih u *ViTu.*

Jedan od ključnih problema *Transformera* je potreba za velikom količinom podataka za obuku ovakvih modela [3]. Razlog tome je što *Transformer* ima manju induktivnu sklonost u usporedbi s konvolucijskim mreža [3]. Njegovo receptivno polje je globalno, što znači da svaki element gleda sve ostale elemente, za razliku od konvolucijskih mreža koje se fokusiraju samo na susjedne elemente. To je također razlog zašto se *Transformer* prvo pokazao uspješnim u obradi prirodnog jezika, dok je u računalnom vidu postao značajniji tek kasnije [46]. Prirodni jezik je lako dostupan i ne zahtijeva puno prostora, što olakšava prikupljanje velike količine podataka za obuku modela. *Transformer* također može biti treniran bez potrebe za označenim podacima, koristeći tehnike predviđanja idućeg tokena u nizu ili maskiranja tokena unutar niza i predviđanja njihovih vrijednosti u odnosu na ostale [4]. Međutim, ovo nije jednako jednostavno u računalnom vidu gdje su slike manje dostupne i zauzimaju više prostora.

U nastavku rada detaljnije ćemo predstaviti *ViT*, kao i *Swin Transformer* [5], varijantu *Transformera* prilagođenu za računalni vid, koji ćemo koristiti kao enkoder za slike.

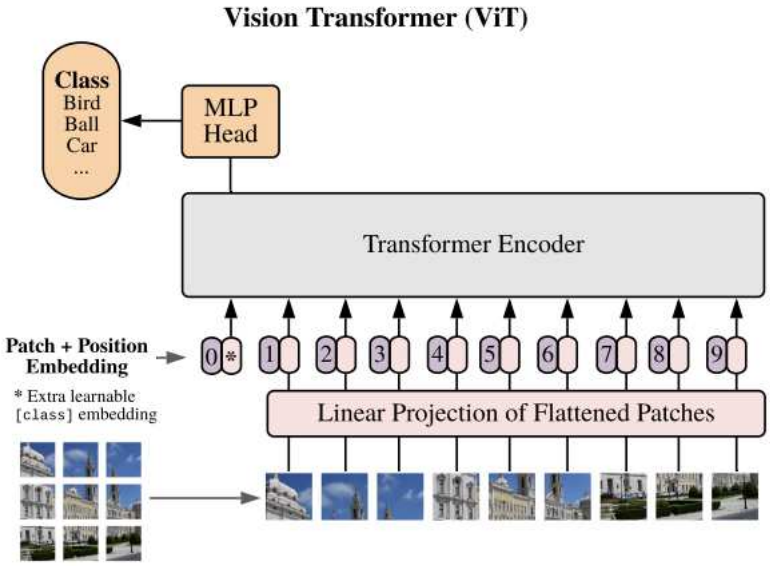
3.1. *Vision Transformer*

ViT [3] je arhitektura *Transformera* koja je postala temelj za sve kasnije primjene *Transformera* u računalnom vidu. Ranije je opisan postupak pretvaranja slike u ulaz pogodan za *Transformer*, što uključuje podjelu slike na nepreklapajuća kvadratna okna te pretvaranje tih okna u vektore koji služe kao ulaz za *Transformer*.

Vision transformer se sastoji od niza *Transformer* kodera, pri čemu izlaz jednog kodera postaje ulaz za sljedeći koder. Broj izlaza jednak je broju ulaza u svaki koder, što znači da iz *ViTa* dobivamo isti broj elemenata kao što smo imali na ulazu. Svaki od tih izlaznih vektora predstavlja dio slike. Slično kao u *BERTu* [4], u ViT se dodaje skupni token (često

označen kao [CLS]) koji predstavlja cijelu sliku. Ovaj vektor se najčešće koristi za klasifikaciju slike [3].

ViT je postigao izuzetne rezultate u raznim zadacima računalnog vida. Međutim, jedan od glavnih nedostataka ovakvih arhitektura jest potreba za velikom količinom podataka kako bi modeli temeljeni na *ViTu* bili konkurentni ili čak nadmašili konvolucijske mreže [4].



Slika 2. Arhitektura *Vision Transformera* kao ulaz prihvaća sliku te ju dijeli na okna koja se pretvaraju u vektore. Tim vektorima se također ugradi i informacija o poziciji [1]. Vektorima iz slike se također dodaje i učeni [CLS] vektor te svi ti vektori prolaze kroz sliku i međusobno djeluju jedno na drugo. Na kraju pomoću klasifikacijske glave se slika može klasificirati. [4]

3.2. *Swin Transformer*

U prethodnom poglavlju predstavljen je *Vision transformer* i istaknut njegov najveći nedostatak - potreba za ogromnom količinom podataka kako bi postigli željene rezultate.

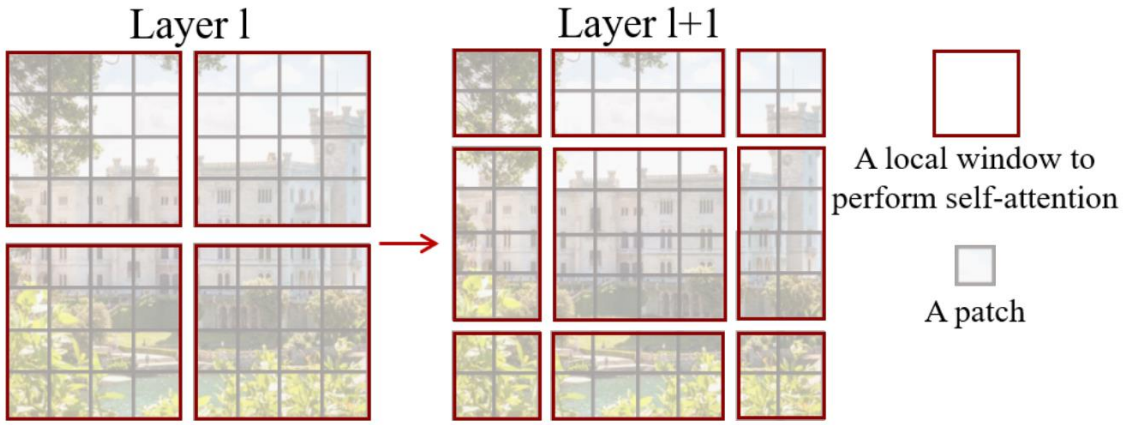
Ova arhitektura je osmišljena kao temelj za primjenu *Transformera* u računalnom vidu, analogno raznim konvolucijskim mrežama koje se koriste za mnoge probleme u tom području [5].

Swin Transformer se temelji na *ViTu*, ali uključuje modificirani koder kako bi se poboljšala primjena *Transformera* u računalnom vidu. Specifično, rješava problem različitih veličina

elemenata u slikama za razliku od teksta. Složenost u vezi s veličinom upita predstavlja značajan izazov u *Transformer* arhitekturama, osobito u kontekstu slika gdje postoji varijacija u veličini slike, rezoluciji i veličini objekata [5]. Na primjer, detekcija objekata na visoko rezolucijskim slikama može uključivati objekte veličine samo 10 piksela. Ovakve primjene mogu predstavljati veliki izazov za *Transformere* u detekciji malih objekata ili semantičkoj segmentaciji, gdje je bitan svaki piksel. [5]

Problem detekcije malih objekata je ono čime se ovaj rad bavi s toga ćemo koristiti *Swin Transformera.* Ovaj model se temelji na izradi hijerarhijske karte značajki i ima linearnu vremensku i prostornu složenost u odnosu na veličinu slike. [5]

Funkcionalnost ovog prikaza jasno je prikazana na slici [Slika 3.].



Slika 3. Prikaz slojeva u S*win Transformeru.* Počinjemo sa podjelom slike po okvirima. Te okvire možemo gledati kao sliku koja ulazi u *ViT*. Ti okviri su podijeljeni u dodatna kvadratna okna koja čine ulaze u sloj pažnje te samo elementi u istom okviru se međusobno vide. U idućim slojevima se mijenjanju pozicije i veličine okvira te novi elementi iz slike vide druge nove elemente. [5]

Swin Transformer počinje s malim okvirima slično kao *ViT*, no razlikuje se po tome što postupno spaja ove manje okvire kako ulaz prolazi kroz dublje slojeve.

Broj okvira u svakom sloju je fiksan, a pažnja se primjenjuje samo unutar lokalnih prozora okvira koji se međusobno ne poklapaju. Ovaj pristup omogućuje *Swin Transformeru* da ostvari linearnu složenost umjesto kvadratne u odnosu na veličinu slike. [5]

Važno je primijetiti da se lokalni prozor okvira, nad kojim se provodi pažnja, mijenja kroz slojeve. To znači da dijelovi slike koji nisu u istim lokalnim prozorima na početku, kasnije mogu međusobno interaktirati putem pažnje. [5]

Originalna formula za pažnju bi bila ovakva:

$\Omega(MSA) = 4hwC^2 + 2(hw)^2C$ [5]

Kada primijenimo W-MSA verziju ovog modula, to jest kada podijelimo okvire u neke prozore i radimo pažnju samo na tim prozorima, a prozori su neovisni jedni od drugih dobijemo ovakvu formulu:

$\Omega(W - MSA) = 4hwC^2 + 2hwM^2C$ [5]

Gdje imamo M^2 tokena u jednom okviru i svaki taj token mora se množit jedan s drugim znači dobijemo M^4C. Pošto ima $\frac{hw}{M^2}$ prozora to pomnožimo i dobijemo $2M^2hwC$. Vidi se da je hw linearano, a ne kvadratno ovisan.

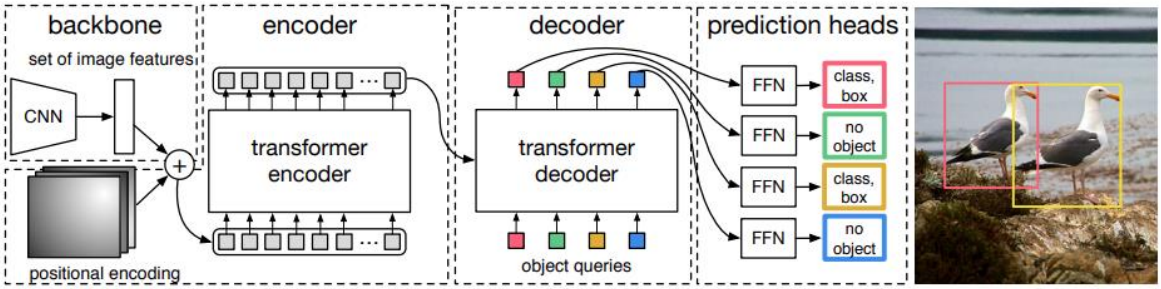
Također *Swin Transformeru* potreban je manji broj podataka kako bi postigao dobre rezultate [5]. To možemo povezati s većom induktivnom sklonošću kod *Swin Transformera* za razliku od *Vision Transformera* te možemo uočiti poveznice između konvolucijskih mreža i *Swin Transformera*.

3.3. DETR

Detection Transformer (DETR) [2] je jedan od prvih radova koji je predstavio rješenje za problem detekcije objekata pomoću *Transformera.*

DETR arhitektura se sastoji od nekoliko dijelova: plitke konvolucijske mreže, kodera, dekodera i potpuno povezanih slojeva koji rade predikcije o koordinatama okvira objekta i kategoriji objekta [2]. U ovom poglavlju fokusirati ćemo se na *Transformer* dekoder i slojeve za predikciju rezultata.

Transformer dekoder u originalnom radu korišten je kao dio modela koji daje predikcije te se zaustavlja kada je predikcija specijalan znak. Taj znak označava da je dekoder dovršio obradu ulaza. U ovoj arhitekturi, dekoder se koristi kao sloj koji predlaže objekte na slici. Za razliku od originalnog rada, ovaj radi paralelno, i u fazi predviđanja, generirajući sve izlaze istovremeno.



Slika 4. Arhitektura DETR modela. Bitna razlika između DETR i ViT modela je dodatak dekodera. Dekoder služi kako bi predviđali objekte iz slike te svaki od prijedloga se klasificira i za svaki se predlaže lokacija. [5]

3.3.1. Ulaz

Ulaz u dekoder su učeni vektori koji se međusobno razlikuju, dok se kao pomoćne vrijednosti koriste vektori slike. Broj učenih vektora je hiperparametara koji se može podešavati i predstavlja maksimalni broj objekata koje model može detektirati. Izlaz iz ovog sloja su vektori koji označavaju potencijalne objekte.

Nakon dobivanja vektora potencijalnih objekata, oni prolaze kroz potpuno povezane mreže. Svaki vektor prolazi kroz klasifikacijsku glavu koja predviđaj kategoriju objekata ili pozadinu. Osim toga, vektor prolazi kroz linearne slojeve generirajući na izlazu 4 vrijednosti koje predstavljaju koordinate okvira normalizirane u odnosu na veličinu slike.

3.3.2. Funkcije gubitka

DETR koristi tri vrste gubitka: klasifikacijski gubitak, gubitak kod predikcije okvira objekta na slici i gubitak sparivanja objekta s predikcijom [2].

Klasifikacijski gubitak je gubitak unakrsne entropiju s obzirom na kategoriju objekta. Samo ulaze u obzir objekti koji su spareni sa ispravnim okvirima detektiranih objekata na slici. [2]

Gubitak kod predikcije okvira objekta na slici se sastoji od L1 gubitka, a to je apsolutna razlika između predviđenih koordinata okvira objekta te stvarnih koordinata okvira objekta. Također uzimamo u obzir i GIoU gubitak koji pruža bolju informaciju o poboljšavanju predikcije okvira detekcije. [2]

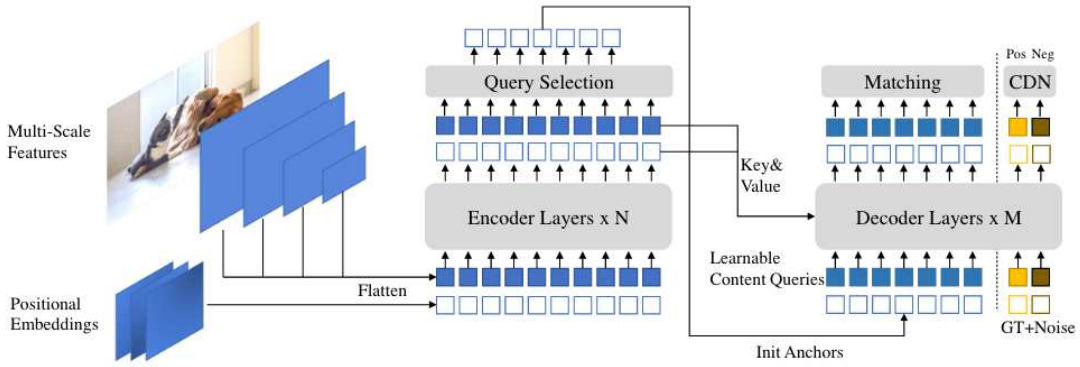
Za gubitak sparivanja objekta s predikcijom koristi se Hungarian gubitak [51]. Za svaki stvaran okvir objekta model mora pronaći predviđeni okvir objekta koji mu odgovara. Za svaki par predviđene detekcije objekta i stvarne detekcije objekta računa se gubitak kao

prethodna dva gubitka opisana. Radi se matrica od tih gubitaka te se koristi Hungarian algoritam [51] kako bi pronašli optimalne parove između predikcija i stvarnih vrijednosti.

3.4. DINO

Kao poboljšanje na DETR, predložen je model nazvan DINO [8]. Najznačajnija razlika između ova dva modela leži u koderu i dekoderu.

U DETR modelu, izlazi iz kodera služe kao pomoćne vrijednosti koje ulaze u dekoder. Na temelju tih vrijednosti dekoder generira prijedloge detektiranih objekata. S druge strane, ulazi u dekoder su učeni vektori. DINO model uvodi dodatni sloj koji iz izlaza kodera odabire *k* najboljih vektora [8]. Ti vektori se zatim koriste kao ulaz u dekoder, uz dodatne učene pozicijske vektore. Pokazalo se da ova metoda daje bolje rezultate, gdje se ulazi u dekoder ne inicijaliziraju izlazima kodera, već su ti ulazi samo učeni parametri.



Slika 5. Arhitektura DETR modela. Ovaj model poboljšava DETR dodavanjem nekoliko komponenata. Usporedbom sa Slika 4. vidi se da izlaz iz koder slojeva prolazi kroz još jedan dodatan sloj te izlazi iz tog dodatnog sloja služe kao vektori ulaza za predikciju objekata na slici korištenjem dekodera. To je najbitnija razlika između DETR i DINO modela. [8]

4. Više-medijski modeli

Opisali smo modele koji rade nad slikama ili nad tekstom, no u ovom radu ćemo se fokusirati na modele koji obrađuju i slike i tekst istovremeno te integriraju oba medija radi donošenja odluka.

Ideja više-medijskih modela često podrazumijeva postavljanje slika i teksta u isti vektorski prostor kako bismo ih mogli uspoređivati i integrirati. U tu svrhu predstavljamo prvi više-medijski model koji služi kao osnova za eksperimente u našem istraživanju, a to je CLIP [7].

4.1. Primjena više-medijskih modela za detekciju objekata

Detekcija objekata obično se provodi na zatvorenom skupu kategorija, što znači da se model trenira da prepozna samo određeni broj kategorija na slici, i ništa više. Takav pristup ograničava sposobnost modela da detektira objekte izvan prethodno definiranih kategorija ili u različitim kontekstima. Na primjer, model može biti treniran da detektira automobile, ali možda neće uspjeti prepoznati automobil u pokretu, parkiran automobil, automobil određene boje, taxi i slično.

Jedan način da se omogući detekcija objekata s otvorenim vokabularom je korištenje prirodnog jezika [6]. Prirodnim jezikom definiramo nazive objekata koji nas zanimaju, što znači da ne postoji unaprijed određena klasifikacija s N kategorija na izlazu modela, već je teoretski broj kategorija neograničen.

Često se povezivanje između teksta i slike postiže stavljanjem vektora koji predstavlja sliku i vektora koji predstavlja tekst u isti vektorski prostor. Cilj je da su ti vektori međusobno usporedivi; na primjer, želimo da vektor slike automobila i vektor teksta "auto" budu bliže u prostoru nego vektor slike stolice i vektor teksta "auto".

Za postizanje ovog cilja nije nužno koristiti *Transformere*, iako se *Transformer* pokazao kao idealna arhitektura za takve probleme. *Transformer* ima svoje nedostatke, što smo spomenuli u više navrata u radu, ali dokazano je da dobro skalira s povećanjem količine podataka [46]. To znači da što više podataka imamo, *Transformer* će bolje raditi, i može čak zaključivati stvari na kojima nije specifično učen. Na primjer, model može biti učen da prepozna zelenu

stolicu među ostalim stolicama, i to znanje može se prenijeti kako bi prepoznao zeleni automobil.

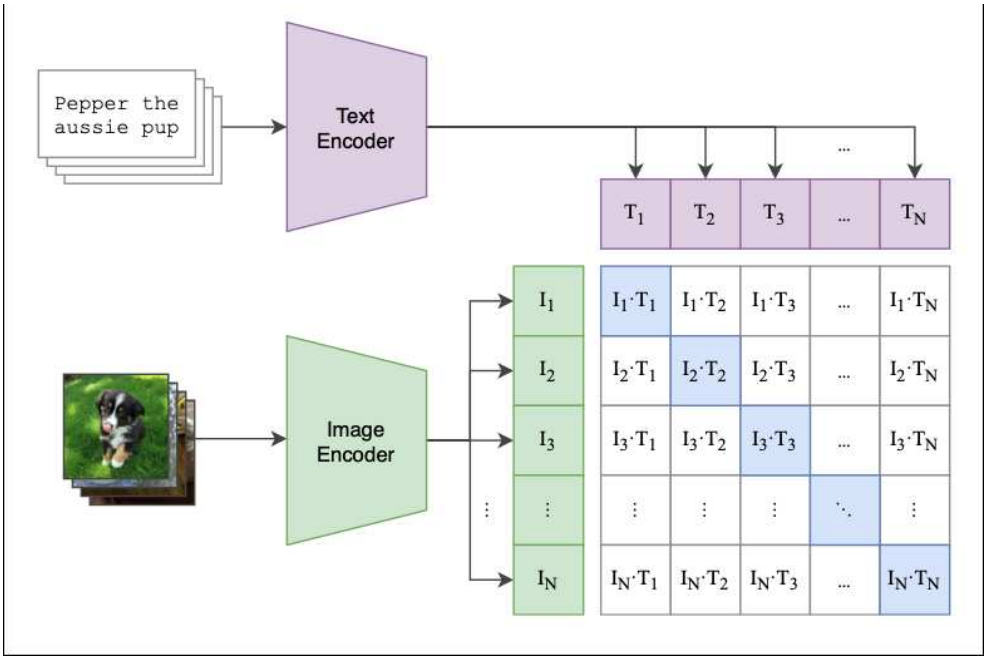
Ova ideja će bolje biti detaljnije objašnjena u opisu modela CLIP [7], koji koristi princip stavljanja teksta i slike u isti vektorski prostor kako bi mogao klasificirati slike, generirati najbolje tekstualne opise slika i obavljati druge slične zadatke.

4.2. CLIP

CLIP [7] je arhitektura sastavljena od kodera prirodnog jezika i kodera slike. Koder prirodnog jezika i koder slike generiraju vektore kao izlaz. Na taj način, tekstualni opis slike dobiva svoj vektor iz kodera, dok slika dobiva svoj vektor iz kodera za slike. Ideja iza CLIPa je da se vektori iz različitih medija nalaze u istom vektorskom prostoru i da su međusobno usporedivi.

Ovo se postiže tako što se koriste različite slike i tekstovi, te se spajaju oni parovi koji idu zajedno, a razdvajaju oni koji ne idu. Pri treniranju treniranju imamo skup slika i skup njima pripadajućih tekstova. Za svaku sliku i za svaki tekst genriramo njihove vektorske reprezentacije pomoću odgovarajućih kodera. Ti vektori se normaliziraju, nakon čega se računa kosinusna sličnost između svih vektora slika i svih vektora teksta. Na taj način dobivamo matricu *NxN* gdje svaki red predstavlja kosinusnu sličnost slike sa svim tekstovima, a svaki stupac kosinusnu sličnost teksta sa svim slikama. Računamo *softmax* svakog reda te računamo gubitak unakrsne entropije gdje su pozitivne oznake na dijagonali matrice. Nadalje, računamo i *softmax* svakog stupca te računamo gubitak unakrsne entropije opet. Time dobivamo dva gubitka: jedan je podudaranje teksta sa slikama i drugi za podudaranje slika sa tekstovima. Ukupni gubitak je prosječna vrijednost ta dva gubitka. Ovakav gubitak se često naziva kontrastni gubitak. [7]

Na ovaj način CLIP postiže to da se vektori slike i vektori teksta nalaze u istom vektorskom prostoru i međusobno su usporedivi.



Slika 6. CLIP se sastoji od kodera prirodnog jezika i kodera slike čiji su izlazi vektori. Ti vektori tvore matricu te se računa kosinusna sličnost između svakog para teksta i slike, maksimizirajući sličnost na dijagonali matrice. [7]

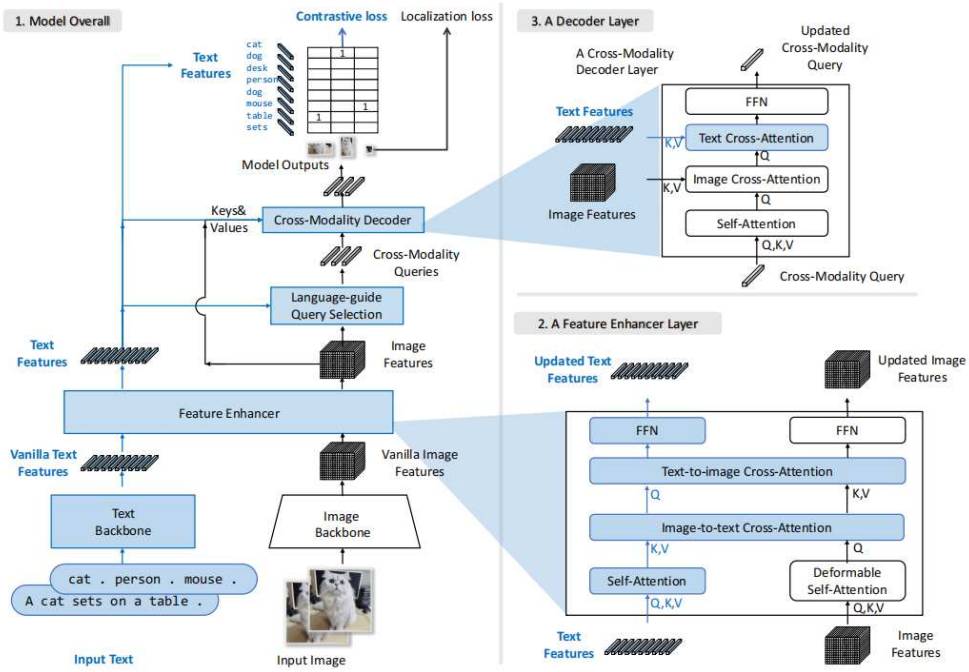
4.3. Grounding Dino

Grounding Dino [6] je model na kojem su izvedeni eksperimenti u radu, a koji je specijaliziran za detekciju objekata nad otvorenim vokabularom.

Grounding Dino je više-medijski model koji povezuje sve modele koje su do sada navedene u radu. Grounding Dino arhitektura kombinira BERTa [4] za obradu prirodnog jezika, Swin Transformera [5] za obradu slike, te princip sličan CLIPu [7] za povezivanje slike i teksta. Princip DINO modela [8] se koristi kako bi model mogao detektirati koji objekti se nalaze na slici i gdje su locirani.

Kao ulaz, Grounding Dino prima sliku i pridruženi tekst. Zadatak modela je detektirati objekte na slici koji su opisani u tekstu.

Opisati ćemo svaki dio Grounding Dino arhitekture, izlaze i ulaze u različite slojeve te funkcije tih slojeva.



Slika 7. Arhitektura Grounding Dino modela. Ulaz u model je slika i tekst. Tekst može biti ili cijela rečenica ili popis objekata za detekciju. [6]

4.3.1. Koder prirodnog jezika

Tekst može biti izražen na različite načine u kontekstu detekcije objekata. Primjeri uključuju izraz "big brown dog", cjelovitu rečenicu poput "A dog standing on the floor", ili jednostavno naziv objekta kao što je "dog". Također, tekst može sadržavati više objekata izraženih kao rečenica, npr. "A dog and birthday cakes", ili kao popis objekata koje želimo detektirati odvojene razmakom i točko, "dog . birthday cake ." [6]. Za svaki objekt naveden u tekstu, koristi BERT model kako bi se dobio njegova vektorska reprezentacija.

4.3.2. Koder slike

U radu su navedeni različiti koderi slike, ali u eksperimentima je korišten Swin Transformer, pa ćemo se fokusirati na njega. Slika ulazi u Swin Transformer te na izlazu dobivamo vektorsku reprezentaciju te slike koja se koristi dalje u modelu.

4.3.3. Pojačivač značajki

Pojačivač značajki je sloj u Grounding Dino koji se sastoji od tri vrste pažnje: pažnje, pažnje gdje tekst utječe na sliku i pažnje gdje slika utječe na tekst. Ulaz u ovaj sloj čine vektori

teksta i vektori slike. Na početku, vektori teksta prolaze kroz klasični sloj pažnje, a isto se događa i s vektorima slike, pri čemu ti vektori trenutno nemaju međusobni utjecaj.

Nakon toga, dolazimo do prvog sloja modela gdje se vektori teksta i vektori slike međusobno povezuju. Od vektora teksta stvaramo ključeve i vrijednosti, svaki kao što je opisano u poglavlju o *Transformeru* i slojevima pažnje. Od vektora slike stvaramo vektore upita. Zatim primjenjujemo pažnju na te vektore upita, ključeve i vrijednosti. Bitno je napomenuti da nije nužno da broj vektora upita odgovara broju vektora ključeva kako bi se izvršila pažnja. Kao izlaz dobivamo isti broj vektora kao i u početnom skupu vektora teksta, koji sada predstavljaju tekst obogaćen informacijama iz slike. [6]

U zadnjem dijelu ovog sloja, postupamo suprotno u odnosu na prethodni korak. Od slike stvaramo ključeve i vrijednosti, a od teksta stvaramo vektore upita. Na taj način na izlazu dobivamo vektore slike obogaćene informacijama iz teksta. [6]

Primjećujemo jedan potencijalni problem pri primjeni ovog modela za detekciju objekata, a to je da će prisutnost određenih objekata u tekstu promijeniti izlaz modela. Na primjer, za sliku čovjeka i tekst "čovjek", izlaz modela neće biti isti kao kada bismo umjesto toga naveli "čovjek . avion", jer će avion također utjecati na vektore slike. Kako bismo ublažili ovaj problem tijekom treninga, u tekst ćemo uvrstiti različite vrste objekata kako bi model naučio ignorirati objekte koji se ne pojavljuju na slici.

4.3.4. Selekcija upita vođena jezikom

Kao izlaz iz prethodnog sloja dobivamo vektore teksta obogaćene informacijama iz slike i vektore slike obogaćene informacijama iz teksta. Ideja ovog sloja je odabrati značajke iz slike koje su najviše povezane s zadanim tekstom.

Definiramo broj upita, što određuje koliko objekata možemo detektirati na slici. Izlazi iz ovog sloja su vektori koji predstavljaju moguće objekte na slici.

Prvo množimo vektore slike i vektore teksta iz prethodnog sloja te dobivamo matricu koja ima broj redaka jednak broju vektora slike i broj stupaca jednak broju vektora teksta.

Zatim za svaki redak te matrice odabiremo najveću vrijednost, što rezultira vektorom koji ima broj elemenata jednak broju vektora slike.

Dalje, iz tih vrijednosti odabiremo broj vrijednosti koji odgovara definiranom broju upita (u ovom radu, kao i u eksperimentima, koristimo 900). Svaka od tih vrijednosti predstavlja poziciju u tom vektoru, što čini izlaz ovog sloja.

4.3.5. Dekoder

Izlaz iz prethodnog sloja je vektor s 900 vrijednosti u rasponu od 0 do *N*. Ove vrijednosti pretvaramo u vektore i dodajemo im informaciju o poziciji, slično kao što je implementirano u DINO modelu.

Ovi vektori su prvi ulaz u sloj pažnje. Izlaz iz tog sloja služi kao ulaz u novi sloj pažnje, pri čemu su upiti izlazni vektori, dok su ključevi i vrijednosti dobiveni iz pojačivača značajki koji sadrže vektore slike.

Nakon toga slijedi još jedan sloj pažnje, ali s ključevima i vrijednostima dobivenim iz vektora teksta.

Izlazi iz ovog sloja su vektori koji predstavljaju potencijalne objekte na slici. Ti vektori integriraju informacije iz slike i teksta.

4.3.6. Predikcija kategorije i pozicije okvira objekta na slici

Na kraju, model mora dati predikcije o lokaciji objekta na slici i povezati ih s odgovarajućim objektom iz teksta. Za to se koristi kombinacija pristupa inspiriranih CLIP-om za usporedbu slike i teksta, te DETR-om za spajanje parova detektiranih objekata na slici sa stvarnim objektima radi dobivanja predikcija o lokacijama objekata.

Koristi se kontrastni gubitak sličan onome u CLIP-u kako bi se model naučio povezivati objekte opisane prirodnim jezikom s odgovarajućim detektiranim objektom na slici. Međutim, postoji razlika u načinu izračuna kontrastnog gubitka u ovom modelu, gdje se koristi fokalni gubitak. Ovo je potrebno zbog velikog broja predloženih objekata (900), pri čemu je broj negativnih sparivanja mnogo veći od pozitivnih. Zato se gubitak unakrsne entropije ($CE(p) = -\log(p)$) množi sa izrazom $(1-p)^{\gamma}$ gdje je γ hiperparametar. Taj izraz fokusira model da daje veće značenje težim primjerima. Kako je vjerojatnost za točnu kategoriju bliža 1 tako će taj gubitak manje pridonositi.

Ovaj pristup omogućuje detekciju objekata nad otvorenim vokabularom koji se sastoji od objekata opisanih prirodnim jezikom, što je u praksi beskonačan skup mogućih objekata.

5. Eksperimentalni dio rada

Cilj ovog rada je napraviti model za detekciju objekta nad otvorenim vokabularom sa slika prikupljenih bespilotnom letjelicom. Trenutno nije pronađen rad koji se bavi ovim specifičnim područjem, što otežava ili čak onemogućava usporedbu našeg modela s postojećim modelima koji rješavaju isti problem u istoj domeni.

Treniran je Grounding Dino nad javno dostupnim podacima prikupljenima sa interneta. Usporedit ćemo rezultate našeg modela s dostupnim modelima za detekciju objekata nad otvorenim vokabularom i s modelima za detekciju objekata nad zatvorenim vokabularom.

5.1. Postavke eksperimenta

Koristiti ćemo biblioteku MMDtection kako bi trenirali Grounding Dino na našem novom skupu podataka. MMDetection je biblioteka za učenje modela za primjenu u detekciji objekata. MMDetection već ima podržan Grounding Dino model te je preko te biblioteke moguće trenirati spomenuti model na proizvoljnom skupu podataka. Postavke za treniranje su preuzete iz rada [61] koji predstavlja metodu treniranja Grounding Dino modela.

Model će biti treniran na dvije lokalne grafičke kartice: NVIDIA RTX A6000 i GeForce RTX 3090.

5.2. Definicija modela

Grounding Dino model kojeg ćemo trenirati će biti verzija Grounding Dino Base modela iz MMDetection biblioteke. Sastoji se od: Swin Transformera, BERTa, 6 slojeva transformer kodera i 6 slojeva transformer dekodera.

Model sadrži 233 milijuna učenih parametara.

Za inicijalizaciju početnih težina koristit ćemo već trenirani model Grounding Dinoa. Taj model je treniran nad ovim skupovima podataka: GoldG [62], V3det [63], COCO2017 [41], LVISV1 [64[, COCO2014 [41], GRIT [65], RefCOCO [45], RefCOCO+ [45], RefCOCOg [45], gRefCOCO [45]. Ovaj trenirani model je izabran kako bi naš model zadržao znanje o svijetu te različitim objektima koji se pojavljuju. Želimo da naš trenirani model ne nauči samo prepoznavati kategorije na kojima ga treniramo nego i prepoznavati objekte pomoću

konteksta primjerice prepoznavati čovjeka koji hoda makar tu kategoriju nemamo u podacima za treniranje.

5.3. Podaci

Veliki problem kod detekcije objekata sa slika dobivenih bespilotnom letjelicom je taj da nemamo puno označenih podataka u toj domeni. Ne postoji ni jedan javno dostupan skup podataka kao što je to RefCOCO [45] koji, osim kategorije objekta, sadrži i detaljniji opis izgleda objekta ili kontekstualne podatke vezane za objekt u slici. Također ne postoji skup podataka koji ima velik broj kategorija kao što je to COCO [41]. Tako ćemo prikupiti što više podataka možemo dobivenih s bespilotnom letjelicom koji imaju označene objekte na slici te ćemo od tih podataka napraviti naš skup za treniranje nad otvorenim vokabularom.

Naš skup za treniranje sadrži ove skupove podataka:

- Cattle Detection and Counting in UAV Images Dataset [10]
- Bird Detection Datasets [11]
- Palm Trees Dataset [12]
- UAVOD-10 Dataset [13]
- Cherry Chevre Dataset [16]
- PUCPR Dataset [17]
- STN PLAD Dataset [18]
- UAV-Vehicle-Detection-Dataset [19]
- Carpark and License Plate Dataset [20]
- POG Dataset [21]
- Plant Detection and Counting Dataset [22]
- Aerial Cows Dataset [23]
- Quantifying the movement, behaviour and environmental context of group-living animals using drones and computer vision Dataset [24]
- Forest Damages – Larch Casebearer Dataset [25]
- Aerial Docks and Boats Dataset [26]
- VisDrone Dataset [27]
- Multispectral Potato Plants Images Dataset [28]
- Aerial Power Infrastructure Dataset [29]
- Overhead Imagery of Wind Turbines (by Duke Dataplus2020) Dataset [30]

- Multi-topography Dataset for Wind Turbine Detection Dataset [31]
- SeaDronesSee Dataset [32]
- LADD: Lacmus Drone Dataset [33]
- Ship Detection from Aerial Images Dataset [34]
- HIT-UAV Dataset [35]
- Car Parking Lot Dataset [17]
- Maize Tassel Detection Dataset [36]
- NTUT 4K Drone Photo Dataset [37]
- Traffic Drone Data – Bangladesh Dataset [38]
- MOBDrone Dataset [39]
- AFO Dataset [41]
- UAVDT Dataset [42]
- Small Object Aerial Person Detection Dataset [43]
- RefCocog [45]

Ovaj skup podataka se sastoji od 32 različita skupa podataka, različitih dimenzija slika dobivenih u različitim uvjetima, različitim visinama bespilotne letjelice i kutovima kamere. Ovaj skup ukupno sadrži više od 20,000 različitih kategorija objekata. Svi podaci su svedeni na isti format. Za svaku sliku uzete su sve kategorije objekata koje se pojavljuju na slici te je napravljena rečenica od njih gdje je svaka jedinstvena kategorija razdvojena od druge sa točkom i razmakom kao što je napravljeno u originalnom radu Grounding Dino modela. Skup podataka sadrži 291,951 označenih slika, od kojih su 103,284 označenih slika iz skupa podataka RefCocog koji ne sadrži slike s bespilotnih letjelica.

Neka imena objekata su promijenjena, a neki objekti su ignorirani kako bi dobili točnije oznake:

- AFO
    - Ignorirano: object, large_obj, small_obj
    - Promijenjeno: human > person in the water
- LADD
    - Promijenjeno: pedestrian > person
- Forest Damages

    - Obogaćena imena objekata: highly damaged larch, lightly damaged larch, healthy larch, highly damaged spruce, lightly damaged spruce, healthy spruce
    - Dodano: tree that is not larch
- Maize Tassel
    - Promijenjeno: tassels > maize tassels
- NTUT
    - walk: person walking
    - stand: person standing
    - riding: person riding
    - sit: person sitting
    - push: person pushing
    - baseball: person playing baseball
    - soccer: person playing soccer
    - watchphone: person watching phone
- Plant Detection
    - maize: maize plant
    - sugarbeet: sugarbeet plant
    - sunflower: sunflower plant
- STN PLAD
    - damper: power line damper
    - insulator: power line insulator
    - plate: power line plate
    - spacer: power line spacer
    - tower: power line tower
- UAVOD-10
    - cable-tower: cable tower
    - cultivation-mesh-cage: cultivation mash cage
    - prefabricated-house: prefabricated house
- VisDrone
    - people: person
    - bicycle: biker
    - motor: motorbike

Za evaluaciju modela rađen je skup podataka za validaciju i testiranje. Validacija i testiranje se provodila putem biblioteke MMDetection te su podaci u formatu COCO skupa podataka.

Skup za testiranje se sastoji od već zadanih skupova za testiranje od ovih skupova podataka:

- Palm Trees Dataset [12]
- Aerial Cows Dataset [23]
- Aerial Power Infrastructure Dataset [29]
- AFO Dataset [41]
- VisDrone Dataset [27]
- SeaDronesSee Dataset [32]
- Small Object Aerial Person Detection Dataset [43]

Svi ovi skupovi podataka sadrže slike s bespilotne letjelice.

5.4. Postavke treniranja

5.4.1. Učitavanje podataka

Slika se učitava i postoji 50% vjerojatnost da će se slika zrcaliti. Slučajno odabiremo jednu od ovih dimenzija slike te mijenjamo dimenziju slike:

- (480, 1333)
- (512, 1333)
- (544, 1333)
- (576, 1333)
- (608, 1333)
- (640, 1333)
- (672, 1333)
- (704, 1333)
- (736, 1333)
- (768, 1333)
- (800, 1333)

Nakon toga slučajno odabiremo jednu od ovih dimenzija i opet mijenjamo dimenzije slike:

- (400, 4200)
- (500, 4200)

- (600, 4200)

Kod zaključivanja postavljamo dimenzije slike na (800, 1333).

Nakon toga uzimamo slučajan odrezak iz slike veličine (384, 600). Onda opet mijenjamo dimenzije slike kao u prvom koraku mijenjanja dimenzija.

Ovo je procedura koja je opisana u postupku treniranja rada kojeg pratimo kod izrade svih eksperimenata [61].

Svakom tekstu dodajemo slučajan broj kategorija između 1 i 50 kategorija. Kategorije za slučajan odabir se uzimaju iz svih kategorija koje se pojavljuju u skupu podataka.

5.4.2. Hiperparametri

Koristit ćemo AdamW kao optimizator. Stopa učenja je postavljena na 0.00005 s propadanjem težina od 0.0001. Gradienti se postavljaju na maksimalnu vrijednost norme od 0.1. Stopa učenja za apsolutna pozicijska ugrađivanja i BERT su postavljena na 0, a stopa učenja za parametre *Swin Transformera* se postavlja na 0.000005.

Koristi se *MultiStepLR* gdje se na 90000 iteraciji te 120000 iteraciji stope učenja množe sa 0.1.

5.5. Rezultati

Treniran je model nad navedenim podacima te u težine prethodno inicijalizirane s težinama već spomenutog modela. Model je treniran na 150000 iteracija sa veličinom grupe od 64.

Testiranje se radi nad skupovima podataka koji su u formatu COCO skupa podataka te se provodi primjenom pycocotools biblioteke.

Skup podataka	mAP@0.1	mAP@0.3	mAP@0.5
Palms	45.40%	44.03%	35.10%
Aerial Cows	40.05%	36.53%	24.15%
API	38.27%	37.07%	34.04%
AFO	42.18%	38.62%	32.84%

SeaDroneSee	35.14%	42.8%	38.72%
VisDrone	22.40%	19.72%	11.13%
SOAP	25.02%	23.34%	17.30%

Tablica 1. Prikaz prosječne srednje vrijednosti nad različitim skupovima podataka. Prosječna srednja preciznosti sadrži i broj koji označava prag sigurnosti. Svaka predikcija sadrži i broj koji označuje koliko je model pouzdan u tu predikciju, ako je taj broj manji od praga onda ta predikcija ne ulazi u obzir.

Iz Tablica 1. vidljivo je da su najveće vrijednosti prosječne srednje preciznosti kada uzmemo prag sigurnosti od 0.1. Slijedno tim rezultatima taj prag ćemo koristiti za ostatak analize modela.

Skup podataka	mAP	mAP mali objekti	mAP srednji objekti	mAP veliki objekti	mAP, iOU=0.5	mAP, iOU=0.75
Palms	45.40%	-1	-1	45.40%	79.01%	48.16%
Aerial Cows	40.05%	27.48%	57.43%	58.31%	80.75%	34.45%
API	38.27%	27.68%	44.25%	30.20%	84.42%	26.03%
AFO	42.18%	3.2%	33.10%	50.00%	75.01%	38.42%
SeaDroneSee	35.14%	30.27%	48.61%	62.20%	78.50%	43.12%
VisDrone	22.40%	12.48%	33.92%	46.92%	37.72%	22.84%
SOAP	25.02%	21.57%	59.21%	80.40%	68.83%	12.96%

Tablica 2. Mali objekti su definirani kao površine manje od 1024 piksela, srednji objekti kao površine manje od 9216 piksela, a veće od 1024 i veliki objekti su ostali. Vrijednost je -1 ako nema objekata te veličine.

Slika 8. Slike iz VisDrone skupa za testiranje. VisDrone skup za testiranje sadrži slike iz Kine te su scene iz gradova. Označeni objekti u ovom skupu podataka su: pješak, ljudi, automobil, bus, kombi, bicikl, motocikl, tricikl, tricikl sa krovom.

Slika 9. Slike iz SOAP skupa za testiranje. SOAP skup za testiranje sadrži različite scene gdje su označeni samo ljudi.

Slika 10. Slike iz SeaDroneSee skupa za testiranje. SeaDroneSee skup za testiranje sadrži slike objekata u moru. Označeni objekti su: plutača, čovjek u vodi, brod, jetski, aparati za spašavanje života koji plutaju.

Slika 11. Slike iz Palms skupa za testiranje. Palms skup za testiranje sadrži slike gdje su označeni različito palme i ostale vrste drveća. Model ima 31% preciznost kod detekcije ostalih drveća, a 60% preciznost kod detekcije palmi.

Slika 12. Slike iz API skupa za testiranje. API skup za testiranje sadrži slike energetske infrastrukture specifično su označeni električni stupovi.

Slika 13. Slike iz AFO skupa za testiranje. AFO sadrži slike objekata koji plutaju u vodi. Označeni objekti su: brod, kajak, plutača, sup daska, čovjek u vodi i jedrilica.

Slika 14. Slike iz Aerial cows skupa za testiranje. Aerial Cows sadrži slike gdje su označene krave.

5.5.1. Usporedba sa dostupnim modelima

U trenutku pisanja ovoga rada nismo uspjeli naći već objavljene modele koji rješavaju problem detekcije objekata nad otvorenim vokabularom sa slike dobivenih bespilotnom letjelicom te zbog toga ne možemo imati jako dobru usporedbu s drugim modelima.

Model	Awn	Bic	Bus	Car	Mot	Ped	Peo	Tri	Tru	Van	mAP50	mAP0:95
SSD [53]	11.2	7.4	49.8	63.2	19.1	18.7	9.0	11.7	33.1	30.0	25.3	14.6
Cascade R-CNN [54]	8.6	7.6	34.9	54.6	21.4	22.2	14.8	14.8	21.6	31.5	23.2	13.4
RetinaNet [9]	4.2	1.4	17.8	45.5	11.8	13.0	7.9	6.3	11.5	19.9	13.9	8.1
ATSS [55]	8.5	**18.8**	52.1	76.6	41.4	42.7	22.3	**28.4**	36.9	41.4	37.3	23.0
Yolov6n [56]	8.1	3.9	23.0	72.2	29.0	28.3	23.0	15.9	21.7	33.1	27.3	15.7
DAMO-YOLO [57]	11.0	6.7	42.3	74.2	33.4	32.8	26.2	19.3	23.5	35.9	30.5	17.6
RTMDET [58]	9.5	4.3	39.6	69.6	29.0	24.6	19.7	17.9	23.4	35.7	27.3	16.1
YOLO-MS [59]	11.9	6.0	39.6	73.5	32.2	31.4	25.2	18.9	22.1	35.4	29.6	16.7
Gold-YOLO [60]	11.5	6.9	44.3	74.3	33.7	32.3	26.3	20.1	26.6	36.9	31.3	18.0
DASSF [52]	16.1	14.4	52.5	**81.0**	**45.9**	44.6	**36.8**	24.7	31.2	**44.4**	**39.6**	**23.5**
Grounding Dino	0.2	0.0	11.6	18.12	2.02	1.5	0.0	1.08	7.6	7.5	5.0	3.1
Naš model	**16.55**	17.86	**62.22**	80.94	37.23	**45.37**	18.86	17.06	**39.85**	41.3	37.7	22.4

Tablica 3. Tablica preuzeta iz rada [52] i nadopunjena našim modelom i Grounding Dino modelom. Usporedba modela na Visdrone2019 testnom skupu podataka. Naš model je najbolji u čak četiri od deset kategorija te ima drugu najveću vrijednost srednje prosječne točnosti, no treba uzeti u obzir da su ostali modeli rađeni za detekciju objekata u stvarnom vremenu dok naš model nije.

Model/Skup podataka	Palms	Aerial Cows	API	AFO	SeaDrone See	VisDrone	SOAP	COCO 2017
Grounding Dino	22.0	18.0	2.4	18.3	11.0	13.6	8.2	59.2
Naš model	45.4	40.0	38.3	42.2	45.1	22.4	25.0	41.1

Tablica 4. Prikazane su vrijednosti mAP 0:95 našeg modela i Grounding Dino modela čije su težine služile za inicijalizaciju našeg modela prije učenja.

Iz Tablica 4. uočavamo poboljšanje nad skupovima za testiranje koji sadrže slike dobivene bespilotnom letjelicom, no uočavamo poprilično pogoršanje nad skupom podataka COCO 2017. Skup podataka COCO 2017 sastoji se od 80 različitih objekata te su slike većinom slikane iz perspektive čovjeka. To nam ukazuje da se model poboljšao nad slikama iz ptičje perspektive, pogoršao se nad slikama iz ljudske nad kojima je treniran Grounding Dino.

Također primjećujemo koliko dobro Grounding Dino nad slikama sa bespilotnih letjelica. Iako nije treniran nad takvim slikama on još uvijek uspijeva detektirati objekte i iz perspektive iz koje ih možda nije vidio. Treba uočiti da puno bolje radi nad skupovima podataka koji imaju neke česte objekte kao što su krave i automobili dok loše radi nad slikama koje imaju možda specifične objekte za ovu domenu, električni stupovi, osobe i oprema koji se nalaze u vodi te jako mali objekti.

5.5.2. Rezultati nad slikama s termalne kamere

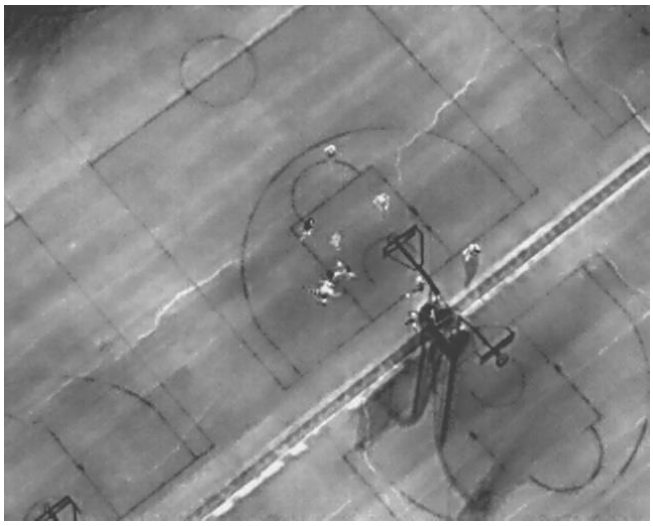
Jedan od često korištenih senzora kod bespilotnih letjelica je termalna kamera. Testirali smo koliko model dobro radi termalnim slikama kada ih nema u skupu podataka za treniranje i kada se dodaju termalne slike u skup podataka za treniranje.

Dodano je 2008 slika u skup podataka za treniranje što čini samo 0.68% skupa podataka za treniranje.

Model koji nije treniran sa skupom podataka Hit-UAV postiže prosječnu preciznost od 5.9%.

Model koji je treniran sa skupom podataka Hit-UAV postiže prosječnu preciznost od 38.4%.

Ovo je indikacija da nam nije potrebna velika količina podataka kako bi se model poboljšao u nekoj potpuno drugoj domeni.



Slika 15. Primjer slike iz skupa podataka Hit-UAV.

5.6. Detekcija objekata uz dodatni kontekst

Uz detekciju objekata po imenu model bi trebao i detektirati objekte uz dodatni opis. Primjerice detektirati plavi automobil, čovjeka s kapom, čovjeka koji vozi motor i tako dalje.

Model bi trebao odvajati slične objekte, kao što su to ljudi bez kape i ljudi sa kapom. Trenutno nemamo način kako testirati detekciju ovakvih deskriptivnih objekata jer za to ne postoji skup podataka nego ćemo prikazati primjere ovakvog korištenja modela.

Slika 16. Lijeva slika prikazuje detektirane objekte kod unosa: „red car .“. Srednja slika prikazuje detektirane objekte kod unosa: „blue car .“. Desna slika pokazuje detektirane objekte kod unosa „orange car .“ Iz sve 3 slike vidimo da je model ispravno odvojio automobile određene boje s obzirom na ostale automobile.

Slika 17. Prikaz detekcije objekata kada je unos „orange car . red car . blue car.“. Jedan od problema koje ovaj model ima je nekonzistentno ponašanje kada se doda još kategorija u unos. Odjednom model prepoznaje druge automobile, a izostavljala predikcije koje je prije imao o narančastom i crvenom automobilu.

Također nailazimo na problem zaboravljanja kod dodatno treniranog modela. Kao što smo opisali u postupku treniranja, trenirali smo model i na RefCocog skupu podataka. To je jedan od originalnih skupova podataka nad kojim je treniran temeljni model te taj skup podataka sadrži jako deskriptivne kategorije objekata. Objekti nisu opisani samo imenom nego i određenim posebnim karakteristikama poput položaja naspram drugih objekata, položaja u prostoru, posebnih fizičkih karakteristika i tako dalje.

Ne dodavanjem ovog skupa podataka primjećujemo da model radi puno lošije nad deskriptivnim objektima te zapravo zaboravlja to znanje koje je dobio trenirajući nad skupovima podataka kao što su RefCocog. Tako dakle model postaje bolji u detektiranju objekata iz ptičje perspektive, ali takav model ne zadovoljava u performansama što se tiče detektiranja objekata s deskripcijom. Ovaj zaključak se temelji na testiranju autora te trenutno nemamo konkretan dokaz.

Zaključak

Detekcija objekata nad otvorenom vokabularu novija je grana računalnog vida koja nudi fleksibilnije modele koje je moguće koristiti u više situacija za razliku od klasičnih modela računalnog vida koji su trenirani prepoznavati objekte nad nekim zatvorenim skupom.

Rezultat ovoga rada je model treniran na domeni koja nije dovoljno istražena u području detekcije objekata nad otvorenim vokabularom, a to su slike dobivene bespilotnim letjelicama. Rezultat pokazuje da ovakav pristup detekciji objekata može dati dobre rezultate nad klasičnom detekcijom objekata, no ovaj rezultat indicira i na druge sposobnosti ovoga modela koje modeli trenirani nad zatvorenim skupom kategorija objekata nemaju.

Ovakvi modeli mogu detektirati iste objekte različitih karakteristika te odvojiti takve objekte jedno od drugih po fizičkim karakteristikama bez da su direktno trenirani nad skupom podataka koji ima takve objekte označene odvojeno. Primjer ovoga bi bila detekcija automobila različitih boja.

Također model pokazuje dobre rezultate na prilagodbe drugim domenama. Sa samo dodatkom samo 0.68% novih slika skupu za treniranje poboljšali smo prosječnu preciznost modela za 34.5% nad skupom podataka koji sadrži termalne slike.

Literatura

[1] Attention Is All You Need, Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin, 2023.

[2] End-to-End Object Detection with Transformers, Nicolas Carion and Francisco Massa and Gabriel Synnaeve and Nicolas Usunier and Alexander Kirillov and Sergey Zagoruyko, 2020.

[3] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, Alexey Dosovitskiy and Lucas Beyer and Alexander Kolesnikov and Dirk Weissenborn and Xiaohua Zhai and Thomas Unterthiner and Mostafa Dehghani and Matthias Minderer and Georg Heigold and Sylvain Gelly and Jakob Uszkoreit and Neil Houlsby, 2021.

[4] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova, 2019.

[5] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, Ze Liu and Yutong Lin and Yue Cao and Han Hu and Yixuan Wei and Zheng Zhang and Stephen Lin and Baining Guo, 2021.

[6] Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection, Shilong Liu and Zhaoyang Zeng and Tianhe Ren and Feng Li and Hao Zhang and Jie Yang and Chunyuan Li and Jianwei Yang and Hang Su and Jun Zhu and Lei Zhang, 2023.

[7] Learning Transferable Visual Models From Natural Language Supervision, Alec Radford and Jong Wook Kim and Chris Hallacy and Aditya Ramesh and Gabriel Goh and Sandhini Agarwal and Girish Sastry and Amanda Askell and Pamela Mishkin and Jack Clark and Gretchen Krueger and Ilya Sutskever, 2021.

[8] DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, Hao Zhang and Feng Li and Shilong Liu and Lei Zhang and Hang Su and Jun Zhu and Lionel M. Ni and Heung-Yeung Shum, 2022.

[9] Focal Loss for Dense Object Detection, Tsung-Yi Lin and Priya Goyal and Ross Girshick and Kaiming He and Piotr Dollár, 2018.

[10] Cattle detection and counting in UAV images based on convolutional neural networks, Wen Shao, Rei Kawakami, Ryota Yoshihashi, Shaodi You, Hidemichi Kawase, Takeshi Naemura International Journal of Remote Sensing, 41:1, 31-52, 2020.

[11] A global model of bird detection in high resolution airborne images using computer vision, Ben Weinstein, Lindsey Garner, Vienna R. Saccomanno, Ashley Steinkraus, Andrew Ortega, Kristen Brush, Glenda Yenni, Ann E. McKellar, Rowan Converse, Christopher D. Lipitt, Alex Wegmann, Nick D. Holmes, Alice J. Edney, Tom Hart, Mark J. Jessopp, Rohan Clarke, Dominik Markowski, Henry Senyondo, Ryan Dotson, … S.K Morgan Ernest, 2021.

[12] Palm Trees, Adel Ammar and Anis Koubaa, 2020.

[13] A context-scale-aware detector and a new benchmark for remote sensing small weak object detection in unmanned aerial vehicle images, Wei Han and Jun Li and Sheng Wang and Yi Wang and Jining Yan and Runyu Fan and Xiaohan Zhang and Lizhe Wang, 2022.

[14] A survey on methods of small weak object detection in optical high-resolution remote sensing images, Wei Han and Jia chen and Lizhe Wang and Ruyi Feng and Fengpeng Li and Lin Wu and Tian Tian and Jining Yan, 2021.

[15] Improving Training Instance Quality in Aerial Image Object Detection with A Sampling-balance based Multi-stage Network, Wei Han and Runyu Fan and Lizhe Wang and Ruyi Feng and Fengpeng Li and Ze Deng and Xiaodao Chen, 2020

[16] CherryChèvre: A Fine-Grained Dataset for Goat Detection in Natural Environments, Vayssade, Jehan-Antoine, 2023.

[17] Drone-based Object Counting by Spatially Regularized Regional Proposal Networks, Meng-Ru Hsieh, Yen-Liang Lin, Winston H. Hsu, 2017.

[18] STN PLAD: A Dataset for Multi-Size Power Line Assets Detection in High-Resolution UAV Images, André Luiz Buarque and de Castro Felix, Heitor and de Menezes Chaves, Thiago and Simões, Francisco Paulo Magalhães and Teichrieb, Veronica and dos Santos, Michel Mozinho and da Cunha Santiago, Hemir and Sgotti, Virginia Adélia Cordeiro and Neto, Henrique Baptista Duffles Teixeira Lott, 2021.

[19] Orientation-and Scale-Invariant Multi-Vehicle Detection and Tracking from Unmanned Aerial Videos, Wang, Jie and Simeonova, Sandra and Shahbazi, Mozhdeh, 2019.

[20] Carpark and License Plate Dataset, Chung Yi Lai, 2024.

[21] POG: People On Grass Dataset, Benjamin Kiefer and Martin Messmer and Leon Varga, 2023.

[22] Plant detection and counting from high-resolution RGB images acquired from UAVs: comparison between deep-learning and handcrafted methods with application to maize, sugar beet, and sunflower crops, David, Etienne, 2021.

[23] aerial cows Dataset, Roboflow 100, 2023.

[24] Quantifying the movement, behaviour and environmental context of group-living animals using drones and computer vision. Journal of Animal Ecology, 92, 1357–1371, Koger, B., Deshpande, A., Kerby, J. T., Graving, J. M., Costelloe, B. R., & Couzin, I. D., 2023.

[25] Forest Damages – Larch Casebearer 1.0, Swedish Forest Agency, 2021.

[26] Aerial Docks and Boats Dataset, Team Roboflow, 2021.

[27] Detection and tracking meet drones challenge, Zhu, Pengfei and Wen, Longyin and Du, Dawei and Bian, Xiao and Fan, Heng and Hu, Qinghua and Ling, Haibin, 2021.

[28] A Dataset of Multispectral Potato Plants Images, Sujata Butte and Aleksandar Vakanski and Kasia Duellman and Haotian Wang and Amin Mirkouei, 2021.

[29] Aerial Power Infrastructure Detection Dataset, Antonis Savva and Rafael Makrigiorgis and Panayiotis Kolios and Christos Kyrkou, 2023.

[30] Overhead Imagery of Wind Turbines, Duke Dataplus2020, 2020.

[31] Multi-topography dataset for wind turbine detection from remote sensi, Hao Mi, 2023.

[32] Seadronessee: A maritime benchmark for detecting humans in open water, Varga, Leon Amadeus and Kiefer, Benjamin and Messmer, Martin and Zell, Andreas, 2022.

[33] LADD: Lacmus Drone Dataset, Mikhail Shuranov and Denis Shurenkov and Dmitry Ruzhitsky and Victoria Martynova and Ekaterina Bykova and Georgy Perevozchikov, 2023.

[34] Ship Detection from Aerial Images, Andrew Maranhão, 2020.

[35] HIT-UAV: A High-altitude Infrared Thermal Dataset for Unmanned Aerial Vehicle-based Object Detection Suo, Jiashun and Wang, Tianyi and Zhang, Xingzhou and Chen, Haiyang and Zhou, Wei and Shi, Weisong, 2023.

[36] Maize tassel detection from UAV imagery using deep learning, Shi, Yeyin and Alzadjali, Aziza and Alali, Mohammed and Veeranampalayam-Sivakumar, Arun-Narenthiran and Deogun, Jitender and Scott, Stephen and Schnable, James, 2021.

[37] NTUT 4K Drone Photo Dataset for Human Detection, Kuan-Ting (K. T.) Lai, 2023.

[38] Traffic Drone Data - Bangladesh Dataset, Raiyaan Abdullah Public, 2023.

[39] MOBDrone: a Drone Video Dataset for Man OverBoard Rescue, Donato Cafarelli and Luca Ciampi and Lucia Vadicamo and Claudio Gennaro and Andrea Berton and Marco Paterni and Chiara Benvenuti and Mirko Passera and Fabrizio Falchi, 2022.

[40] MOBDrone: a large-scale drone-view dataset for man overboard detection, Donato Cafarelli and Luca Ciampi and Lucia Vadicamo and Claudio Gennaro and Andrea Berton and Marco Paterni and Chiara Benvenuti and Mirko Passera and Fabrizio Falchi, 2022.

[41] An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance, Gąsienica-Józkowy, Jan and Knapik, Mateusz and Cyganek, Boguslaw, 2021.

[42] UAVDT Dataset, Dawei Du and Yuankai Qi and Hongyang Yu and Yifan Yang and Kaiwen Duan and Guorong Li and Weigang Zhang and Qingming Huang and Qi Tian, 2018.

[43] Small Object Aerial Person Detection Dataset, Rafael Makrigiorgis, Christos Kyrkou, & Panayiotis Kolios, 2023

[44] Microsoft COCO: Common Objects in Context, Tsung-Yi Lin and Michael Maire and Serge Belongie and Lubomir Bourdev and Ross Girshick and James Hays and Pietro Perona and Deva Ramanan and C. Lawrence Zitnick and Piotr Dollár, 2015.

[45] ReferItGame: Referring to Objects in Photographs of Natural Scenes, Kazemzadeh, Sahar, et al., 2014.

[46] Prepoznavanje slojeva primjenom slojeva pažnje, Bruno Ćorić, 2021.

[47] Fast WordPiece Tokenization, Xinying Song and Alex Salcianu and Yang Song and Dave Dopson and Denny Zhou, 2021.

[48] Long short-term memory. Sepp Hochreiter i Jürgen Schmidhuber. 1997.

[49] Gemini: A Family of Highly Capable Multimodal Models, Gemini Team at al., 2024.

[50] Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions,5 Gokul Yenduri and Ramalingam M and Chemmalar Selvi G and Supriya Y and Gautam Srivastava and Praveen Kumar Reddy Maddikunta and Deepti Raj G and Rutvij H Jhaveri and Prabadevi B and Weizheng Wang and Athanasios V. Vasilakos and Thippa Reddy Gadekallu, 2023.

[51] The hungarian method for the assignment problem, Kuhn, H.W., 1955.

[52] DASSF: Dynamic-Attention Scale-Sequence Fusion for Aerial Object Detection, Haodong Li and Haicheng Qu, 2024.

[53] Ssd: Single shot multibox detector, Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, 2016.

[54] Cascade r-cnn: Delving into high quality object detection, Zhaowei Cai and Nuno Vasconcelos, 2018.

[55] Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li, 2020.

[56] Yolov6 v3. 0: A full-scale reloading, Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu, 2023.

[57] Damoyolo: A report on real-time object detection design, Xianzhe Xu, Yiqi Jiang, Weihua Chen, Yilun Huang, Yuan Zhang, and Xiuyu Sun, 2022.

[58] Rtmdet: An empirical study of designing real-time object detectors, Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen, 2022.

[59] Yolo-ms: rethinking multi-scale representation learning for real-time object detection, Yuming Chen, Xinbin Yuan, Ruiqi Wu, Jiabao Wang, Qibin Hou, and Ming-Ming Cheng, 2023.

[60] Gold-yolo: Efficient object detector via gather-and-distribute mechanism, Chengcheng Wang, Wei He, Ying Nie, Jianyuan Guo, Chuanjian Liu, Yunhe Wang, and Kai Han, 2024.

[61] An Open and Comprehensive Pipeline for Unified Object Grounding and Detection, Xiangyu Zhao and Yicheng Chen and Shilin Xu and Xiangtai Li and Xinjiang Wang and Yining Li and Haian Huang, 2024.

[62] V3Det: Vast Vocabulary Visual Detection Dataset, Jiaqi Wang and Pan Zhang and Tao Chu and Yuhang Cao and Yujie Zhou and Tong Wu and Bin Wang and Conghui He and Dahua Lin, 2023.

[63] LVIS: A Dataset for Large Vocabulary Instance Segmentation, Agrim Gupta and Piotr Dollár and Ross Girshick, 2019.

[64] A Spoken Language Dataset of Descriptions for Speech-Based Grounded Language Learning, Gaoussou Youssouf Kebe and Padraig Higgins and Patrick Jenkins and Kasra Darvish and Rishabh Sachdeva and Ryan Barron and John Winder and Donald Engel and Edward Raff and Francis Ferraro and Cynthia Matuszek, 2021.

[65] GRIT: General Robust Image Task Benchmark, Tanmay Gupta and Ryan Marten and Aniruddha Kembhavi and Derek Hoiem, 2022.

Sažetak

Detekcija objekata nad otvorenim vokabularom otvara nova područja za korištenje računalnog vida. Klasični pristupi za detekciju objekata se sastoje od treniranja modela modela računalnog vida nad zatvorenim skupom kategorija te takvi modeli nisu fleksibilni što se tiče korištenja u nove svrhe. Ovakvim pristupom broj kategorija koje model može detektirati je beskonačan te model ne mora u fazi treninga biti učen na određenoj kategoriji objekata kako bi ih znao prepoznati. U ovom radu trenirali smo model nad samostalno napravljenim skupom podataka koji sadrži slike s bespilotnih letjelica. Prikazani su rezultati modela nad skupom podataka VisDrone2019 u usporedbi sa drugim modelima koji su samo trenirani kako bi radili nad VisDrone2019 skupu podataka. Također je prikazano korištenje ovakvih modela nad deskriptivnom kategorijama objekata te prilagođavanje ovako treniranog modela potpuno drugim domenama.

Ključne riječi: transformer, detekcija objekata, bespilotne letjelice

Summary

Object detection with open vocabulary opens new areas for the use of computer vision. Classical approaches to object detection involve training computer vision models on a closed set of categories, and such models are not flexible for new purposes. With this approach, the number of categories that the model can detect is infinite, and the model does not need to be trained on a specific category of objects during the training phase to be able to recognize them. In this paper, we trained a model on a custom-made dataset containing images from unmanned aerial vehicles. The results of the model on the VisDrone2019 dataset are presented and compared with other models that were only trained to work on the VisDrone2019 dataset. The use of such models with descriptive categories of objects is also demonstrated, as well as the adaptation of a model to completely different domains.

Keywords: transformer, object detection, unmanned aerial vehicles

Skraćenice

ViT *Vision Transformer* *Transformer* arhitektura za računalni vid

BERT *Bidirectional Encoder Representations from Transformers* *Transformer* model za kodiranje prirodnog jezika

DETR *DEtection TRansformer* *Transformer* arhitektura za detekciju objekata

CLIP *Contrastive Language-Image Pre-Training model* Model koji prepoznaje tekst koji je najpovezaniji sa slikom

DINO *DETR with improved denoising anchor boxes* Arhitekutra za detekciju objekata građena nad DETR pristupom