S V E U Č I L I Š T E U S P L I T U

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

POSLIJEDIPLOMSKI DOKTORSKI STUDIJ ELEKTROTEHNIKE I INFORMACIJSKE TEHNOLOGIJE

KVALIFIKACIJSKI ISPIT

PRIMJENA METODA DUBOKOG UČENJA ZA DETEKCIJU I PRAĆENJE PLOVILA NA VIDEOZAPISIMA

Ivana Marin

Split, travanj 2024.

Sadržaj

1. Uvod

Automatska detekcija i praćenje plovila ključne su komponente različitih pomorskih sustava. One omogućuju učinkovit nadzor i upravljanje pomorskim prometom, doprinoseći istovremeno sigurnosti na moru, očuvanju ekološke ravnoteže te poboljšanju učinkovitosti logističkih operacija. Osim navedenog, automatska detekcija i praćenje plovila omogućuju brže reakcije u hitnim situacijama poput nesreća, ilegalnih aktivnosti i kršenja pomorskih zakona. Tradicionalne metode praćenja često ovise o ljudskoj intervenciji, što može biti skupo i neefikasno. Složenost pomorskog okruženja otežava ljudima da se usredotoče na videozapise tijekom duljeg vremenskog razdoblja, čime njihova prosudba može postati nepouzdana. Automatizacija ovih procesa omogućila bi efikasno i precizno praćenje plovila koje ne zahtijeva ljudsku kontrolu ili barem uvelike smanjuje potrebu za njom.

Klasične metode za detekciju objekata su se posljednjih godina pokazale inferiorne u usporedbi s detektorima baziranim na dubokim konvolucijskim mrežama, koji postižu zavidne rezultate na različitim zadacima računalnog vida, uključujući i detekciju plovila. Duboki modeli pokazali su se robusnijima i preciznijima od klasičnih. Oni uče značajke direktno iz sirovih podataka te ne iziskuju "ručni" odabir i konstrukciju prikladnih značajki poput klasičnih modela. Međutim, za treniranje dubokih modela potrebna je jako velika količina označenih podataka koju za određene praktične primjene ponekad nije jednostavno pribaviti, ili to uopće nije moguće. Zbog toga se u praksi uglavnom koriste duboki modeli koji su prethodno trenirani na velikim skupovima podataka i naknadno prilagođeni konkretnom zadatku od interesa.

Detekcija objekata jedan je od ključnih koraka u algoritmima za praćanje. No, za razliku od detektora koji pravokutnim graničnim okvirima samo lociraju objekte pojedine klase u okvirima videozapisa, algoritmi za praćenje dodatno svakom graničnom okviru pridjeljuju i jedinstveni identifikator. Time je omogućena distinkcija različitih objekata iste klase. Na primjer, u okviru videozapisa nadzorne kamere neke pomorske luke, algoritam za detekciju locirati će sve plovila i označiti ih labelom *"plovilo"*. Algoritam za praćenje dodijeljuje svakom plovilu i odgovarajući identifikator uz oznaku klase: *"plovilo 1"*, *"plovilo 2"*, ..., *"plovilo n"*. Sustav sada može razlikovati plovila, što omogućuje analizu njihovog kretanja. Nadalje, temeljem prethodnih zapažanja, sustav je u mogućnosti predvidjeti kretanje pojedinih plovila u budućnosti, što je ključno za prevenciju sudara, identifikaciju prijetnji i optimizaciju pomorskog prometa.

Ostatak rada strukturiran je na sljedeći način. Poglavlje 2 uvodi osnovne pojmove vezane za konvolucijske neuronske mreže i detekciju objekata te daje pregled popularnih detektora koji se temelje na dubokom učenju. U Poglavlju 3 detaljno su opisani koraci koji čine osnovu većine algoritama praćenja, obuhvaćen je pregled popularnih algoritama te su predstavljene metrike koje se koriste za njihovu evaluaciju. Nadalje, primjena metoda dubokog učenja razložena je i analizirana po koracima algoritama praćenja, ali i općenito. Konačno, Poglavlje 4 sadrži pregled dostupnih skupova podataka iz pomorskih okruženja koji uključuju slike i/ili videozapise plovila i relevantnih radova iz područja detekcije i praćenja plovila.

2. Detekcija objekata

Detekcija objekata važan je zadatak iz domene računalnog vida koji uključuje lokalizaciju svih objekata od interesa na digitalnoj slici i njihovu klasifikaciju u jednu od predefiniranih kategorija. Osnovno pitanje na koje se pokušava dati odgovor je: *Koji objekti se nalaze na slici i gdje se oni nalaze?* Lokacija pojedinog objekta na slici precizira se pravokutnim graničnim okvirom (engl. *bounding box*) najčešće reprezentiranim uređenom četvorkom sljedećih formata: (a) PASCAL VOC $(x_{min}, y_{\min}, x_{\max}, y_{\max})$, (b) COCO $(x_{\min}, y_{\min}, w, h)$, (c) YOLO $(\frac{x_c}{w_{img}}, \frac{y_c}{h_{img}}, \frac{w}{w_{img}}, \frac{h}{h_{img}})$, pri čemu su x_{min} i y_{min} koordinate gornjeg-lijevog vrha graničnog okvira, x_{max} i y_{max} koordinate donjeg-desnog vrha graničnog okvira, x_c i y_c koordinate središta graničnog okvira, w i h širina i visina graničnog okvira, w_{img} i h_{img} širina i visina slike. Algoritam za detekciju kao ulaz ima digitalnu sliku, a kao izlaz detekcije objekata od interesa na toj slici (ako ih ima) pri čemu se svaka detekcija opisuje s tri atributa: klasom kojoj detektirani objekt pripada, graničnim okvirom koji lokalizira objekt na slici te pouzdanošću (engl. *confidence score*) detektora u dano predviđanje[1].

2.1. Tradicionalan pristup detekciji objekata

Detekcija objekata u tradicionalnim pristupima obično uključuje tri osnovne faze: predlaganje regija koje bi mogle sadržavati objekte, ekstrakciju značajki i klasifikaciju [1]. U prvoj fazi, koja obuhvaća predlaganje kandidatnih regija, najčešće se koriste pomični prozori (engl. *sliding windows*), pravokutni prozori fiksne visine i širine kojima se horizontalno i vertikalno prolazi po slici koristeći unaprijed definiranu veličinu koraka. Nakon toga, svaka kandidatna regija klasificira se primjenom algoritama poput stroja potpornih vektora [2] ili AdaBoost [3] algoritma, kako bi se utvrdilo sadrži li objekt od interesa i, ako da, kojoj klasi pripada. Umjesto originalnih vrijednosti piksela, za uspješnu klasifikaciju koriste se vektori ručno dizajniranih (engl. *hand-crafted*) značajki poput HOG [4], SIFT [5], SURF [6] i Haarovih značajki [7], koji kodiraju semantička i vizualna svojstva objekata. Budući da algoritam treba detektirati objekte različitih veličina, ovaj postupak potrebno je ponavljati s pomičnim prozorima različitih veličina ili, u slučaju fiksnog pomičnog prozora, sa skaliranim verzijama originalne slike (tzv. piramida slika) [8]. Ilustracija tradicionalnog pristupa detekciji

[1] Pouzdanost detektora u dano predviđanje se najčešće prikazuje kao realan broj između 0 i 1.

objekata prikazana je na Slici 2.1.

Slika 2.1: Tradicionalan pristup detekciji objekata.

S vremenom su postala evidentna višestruka ograničenja tradicionalnog pristupa detekciji objekata [9]. Tradicionalan pristup detekciji pomoću pomičnih prozora računalno je zahtjevan te zbog preklapanja velikog broja prozora dolazi do redundantnosti i značajnog broja lažno pozitivnih detekcija. Nadalje, ručno dizajnirane značajke obično su usmjerene na jednostavne vizualne obrasce te nisu dovoljno robusne na promjene osvjetljenja i pozadine slike, kao ni na varijacije unutar iste klase objekata [10]. Dodatno, njihovo dizajniranje zahtijeva stručno znanje i podložno je ljudskim pogreškama. Moderni algoritmi za detekciju objekata stoga se okreću metodama dubokog učenja, zamjenjujući ručno dizajnirane značajke značajkama dobivenim dubokim konvolucijskim neuronskim mrežama [11, 12] koje automatski kodiraju složene semantičke informacije te su se pokazale znatno robusnijima od ručno dizajniranih značajki [10]. Detektori bazirani na dubokom učenju također omogućuju optimizaciju svih faza detekcije istovremeno, za razliku od tradicionalnih detektora u kojima se svaka faza oblikuje i optimizra zasebno [9].

2.2. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (engl. *Convolutional Neural Networks*, CNNs) [11] jedan su od ključnih pokretača razvoja i uspjeha metoda dubokog učenja u radu s vizualnim podacima poput slika i videozapisa. Arhitektura konvolucijskih neuronskih mreža inspirirana je receptivnim poljima neurona u vizualnom korteksu životinja [13, 14] te je dizajnirana tako da se značajke uče hijerarhijski komponirajući jednostavnije značajke u one složenije. Konvolucijska mreža sastoji se od niza slojeva koji transformiraju dobiveni ulaz i dobivene vrijednosti prosljeđuju naprijed višim slojevima sve do posljednjeg sloja mreže. Trodimenzionalni ulazi i izlazi pojedinih slojeva konvolucijske mreže često se nazivaju *volumenima*. Osnovne vrste slojeva koji se javljaju u konvolucijskim mrežama su konvolucijski slojevi, slojevi sažimanja i potpuno povezani slojevi.

Konvolucijski slojevi računaju mape značajki pomicanjem konvolucijskih filtera horizontalno i vertikalno duž ulaznog volumena koristeći unaprijed definirani korak te računanjem skalarnih produkata lokalnog dijela ulaza i težina filtera. Svaki filter konvolucijskog sloja producira jednu, odgovarajuću mapu značajki. Niži konvolucijski slojevi izdvajaju generičke značajke poput linija i rubova, dok oni viši kodiraju složenije vizualne značajke. Nelinearna

aktivacijska funkcija, obično *ReLU* (engl. *Rectified Linear Unit*) [15], se zatim primjenjuje na elemente dobivenih mapa značajki kako bi se uvela nelinearnost poželjna za detekciju nelinearnih značajki [16]. U jednom konvolucijskom sloju se primjenjuje više različitih filtera kako bi se omogućila detekcija različitih značajki u ulazu. Izlaz konvolucijskog sloja je volumen nastao slaganjem mapa značajki dobivenih različitim filterima.

Slojevi sažimanja (engl. *pooling* layers) smanjuju visinu i širinu ulaza zamjenjujući manja kvadratna područja u ulaznim mapama njihovom maksimalnom (*Max-Pooling sloj*) ili srednjom (*Average-Pooling sloj*) vrijednošću kako bi se postigla invarijantnost na male pomake i deformacije u ulazu. Obično se prvi dio konvolucijske mreže sastoji od blokova koji sadrže jedan ili više konvolucijskih slojeva nakon kojih slijedi sloj sažimanja, dok se potpuno povezani slojevi dodaju pri samom kraju mreže.

2.3. Učenje prijenosom znanja

Treniranje dubokih neuronskih mreža zahtjeva velike količine *označenih* podataka. Prikupljanje dovoljno velikog i raznolikog skupa podataka potrebnog za treniranje neuronske mreže je "skupo", a ponekad nije niti izvedivo. Ono također iziskuje znatne računske resurse te u slučaju kompleksnijih zadataka može trajati danima, čak i tjednima. Uobičajen pristup treniranju neuronske mreže u slučaju kada na raspolaganju imamo samo ograničen skup podataka za treniranje ili ograničene resurse za treniranje dubokih modela je učenje prijenosom znanja (engl. *transfer learning*).

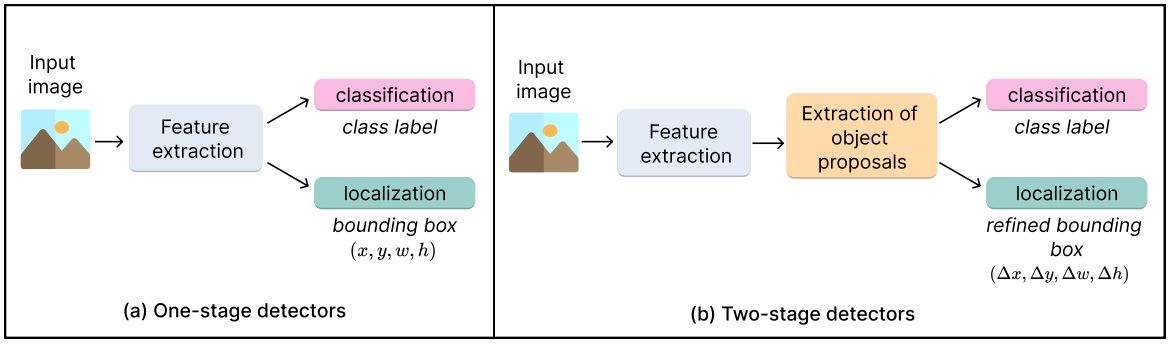
Ljudi imaju uređenu sposobnost "prijenosa" naučenog znanja iz jedne domene u drugu. *Transfer learning* se zasniva na sličnoj ideji; znanje, odnosno značajke, koje je neuronska mreža naučila na jednom zadatku koriste se za rješavanje nekog novog (sličnog) zadatka što u konačnici rezultira bržim i efikasnijim procesom učenja. Za prijenos znanja najčešće se koristi neuronska mreža predtrenirana na jako velikom skupu podataka poput klasifikatora treniranog na ImageNet [17] skupu podataka s 1.2 milijuna slika iz 1000 različitih kategorija ili detektora terninranog na COCO [18] skupu podataka s 2.5 milijuna označenih objekata (na 328 tisuća slika) među kojima je 91 vrsta objekata. Težine modela predtreniranog na velikom skupu podataka mogu se iskoristiti za incijalizaciju težina novog modela. Za vrijeme treniranja te težine se ažuriraju propagirajući pogreške mreže na zadatku od interesa natrag kroz mrežu kako bi se model prilagodio novom zadatku. Primjena učenja prijenosom znanja dala je rezultate na različitim zadacima računalnog vida u različitim domenama kao što su primjerice medicinska dijagnostika [19], daljinska istraživanja (engl. *remote sensing*) [20] te ekologija [21].

2.4. Algoritmi dubokog učenja za detekciju objekata

Popularni algoritmi za detekciju objekata mogu se grubo podijeliti u dvije kategorije: detektori koji objekte detektiraju u dvije faze (engl. *two-stage detectors*) i detektori koji objekte detektiraju u jednoj fazi (engl. *one-stage detectors*). Grafički prikaz dvije navedene vrste detektora dan je na Slici 2.2.

(a) **Detektori koji objekte detektiraju u dvije faze**
U prvoj fazi se iz danog ulaza generiraju područja od interesa (engl. *Region of Interest*, *RoI*). Nakon toga, u drugoj fazi, generirana područja od interesa se klasificiraju u jednu od unaprijed definiranih kategorija uključujući i pozadinu, te se regresijom dodatno korigiraju predloženi granični okviri.

(b) **Detektori koji objekte detektiraju u jednoj fazi**
Izostavlja se ekstrakciju područja od interesa te se granične okviri i klase objekata predviđaju direktno iz ulaza.



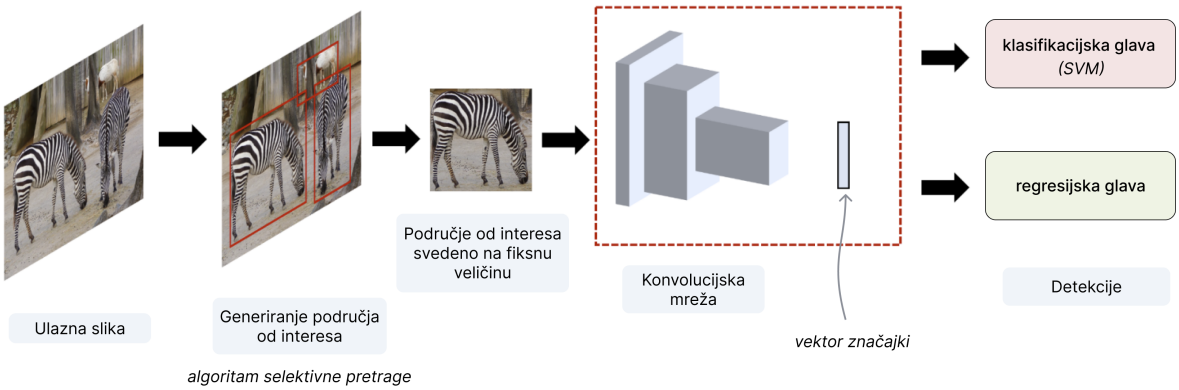
Slika 2.2: Ilustracija detektora koji objekte detektiraju u jednoj (a) i dvije (b) faze.

Dok detektori u dvije faze stavljaju naglasak na točnost detekcije modela, detektori koji objekte detektiraju u jednoj fazi prioritiziraju jednostavnost i brzinu izvršavanja nauštrb točnosti. U kontekstu detektora, pojam "okosnica" (engl. *"backbone"*) često se koristi za konvolucijsku neuronsku mrežu koje se koristi za ekstrakciju značajki iz ulazne slike.

2.4.1. Detekcija objekata u dvije faze

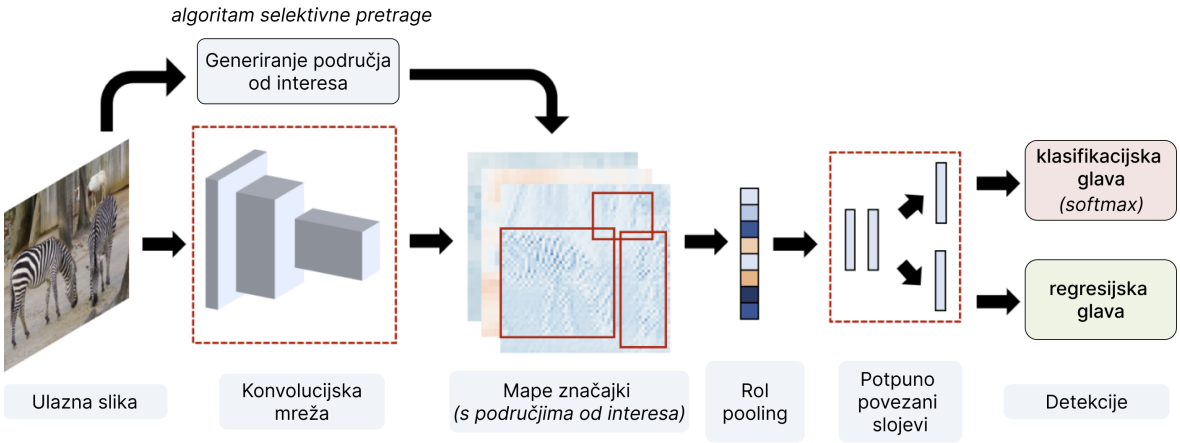
Jedan od prvih algoritama koji je u proces detekcije uključio konvolucijske neuronske mreže bio je **R-CNN** (engl. *Region-based Convolutional Neural Network*) [22] detektor u dvije faze. Osnovna ideja je jednostavna. Prvo se pomoću algoritma selektivne pretrage (engl. *selective search*) [23] iz ulazne slike generira dvije tisuće područja od interesa. Sva predložena područja se zatim svode na istu fiksnu veličinu te se iz njih, pomoću predtrenirane konvolucijske neuronske mreže, izdvajaju vektori značajki. Dobivene značajke se potom

prosljeđuju u linearni stroj potpornih vektora [2] za klasifikaciju područja od interesa i u regresijsku glavu koja fino podešava predložene granične okvire kako bi lokalizacija objekata bila preciznija. Naposlijetku metodom ne-maksimalnog potiskivanja (engl. *Non-Maximum Suppression*, NMS) uklanjaju se redundantne detekcije istog objekta kako bi se smanjio broj lažno pozitivnih detekcija. Arhitektura R-CNN detektora prikazana je na Slici 2.3.



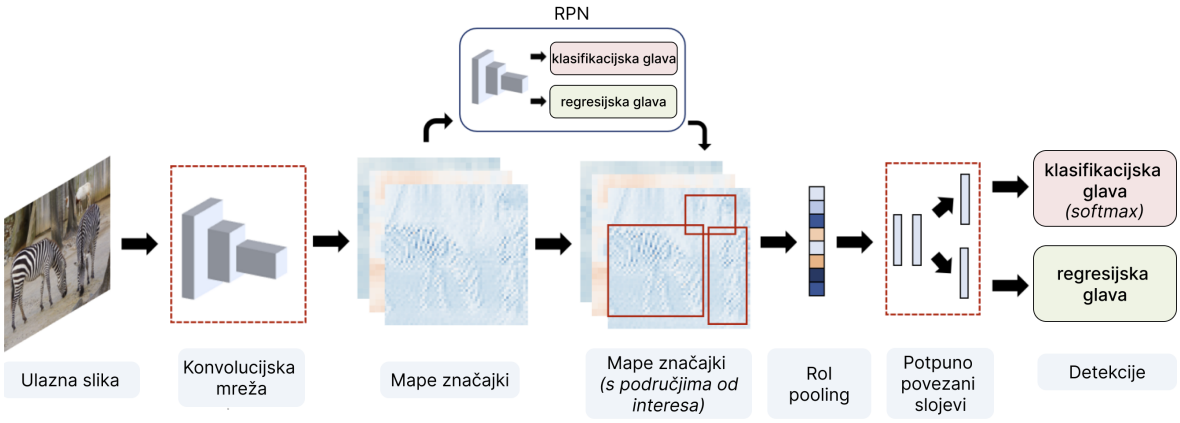
Slika 2.3: R-CNN detektor (slika s izmjenama preuzeta iz [24]).

R-CNN model zahtijeva da se svako generirano područje od interesa, njih dvije tisuće, zasebno provuče kroz konvolucijsku neuronsku mrežu. U radu [25], predložena je učinkovitija varijanta R-CNN modela, **Fast R-CNN**, koja prvo ulaznu sliku provlači kroz konvolucijsku neuronsku mrežu (samo jedan prolaz) koja generira mape značajki. Zatim, algoritam selektivne pretrage iz dobivenih mapa značajki generira područja od interesaa koja se pomoću *RoI* sloja sažimanja (engl. *RoI* pooling) svode na područja unaprijed definirane, fiksne visine i širine. Značajke dobivene RoI sažimanjem se dalje prosljeđuju kroz potpuno povezane slojeve koji se granaju na *softmax* klasifikacijsku glavu koja predviđa vjerojatnost pojedine klase (i pozadine) i regresijsku glavu koja se koristi za korekciju graničnog okvira. Fast R-CNN detektor ilustriran je na Slici 2.4.



Slika 2.4: Fast R-CNN detektor (slika s izmjenama preuzeta iz [24]).

Prilikom predviđanja Fast-RCNN modela, značajan dio vremena troši se na generiranje područja od interesa sporim algoritmom selektivne pretrage. Ren *et al.* [26], predstavljaju novi model, **Faster R-CNN**, koji algoritam selektivne pretrage zamjenjuje zasebnom neuronskom mrežom za predlaganje područja od interesa (engl. *Region Proposal Network*, *RPN*), koja se sastoji isključivo od konvolucijskih slojeva. Generiranje područja od interesa RPN mrežom temelji se na tzv. baznim graničnim okvirima (engl. *anchors*): svakoj lokaciji na konvolucijskoj mapi značajki pridružuje se devet predefiniranih pravokutnih regija različitih veličina i omjera visine i širine, koristeći metodu pomičnog prozora. Za svaki bazni okvir, RPN mreža daje ocjenu vjerojatnosti da sadrži objekt od interesa koja se kasnije koristi za filtriranje prijedloga regija zajedno s metodom ne-maksimalnog potiskivanja. RPN mreža također sadrži i regresijski dio koji korigira bazne okvire. Predloženi pristup generiranja područja od interesa RPN mrežom, ne samo da ubrzava proces detekcije, već u konačnici rezultita i većom točnošću detektora [26]. Arhitektura Faster R-CNN detektora prikazana je na Slici 2.5.



Slika 2.5: Faster R-CNN detektor (slika s izmjenama preuzeta iz [24]).

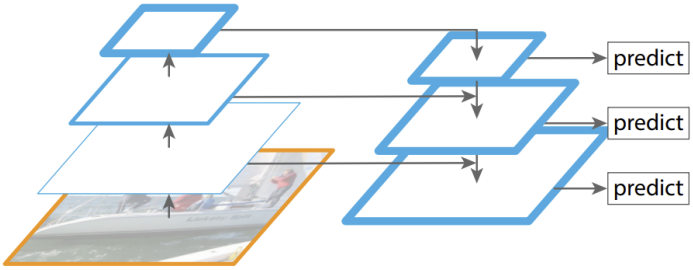
Uz tri navedena detektora u dvije faze, još neki od popularnih detektora su:

(1) **SPP-Net** [27] detektor koji prvo koristi konvolucijsku neuronsku mrežu za ekstrakciju mapu značajki, a zatim se algoritmom selektivne pretrage generiraju područja od interesa. SPP-Net uvodi sloj prostornog piramidalnog sažimanja (engl. *spatial pyramid pooling*) između posljednjeg konvolucijskog i potpuno povezanih slojeva, kojim se uklanja potreba za fiksnom veličinom ulazne slike.

(2) **Mask R-CNN** [28] detektor koji unaprijeđuje Faster R-CNN dodajući na potpuno povezane slojeve, paralelno s klasifkacijskom i regresijskom granom, još jednu granu koja predviđa semantičku masku objekta. Također, umjesto *RoI Pooling* sloja koristi *RoI Align* sloj.

(3) **FPN** [29] detektor koji modul ekstrakcije značajki u Faster R-CNN detektoru zamjenjuje piramidalnom mrežom značajki (engl. *Feature Pyramid Network*) i time postiže

state-of-the-art rezultate. Piramidalna mreža značajki prvo konvolucijskom neuronskom mrežom ("odozdo prema gore") izdvaja piramidalnu hijerarhiju mapa značajki iz danog ulaza - od jednostavnijih značajki više rezolucije na nižim razinama prema kompleksnijim značajkama niže rezolucije. Zatim se ("odzgo prema dole") naduzorkovanjem povećava rezolucija značajki višeg semantičkog značenja s vrha piramide te se dobivenim mapama pridružuju odgovarajuće mape značajki iz prolaza "odozdo prema gore". Navedeni proces ilustriran je na Slici 2.6.



Slika 2.6: Piramidalna mreža značajki (slika preuzeta iz [29]).

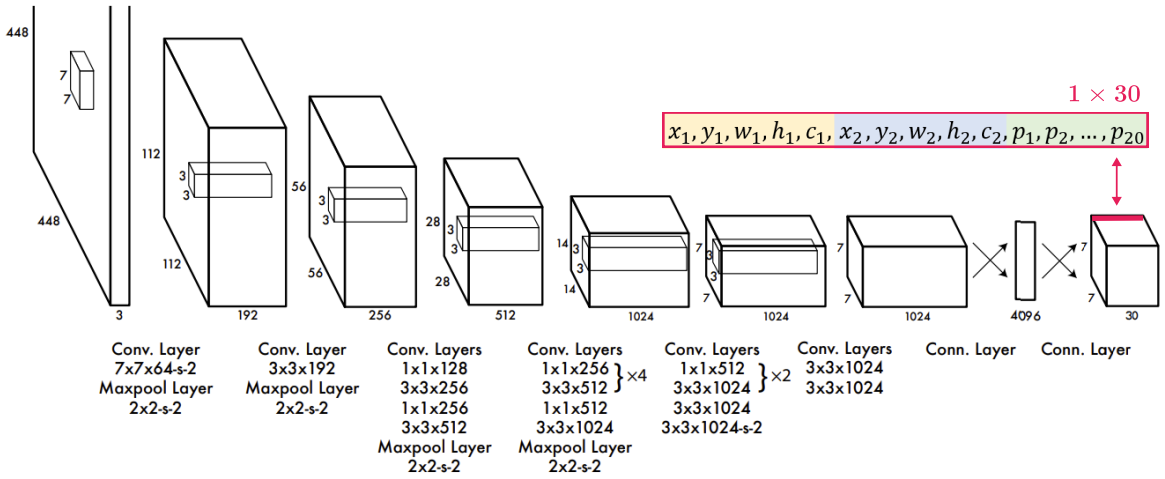
2.4.2. Detekcija objekata u jednoj fazi

Zahvaljujući optimalnom balansu između brzine izvršavanja i točnosti detekcije, detektori iz **YOLO** (engl. *You Only Look Once*) [30, 31, 32, 33, 34, 35, 36, 37, 38, 39] familije detektora ističu se kao vodeći predstavnici detektora u jednoj fazi, ali također i među detektorima općenito. Među ostalim detektorima koji objekte detektiraju u jednoj fazi popularni su još i **SSD** (engl. *Single Shot MultiBox Detector*) [40] , **EfficentDet** [41], **RetinaNet** [42] te **CenterNet** [43] detektor.

YOLO

Redmon *et al.* [30] su 2016. godine predstavili prvu verziju YOLO detektora. Predložena verzija YOLO detektora koristi jednu neuronsku mrežu koja direktno iz ulazne slike predviđa granične okvire i vjerojajatnosti klasa. Algoritam je vrlo jednostavan. Ulazna slika dijeli se na mrežu od $S \times S$ ćelija, pri čemu je svaka ćelija zadužena za detekciju objekta kojemu se središte nalazi u njoj. Za svaku ćeliju, model predviđa B graničnih okvira i C vjerojatnosti klasa. Za svaki granični okvir, predviđa se pet vrijednosti: x, y, w, h i c gdje su (x, y) koordinate središta graničnog okvira (relativne ćeliji u kojoj se središte nalazi), w i h visina i širina graničnog okvira (relativne visini i širini cijele slike), a p_c vrijednost koja predstavlja sigurnost modela da granični okvir sadrži objekt i sigurnost u predviđeni granični okvir. Dakle, za svaku ćeliju, predviđa se $B \cdot 5 + C$ vrijednosti. Izlaz neuronske mreže za detekciju tada

je $S \times S \times (B \cdot 5 + C)$ volumen. Ahitektura mreže, inspirirana GoogleLeNet [44] modelom, ilustrirana je na Slici 2.7.



Slika 2.7: Arhitektura originalnog YOLO detektora: $S = 7, B = 2, C = 20$ (slika s izmjenama preuzeta iz [30]).

Prednosti predloženog YOLO detektora su višestruke: jednostavnost implementacije, brzina izvršavanja (pionir brze detekcije u stvarnom vremenu), implicitno kodiranje kontekstualanih i vizualnih informacija klasa budući da za vrijeme treniranja i testiranja YOLO vidi cijelu sliku, te *end-to-end* optimizacija budući da se koristi jedna mreža za detekciju. Međutim, YOLO detektor zaostaje za drugim *state-of-the-art* detektorima u pogledu točnosti detekcije; nešto je manje precizan u lokalizaciji objekata te ima poteškoća s detekcijom manjih objekata i objekata u grupama. Nadalje, svaka ćelija može predvidjeti samo dva granična okvira ($B = 2$) i jednu klasu.

YOLOv2 [31], također poznat i kao YOLO900, poboljšana je verzija originalnog YOLO detektora koja: (1) uvodi normalizaciju podataka po mini-grupama (engl. *batch normalization*) [45] za pobljšanje konvergencije modela i njegovu regularizaciju, (2) trenira klasifikator s ulaznim slikama veće rezolucije, (3) uklanja potpuno povezane slojeve iz YOLO detektora i uvodi bazne granične okvire, (4) koristi K-Means [46] algoritam za automatski odabir inicijalnih baznih okvira, (5) implementira Darknet-19 okosnicu koja se temelji na VGG [47] arhitekturi, (6) za vrijeme treniranja koristi ulaze različitih dimenzija. Treća varijanta YOLO detektora, **YOLOv3** [32], koristi novu Darknet-53 okosnicu s rezidualnim blokovima [48], te uvodi koncept sličan piramidalnoj mreži značajki kako bi se poboljšala detekcija objekata različitih veličina.

YOLOv3 posljednja je verzija YOLO detektora koju je kreirao originalni autor YOLO-a, Joseph Redmonom koji se iz etičkih razloga povukao iz područja računalnog vida i umjetne inteligencije. U međuvremenu, predstavljene su verzije YOLO detektora od **YOLOv4** [33] do najrecentnije verzije **YOLOv9** [36], te verzije poput **PP-YOLO** [37], **YOLOR** [38], **YOLOX** [39] i **YOLO-NAS** [49] detektora. Verzije YOLO algoritma od YOLOv1 do YOLOv4

koriste DarkNet okvir otvorenog koda koji je napisan u programskom jeziku C i CUDA-i. Verzija **YOLOv5** [34], koju je razvio tim iz tvrtke Ultralytics, prva je verzija YOLO-a koja je umjesto u DarkNet okviru implementirana u PyTorch-u. Budući da je YOLOv5 objavljen samo kao GitHub repozitorij, a ne kao recenzirano istraživanje, postojale su sumnje u autentičnost i učinkovitost tog modela. Iako odgovarajući sitraživački rad nije bio dostupan, činjenica da je YOLOv5 kasnije primjenjen u brojnim aplikacijama s učinkovitim rezultatima počela je graditi vjerodostojnost modela [24]. Isti tim je kasnije (2023. godine) objavio noviju verziju YOLO detektora - **YOLOv8** [35].

Najnovija iteracija YOLO detektora, **YOLOv9** [36], ističe se inovativnim pristupom sprječavanja potencijalnog gubitka informacija tijekom sukcesivnog prolaska podataka kroz slojeve neuronske mreže. Autori predstavljaju: (1) koncept "programabilne informacije o gradijentu" (engl. *programmable gradient information, PGI*) koji osigurava očuvanje bitnih informacija kroz slojeve dubokih neuronskih mreža kako bi se generirali pouzdani gradijenati za ažuriranje težina, (2) efikasnu "mrežu generalizarne učinkovite agregacije slojeva" (engl. *Generalized Efficient Layer Aggregation Network, GELAN*) koja omogućuje fleksibilnu integraciju različitih računalnih blokova čineći YOLOv9 detektor prikladnim za širok raspon aplikacija. Kombinirajući navedene koncepte, YOLOv9 uvelike nadmašuje postojeće detektore prikladne za izvođenje u stvarnom vremenu na MS COCO [50] skupu podataka.

CenterNet

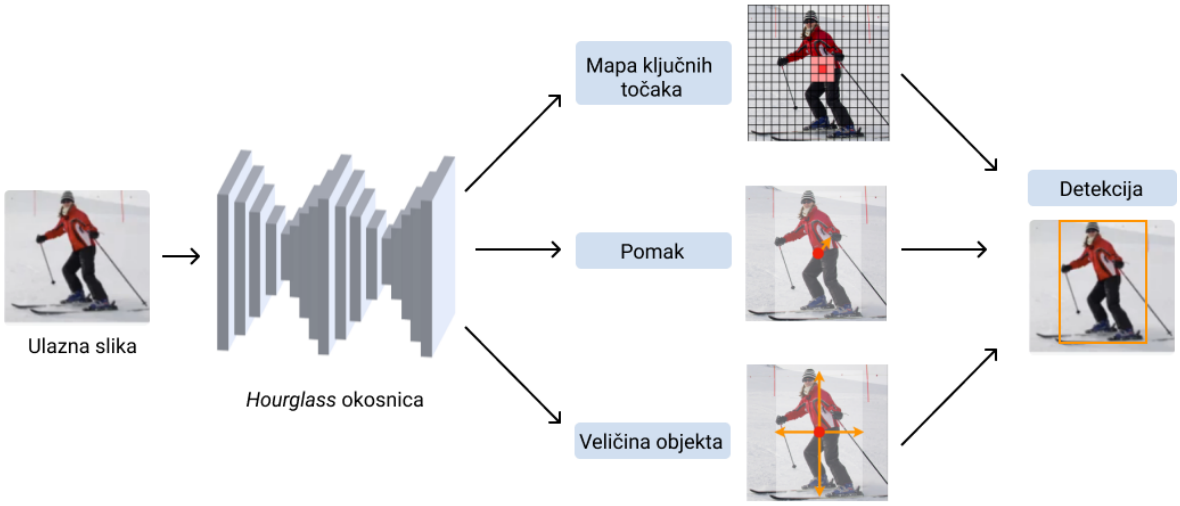
Zhou et al. [43] predlažu novi pristup detekciji koji objekte predstavlja samo *jednom točkom*: središtem pripadajućeg graničnog okvira. Njihov **CenterNet** detektor ne koristi predefinirane bazne okvire i ne zahtijeva dodatno postprocesiranje NMS metodom koja prolongira vrijeme izvršavanje detektora. Neka su R unaprijed definirana vrijednost faktora redukcije izlaza i C broj promatranih kategorija. CenterNet detektor kao ulaz prima sliku I širine W i visine H te za nju generira:

(1) mapu ključnih točaka (engl. *keypoint heatmap*) $\hat{Y} \in [0,1]^{\frac{W}{R} \times \frac{H}{R} \times C}$ koja daje informaciju o lokaciji središta objekta,

(2) predviđeni pomak (engl. *offset*) $\hat{O} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$,

(3) predviđenu visinu i širinu graničnog okvira $\hat{S} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$.

Među promatranim okosnicama za predviđanje $\hat{Y}$, najbolje rezultate dala je *Hourglass-104* [51] arhitektura. Iz mape ključih točaka $\hat{Y}$, za svaku klasu c zasebno, se kao ključne točke (središta) detektiraju lokalni maksimumi (engl. *peaks*) $(\hat{x}_i, \hat{y}_i)$ koji imaju najveću vrijednost u 3×3 susjedstvu. Vrijednost $\hat{Y}_{x_i,y_i,c}$ tada mjeri pouzdanost u danu detekciju, dok je rezultirajući granični okvir dan s $(\hat{x}_i + \delta\hat{x}_i - \hat{w}_i/2,\ \hat{y}_i + \delta\hat{y}_i - \hat{h}_i/2,\ \hat{x}_i + \delta\hat{x}_i + \hat{w}_i/2,\ \hat{y}_i + \delta\hat{y}_i + \hat{h}_i/2)$ gdje je $(\delta\hat{x}_i, \delta\hat{y}_i) = \hat{O}_{\hat{x}_i,\hat{y}_i}$ predviđena vrijednost pomaka te $(\hat{w}_i, \hat{h}_i) = \hat{S}_{\hat{x}_i,\hat{y}_i}$ predviđene vrijednosti širine i visine graničnog okvira. Arhitektura CenterNet detektora prikazana je na Slici 2.8.



Slika 2.8: Arhitektura CenterNet detektora.

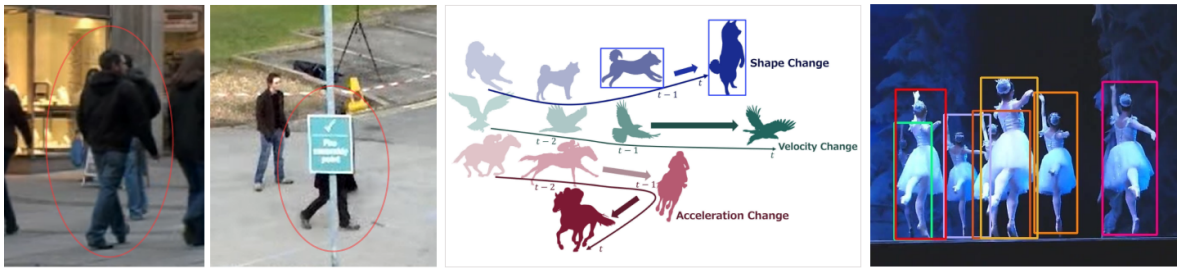
3. Praćenje više objekata

Zahvaljujući velikom komercijalnom i akademskom potencijalu, praćenje objekata postalo je jednim od najrelevantnijih problema u domeni računalnog vida [52, 53]. Nakon detekcije, praćenje je najčešće prvi sljedeći korak koji podrazumijeva precizno praćenje putanje objekata na nizu slika, obično videzapisu [54]. Kontinuirana informacija o kretanju objekata kroz vrijeme od izuzetne je važnosti u mnogim aplikacijama kao što su video nadzor, analiza ponašanja, autonomna vožnja i robotika. Sama detekcija objekata iz okvira u okvir ne pruža tu informaciju, stoga je nužno primijeniti algoritme praćenja kako bi se bolje razumjela dinamika kretanja objekata.

S obzirom na broj objekata koji se prati razlikujemo metode praćenja jednog objekta (engl. *Single Object Tracking, SOT*) i metode praćenja više objekata (engl. *Multiple Object Tracking, MOT*). Kod metoda **praćenja jednog objekta** cilj je pratiti jedan konkretan objekt tijekom cijelog videozapisa. Objekt specificiran u prvom okviru videozapisa se detektira i prati kroz sve ostale okvire [52]. S druge strane, metodama za **praćenje više objekata** cilj je locirati više različitih objekata te pratiti njihov identitet i putanje kroz dani videozapis [55]. Pratiti se primjerice može ljude na javnim mjestima (ulicama, trgovima, trgovačkim centarima, aerodromima) [56, 57], igrače i druge objekte na terenu za vrijeme sportskih događanja [58, 59], vozila na cestama i raskrižjima [60, 61], stanice i mikroorganizme unutar bioloških sustava [62] ili grupe životinja i insekata [63, 64, 65].

Praćenje više objekata složen je zadatak obilježen brojnim izazovima. U odnosu na praćenje jednog objekta, ono podrazumijeva dva dodatna zadatka: utvrđivanje broja objekata koji se prate, a koji se često mijenja tijekom vremena, i održavanje konzistentnog identiteta objekata tijekom cijelog videozapisa [55]. Ovi zadaci dodatno se kompliciraju čestim preklapanjima objekata s drugim objektima ili pozadinom, kao i kolizijama i interakcijama među objektima. Dodatan izazov predstavlja sličnost u izgledu različitih objekata te nepredvidive promjenama u njihovom kretanju poput iznenadnih promjena smjera i brzine te naglih zaustavljanja. Nadalje, izgled istog objekta može se znatno razlikovati u različitim dijelovima videozapisa zbog varijacija u osvjetljenju, pozadini ili položaju objekta. Algoritmi za praćenje više objekata također moraju biti sposobni reidentificirati izgubljene objekte te identificirati nove objekte koji se tek pojavljuju u sceni. Primjeri nekih od navedenih izazova prikazani su na Slici 3.1. Pored navedenoga, većina primjena zahtijeva praćenje objekata u stvarnom vremenu, što implicira potrebu za brzim algoritmima. S druge strane, ograničenja hardverskim

resursima mogu značajno ograničiti kompleksnost i učinkovitost ovih algoritama.



(a) (b) (c)

Slika 3.1: Primjeri izazova s kojima se MOT algoritam susreće. (a) lijevo: preklapanje objekta koji se prati drugim objektom, desno: preklapanje objekta predmetom iz pozadine (slike iz MOT15 videozapisa [66]), (b) promjene u izgledu i kretanju objekta koji se prati (slika preuzeta iz [67]), (c) sličan izgled objekata koji se prate (slika preuzeta iz [68]).

Algoritmi za praćenje mogu se primijeniti s dvodimenzionalnim [56, 63, 67, 69] i s trodimenzionalnim [70, 71, 72, 73] podacima. Videozapisi koji se koriste prilikom praćenja mogu biti snimljeni s jednom kamerom ili s više kamera postavljenih na različitim položajima kako bi se dobila veća pokrivenost scene i bolja percepcija. Korištenjem više kamera, a samim time i različitih pogleda, problem preklapanja objekata može se reducirati [60]. Međutim, praćenje s više kamera dolazi s dodatnim izazovima poput varijacije u pogledima kamera i njihovih mogućih preklapanja, sinkronizacije i precizne kalibracije kamera te integracije podataka različitih kamera [74]. Obrada podataka s više kamera također može zahtijevati veće računalne resurse [60]. Nadalje, podaci dobiveni iz kamera mogu se kombinirati s podacima koji su dobiveni drugim senzorima poput LIDAR-a, radara ili ultrazvučnih senzora [72, 75].

Zbog široke primjene u područjima poput robotike i autonomne vožnje, istaknuti smjerovi budućeg istraživanja u praćenju objekata uključuju otvorena pitanja vezana za praćenje objekata putem više različitih kamera te trodimenzionalno multimodalno praćenje [76, 77]. Dvodimenzionalno praćenje na videozapisima snimljenim jednom kamerom trenutno je dominantna pradigma u praćenju više objekata te je u fokusu ostatka ovog rada.

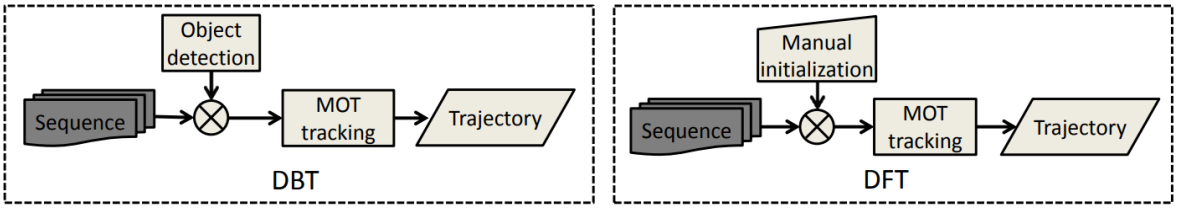
3.1. Kategorizacija MOT algoritama

S obzirom na način obrade videozapisa tijekom praćenja, MOT algoritmi se dijele na *online* algoritme koji prilikom obrade trenutnog okvira koriste samo informacije iz prošlih okvira i *offline (batch)* algoritme koji koriste informacije iz prošlih i iz budućih okvira videozapisa [55]. Offline metode generalno daju bolje rezultate zbog dostupnosti globalne informacije iz svih okvira videozapisa, ali one nisu primjenjive u aplikacijama koje zahtijevaju izvršavanje u stvarnom vremenu jer tada nisu dostupni budući okviri videozapisa [78, 79].

MOT algoritmi se dalje mogu kategorizirati na algoritme koji prate *samo jednu klasu objekata* u videozapisu, primjerice samo pješake, i na algoritme koji istovremeno prate

objekte više različitih klasa, primjerice pješake, vozila i bicikliste [80]. U slučaju praćenja objekata više različitih klasa, osim lokalizacije i praćenja objekata kroz videozapis, algoritam također treba svaki objekt dodatno i klasificirati.

Ovisno o načinu incijalizacije objekata, MOT algoritmi se dijele na *algoritme praćenja koji za incijalizaciju koriste detekcije* (engl. *Detection-Based Tracking, DBT*) i *algoritme koji ne koriste detekcije* (engl. *Detection-Free Tracking, DFT*), već u prvom okviru iziskuju ručnu inicijalizaciju fiksnog broja objekata koji se dalje prate kroz videozapis [55, 57]. Ove dvije vrste algoritama ilustrirane su na Slici 3.2. Za razliku od DFT algoritama, algoritmi koji koriste detekcije automatski mogu otkriti nove objekte koji ulaze u scenu i prestati pratiti one koji izlaze iz nje. S druge strane, kvaliteta samog praćenja značajno ovisi o kvaliteti detekcija koje se pri tom koriste. Identičan algoritam može dati znatno različite rezultate praćenja koristeći različite detekcije [57].



Slika 3.2: Dijagram algoritma koji koristi detekcije za inicijalizaciju objekata (lijevo) i algoritma koji ne koristi detekcije (desno) [55].

Algoritmi koji koriste detekcije mogu se dalje podijeliti na *algoritme temeljene na detekciji* (engl. *Tracking-By-Detection*, *TBD*), koji detekciju i praćenje promatraju kao dvije nezavisne komponente, i *algoritme zajedničke detekcije i praćenja* (engl. *Joint-Detection and Tracking*, *JDT*) koji objedinjuju detekciju i praćenje u jedan integrirani algoritam koji istovremeno obavlja oba zadatka.

3.2. Osnovni koraci MOT algoritma

S obzirom na značajan napredak i izvanredna postignuća u području detekcije objekata, metode koje koriste detekcije postale su standardnom u domeni praćenja više objekata [81]. Stoga je u nastavku fokus isključivo na njima. Unatoč velikoj raznolikosti, većina MOT algoritama na određen način kombinira sljedeće korake (dio njih ili sve) [78, 57, 57]:

(a) ***Detekcija** objekata*: u danom ulaznom okviru, detektor lokalizira objekte od interesa koristeći pravokutne granične okvire;

(b) ***Predviđanje sljedeće pozicije** objekta*: za svaku putanju, buduća pozicija objekta predviđa se temeljem prethodnih kretanja i brzine;

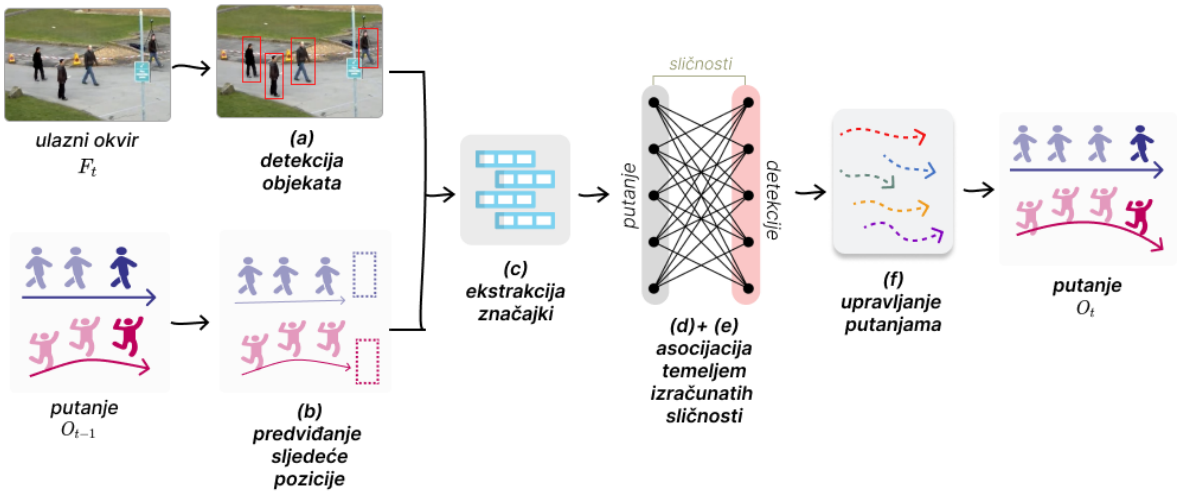
(c) ***Ekstrakcija značajki***: vizualne značajke i značajke o kretanju "izvlače" se iz detekcija i/ili putanja praćenih objekata pomoću jednog ili više ekstraktora;

(d) ***Izračun sličnosti***: značajke i predviđene pozicije objekata koriste se za izračun sličnosti (ili udaljenosti) između detekcija i putanja;

(e) ***Asocijacija***: izračunate vrijednosti sličnosti (ili udaljenosti) koriste se za povezivanje detekcija i putanja, dodjeljujući detekcijama identifikator odgovarajućeg objekta;

(f) ***Upravljanje putanjama***: ažuriranje stanja postojećih putanja, incijalizacija novih putanja i završavanje neaktivnih putanja.

Korak (a) i koraci (b) - (f) mogu se promatrati kao dvije nezavisne komponente algoritama temeljenih na detekciji: komponenta za *detekciju* i komponenta za *praćenje*. Kvaliteta detekcija koje se koriste za praćenje ima značajan utjecaj na performanse TBD algoritma. Kako bi se omogućila transparentna usporedba različitih komponenti za praćenje, neki MOT izazovi pružaju pristup javnim detekcijama [82, 83]. Time se stavlja fokus na razvoj inovativnih komponenti za praćenje, umjesto na implementaciju moćnih detektora. Slika 3.3 ilustrira uobičajen redoslijed navedenih koraka u algoritmu za praćenje. S druge strane, u algoritmima zajedničke detekcije i praćenja, određeni koraci (b) - (f) se integriraju s detekcijom iz koraka (a). Najčešće je to integracija detekcije i ekstrakcije značajki [84, 85, 86] ili detekcije i predviđanje kretanja tj. sljedeće pozicije objekta [87, 88, 89].



Slika 3.3: Uobičajen proces MOT algoritma temeljenog na detekciji (slika ulaznog okvira preuzeta iz MOT15 [66] skupa podataka).

3.2.1. Detekcija objekata

Iako se u koraku (a) može koristiti tradicionalne metode detekcije objekata koje se zasnivaju na tradicionalnim, ručno generiranim značajkama [90, 91], suvremeni algoritmi za

praćenje preferiraju upotrebu dubokih modela zbog njihove superiornosti u detekciji objekata od interesa. Nekoliko metoda praćenja implementira Faster R-CNN detektor u dvije faze [92, 93, 94, 95] koji obično daje bolje rezultate u pogledu točnosti detekcije od detektora koji detektiraju objekte u jednoj fazi. S druge strane, radovi poput [96, 97, 98] koriste jednofazni SSD detektor prilikom praćenja objekata. Zbog optimalnog balansa točnosti i brzine, mnoge metode implementiraju varijante YOLO detektora [67, 99, 100, 101]. CenterNet detektor se također dosta često koristi zbog svoje efikasnoti i jednostavnosti [69, 86, 89].

3.2.2. Predviđanje sljedeće pozicije objekta

Detektori koji se koriste u MOT algoritmima nisu savršeni. Često se suočavaju s izazovima poput lažno pozitivnih i nepreciznih detekcija koje se javaljaju zbog lošeg osvijetljenja, prisutnosti sjena, djelomične zaklonjenosti i sličnih faktora. Također, moguće je da detektor uopće ne registrira objekt, osobito kada je potpuno zaklonjen, što dovodi do prekida putanje tog objekta. Kako bi se prevladali navedeni problemi i poboljšali rezultati samog praćenja, primjenjuju se različite metode za predviđanje budućeg stanja objekta. One omogućuju nadopunjavanje putanje objekta u slučajevima kada detekcija nedostaje ili korigiranje putanje u slučajevima neprecizne lokalizacije objekata pomoću detektora [102]. Za predviđanje stanja objekta u sljedećem okviru videozapisa najčešće se koristi Kalmanov filter i njegove modifikacije [67, 93, 94, 99, 100, 101, 103, 92, 86, 85].

Kalmanov filter [104] je rekurzivni matematički algoritam koji se koristi za procjenu stanja diskretnog dinamičkog sustava na temlju niza mjerenja koja u sebi sadrže šum. Za predviđanje sljedećeg stanja sustava Kalmanov filter zahtijeva informacije o predviđenom stanju sustava iz prethodnog koraka i trenutnim (novim) mjerenjima. U kontekstu praćenja više objekata, mjerenja u koraku t bi bile detekcije objekata u okviru t danog videozapisa. Primjerice, detekcija $\boldsymbol{z}_t = [u, v, w, h, c]^\top$ gdje su (u, v) koordinate centra, w visina, a h širina graničnog okvira, te c pouzdanost u danu detekciju [93, 105]. Iteracija algoritma Kalmanovog filtera sastoji se od dva koraka [106]:

(1) ***prediktivni korak*** u kojem se temeljem prethodnog stanja sustava iz koraka $t-1$ računa *apriori procjenitelj sljedećeg stanja* sustava $\hat{\boldsymbol{x}}_{t|t-1} \in \mathbb{R}^{n_x \times 1}$ i *matrica kovarijance apriori pogreške* procjenitelja $\boldsymbol{P}_{t|t-1} \in \mathbb{R}^{n_x \times n_x}$,

(2) ***korektivni korak*** u kojem algoritam korigira apriori pretpostavke temeljem pristiglih, novih mjerenja (*aposteriori procjenitelj stanja* $\hat{\boldsymbol{x}}_{t|t} \in \mathbb{R}^{n_x \times 1}$ i *matrica kovarijance aposteriori pogreške* procjenitelja $\boldsymbol{P}_{t|t} \in \mathbb{R}^{n_x \times n_x}$).

Jednadžbe Kalmanovog filtera tada se mogu podijeliti na dvije grupe:

$$\textit{predikcija} \quad \begin{cases} \hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{F}_t \hat{\boldsymbol{x}}_{t-1|t-1} \\ \boldsymbol{P}_{t|t-1} = \boldsymbol{F}_t \boldsymbol{P}_{t-1|t-1} \boldsymbol{F}_t^\top + \boldsymbol{Q}_t \end{cases} \tag{3.1}$$

$$\textit{korekcija} \quad \begin{cases} \boldsymbol{K}_t = \boldsymbol{P}_{t|t-1} \boldsymbol{H}_t^\top (\boldsymbol{H}_t \boldsymbol{P}_{t|t-1} \boldsymbol{H}_t^\top + \boldsymbol{R}_t)^{-1} \\ \hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{K}_t (\boldsymbol{z}_t - \boldsymbol{H}_t \hat{\boldsymbol{x}}_{t|t-1}) \\ \boldsymbol{P}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t) \boldsymbol{P}_{t|t-1} \end{cases} \tag{3.2}$$

pri čemu je $\boldsymbol{F}_t \in \mathbb{R}^{n_x \times n_x}$ tranzicijska matrica iz stanja u koraku $t-1$ do stanja u koraku t, $\boldsymbol{Q}_t$ matrica kovarijance šuma procesa, $\boldsymbol{z}_t \in \mathbb{R}^{n_z \times 1}$ mjerenja pristigla u koraku t, $\boldsymbol{R}_t$ matrica kovarijance šuma mjerenja, $\boldsymbol{H}_t \in \mathbb{R}^{n_z \times n_x}$ opservacijska matrica koja preslikava stanje sustava u prostor mjerenja, te $\boldsymbol{K}_t \in \mathbb{R}^{n_x \times n_z}$ Kalmanovo pojačanje u koraku t je koje je definirano tako da minimizira kovarijancu pogreške procjenitelja.[1]

Kalmanov filter omogućava optimalnu procjenu stvarnog stanja *diskretnog linearnog* sustava, minimizirajući utjecaj procesnog i mjernog šuma. **Prošireni Kalmanov filter** [107] je modifikacija originalnog Kalmanovog filtera prikladna za procjenu stanja nelinearnih sustava. U sklopu GIAOTracker [108] algoritma praćenja predstavljena je i modifikacija Kalmanovog Filtera, **NSA** (*Noise Scale Adaptive*) **Kalman**, koja prilagođava kovarijancu šuma mjerenja u skladu s pouzdanošću detekcija:

$$\widetilde{\boldsymbol{R}}_t = (1 - c_t) \boldsymbol{R}_t \tag{3.3}$$

gdje je $\boldsymbol{R}_t$ unaprijed postavljena konstantna kovarijanca šuma, a c_t pouzdanost detekcije. Što je pouzdanost c_t veća, to $\widetilde{\boldsymbol{R}}_t$ poprima manje vrijednosti, što implicira da će u korektivnom koraku, tijekom ažuriranja stanja sustava, veća težina biti stavljena na mjerenje, odnosno detekciju [99].

Umjesto Kalmanovog filtera, za predviđanje sljedeće pozicije objekta također se može koristiti i čestični filter (engl. *particle filter)* [90, 91, 109] ili RNN i LSTM arhitekture neuronskih mreža [110, 111, 112, 102].

3.2.3. Ekstrakcija značajki

Korak ekstrakcije značajki predstavlja ključnu fazu u procesu identifikacije i razlikovanja objekata u MOT algoritmima. Stoga je od presudne važnosti kreirati značajke koje su robousne na promjene istog objekta tijekom vremena i istovremeno sposobne jasno razlikovati različite objekte. Najčešće se koriste značajke koje sadrže informacije o *vizualnim karakteristikama* objekta, kao što su boja, oblik ili tekstura, te značajke koje kodiraju *svojstva*

[1] n_x označava broj stanja u vektoru stanja, n_z broj različitih mjerenja koji pristiže u svakom koraku.

kretanja objekta, poput brzine i pozicije objekta. Iako vizualne karakteristike i karakteristike kretanja pružaju komplementarne informacije, one se obično promatraju zasebno, a potom se prilikom izračuna sličnosti kombiniraju na neki način, primjerice kroz jednostavne linearne kombinacije [102].

Vizualne značajke dobivene konvolucijskim neuronskim mrežama

Zbog njihove sposobnosti da iz ulaznih podataka automatski izvlače kompleksne značajke, konvolucijske neuronske mreže i njihove modifikacije postale su dominantan pristup ekstrakciji značajki u MOT algoritmima [78]. Jedan od ranijih primjera primjene konvolucijskih značajki za praćenje objekata opisan je u [113], gdje autori koriste predtreniranu konvolucijsku neuronsku mrežu kako bi izvukli 4096-dimenzionalni vektor značajki iz svakog graničnog okvira. Dobiveni vektori se zatim reduciraju na 256 dimenzija korištenjem PCA algoritma [114]. Neki autori za ektrakciju vizualnih značajki u MOT algoritmima koriste standardne arhitekture konvolucijskih neuronskih mreža, poput GoogLeNet [92, 115, 116, 117], ResNet [112, 118, 119] i VGG [120, 121, 102] arhitektura.

Sijamske neuronske mreže za ekstrakciju značajki

Alternativni pristup ekstrakciji značajki, koji se također zasniva na konvolucijskim neuronskim mrežama, je ekstrakcija značajki pomoću sijamskih neuronskih mreža koje se koriste za učenje sličnosti. *Sijamske neuronske mreže* [122] najčešće se sastoj od dvije ili tri identične podmreže koje se zajedno treniraju i kojima se težine za vrijeme treniranja zrcalno ažuriraju. Ilustracija sijamskih nueronskih mreža prikazana je na Slici 3.4.

Kada se koriste dvije podmreže, sijamska neuronska mreža na ulaz prima dvije slike i računa kontrastivni gubitak (engl. *contrastive loss*) [123]:

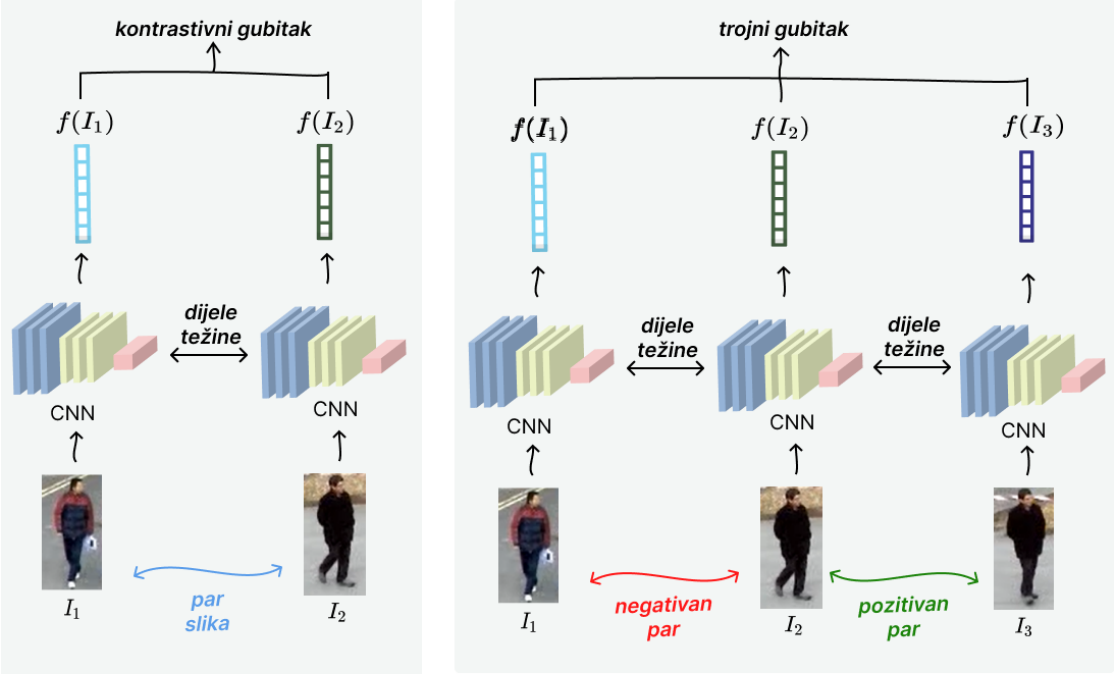
$$\mathcal{L}(I_1, I_2) = (1-y)\cdot\frac{1}{2}\|f(I_1)-f(I_2)\|_2^2 + y\cdot\frac{1}{2}\max\{0,\ m-\|f(I_1)-f(I_2)\|_2\}^2, \tag{3.4}$$

gdje je $m > 0$ margina koja definira radijus sličnosti, $y = 1$ ako se radi o sličnim slikama (istom objektu), 0 u protivnom. U slučaju tri podmreže, na ulaz sijamske mreže šalju se tri slike: jedna temeljna slika (engl. *anchor*) (I_2), pozitivni primjer koji je sličan temeljnoj slici (I_3) i negativan primjer koji joj nije sličan (I_1). Za vrijeme treniranja minimizira se trojni gubitak (engl. *triplet loss*) [124]:

$$\mathcal{L}(I_1, I_2, I_3) = \max\{0,\ \|f(I_2)-f(I_3)\|_2^2 - \|f(I_2)-f(I_1)\|_2^2 + m\} \tag{3.5}$$

U oba slučaja, cilj je minimizirati udaljenost vektora značajki sličnih (pozitivnih) primjera i istovremeno maksimizirati udaljenost vektora značajki različitih (negativnih) primjera.

U kontektu algoritama praćenja više objekata, Kim *et al.* [125] koriste konvolucijsku podmrežu sijamske neuronske mreže trenirane s kontrastivnim gubitkom za ekstrakciju vek-



Slika 3.4: Sijamska neuronska mreža s dvije podmreže (a) i s tri podmreže (b).

tora značajki iz ulazne slike. Varijante sijamske neuronske mreže koje kao ulaz primaju dvije ulazne slike, ali ne koriste klasični kontrastivni gubitak korištene su u [126, 127]. S druge strane, Zhou *et al.* [128] pomoću podmreže sijamske neuronske mreže trenirane s trojnim gubitkom iz danog uzlaza "izvlači" 128-dimenzionalni vektor značajki. Long *et al.* u [129], za ekstrakciju vektora značajki koriste GoogLeNet podmrežu sijamske neuronske mreže s trojnim gubitkom, dok Zhang *et al.* u [130] predlažu korištenje podmreže sijamske mreže s poboljšanom verzijom trojnog gubitka (*SymTrimplet*) koja dodatno uzima u obzir i udaljenost vektora značajki negativnog i pozitivnog primjera $\|f(I_1) - f(I_3)\|_2$.

Kombinacija različitih vrsta značajki

U nastojanju da unaprijede robusnosti i preciznosti algoritama praćenja objekata, istraživači u radovima [92, 103, 131] prilikom izračuna sličnosti kombiniraju vizualne značajke dobivene konvolucijskm neuronskim mrežama s informacijama o kretanju i obliku objekta. Yu *et al.* [92] integriraju značajke dobivene GoogLeNet konvolucijskom mrežom s prostornim značajkama koje opisuju kretanje i oblik objekta dobivenim primjenom Kalmanovog filtera. U radu [103], vizualne značajke dobivene konvolucijskom neuronskom mrežom koriste se zajedno s dodatnim značajkama koje karakteriziraju veličinu objekta, njegovu poziciji i dinamiku kretanja. Bae i Yoon [131] također predlažu kombinaciju konvolucijskih značajki s značajkama modela kretanja i oblika objekta koji se prati. Vizualne značajke dobivene rezidualnom konvolucijskom mrežom se u [94] koriste zajedno s informacijama o kretanju. U radu [102], vizualne značajke i značajke kretanja izdvajaju se zasebno koristeći konvolu-

cijsku VGG16 mrežu i LSTM mrežu te se uče integrirati u jedan vektor značajki pomoću Metric-Net mreže s trostrukim gubitkom.

3.2.4. Mjere sličnosti/udaljenosti

Kako bi se detekcije u koraku asocijacije mogle pridružiti odgovarajućim objektima, potrebno je na neki način izračunati sličnost detekcija iz novog okvira i postojećih putanja. Za izračun sličnosti u obzir se može uzimati samo jedna relevantna komponenta poput kretanja [93, 132] ili kombinirati više različitih komponenti poput vizualnog izgleda, dinamike kretanja i oblika objekta [131, 92, 103, 94, 67].

Jedna od najčešće korištenih mjera prilikom izračuna sličnosti je ***IoU*** detektiranog graničnog okvira A i predviđenog graničnog okvira B:

$$IoU(A,B) = \frac{|A \cup B|}{|A \cap B|}, \tag{3.6}$$

gdje $|\cdot|$ označava površinu. U [93, 132, 133] IoU se koristi kao jedina mjera sličnosti, a u [125, 94, 101, 105, 67] se nadopunjuje dodatnim komponentama poput mjera vizualne sličnosti ili dodatnih informacijama o kretanju, obliku ili poziciji objekta.

Yu *et al.* [92] predlažu mjeru sličnosti koja kombinira vizualne značajke sa značajkama kretanja i oblika detekcije A i putanje B. Neka (x,y), w i h redom označavaju koordinate centra, visinu i širinu odgovarajućih graničnih okvira. Tada se ***Yu sličnost*** definira na sljedeći način:

$$s(A,B) = s_{app}(A,B) \cdot s_{mot}(A,B) \cdot_{shp} (A,B), \tag{3.7}$$

gdje je

$$s_{app}(A,B) = \cos(feat_A, feat_B), \tag{3.8}$$

$$s_{mot}(A,B) = e^{-w_1\left(\left(\frac{x_A - x_B}{w_A}\right)^2 + \left(\frac{y_A - y_B}{h_A}\right)^2\right)}, \tag{3.9}$$

$$s_{shp}(A,B) = e^{-w_2\left(\frac{|h_A - h_B|}{h_A + h_B} + \frac{|w_A - w_B|}{w_A + w_B}\right)}. \tag{3.10}$$

Značajke $feat_A$ i $feat_B$ koje se koriste u izračunu s_{app} su zapravo 128-dimenzionalni vektori dobiveni pomoću konvolucijske neruonske mreže slične GoogLeNet mreži, a za w_1 i w_2 se redom koriste vrijednosti 0.5 i 1.5.

Kosinusna sličnost vektora značajki, definirana kao

$$\cos(feat_A, feat_B) = \frac{\langle feat_A,\ feat_B \rangle}{\|feat_A\|_2 \, \|feat_B\|_2}, \tag{3.11}$$

često se koristi kao mjera vizualne sličnosti [92, 128, 94, 116, 85, 86].[2] Pored kosinusne

[2]Umjesto kosinusne sličnosti, u nekim radovima se računa kosinusnu udaljenost $d_{\cos}(feat_A, feat_B)$ defini-

sličnosti, još se često koristi i ***euklidska udaljenost*** vektora značajki [125, 130, 94, 129]. Prednost kosinusne sličnosti u odnosu na euklidsku udaljenost je u njenoj neosjetljivosti na skaliranje podataka i efikasnosti u visokodimenzionalnim prostorima. Vektori se smatraju sličnima ako imaju istu orijentaciju u prostoru, bez obzira na njihovu veličinu. S druge strane, euklidska udaljenost predstavlja stvarnu geometrijsku udaljenost obuhvaćajući razlike kako u veličini tako i u orijentaciji vektora.

Kako bi integrirali informaciju o kretanju u izračun sličnosti, u radovima [85, 94, 134, 86] koriste ***Mahalanobisovu udaljenost*** [135] koja uzima u obzir korelaciju promatranih varijabli i njihove varijance. Mahalonobisova udaljenost definirana je s

$$d_M(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^\top S^{-1}(\boldsymbol{x}-\boldsymbol{y})}, \tag{3.12}$$

gdje su $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$, a S matrica kovarijance.

3.2.5. Asocijacija

U koraku asocijacije, cilj je odrediti kojem od do sada praćenih objekata odgovara detekcija iz trenutnog okvira, ili alternativno, predstavlja li detekcija neki novi objekt koji je tek potrebno početi pratiti. Problem optimalnog pridruživanja detekcija postojećim putanjama može se formulirati kao problem pridruživanja maksimalne težine u potpunom težinskom bipartitnom grafu.[3]

Neka je $O = \{o_1,\ldots,o_n\}$ skup postojećih putanja objekata, a $D = \{d_1,\ldots,d_m\}$ skup detekcija. Nadalje, neka je $G = (V,E)$, gdje je $V = O \cup D$ skup vrhova, a $E = O \times D$ skup bridova, potpuni težinski bipartitni graf s funkcijom težine $w : E \to \mathbb{R}_0^+$ koja svakom bridu (o_i,d_j) pridruži izračunatu sličnost putanje o_i i detekcije d_j, odnosno $w(o_i,d_j) = s(o_i,d_j)$, gdje je s odabrana mjera sličnosti. Pridruživanje M u grafu G je podskup bridova $M \subseteq E$ takav da za svaki vrh $v \in V$ vrijedi da je incidentan najviše jednom bridu iz M. Težina pridruživanja M jednaka je sumi težina svih bridova $e \in M$:

$$w(M) = \sum_{e \in M} w(e). \tag{3.13}$$

Cilj je za dani bipartitni graf G odrediti pridruživanje M maksimalne težine.

Navedeno, optimalno pridruživanje može se pronaći pomoću mađarskog algoritma. ***Mađarski algoritam*** [136], također poznat i pod nazivom *Kuhn-Munkersov* algoritam, problem pridruživanja rješava u polinomijalnom vremenu $O(n^3)$ gdje je $n = |O| = |D|$ [137]. Iako datira još iz 1955. godine, mađarski algoritam je najčešće korišteni algoritam za asocijaciju detekcija i putanja u algoritmima praćenja [92, 130, 103, 131, 93, 101, 100, 96, 85, 102, 86,

rana s $d_{\cos}(feat_A, feat_B) = 1 - \cos(feat_A, feat_B)$.

[3] Navedeno se lako može prilagoditi problemu pridruživanja minimalne težine kada se umjesto sličnosti koriste različite mjere udaljenosti.

133]. U slučaju da pretpostavka $|O| = |D|$ ne vrijedi, u particiju vrhova manje kardinalnosti može se dodati odgovarajući broj fiktivnih čvorova koji se onda povezuju sa svim vrhovima iz druge particije bridovima težine 0, odnosno minimalne vrijednosti sličnosti.

Dani bipartitni graf može se reprezentirati $n \times n$ matricom susjedstva $C = [c_{i,j}]$ kojoj retci odgovaraju putanjama, a stupci detekcijama pri čemu je $c_{i,j} = w(o_i, d_j)$. Tada je mađarska metoda dana sljedećim algoritmom.[4]

Algoritam 1 Mađarski algoritam

Ulaz: $n \times n$ matrica susjedstva $C = [c_{i,j}]$
Izlaz: optimalno pridruživanje redaka i stupaca matrice

1. *Svođenje problema maksimizacije na problem minimizacije:*

   Vrijednost $c_{i,j}$ svake ćelije zamijeniti razlikom $C_{max} - c_{i,j}$ gdje je C_{max} maksimalna vrijednost dane matrice susjedstva.

2. *Redukcija redaka*:

   Od svake vrijednosti u retku oduzeti minimalnu vrijednost tog retka.

3. *Redukcija stupaca*:

   Od svake vrijednosti u stupcu oduzeti minimalnu vrijednost tog stupca.

4. *Minimalna pokrivenost*:

   Minimalnim brojem vertikalnih i horizontalnih linija prectrtati retke i stupce matrice tako da sve nule budu precrtane.

5. **Ako** je broj nacrtanih linija jednak n:
6. **vrati** $\{(i, j) : c_{i,j} = 0\}$
7. **Inače**:
8. Pronađi najmanji element matrice koji nije precrtan linijama.
9. Taj element **oduzmi** od svih elemenata redaka koji **nisu** precrtani.
10. Taj element **dodaj** svim elemenatima stupaca koji **su** precrtani.
11. Vrati se na liniju 4.

Umjesto mađarskog algoritma, u [125, 138] koriste jednostavn pohlepni algoritam koji u svakom koraku pridružuje parove detekcija i putanja koji imaju najveću vrijednost izračunate sličnosti. Zbog njene efikasnosti, pohlepna metoda asocijacije koristi se i u [139] za postizanje online praćenje u stvarnom vremenu. Azizpour *et al.* [140] predstavljaju algoritam asocijacije koji se temelji na problemu pridruživanja grafa putanja i grafa detekcija, koristeći pri tom kvadratno programiranje i graf neuronske mreže (engl. *graph neural networks*). Liu *et al.* [141] i Milan *et al.* [110] koriste LSTM mrežu za asocijaciju, dok Yoon *et al.* [142] koriste neuronsku mrežu koja se sastoji od enkodera s potpuno povezanim slojevima i

[4] Ako matrica susjedstva C sadrži udaljenosti, a ne sličnosti, onda se traži pridruživanje minimalne težine te se preskaže prvi korak algoritma.

dvosmjerne LSTM mreže u dekoderu.

3.2.6. Upravljanje putanjama

U koraku upravljanja putanjama provode se sljedeće operacije: *ažuriranje stanja postojećih putanja* kojima je uspješno pridružena detekcija u koraku asocijacije, *incijalizaciju putanja* za detekcije koje nisu uspješno pridružene postojećim putanjama, *završavanje putanja* objekata koji su napustili scenu.

- ***Ažuriranje stanja putanja:*** Ovaj proces uključuje osvježavanje vrijednosti promatranih varijabli putanja temeljem vrijednosti pridruženih detekcija, odnosno novih mjerenja. Primjerice, ažuriranje stanja koje opisuje kretanje objekta pomoću Kalmanovog filtera ili prilagodba vizualnih značajki putanja vizualnim značajkama pridruženih detekcija [85].

- ***Inicijalizacija novih putanja:*** Svaka nepridružena detekcija ne rezultira odmah stvaranjem nove putanje. Nova putanja se obično ne dodaje odmah u skup postojećih i aktivnih putanja budući, budući da se može raditi o lažno pozitivnoj detekciji. Često se za nepridružene detekcije stvaraju probne putanje, koje se kasnije inicijaliziraju ako prežive testno razdoblje [93, 85]. Na primjer, u [85], nova putanja se stvara samo ako odgovarajuća detekcija preživi dva uzastopna okvira videozapisa. Nasuprot tome, Zhang *et al.* [100] odmah inicijaliziraju putanje, ali samo za nepridružene detekcije visoke pouzdanosti.

- ***Završavanje putanja:*** Postojeće putanje obično se ne prekidaju odmah ako im se u jednom okviru nije uspjelo pridružiti detekciju, jer to može biti rezultat privremene zaklonjenosti praćenog objekta ili neuspješne detekcije. Umjesto toga, putanje se obično završavaju nakon određenog broja uzastopnih neuspješnih pokušaja pridruživanja detekcija [93, 138, 85, 101, 86, 133]. Na primjer, u [85, 101, 86], putanje se završavaju nakon 30 uzastopnih neuspješnih pridruživanja. S druge strane, u radu Kim *et al.* [139], putanja se završava ako joj u nizu od N_{miss} uzastopnih okvira nije uspješno pridružena niti jedna detekcija ili ako je broj pridruženih detekcija manji od broja nedostajućih detekcija za tu putanju.

Mahmoudi *et al.* [103] kao izlaz prikazuju samo stabilne putanje koje ispunjavaju sljedeće kriterije: (a) najviše τ_{inv} okvira putanji nije uspješno pridružena detekcija, (b) omjer broja okvira tijekom postojanja putanje u kojima je putanji pridružena detekcija i broja okvira u kojima to nije bio slučaj veći je od τ_{vis}, (c) prosječna cijena pridruživanja detekcije putanji manja je od τ_{cost}. Rad [110] koristi rekurentnu neuronsku mrežu za predviđanje vjerojatnosti ε postojanja putanje u sljedećem okviru na temelju prethodno prikupljenih informacija. Vrijednost ε koristi se za odluku o inicijalizaciji ili završavanju putanje objekta.

3.3. Popularni algoritmi

U nastavku se opisuju neki od najpopularnijih MOT algoritama koji se: (1) temelje na detekciji, (2) integriraju detekciju i određene korake praćenja, (3) zasnivaju na transformer arhitekturama neuronskih mreža.

3.3.1. Algoritmi temeljeni na detekciji

Trenutno, algoritmi praćenja temeljeni na detekciji ističu se kao najučinkovitija paradigma za zadatak praćenja više objekata [100, 101]. Njih karakterizira jednostavna struktura i interpretabilnost s jedne strane, dok s druge strane pokazuju preveliku ovisnost o performansama korištenog detektora

SORT

SORT [93] je brz i efikasan algoritam praćenja pogodan za izvođenje aplikacija u stvarnom vremenu koji se temelji na jednostavnim konceptima poput Kalmanovog filtera za predviđanje putanja i mađarskog algoritma za asocijaciju putanja i novih detekcija. Stanje objekta u Kalmanovom filteru definira se vektorom $\boldsymbol{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^\top$, gdje su (u, v) koordinate centra, s površina i r omjer širine i visine graničnog okvira. Pretpostavlja se da je vrijednost r konstantna. Preostale tri varijable, $\dot{u}$, $\dot{v}$ i $\dot{s}$, su odgovarajuće derivacije po vremenu. Za detekciju objekata u svakom okviru videozapisa koristi se Faster R-CNN [26] detektor. Koristeći isključivo *IoU* vrijednost detektiranih graničnih okvira i graničnih okvira predviđenim Kalmanovim filterom, mađarskim algoritmom se detekcije pridružuju odgovarajućim putanjama pri čemu se za pridruživanje dodatno zahtijeva da je vrijednost *IoU* veća od predefinirane minimalne vrijednosti IoU_{min}. Nove putanje se inicijaliziraju za svaku nepridruženu detekciju nakon probnog perioda, a završavaju čim im u jednom koraku nije pridružena detekcija kako bi se očuvala efikasnost algoritma. Ako se objekt čija je putanja završena ponovno pojavi u videozapisu, praćenje se nastavlja s novim identifikatorom.

DeepSORT

Wojke *et al.* predlažu poboljšanu verziju SORT algoritma, *DeepSORT* algoritam, koja u SORT integrira vizualnu informaciju kodiranu 128-dimenzionalnim vektorom značajki dobivenim pomoću predtrenirane konvolucijske mreže s rezidualnim blokovima. Time je omgućeno praćenje objekata i kroz dulje periode zaklonjenosti. Nadalje, za razliku od SORT algoritma, DeepSORT koristi kaskadno pridruživanje u kojem se detekcije prvo pridružuju mlađim putanjama kojima su nedavno pridružene detekcije, a zatim starijim putanjama kojima već određen broj uzastopnih okvira nije pridružena detekcija. Prilikom pridruživanja koristi se metrika koja kombinira informaciju o kretanju objekta i vizualnu informaciju:

$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda)d^{(2)}(i,j)$, gdje je $d^{(1)}(i,j)$ kvadrat Mahalanobisove udaljenosti predviđenog Kalmanovog stanja za putanju i i novog mjerenja (detekcije) j, a $d^{(2)}(i,j)$ minimalna kosinusna udaljenost vizualnih značajki detekcije j i značajki zadnjih 100 pridruženih detekcija putanji i. Koristeći unaprijed definirane granične vrijednosti $t^{(1)}$ i $t^{(2)}$ za $d^{(1)}(i,j)$ i $d^{(2)}(i,j)$ redom, filtriraju se neprihvatljiva pridruživanja, odnosno pridruživanja za koja vrijedi $d^{(1)}(i,j) > t^{(1)}$ ili $d^{(2)}(i,j) > t^{(2)}$. Budući da se u eksperimentima koristi $\lambda = 0$, informacija o kretanju $d^{(1)}$ se koristi samo za filtriranje neprihvatljivih pridruživanja. U završnom koraku, nakon kaskadnog pridruživanja, koristeći samo *IoU* udaljenost, pokušavaju se međusobno pridružiti nepridružene detekcije i putanje. Putanje se završavaju tek ako im 30 uzastopnih okvira nije uspiješno pridružena detekcija, a nove putanje za nepridružene detekcije se inicijaliziraju nakon probnog perioda od tri uzastopna okvira.

StrongSORT

StrongSORT [99] algoritam unaprijeđuje DeepSORT na različite načine: (1) zamjenjujuje Faster R-CNN detektor s YOLOX-X [39] detektorom, (2) koristi ECC [143] model za kompenzaciju pokreta kamere, (3) implementira NSA kalmanov filter umjesto običnog Kalmanovog filtera, (4) za ekstrakciju vizualnih značajki koristi BoT ekstraktor [144] s ResNeSt50 okosnicom umjesto jednostavne konvolucijske mreže, (5) za opis vizualnog izgleda putanje koristi se eksponencijalni pomični prosjek vizualnih značajki pridruženih detekcija, (6) kao mjeru udaljenosti koristi linearnu kombinaciju $c = \lambda d_{izgled} + (1-\lambda)d_{kretanje}$, pri čemu je $\lambda = 0.98$, te se vrijednost $d_{kretanje}$ ne koristi samo za filtriranje neprihvatljivih pridruživanja već se i direktno integrira u konačnu udaljenost, (7) za asocijaciju koristi jednostavno linearno pridruživanje, umjesto kaskadnog pridruživanja. Dodatno, predložena su dva jednostavna i učinkovita algoritma za post-procesiranje: *AFLink* metoda za globalnu asocijaciju putanja koja koristi isključivo prostorno-vremenske informacije i *GSI* algoritam za interpolaciju putanja baziran na Gaussovoj regresiji procesa, koji se koristi za ublažavanje nepravilnosti nastalih zbog nedostajućih detekcija. *StrongSORT++* predstavlja nadogradnju StrongSORT algoritma u kojoj su impmlementirani navedeni post-procesni koraci.

ByteTrack

U [100], autori predložu novu metodu asocijacije koja u obzir uzima gotovo sve detektirane granične okvire, čak i one s malom pouzdanošću, i implementiraju je u **ByteTrack** algoritam. Granični okviri s malom pouzdanošću mogu indicirati postojanje objekata koji su djelomično zaklonjeni pa njihovo filtriranje može dovesti do fragmentacije putanja praćenih objekata. Predložena metoda asocijacije, koja se temelji na mađarskom algoritmu, odvija se u dvije faze. U prvoj fazi putanjama se pokušavaju pridružiti detekcije visoke pouzdanosti koristeći ili *IoU* ili udaljenost vektora vizualnih značajki detekcija i predviđanja Kalmanovog filtera. U drugoj fazi se neuparenim putanjama pridružuju detekcije koje imaju nižu pouzdanost

koristeći isključivo *IoU* kao mjeru sličnosti budući da takvi granični okviri obično sadrže zaklonjene objekte. Nakon asocijacije, inicijaliziraju se nove putanje za neuparene detekcije visoke pouzdanosti, a završavaju putanje kojima u posljednjih 30 okvira nije pridružena detekcija.

BoT-SORT

BoT-SORT algoritam [101] integrira sljedeće modifikacije u ByteTrack: (1) dodatno se upotrebljava kompenzacija pokreta kamere, (2) koristi se poboljšana verzija Kalmanovog filtera koja u vektoru stanja koristi direktno visinu i širinu graničnog okvira, umjesto visine i omjera širine i visine graničnog okvira, (3) u prvoj fazi asocijacije, u kojoj se putanjama pridružuju detekcije visoke pouzdanosti, koristi se mjera udaljenosti koja povezuje vizualne značajke i značajke kretanja. Cijena $c_{i,j}$ pridruživanja detekcije j putanji i u prvoj fazi asocijacije dana je s $c_{i,j} = \min\{d_{i,j}^{IoU}, \widehat{d}_{i,j}^{cos}\}$, gdje je $d_{i,j}^{IoU}$ IoU udaljenost predviđenog graničnog okvira za putanju i i detektiranog graničnog okvira j, a $\widehat{d}_{i,j}^{cos}$ predložena mjera vizualne udaljenosti. Nova mjera vizualne udaljenosti definira na sljedeći način:

$$\widehat{d}_{i,j}^{cos} = \begin{cases} 0.5 \cdot d_{i,j}^{cos}, & (d_{i,j}^{cos} < \theta_{emb}) \ \& \ (d_{i,j}^{IoU} < \theta_{IoU}) \\ 1, & \text{inače} \end{cases}, \tag{3.14}$$

pri čemu je $d_{i,j}^{cos}$ kosinusna udaljenost vektora značjki detekcije j i eksponencijalnog pomičnog prosjeka vektora značajki detekcija pridruženih putanji i, $\theta_{IoU} = 0.5$ i $\theta_{emb} = 0.25$ granične vrijednosti koje se koriste za odbacivanje pridruživanja koja su slabo vjerojatna.

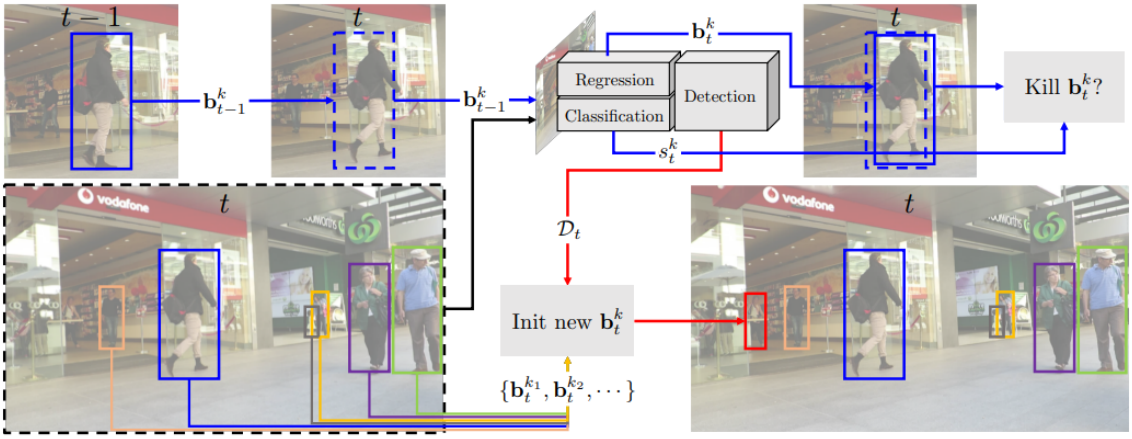
3.3.2. Algoritmi zajedničke detekcije i praćenja

Razvojem simultanog učenja neuronskim mrežama (engl. *multi-task learning*), algoritmi zajedničke detekcije i praćenja koji integriraju detekciju s različitim koracima praćenja u jedinstvenu mrežu, postali su sve privlačniji istraživačkoj zajednici [86]. Ovi algoritmi ističu se svojom unificiranom arhitekturom, efikasnošću i performansama koje su usporedive s performansama algoritama temeljenih na detekciji [69, 99, 101].

Tracktor, Tracktor++

Bergman *et al.* [87], predlažu algoritam koji detektor poput Faster R-CNN-a pretvara u model za praćenje naziva *Tracktor*. Osnovna ideja je iskoristiti regresijsku glavu detektora za dobivanje sljedeće pozicije praćenog objekta. U prvom okviru $t = 0$ videozapisa nove putanje se inicijaliziraju za sve detekcije $B_0 = \{\boldsymbol{b}_0^1, \boldsymbol{b}_0^2, \ldots, \boldsymbol{b}_0^n, \}$. Nakon prvog okvira skup aktivnih putanja dan je s $\mathcal{T}_{active} = \{T_1, \ldots, T_n\}$, gdje je $T_k = \{\boldsymbol{b}_0^k\}$ putanja objekta k. U svakom sljedećem okviru $t > 0$, za svaku aktivnu putanju T_k dohvaća se odgovarajući granični okvir

$\boldsymbol{b}_{t-1}^k$ iz prethodnog koraka. $\boldsymbol{b}_{t-1}^k$ se postavlja na okvir t danog videozapisa[5] te se *RoI pooling* primjenjuje na danu regiju okvira t kako bi se dobile mape značajki koje se onda kao ulaz šalju u regresijsku i klasifikacijsku glavu detektora. Izlaz regresijske glave je fino podešeni granični okvir $\boldsymbol{b}_t^k$. Dobiveni fino podešeni okviri svih aktivnih putanja se filtriraju koristeći NMS algoritam, a zatim se preostali granični okviri $\boldsymbol{b}_t^k$ dodaju odgovarajućim putanjama ($T_k = T_k \cup \{\boldsymbol{b}_t^k\}$) ako je vjerojatnost s_t^k da se u predloženoj regiji okvira t nalazi objekt od interesa, koju predviđa klasifikacijska glava, veća od unaprijed definirane granične vrijednost σ_{active}. U protivnom, putanja se završava. Dodatno, da bi se omogućilo praćenje objekata koji se naknadno pojavljuju u sceni, detektor također predviđa skup detekcija $\mathcal{D}_t$ za okvir t. Za sve detekcije $d \in \mathcal{D}_t$ za koje vrijedi $IoU(d, \mathbf{b}_t^k) < \lambda_{new}$ za sve granične okvire $\mathbf{b}_t^k$ aktivnih trajektorija iz $\mathcal{T}_{active}$. Ilustracija navedenog procesa prikazana je na Slici 3.5.



Slika 3.5: Dijagram toka Tractor algoritma (slika preuzeta iz [87]).

Tractor++ [87] nadograđuje osnovni *Tractor* algoritam uvođenjem *modela kretanja* koji kompenzira nizak broj okvira po sekundi (pretpostavka konstantne brzine) i veće pomake kamere (ECC algoritam za kompenzaciju pokreta kamera) i *vizualnih značajki* koje omogućuju reidentifikaciju objekata čije su putanje deaktivirane u posljednjih F_{reID} okvira.

CenterTrack

CenterTrack [89] algoritam praćenja zasniva se na arhitekturi CenterNet detektora s DLA [145] okosnicom, koja je modificirana tako da osim trenutnog okvira videozapisa $I^{(t)}$, kao ulaz dodatno prima i prethodni okvir $I^{(t-1)}$ te informaciju o praćenim objektima iz prethodnog okvira $T^{(t-1)} = \{t_0^{(t-1)}, t_1^{(t-1)}, \dots\}$. Svaki praćeni objekt opisan je uređenom četvorkom $t = (\boldsymbol{p}, \boldsymbol{s}, c, id)$, gdje su $\boldsymbol{p} = (x, y)$ koordinate središta graničnog okvira, $\boldsymbol{s} = (w, h)$ širina i visina graničnog okvira, c pouzdanost detekcije, te id jedinstveni identifikator objekta. Također, dodana je grana koja predviđa pomak objekta detektiranog na lokaciji $\hat{\boldsymbol{p}}^{(t)}$ između

[5] Pretpostavka je da su pomaci objekata između susjednih okvira videozapisa minimalni, odnosno da je broj okvira u sekundi danog videozapisa velik.

trenutnog i prethodnog okvira: $\hat{\boldsymbol{d}}^{(t)} = \hat{\boldsymbol{p}}^{(t)} - \hat{\boldsymbol{p}}^{(t-1)}$. Za asocijaciju objekata kroz vrijeme koristi se jednostavno pohlepno pridruživanje: svakoj detekciji u trenutnom okviru pridružuje se najbliža neuparena detekcija iz prethodnog okvira na poziciji kompenziranoj za predviđeni pomak. Za neuparene detekcije iz trenutnog okvira, inicijaliziraju se nove putanje ako u zadanom radijusu $\mathcal{K}$ dane detekcije, ne postoji neuparena detekcija iz prethodnog okvira. S obzirom na to da su arhitekturne promjene CenterTrack algoritma u odnosu na CenterNet detektor minorne (četri dodatna ulazna kanala i dva izlazna), težine detekcijskog dijela moguće je inicijalizirati pomoću težina predtreniranog CenterNet detektora, dok se težine dodatnih grana inicijaliziraju na slučajan način. CenterTrack algoritam za praćenje objekata svodi se na propagaciju odgovarajućih identiteta objekata među susjednim okvirima videozapisa, čime se postiže jednostavnost i brzina izvođenja, nauštrb sposobnosti reidentifikacije davno izgubljenih objekata.

FairMOT

Zhang *et al.* [86] predlažu *FairMOT* algoritam za praćenje objekata koji se također temelji na CenterNet detektoru. Za razliku od CenterTrack algoritma, FairMOT integrira i vizualne značajke za reidentifikaciju objekata. Koristi se unaprijeđena DLA-34 [145] okosnica s homogenim granama za detekciju i reidentifikaciju kako bi se eliminirala nepravedna prednost detekcije nad reidentifikacijom koja je često prisutna kod sličnih algoritama praćenja. Grana za detekciju na okosnicu dodaje tri paralelne grane za predviđanje mapa ključnih točaka - središta, pomaka i veličine graničnih okvira. Grana za reidentifikaciju šalje izlaz okosnice u konvolucijski sloj s 128 filtera, dodjeljujući tako svakom pikselu 128-dimenzionalni vektor značajki koji opisuje objekt čije je središte u tom pikselu. Tijekom treniranja, obe grane se simultano treniraju s gubitkom koji automatski balansira zadatak detekcije i reidentifikacije. Sami proces praćenja objekata obuhvaća sljedeće korake: (1) inicijalizacija novih putanja za sve detektirane objekte u prvom okviru, (2) korištenje Kalmanovog filtera za predviđanje sljedeće pozicije praćenih objekata, (3) *prva faza pridruživanja*: mađarskim algoritmom se nove detekcije povezuju s putanjama koristeći pri tom, kao u DeepSORT-u, mjeru udaljenosti koja kombinira Mahalanobisovu udaljenost predviđenog i detektiranog graničnog okvira te kosinusnu udaljenost reID vektora značajki, s graničnom vrijednosti pridruživanja $\tau = 0.4$, (4) *druga faza pridruživanja*: pridružuju se preostale neuparene detekcije i putanje koristeći IoU predviđenog i detektiranog graničnog okvira kao mjeru udaljenosti uz graničnu vrijednosti pridruživanja $\tau = 0.5$, (5) inicijalizacija novih putanja za neuparene detekcije, (6) završavanje putanja objekata kojima 30 uzastopnih okvira nije pridružena detekcija.

JDE

Wang *et al.* [85] predlažu *Joint Detector and Embeddings (**JDE**)* algoritam praćenja koji zadatak detekcije objekata i ekstrakcije vizualnih značajki kombinira u jednoj zajedničkoj

mreži. Kao osnovna arhitektura koristi se piramidalna mreža značajki (FPN) s mapama značajki na tri različite skale za obradu objekata različith veličina. Osim glava za klasifikaciju i regresiju graničnih okvira, JDE također koristi i glavu za generiranje vektora vizualnih značajki detektiranih objekata. Za vrijeme treniranja dane mreže minimizira se gubitak koji automatski kombinira gubitke tri glave neuronske mreže koristeći pri tom parametre nesigurnosti pojedinih zadataka koji se uče. Sami algoritam praćenja obuhvaća sljedeće korake: (1) inicijalizaciju novih putanja za detektirane objekte u prvom okviru videozapisa pri čemu se početno stanje putanje i koje odgovara vizualnim značajkama e_i postavlja na vrijednost vektora značaajki f_i^0 dane detekcije, (2) Kalmanovim filterom predviđa se sljedeća pozicija praćenih objekata, (3) *mađarskim algoritmom* se nove detekcije povezuju s putanjama koristeći mjeru udaljenosti $C = \lambda d_{app} + (1-\lambda) d_{mot}$ koja kombinira Mahalanobisovu udaljenost predviđenog i detektiranog graničnog okvira (d_{app}) te kosinusnu udaljenost vizualnih značajki putanje i dane detekcije, (4) varijable vezane za kretanje objekta ažuriraju se koristeći Kalmanov filter, a vizualne značajke putanje ažuriraju se vektorom značajki pridružene detekcije $e_i^t = \alpha e_i^{t-1} + (1-\alpha) f_i^t$, $\alpha = 0.9$, (5) nove putanje inicijaliziraju se za neuparene detekcije koje se uzastopno pojavljuju u dva okvira, (6) završavaju se putanje kojima 30 uzastopnih okvira nije pridružena detekcija.

3.3.3. Algoritmi koji koriste transformere

Transformer [146], arhitektura neuronskih mreža koja se zasniva na *ključ-vrijednost* mehanizmu pozornosti (engl. *attetnion*), istaknula se u području obrade prirodnog jezika [81]. U usporedbi s klasičnim LSTM i RNN modelima koji elemente niza obrađuju sekvencijalno, transformeri omogućuju značajniju paralelizaciju. Nadalje, mehanizam pozornosti omogućuje modelu da se fokusira na relevantne dijelove ulaza i učinkovitije modelira daleke ovisnosti u podacima.

Zahvaljujući svojoj elegantnoj strukturi i impresivnim performansama, primjena transformera postala je atraktivna i u domeni računalnog vida. Njihova sposobnost prijenosa informacija kroz vremensku dimenziju otvara mogućnost za doprinos u različitim zadacima koji obuhvaćaju obradu vizualnih podataka tijekom vremena, poput praćenja objekata [133].

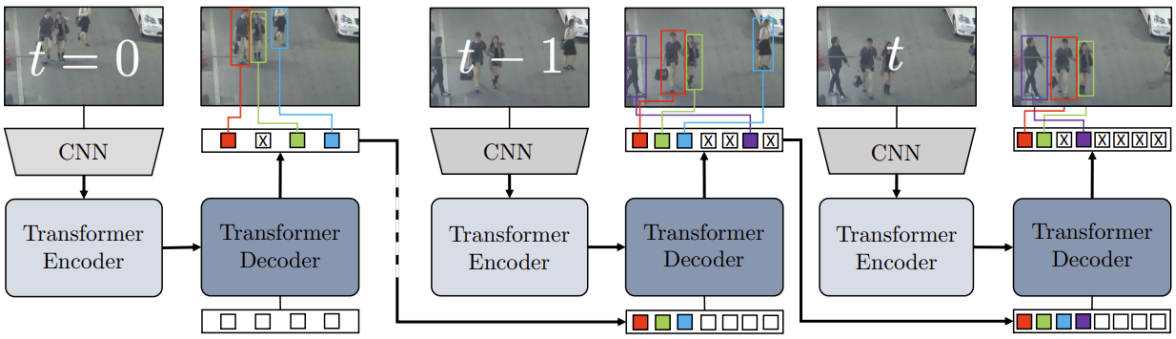
TransTrack

TransTrack algoritam [133] prvi primjenjuje transformer arhitekturu za izazovni zadatak praćenja više objekata u video zapisima. Predstavljeni model sastoji se od sljedećih komponenti: (1) *konvolucijske okosnice* koja iz trenutnog okvira videozapisa ekstrahira mapu vizualnih značajki, (2) *kodera* koji prima mape značajki trenutnog i prethodnog okvira te generira kompozitne značajke, (3) *dva paralelna dekodera* koji kao ulaz primaju specifične upite za svoju zadaću i koriste zajedničke ključeve, odnosno mape značajki koje generira koder. Prvi dekoder prima naučene upite vezane uz objekte (*"object query"*) kao ulaz i producira gra-

nične okvire objekata (detekcije) u trenutnom okviru. Značajke koje sadrže informacije o vizualnom izgledu i kretanju objekata detektiranih u prethodnom okviru proslijeđuju se na ulaz drugog dekodera kao upit putanja (*"track query"*). Temeljem tih informacija, drugi dekoder predviđa lokacije prethodno praćenih objekata u trenutnom okviru. Pridruživanje detektiranih graničnih okvira prvog dekodera i predviđenih graničnih okvira putanja izvršava se primjenom mađarskog algoritma, pri čemu se IoU koristi kao mjeru sličnosti. Za detekcije koje nisu uparene, incijaliziraju se nove putanje. Putanje se završavaju ako im u 32 uzastopna okvira nije pridružena detekcija.

TrackFormer

Meinhardt *et al.* [147] predlažu *TrackFormer* model zasnovan na koder-dekoder arhitekturi transformera koji istovremeno obavlja detekciju i praćenje te za asocijaciju objekata također koristi mehanizam pozornosti. Praćenje se odvija u četiri uzastopna koraka: (1) konvolucijskom neuronskom mrežom generira se mapa značajki ulaznog okvira videozapisa, (2) koder transformera kodira značajke okvira koristeći mehanizam samopozornosti (engl. *self-attention*), (3) pomoću mehanizama samopozornosti i unakrsne pozornosti (engl. *cross-attention*) dekodera transformera ulazni upiti se dekodiraju u izlazne vektore značajki, (4) izlaz dekodera šalje se u višeslojni perceptron koji predviđa granične okvire i klase objekata. Dekoder transformera koristi dvije vrste ulaznih upita: *upiti objekata* koji omogućuju inicijalizaciju novih putanja i *upiti putanja* odgovorni za praćenje objekata kroz vrijeme. U okviru $t = 0$ dekoder inicijalizira nove upite putanja iz izlaznih vektora za N_{object} *upita objekata* s rezultatom klasifikacije većim od σ_{object}. U svakom sljedećem okviru $t > 0$, kao ulaz u dekoder šalje se N_{object} upita objekata i N_{track} upita putanja. Broj upita putanja N_{tracks} se mijenja tijekom vremena kako se nove putanje inicijaliziraju, a neke stare završavaju. Ilustacija algoritma prikazana je na Slici 3.6.



Slika 3.6: TrackFormer algoritam. Ulaz u dekoder transformera je N_{object} upita objekata (bijelo) i N_{tracks} upita putanja (obojano). Izlaz dekodera su značajke koje inicijaliziraju nove putanje objekata (obojano) ili predviđaju "pozadinu" (prekriženo) (slika preuzeta iz [147]).

3.4. Evaluacija MOT algoritama

Za efikasnu i objektivnu evaluaciju MOT algoritama potrebne su nam kvantitativne metrike koje mjere sposobnost algoritama da u svakom okviru videozapisa pronađu i precizno lokaliziraju sve objekte od interesa te konzistentno prate njihove putanje i jedinstvene identifikatore kroz vrijeme [148]. S druge strane, javno dostupni referentni skupovi podataka (engl. *benchmark datasets*) pružaju standardiziranu osnovu za usporedbu različitih modela i tehnika potičići transparentnost, reproduktibilnost i napredak u istraživanju [60].

3.4.1. Referentni skupovi podataka

Fokus velikog broja referentnih skupova podataka za praćenje više objekata je na **detekciji i praćenju pješaka**. Jedan od prvih, ali i jednostavnijih, takvih skupova podataka je PETS2009 [149] skup podataka za evaluaciju nadzornih sustava za praćenje osoba na javnim površinama na otvorenom. Nešto recentniji i generalno najpopularniji, su skupovi podataka iz MOT izazova [66, 82, 150, 83].[6] MOT15 [66] prvi je opsežniji referentni skup podataka za praćenje koji obuhvaća videozapise iz drugih prethodno objavljenih skupova, uključujući i PETS2009. Skupovi MOT16/17 [82] i MOT20 [83] koji nasljeduju MOT15, sadrže nešto izazovnije videozapise u kojima je gustoća pješaka veća pa su objekti koji se prate češće zaklonjeni.

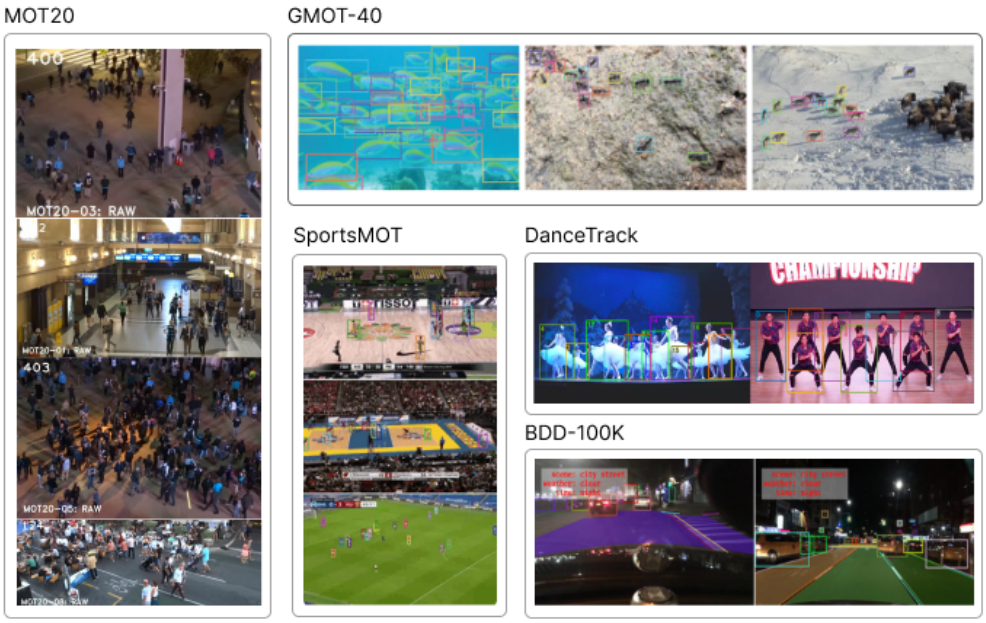
Pionir među referentnim skupovima za praćenje u domeni **autonomne vožnje** bio je KITTI [72, 151] skup podataka. Primjeri novijih i većih skupova za detekciju i praćenje pješaka i vozila uključuju UA-DETRAC [152], BDD100K [153], NuScenes [154], Waymo [155] i KITTI360 [156] skupove podataka. Većina navedenih skupova podataka, uz same videozapise, sadrži i dodatne informacije prikupljene iz okoline različitim senzorima (LIDAR, RADAR, GPS, IMU) [151, 153, 154, 155, 156].

Neki skupovi podataka objavljeni u posljednje vrijeme fokus stavljaju na određene aspekate MOT algoritama koji nisu adekvatno pokriveni drugim referentnim skupovima. GMOT-40 [157] se fokusira na praćenje različitih generičkih objekata deset različitih klasa[7] koje karakteriziraju iznenadnije i brže promjene stanja tijekom praćenja [67]. DanceTrack [68] uglavnom sadrži videozapise plesnih skupina u kojima pojedinci koji se prate imaju jako slične vizualne značajke, ali različite pokrete i artikulaciju, a SportsMOT [158] videozapise igrača tri različita sporta (košarka, odbojka, nogomet) koje karakterizira brzo i promjenjivo kretanje različitih brzina te sličan, ali razlikovan izgled [8].

[6] https://motchallenge.net/

[7] klase: avion, lopta, balon, ptica, brod, auto, riba, insekt, stoka, osoba

[8] Primjerice, svi igrači istog tima nose iste dresove, ali svaki igrač na svom dresu ima svoj broj.



Slika 3.7: Primjeri okvira videozapisa iz MOT20 [83], GMOT-40 [157], SportsMOT [158], DanceTrack [68] i BDD-100K [153] skupova podataka.

3.4.2. Metrike

Neka je $O = \{o_1, \ldots, o_n\}$ skup stvarnih putanja objekata, a $H = \{h_1, \ldots, h_m\}$ skup hipoteza (predviđanja) MOT algoritma. Stvarne putanje objekata i hipoteze reprezentirane su skupom detekcija O_{det} i H_{det} u svakom okviru videozapisa pri čemu je svakoj detekciji pridjeljen jedinstveni ID za taj okvir koji je konzistentan kroz vrijeme s ID vrijednostima detekcija iste putanje.[9] Detekciju objekta/hipoteze o_i/h_j u okviru t označavamo s $o_i^{(t)}/h_j^{(t)}$ pri čemu indeks i/j predstavlja odgovarajući ID detekcije.

Pogreške algoritama za praćenje mogu se klasificirati u tri kategorije: 1) *pogreške lokalizacije* koje nastaju kada predviđene detekcije nedovoljno precizno određuju položaje stvarnih objekata, 2) *pogreške detekcije* koje se javljaju kada algoritam predviđa detekcije koje u stvarnosti ne postoje ili propušta detektirati stvarne objekte, 3) *pogreške asocijacije* koje su rezultat pogrešnog povezivanja detekcija između okvira videozapisa bilo da algoritam isti ID pridjeli različitim stvarnim putanjama ili više različitih ID vrijednosti jednoj stvarnoj putanji.

MOTA

Bernardin *et al.* u [148] predstavljaju dvije **CLEAR MOT** metrike: **MOTA** (engl. *Multi-Object Tracking Accuracy*) metriku koja mjeri koliko dobro algoritam detektira objekte i predviđa putanje i **MOTP** (engl. *Multi-Object Tracking Precision*) metriku koja mjeri preciz-

[9] Kad se prati objekte iz više klasa, svakoj putanji se dodatno pridjeljuje i oznaka klase. Budući da se u slučaju višeklasne klasifikacije kod većine metrika računa srednja vrijednost metrike dobivene po klasama [159] u nastavku se pretpostavlja da se prate objekti samo jedne klase.

nost lokalizacije praćenih objekata. Unatoč svojim nedostacima, MOTA se ubrzo afirmirala kao primarna metrika za evaluaciju MOT algoritama [159, 55]. Kako bi se mogle izračunati vrijednost MOTA i MOTP metrika, u svakom okviru videozapisa potrebno je pridružiti stvarne detekcije objekata detekcijama hipoteza. Navedeno se radi na sljedeći način:

(1) Ako je sličnost s između detekcija $o_i^{(t)}$ i $h_j^{(t)}$ manja od unaprijed definirane granične vrijednosti α, onda pridruživanje detekcija $(o_i^{(t)}, h_j^{(t)})$ ***nije valjano***.[10]

(2) Sva pridruživanja detekcija $(o_i^{(t)}, h_j^{(t)})$ iz okvira t koja su valjana u okviru $t+1$, odnosno za koja vrijedi $s\left(o_i^{(t+1)}, h_j^{(t+1)}\right) \geq \alpha$, ostaju očuvana i u okviru $t+1$.

(3) Detekcije koje su ostale neuparene nakon (2), pokušavaju se pridružiti jedne drugima na način da se maksimizira njihova ukupna sličnost.[11] Navedeno se može napraviti pomoću *mađarskog (Kuhn-Munkresovog)* [136] *algoritma*.

Neka je $TP \subseteq O_{det} \times H_{det}$ skup svih pridruženih parova detekcija stvarnih objekata i hipoteza, $FP \subseteq H_{det}$ skup **lažno pozitivnih** (preostalih, neuparenih) detekcija hipoteza te $FN \subseteq O_{det}$ skup **lažno negativnih** (preostalih, neuparenih) detekcija stvarnih objekata. Nadalje, neka ID_s označava broj **zamjena identita** (engl. *Identity Switch*) tj. koliko je puta detekciji stvarnog objekta pridružena detekcija hipoteze čiji identifikator nije konzistentan identifikatoru hipoteze koja je tom istom objektu pridružena u prethodnim okvirima. Tada je

$$MOTA = 1 - \frac{|FN| + |FP| + ID_s}{|O_{det}|}, \quad MOTP = \frac{1}{|TP|} \sum_{(o,h) \in TP} s(o,h). \tag{3.15}$$

Prethodno definirani skupovi korišteni u izračunu CLEAR MOT metrika vizualizirani su na Slici 3.8.

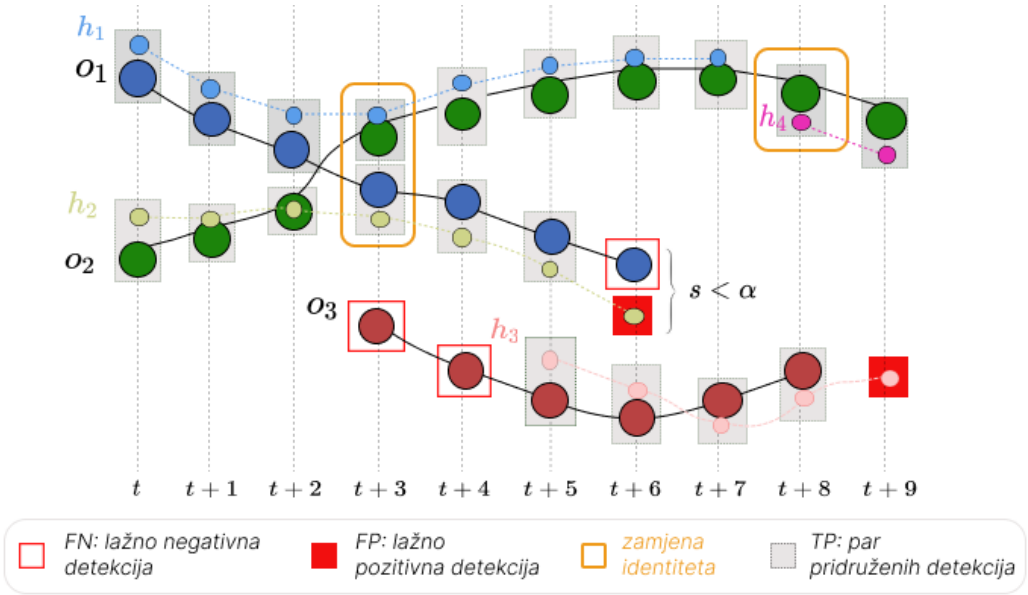
HOTA

Budući da MOTA ne mjeri pogrešku lokalizacije te prenaglašava važnost detekcije nauštrb asocijacije, u [159] je predložena nova **HOTA** (engl. *Higher Order Tracking Accuracy*) metrika koja na uravnotežen način kombinira sve aspekte evaluacije algoritama za praćenje.

Napomena: *U definicijama skupova* (3.16)*,* (3.17) *i* (3.18) *koje slijede, zbog preglednosti se izostavlja* $t' \in \{1, \ldots, N\}$ *gdje je* N *broj okvira danog videozapisa. Nadalje, ako je* z *skup,* $\{x \in z \mid \psi\}$ *označava skup* $\{x \mid x \in z \wedge \psi\}$.

Za dani par pridruženih detekcija $(o_i^{(t)}, h_j^{(t)}) \in TP$ skupovi točnih (TPA), lažno negativnih

[10] U slučaju 2D praćenja kao mjera sličnosti s najčešće se koristi IoU odgovarajućih graničnih okvira [78].
[11] Sva pridruživanja moraju biti valjana.



Slika 3.8: Vizualizacija CLEAR MOT koncepata. Na slici je prikazano deset uzastopnih okvira praćenja. o_1, o_2 i o_3 su stvarne putanje objekata, a $h_1, \ldots, h_4$ hipoteze algoritma.

(*FNA*) i lažno pozitivnih (*FPA*) asocijacija[12] definiraju se na sljedeći način:

$$TPA\left((o_i^{(t)}, h_j^{(t)})\right) = \left\{\left(o_i^{(t')}, h_j^{(t')}\right) \in TP\right\}, \tag{3.16}$$

$$FNA\left((o_i^{(t)}, h_j^{(t)})\right) = \left\{\left(o_i^{(t')}, h_k^{(t')}\right) \in TP \mid k \neq j\right\} \cup \left\{o_i^{(t')} \in FN\right\}, \tag{3.17}$$

$$FPA\left((o_i^{(t)}, h_j^{(t)})\right) = \left\{\left(o_k^{(t')}, h_j^{(t')}\right) \in TP \mid k \neq i\right\} \cup \left\{h_j^{(t')} \in FP\right\}. \tag{3.18}$$

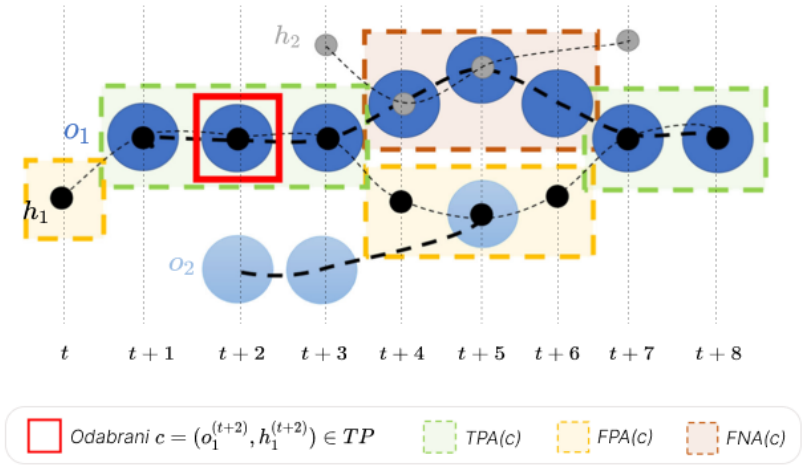
Definirani skupovi vizualno su pojašnjeni na Slici 3.9. Tada je $HOTA_\alpha$ metrika za zadanu graničnu (lokalizacijsku) vrijednost α dana s

$$HOTA_\alpha = \sqrt{\frac{\sum_{c \in TP} \mathcal{A}(c)}{|TP| + |FN| + |FP|}}, \quad \mathcal{A}(c) = \frac{|TPA(c)|}{|TPA(c)| + |FNA(c)| + |FPA(c)|}, \tag{3.19}$$

pri čemu TP, FN i FP mjere uspjeh, odnosno pogrešku detekcije, a TPA, FNA i FPA asocijacije. Kako bi se dodatno uzeo i aspekt lokalizacije, HOTA se definira kao vrijednost intergla po valjanim graničnim vrijednostima α, a u praksi se aproksimira aritmetičkom sredinom vrijednosti $HOTA_\alpha$ metrike za $\alpha \in \{0.05, 0.1, \ldots, 0.95\}$:

$$HOTA = \int_0^1 HOTA_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\substack{\alpha=0.05 \\ \alpha+=0.05}}^{0.95} HOTA_\alpha. \tag{3.20}$$

[12] TPA (engl. *True Positive Associations*), FNA (engl. *False Negative Associations*), FPA (engl. *False Positive Associations*).



Slika 3.9: HOTA: Ilustracija TPA, FPA i FNA skupova. Slika preuzeta iz [159] i modificirana.

IDF1

Za evaluaciju algoritama za praćenje uglavnom se istovremeno koristi više različitih metrika. Uz MOTA i HOTA metrike, često se koristi i $IDF1$ [160] metrika koja se fokusira na točnost identifikacije objekata tokom praćenja. Dok MOTA i HOTA rade pridruživanja na razini detekcija, $IDF1$ to čini na razini putanja. Definiraju se novi skupovi: $IDTP$ (engl. *Identity True Positives*) kao skup parova pridruženih detekcija $(o_i^{(t)}, h_j^{(t)})$ na preklapajućim dijelovima putanja koje su pridružene, $IDFN$ (engl. *Identity False Negatives*) i $IDFP$ (engl. *Identity True Positives*) kao skupovi preostalih stvarnih detekcije iz O_{det} te preostalih predviđenih detekcija iz H_{det} koje se nalaze na putanjama koje nisu uspješno pridružene ili na nepreklapajućim dijelovima pridruženih putanja. Tada je,

$$\textit{ID-Recall} = \frac{|IDTP|}{|IDTP| + |IDFN|}, \quad \textit{ID-Precision} = \frac{|IDTP|}{|IDTP| + |IDFP|}, \tag{3.21}$$

$$IDF1 = \frac{|IDTP|}{|IDTP| + 0.5|IDFN| + 0.5|IDFP|}. \tag{3.22}$$

ID-Recall je udio stvarnih detekcija koje su ispravno identificirane, a *ID-Precision* udio detekcija hipoteza koje su ispravno identificirane. $IDF1$ kombinira *ID-Recall* i *ID-Precision* u jedan broj računajući njihovu harmonijsku sredinu [160]. IDF1 metrika dolazi s nekim nedostacima uključujući prenaglašavanje asocijacija, neintuitivno i nemonotono ponašanje u slučaju detekcija, izostanak evaluacije pogreške lokalizacije te ne razmatranje točnosti asocijacije van preklapajućih dijelova pridruženih putanja [159].

Klasične metrike

Prethodno navedene metrike često se komplemetiraju i rezultatima **klasičnih metrika** [138] poput broja putanja stvarnih objekata koji je točno praćenen u barem 80% okvira videozapisa (***MT*** - *Mostly Tracked*), broja putanja stvarnih objekata koji je točno praćen u manje od 20% okvira (***ML*** - *Mostly Lost*) te broja **fragmenata**, odnosno hipoteza koje pokrivaju manje od 80% stvarne putanje objekta.

3.4.3. Usporedba popularnih MOT algoritama

Budući da se zadatak praćenja objekata sastoji od niza zasebnih koraka, različiti algoritmi praćenja primjenjuju različite pristupe u tim koracima. Stoga, objektivna usporedba performansi različitih algoritama praćenja i doprinosa njihovih komponenti postaje izazovna. Nadalje, kvaliteta detekcija koje se koriste prilikom praćenja ima značajan utjecaj na konačne performanse algoritma. Također, pojedini algoritmi koriste određene "trikove" tijekom faza treniranja i predviđanja kako bi poboljšali performanse praćenja [99].

Tablica 3.1: Rezultati evaluacije popularnih algoritama za praćenje više objekata na skupu za testiranje MOT17 referentnog skupa podataka.

Algoritam	**Godina**	**Vrsta**	**HOTA**(↑)	**MOTA**(↑)	**IDF1**(↑)	**FPS**(↑)
SORT [93]	2016	TBD	34.0	43.1	39.8	**143.3**
DeepSORT [94]	2017	TBD	61.2	78.0	74.5	13.8
ByteTrack [100] (s1=IoU, s2=IoU)	2022	TBD	63.1	80.3	77.3	29.6
ByteTrack [100] (s1=Re-ID, s2=IoU)	2022	TBD	-	76.3	**80.5**	11.8
BoT-SORT [101]	2022	TBD	**64.6**	**80.6**	79.5	6.6
BoT-SORT-ReID[101]	2022	TBD	**65.0**	**80.5**	**80.2**	4.5
StrongSORT [99]	2023	TBD	63.5	78.3	78.5	7.5
StrongSORT++ [99]	2023	TBD	64.4	79.6	79.5	7.1
Tracktor++ [87]	2019	JDT	44.8	56.3	55.1	1.5
FairMOT [86]	2020	JDT	59.3	73.7	72.3	25.9
CenterTrack [89]	2021	JDT	52.2	67.8	64.7	3.8
JDE [85]	2020	JDT	–	62.1	56.9	30.3
TransTrack [133]	2021	Transformer	–	74.5	63.9	–
TrackFormer [147]	2022	Transformer	57.3	74.1	68.0	7.4

U Tablici 3.1 prikazani su rezultati evaluacije popularnih metoda praćenja na MOT17 referentnom skupu podataka. Uz vrijednosti HOTA, MOTA i IDF1 metrika, također je dana i FPS (Frames Per Second) vrijednost kako bi se omogućila usporedba brzine izvršavanja pojedinih algoritama. Simbol ”↑” označava da veća vrijednost metrike odgovara boljim performansama algoritma, a ”-” označava vrijednosti koje nisu prijavljene. Navedene rezultate treba promatrati s oprezom, budući da oni u nekim radovima odgovaraju javnim detekcijama,

dok se u drugim koriste privatne detekcije. Nadalje, FPS vrijednost algoritma uvelike ovise o korištenom hardveru, a vrijeme potrošeno na detekciju najčešće nije uključeno.

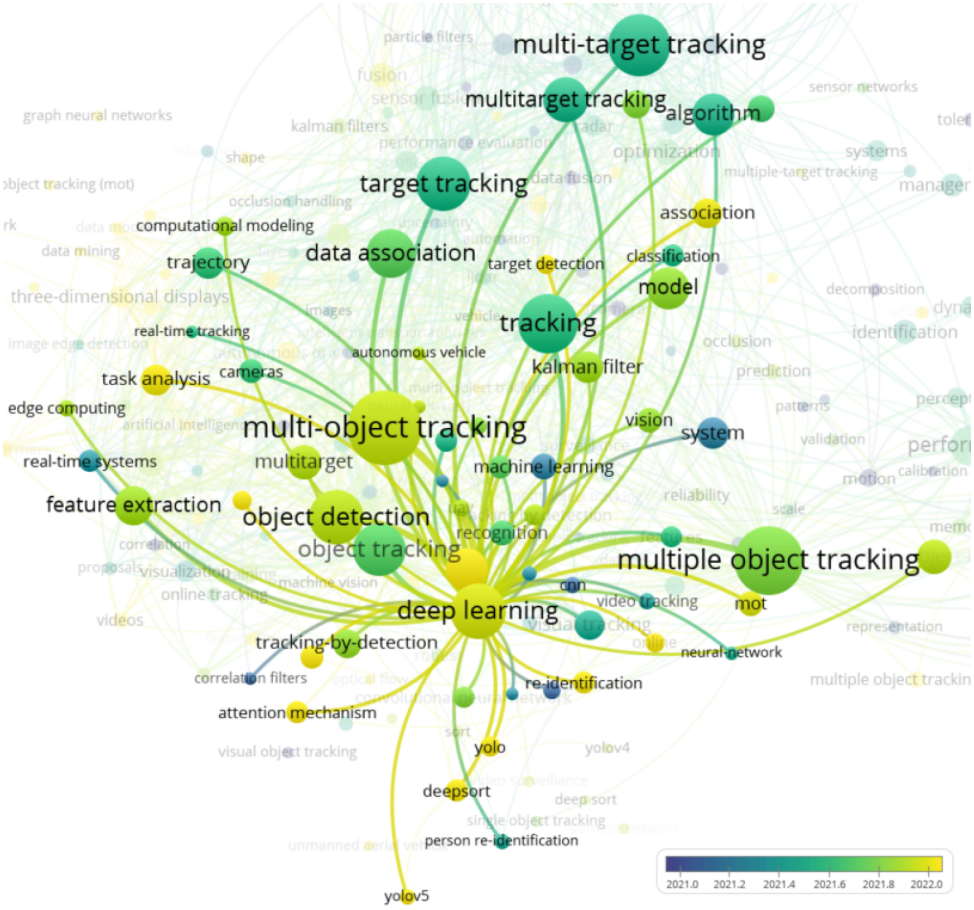
Rezultati evaluacije ByteTrack i BoT-SORT algoritma prikazani su za oba scenarija: prvo, kada se ne koristi vizualna informacija (ByteTrack: s1=s2=IoU, BoT-SORT), i drugo, kada se vizualna informacija koristi u prvoj fazi asocijacije (ByteTrack: s1=Re-ID, s2=IoU, BoT-SORT-ReID). Navedeni algoritmi postižu izuzetne rezultate, čak i kada se potpuno izostavi vizualna informacija te se algoritmi oslanjaju na dobre performanse detektora, informacije o kretanju i primjenu asocijacije u dvije faze. Budući da DeepSORT algoritam nije evaluiran na MOT17 skupu podataka u originalnom radu [94], u Tablici 3.1 prikazani su rezultati reproducirane implementacije algoritma iz rada [99]. Iako SORT algoritam postiže najlošije rezultate u smislu HOTA, MOTA i IDF1 metrika, on i dalje ostaje jedan od preferiranih algoritama za praćenje objekata u stvarnom vremenu kada je brzina izvršavanja prioritet.

3.5. Duboko učenje u MOT algoritmima

Kao što je naznačeno u potpoglavlju 3.2 *"Osnovni koraci MOT algoritma"*, metode dubokog učenja pronašle su primjenu u mnogim koracima MOT algoritma. Superiornosti dubokih detektora u odnosu na tradicionalne metode, dovela je do toga da su oni postali standardom u koraku detekcije novijih algoritama praćenja. Metode dubokog učenja mogu se primijeniti i na preostale korake MOT algoritma. Primjerice, korištenjem LSTM mreža i transformera za predviđanje budućih pozicija objekata ili sijamskih mreža za učenje sličnosti. Međutim, ekstrakcija značajki često se ističe kao preferirani korak za primjenu dubokih neuronskih mreža zbog njihove izuzetne sposobnosti izdvajanja složenih semantičkih i vizualnih karakteristika [78].

Za istraživanje osnovnih karakteristika i glavnih smjerova istraživanja u najnovijim radovima o praćenju više objekata pomoću metoda dubokog učenja, analizirani su radovi unutar WoS Core Collection-a unutar Web of Science (WoS) bibliografske baze podataka. U obzir su uzeti znanstveni radovi objavljeni od 2020. godine do danas koji sadrže ključne riječi: *"multiple object tracking"*, *"MOT"*, *"multi-object tracking"*, *"multi-target tracking"*, *"deep learning"*. Ukupno je pronađeno 2994 rezultata. Slika 3.10 prikazuje mrežu istovremenog pojavljivanja ključnih riječi u pronađenim radovima, uzimajući u obzir samo one ključne riječi koje se pojavljuju najmanje pet puta, što odgovara 609 ključnih riječi. Ključne riječi koje nisu relevantne "na ruke" su izuzete iz vizualizacije.

Čvorovi grafa na vizualizaciji predstavljaju ključne riječi. Veličinu čvora i odgovarajuće labele određuje težina koja ukazuje na važnost dane ključne riječi, a koja ovisi o stupnju čvora, snazi/težini pojedine veze/brida, citiranosti i sl. Ključna riječ veće težine smatra se važnijom te je na vizualizaciji istaknutija od ključnih riječi manje težine. Bridom su povezane ključni pojmovi koji se istovremeno pojavljuju u radovima. Na vizu-



Slika 3.10: Mreža ključnih pojmova recentnih znanstvenih radova ne temu praćenja više objekata primjenom dubokog učenja.

alizaciji se ističu riječi poput *"object detection"*, *"feature extraction"*, *"re-identification"*, *"person-reidentification"*, *"classification"*, *"tracking-by-detection"*, *"deepsort"*, *"yolo"*, *"yolov5"*, koje su (jako) povezane s pojmom dubokog učenja - *"deep learning"*. Iz analize je vidljivo da se u relevantnim radovima najčešće koristi YOLO detektor među spomenutim detekcijskim algoritmima, dok se DeepSORT izdvaja kao najkorišteniji MOT algoritam. Nadalje, algoritmi temeljeni na detekciji i dalje dominiraju kao glavna paradigma praćenja više objekata. Trendovi po godinama mogu se iščitati iz boja čvorova. Pojmovi kao što su *"deepsort"*, *"yolo"*, *"yolov5"* i *"attention mechanism"* odražavaju najnovije trendove u praćenju više objekata.

Iako su metode dubokog učenja privukle značajnu pažnju posljednjih godina, tradicionalni algoritmi i dalje imaju važnu ulogu u algoritmima praćenja više objekata [57]. Na primjer, iz grafa se može primijetiti često pojavljivanje ključne riječi "Kalman filter" zajedno s pojmom "deep learning", što je djelomično zbog toga što se u mnogim algoritmima praćenja, uz primjenu raznih metoda dubokog učenja, u koraku predikcije budućeg stanja i dalje obično koristi Kalmanov filter i njegove varijante.

4. Pregled područja: detekcija i praćenje plovila

Zadaci detekcije i praćenja plovila od ključne su važnosti za sigurnost plovidbe i upravljanje pomorskim operacijama. Korištenjem sustava za detekciju i praćenje, brodovi mogu identificirati druge objekte u svojoj okolini, što omogućuje pravovremenu reakciju na prisutnost plovila u blizini i efikasno sprječavanje sudara. Ovi sustavi također pomažu nadzornim tijelima da osiguraju usklađenost s pomorskim propisima te omogućuju brz odgovor na hitne situacije, poput nesreća i onečišćenja. Uz navedeno, oni se koriste u spasilačkim operacijama za lociranje i praćenje plovila u nevolji, za upravljanje prometom u prometnim lukama te nadzor ribarskih aktivnosti radi sprečavanja ilegalnog ribolova.

U tradicionalnim sustavima za upravljanje pomorskim prometom, često se oslanja na tehnologije poput AIS-a (*Automatskog Identifikacijskog Sustava*) i radara za detekciju, identifikaciju i praćenje brodova [161, 162]. Međutim, razne prepreke poput obližnjih građevina, brda ili drugih plovila, zajedno s nepovoljnim vremenskim uvjetima, ograničavaju domet i smanjuju pouzdanost radara [161]. Nadalje, sva plovila ne moraju nužno imati ugrađen AIS sustav, primjerice brodovi za sport i razonodu te ribarski i ratni brodovi. AIS sustav pojedinih brodova također može biti neispravan ili isključen, što otežava ili onemogućuje njihovu identifikaciju i praćenje [163]. Kako bi se prevladali nedostaci tradicionalnih radarskih i AIS sustava, nedavni radovi istražuju mogućnosti primjene kamera [164, 165, 161] i slika snimljenih iz zraka/satelitskih snimaka [166, 167, 168, 169] za automatski nadzor, detekciju i praćenje plovila. Prednosti u smislu ekonomske isplativosti i bogatstva vizualnih informacija koje pružaju, čine RGB kamere ključnim izvorom za unapređenje performansi sustava za detekciju i praćenje [162].

Praćenje plovila na videozapisima složen je zadatak popraćen nizom specifičnih izazova. Sustav za detekciju i identifikaciju plovila mora biti otporan na promjene vremenskih uvjeta i uvjeta na moru, kao i na varijacije osvjetljenja. Refleksije sunca na površini vode, valovi te prisutnost drugih objekata, poput biljaka, životinja ili otpada, dodatno otežavaju detekciju plovila. Nadalje, isto plovilo može izgledati znatno drugačije na različitim udaljenostima od kamere i u različitim položajima. Povrh navedenog, većina praktičnih aplikacija zahtijeva izvršavanje u stvarnom vremenu uz ograničene hardverske resurse [170, 162].

Primjena metoda dubokog učenja za detekciju i praćenje pomorskih objekata nije trivijalna. Velike razlike u pomorskom okruženju i okruženjima u kojem su razvijeni *state-of-the-art* algoritmi otežavaju izravnu primjenu tih algoritama za preciznu detekciju i praćenje plovila. Ovi algoritmi ne uzimaju u obzir specifične karakteristike plovila. Primjerice, prilikom konstrukcije baznih graničnih okvira ne razmatraju se mogući oblici i veličine plovila, već nekih drugih objekata. Dodatno, složene pozadine i promjenjivi vremenski i morski uvjeti zahtijevaju veću diskriminativnu sposobnost algoritama, dok su robusniji ekstraktori značajki nužni kako se ne bi ignoriralo bitne značajke plovila u daljini, koja mogu zauzimati samo nekoliko piksela [171].

4.1. Skupovi podataka

Performanse algoritama baziranih na dubokom učenju uvelike ovise o broju kvalitetnih primjera dostupnih za učenje [172]. Razvoj pouzdanog sustava za detekciju i praćenje različitih vrsta plovila stoga zahtijeva velik i realističan skup podataka koji bilježi svu heterogenost pomorskog okruženja [161, 162]. Nedostatak kvalitetnih, javno dostupnih skupova podataka za specifično područje primjene jedan je od osnovnih problema u domeni računalnog vida, a slučaj detekcije i praćenja plovila nije iznimka [173].

4.1.1. Opći skupovi podataka

Iako neki opći, javno dostupni skupovi podataka sadrže slike s primjerima plovila, ona su najčešće samo generalizirana kao *"ships"* ili *"boat"* (CIFAR10 [174], PASCAL VOC [175], MS COCO [50]). Caltech-256 [176] skup podataka sadrži četiri kategorije plovila (*"canoe"*, *"kayak"*, *"ketch"*, *"speed boat"*), a ImageNet [17] njih šest (*"fireboat"*, *"lifeboat"*, *"speedboat"*, *"submarine"*, *"pirate"*, *"container ship"*). Međutim, dane klase ne obuhvaćaju svu raznolikost plovila, broj primjera je ograničen, a plovila se obično nalaze u središtu slike zauzimajući njezin veći dio. MS COCO [50] skup podataka sadrži veći broj instanci objekata koji predstavljaju plovila, no kategorija *"boat"* nije dalje podijeljena na potkategorije. U Tablici 4.1 prikazan je broj klasa plovila, slika koje odgovaraju plovilima i odgovarjućih instanci objekata u općim skupovima podataka.

Tablica 4.1: Opći skupovi podataka i plovila [164, 172].

Skup podataka	Klase plovila	Slike	Objekti	Zadatak
CIFAR10 [174]	1	6000	–	klasifikacija
Caltech-256 [176]	4	418	–	klasifikacija
ImageNet [17]	6	525	613	klasifikacija/detekcija
PASCAL VOC [175]	1	363	791	detekcija
MS COCO [50]	1	3025	10759	detekcija

4.1.2. Skupovi podataka iz pomorskih okruženja

Do sada je predstavljeno nekoliko skupova podataka usredotočenih na klasifikaciju, detekciju i praćenje u pomorskim okruženjima. Pregled takvih skupova podataka koji koriste RGB slike i/ili videozapise dan je u Tablici 4.2. Klase plovila koje se javljaju u pojedinim skupovima podataka specificirane su u Tablici 4.3.

Tablica 4.2: Skupovi podataka za klasifikaciju, detekciju i praćenje plovila.

Skup podataka	Godina	Slike	Objekti	Klase	Rezolucija	Zadaci
VAIS [177]	2015.	1623	–	6	5056×5056	klasifikacija
SMD [178]	2017.	20367	157668	10	1920×1080	detekcija, praćenje
MARVEL [179]	2017.	2M (140000)	–	109 (26)	razne rezolucije	klasifikacija
SeaShips [164]	2018.	31455	40077	6	1920×1080	detekcija
Harbor Surveillance [161]	2018.	48966	70513	1	2048×1536	detekcija
Airbus ship [180]	2018.	208162	N/A	1	768×768	detekcija, segmentacija
Game of DL: ship dataset [181]	2019.	8932	–	5	razne rezolucije	klasifikacija
McShips [182]	2020.	14709	26259	13	razne rezolucije	detekcija
ABOships [173]	2021.	9880	41967	11	1920×720	detekcija
GLSD [172]	2021.	152576	212357	13	razne rezolucije	detekcija
LMD-TShip [183]	2021.	40240	N/A	5	razne rezolucije	detekcija, praćenje
MarSyn [184]	2022.	25000	34000	6	1280×720 550×550	detekcija segmentacija
SeaSAw [162]	2022.	1.9 M	14.6 M	12	7680×1408, 3840×2056, 3648×2052, 1920×1080	detekcija, praćenje
SPSCD [185]	2023.	19337	27849	12	1920×1080	detekcija

VAIS [177], *MARVEL* [179] i *Game od Deep Learning: Ship Dataset* [181] skupovi podataka fokus stavljaju na klasifikaciju različitih vrsta plovila. **VAIS** [177] skup podataka obuhvaća uparene RGB i infracrvene slike brodova prikupljene tijekom devet dana na šest različitih gatova. Ukupno je 2865 slika (1623 RGB i 1242 infracrvenih), od kojih je 1088 parova. Slike sadrže šest kategorija brodova koje se dalje mogu podijeliti na 15 manjih pot-

kategorija. MARitime VEsseLs (**MARVEL**) [179] skup podataka sadrži 2 milijuna slika 109 različitih tipova plovila, prikupljenih sa *Shipspotting*[1] web stranice. Korištenjem polunadzirane metode grupiranja, konstruirano je 26 superklasa plovila. **Game od Deep Learning: Ship Dataset** [181] je javno dostupan skup podataka predstavljen u sklopu *Game of Deep Learning: Computer Vision Hackathon-a* koji je održan 2019. godine. Skup podataka sadrži 6252 označene slike za treniranje i 2680 neoznačenih slika za konačnu evaluaciju iz skupa za testiranje.

Tablica 4.3: Klase plovila koje se javljaju u pojedinim skupovima podataka.

Skup podataka	Broj klasa	Klase
VAIS [177]	6	merchant, sailing, medium passenger, medium other, tugboat, small boat
SMD [178]	10	ferry, buoy, vessel/ship, speed boat, boat, kayak, sail boat, swimming person, flying bird/plane, other
MARVEL [179]	26 superklasa	container ship, bulk carrier, passengers ship, ro-ro/passenger ship, ro-ro cargo, tug, vehicles carrier, reefer, yacht, sailing vessel, heavy load carrier, wood chips carrier, fire fighting vessel, patrol vessel, platform, standby safety vessel, combat vessel, training ship, icebreaker, replenishment vessel, tankers, fishing vessels, supply vessels, carrier/floating, dredgers
SeaShips [164]	6	ore carrier, bulk cargo carrier, general cargo ship, container ship, fishing boat, passenger ship
Harbor Surveillance [161]	1	vessel
Airbus ship [180]	1	ship
Game of DL: ship dataset [181]	5	cargo, carrier, cruise, military, tankers
McShips [182]	13	aircraft carrier, submarine, landing ship, auxiliary ship, destroyer, missile boat, speedboat, fishing boat, passenger ship, container ship, tugboat, sailboat, support ship
ABOships [173]	11	boat, cargoship, cruiseship, ferry, militaryship, miscboat, miscellaneous, motorboat, passengership, sailboat, seamark
GLSD [172]	13	sailing boat, fishing boat, warship, passenger ship, general cargo ship, container ship, bulk cargo carrier, barge, ore carrier, speed boat, canoe, oil carrier, tug
LMD-TShip [183]	5	cargo ships, fishing ships, passenger ships, speed boats, unmanned ships
MarSyn [184]	6	cargo ships, military ships, fishing boats, speed boats, rescue rafts, other
SeaSAw [162]	12	ship, recreational vessel, manual craft, sailing vessel, work boat, fishing vessel, towing vessel, dredge, wind turbine, marker, mooring buoy, miscellaneous
SPSCD [185]	12	small craft, small fishing boat, small passenger ship, fishing trawler, large passenger ship, sailing boat, speed craft, motorboat, pleasure yacht, medium ferry, large ferry, high speed craft

Iako sadrže isključivo slike iz pomorskih okruženja, skupovi podataka za detekciju plovila *Harbor Surveillance* [161] i *Airbus Ship Dataset* [180] ne rade distinkciju između razli-

[1] https://www.shipspotting.com/

čitih kategorija plovila. **Harbor Surveillance** [161] sadrži 48966 slika dobivenih iz videozapisa snimljenih tijekom šestomjesečnog razdoblja s deset različitih pogleda kamere na luku. Za svaki od pogleda, odabrano je i označeno više slika brodova, osiguravajući raznolikost pozadina i orijentacija. **Airbus Ship Dataset** [180] za lokalizaciju brodova na satelitskim slikama javno je dostupan na *Kaggle* platformi[2]. Ovaj skup obuhvaća širok spektar slika, od onih na kojima uopće nema brodova do onih s više brodova. Brodovi na slikama se mogu značajno razlikovati u veličini, te se nalaze u različitim okruženjima poput otvorenog mora, luka, marina i slično.

Neki skupovi podataka uključuju i primjere prikupljene putem interneta [172, 182], dok drugi sadrže isključivo stvarne podatke snimljene kamerama postavljenim u lukama, duž obale ili na samim plovilima [161, 178, 164, 173, 162, 185]. Većina slika u **GLSD** [172] skupu podataka prikupljena je s interneta te obuhvaća različite svjetske luke, dok manji dio dolazi s nadzornog sustava Zhuhai Hengqin New Area, u Kini. Ovaj skup podataka obuhvaća širok spektar slika koje sadrže male objekte (manje od 32×32 piksela) i objekte srednje veličine (između 32×32 i 96×96 piksela), obuhvaća također i neke neobične situacije, poput slika plovila u plamenu te mozaike slika plovila. **McShips** [182] skup podataka obuhvaća 14709 slika podijeljenih u skup za treniranje (10297) i skup za testiranje (4412). Slike sadrže šest kategorija ratnih i sedam kategorija civilnih brodova. Podaci su prikupljeni putem različitih izvora poput web-tražilica, foruma, portala te videozapisa i nadzornih kamera, osiguravajući prisutnost različitih pozadina, osvjetljenja te atmosferskih uvjeta. **LMD-TShip** [172] skup podataka sadrži 40240 označenih okvira iz 191 videozapisa, prikupljenih pomoću fiksnih kamera postavljenih na dva plovila te kamerama i mobilnim telefonima s obale. Podaci su podijeljeni u skup za treniranje (152 videozapisa, 31527 okvira) i skup za testiranje (39 videozapisa, 8713 okvira).

Singapore Maritime Dataset (SMD) [178] sadrži videozapise visoke rezolucije snimljene u vodama oko Singapura na različitim lokacijama i rutama. Od ukupno 51 videozapisa, 40 je snimljeno kamerom postavljenom na obali na fiksnom postolju, dok je 11 snimljeno kamerom postavljenom na brodu u pokretu. Snimci su zabilježeni u različitim vremenskim uvjetima: prije izlaska sunca, ujutro, popodne, predvečer, nakon zalaska sunca, za vrijeme kiše, magle i slično. Osim RGB videozapisa, ovaj skup podataka sadrži i podatke u bliskom infracrvenom spektru (*NIR*). **SeaShips** [164] skup podataka sadrži slike prikupljene s nadzornih kamera s 45 različite lokacije na obali otoka Hengqin (Kina), snimljene u siječnju, travnju, kolovozu i listopadu 2017. i 2018. godine, svaki dan od 6:00 do 20:00 sati. Odabrane su slike koje prikazuju plovila različitih vrsta i veličina te obuhvaćaju različite poglede, osvjetljenja i razine zaklonjenosti objekata. Skup podataka **ABOships** [173] obuhvaća 9880 slika koje prikazuju čak 41967 označenih objekata. Slike su dobivene iz videozapisa snimljenih kamerom na plovilu za razgledavanje znamenitosti na ruti od od grada Turku do grada Rusissalo u Finskoj tijekom 13 dana u lipnju i srpnju 2018. godine. Ovaj skup podataka

[2] https://www.kaggle.com/

obuhvaća različite vremenske uvjete tijekom dana, te uključuje slike otvorenog mora, luka i urbanih krajolika.

Jedan od najrecentnijih i najopsežnijih skupova, **Sea Situational Awareness (SeaSAw)** [162] sadrži podatke prikupljene kamerama postavljenim na brodovima u pokretu na nekoliko geografskih lokacija duž Isotčne obale SAD-a, u Bostonskoj luci i Europi. Korištene su kamere različitih vidnih polja i rezolucija. Slike su snimane pri različitim vremenskim uvjetima i osvjetljenjima, te obuhvaćaju snimke iz luka, uz obalu i na otvorenom moru. Nedavno je predstavljen i **Split Port Ship Classification Dataset (SPSCD)** [185] koji obuhvaća slike splitske luke snimljene LR (*long-range*) kamerom u razdoblju od veljače 2020. do prosinca 2022. godine. Slike su snimane tijekom različitih godišnjih doba, u različitim razdobljima dana i u raznim vremenskim uvjetima. Ovaj skup podataka odražava specifičnosti meditetranske (splitske) luke u kojoj pomorski promet varira od manjih plovila do velikih putničkih trajekata i kruzera. Osim toga, uključuje velik broj manjih plovila i plovila srednje veličine koji se često ne prate uobičajenim sustavima za nadzor pomorskog prometa poput AIS-a.

U [184], autori u Blenderu generiraju sintetički **MarSyn** skup podataka koji obuhvaća 25 različitih foto-realističnih videozapisa, pri čemu se svaki sastoji od 1000 okvira. Cilj ovog skupa podataka je simulirati raznolike pomorske scenarije i uvjete, uključujući varijacije vremenskih uvjeta, slike u blizini obale te refleksije na površini vode. Plovila na slikama su različitih vrsta (teretni brodovi, vojni brodovi, ribarski čamci, gliseri, splavi za spašavanje i dr.), duljina (od 3 m do 125 m), oblika i boja.

4.2. Pregled relevantnih radova

U ovom poglavlju, istaknuti su relevantni radovi koji istražuju detekciju i praćenje plovila na RGB slikama i videozapisima, koristeći pritom tehnike dubokog učenja. Broj istraživačkih radova posvećenih praćenju značajno zaostaje za onima usmjerenima isključivo na detekciju. Ovaj nesrazmjer djelomično proizlazi iz nedostatka javno dostupnih i prikladno anotiranih skupova podataka za praćenje plovila, koji su ključni za razvoj i evaluacija algoritama. Većina radova koristi privatne skupove podataka, što dodatno otežava generalizaciju rezultata i objektivnu usporedbu različitih modela.

4.2.1. Detekcija plovila

U radovima koji se bave detekcijom plovila u pomorskim okruženjima, Faster R-CNN se ističe kao najčešće korišteni detektor u dvije faze. U radu [168], autori predstavljaju unaprijeđenu verziju Faster R-CNN mreže za detekciju brodova na satelitskim slikama. Prvo, provodi se segmentacija vodene površine slike od ostalih površina korištenjem SVM algoritma te ekstrakcija područja od interesa koja potencijalno sadrže brodove. Nakon toga, na identificiranim područjima detektiraju se brodovi pomoću Faster R-CNN detektora prilago-

đenog za detekciju manjih brodova i brodova koji se nalaze u blizini jedan drugoga. Qi *et al.* [186] također koriste modificiranu verziju Faster R-CNN mreže. Prije same detekcije, provodi se postupak smanjenjenja veličine slike i semantičkog sužavanje scene kako bi se istaknule bitne informacije slike i pažnja usmjerila prema ciljnim područjima gdje bi se brodovi mogli nalaziti. Za detekciju koriste Faster R-CNN mreža preoblikovanu u hijerarhijsku mrežu sužavanja, s ciljem smanjenja opsega pretrage detektora i poboljšanja brzine detekcije. Poboljšane varijante Faster R-CNN detektora za detekciju brodova koriste se i u [187, 188, 189, 169].

U kontekstu detektora u jednoj fazi, ali i općenito, YOLO detektor se preferira u radovima posvećenim detekciji plovila, zahvaljujući svojoj sposobnosti brze i precizne detekcije objekata u stvarnom vremenu. U radu [190], ispituju se performanse YOLOv2 detektora za detekciju i klasifikaciju plovila, uspoređujući varijantu detektora predtreniranu na PASCAL VOC skupu podataka s onom koja je trenirana na SMD skupu podataka. Za detekciju objekata na površini mora, u radu [191] primjenjuje se unaprijeđena verzija YOLOv3 detektora, koja inkorporira DenseNet model u Darknet-53 okosnicu, s ciljem poboljšanja prilagodljivosti bespilotnih plovila tijekom dugotrajnih misija. U radu [192], predstavljena je modificirana varijanta detektora YOLOv5 koja koristi K-means algoritam za optimizaciju inicijalnih baznih okvira, uz dodatak Ghost modula i transformera, također namijenjena detekciji plovila na snimcima bespilotnih površinskih plovila. Wu *et al.* [193] predstavljaju poboljšanu varijantu YOLOv7 detektora koja koristi bazne granične okvire koji su bolje prilagođeni različitim veličinama i oblicima brodova, integrira modul za fuziju značajki različitih skala te uključuje agregacijsku mrežu za fuziju mapa značajki s različitih razina, što rezultira poboljšanom točnošću i robusnošću detekcije. Unaprijeđena verzija YOLOv7-Tiny detektora, YOLOv7-Ship, prikladna za detekciju brodova u kompleksnim pomorskim okruženjima predložena je u [194]. Uspoređujući je s baznom verzijom, YOLOv7-Ship pokazuje unaprijeđenu točnost u detekciji objekata različitih veličina, malih objekata i objekata koji su djelomično zaklonjeni. Zhao i Song [195] predlažu ekstenziju YOLOv8 detektora koja standardnu okosnicu za ekstrakciju značajki zamjenjuje kombinacijom efikasnog MobileViTSF vizualnog transformera i MobileNetv2 mreže, klasične konvolucijske blokove zamjenjuje GSConv blokovima i koristi redizajnirani C2f blok YOLOv8 mreže. Primjena različitih verzija YOLO detektora za detekciju objekata na moru (uključujući i različite vrste plovila) na krajnjim uređajima ugradbenih sustava istražena je u [196].

U članku autora Iancu *et al.* [197], evaluiraju se performansa CenterNet detektora s različitim ekstraktorima značajki na prilagođenoj varijanti ABOships skupa podataka. Ova varijanta uključuje izbacivanje objekata koji zauzimaju manje od 16^2 piksela, te agregaciju originalnih trinaest klasa ABOship skupa podataka u četiri superklase radi ublažavanja neuravnoteženosti klasa. SSD detektor, prilagođen opažanju ekstremnih varijacija u veličini i obliku brodova, koristi se za detekciju brodova u Harbour Surveillance skupu podataka [161]. Li *et al.* [198] predlažu algoritam za detekciju objekata na vodenoj površini na pano-

ramskim slikama temeljen na poboljšanoj varijanti SSD-a u kojoj je VGG16 zamijenjena s ResNet-50 mrežom te je dodano pet slojeva za ekstrakciju značajki.

Studije koje uspoređuju performanse različitih detektora provedene su u nekoliko istraživačkih radova. U [199] je dana usporedba Faster R-CNN i Mask R-CNN detektora (s ResNet101 okosnicom), koji su prethodno trenirani na skupovima podataka ImageNet i MS COCO, primijenjenih na SMD skupu podataka. Autori rada [164] evaluiraju detektore Fast R-CNN, Faster R-CNN s više različitih okosnica, SSD i YOLOv2 na SeaShips skupu podataka. Usporedba Faster R-CNN, SSD, YOLOv2, YoLOv3 i YOLOv3SPP detektora na McShips skupu podataka dana je u [182]. Detektori Faster R-CNN, SSD, EfficientDet i RFCN su evaluirani na ABOships skupu podataka u [173]. Rad [172] daje usporedbu devet različitih detektora, uključujući Faster R-CNN i RetinaNet detektore, na GLSD skupu podataka. Zhao *et al.* [200] uspoređuju dvanaest različitih detektora, među kojima su Faster R-CNN, SSD, RetinaNet i razne varijante YOLO detektora, na skupu podataka koji sadrži slike snimljene bespilotnim letjelicama.

4.2.2. Praćenje plovila

U istraživačkim radovima koji se bave praćenjem plovila u RGB videozapisima, često se primjenjuje kombinacija YOLO detektora i DeepSORT algoritma. U [167], predložena je kombinacija poboljšane verzije YOLOv3 detektora i DeepSORT algoritma za praćenje brodova na slikama snimljenim iz zraka. U YOLOv3 algoritmu se koriste modificirani bazni okviri koji su prilagođeni specifičnom obliku brodova, izostavlja se 52×52 skala detekcije, te se umjesto originalnog gubitka koristi fokalni gubitak kako bi se uravnotežile nejednake proporcije pozitivnih i negativnih primjera. Nadalje, koristi se mreža višestruke granularnosti (engl. *multiple granularity network*) [201] za ekstrakciju bogatijih i preciznijih vizualnih značajki. Modificiranu verziju YOLOv3 algoritma u kombinaciji s DeepSORT algoritmom koriste i u [202] za detekciju i praćenje brodova u unutarnjim plovnim putevima, preciznije na rijeci Yangtze u Kini. Uz optimizaciju inicijalnih baznih okvira, autori zamjenjuju sigmoidnu aktivacijsku funkciju klasifikatora softmax aktivacijom, te predlažu korištenje poboljšane verzije NMS algoritma za učinkovitije uklanjanje redundantnih graničnih okvira. U [203], fokus je na praćenju brodova za maglovita vremena. Prvo se pomoću konvolucijske neuronske mreže iz svakog okvira videozapisa ukloni magla, a zatim se za detekciju i praćenje koriste YOLOv5 i DeepSORT. Poboljšane verzije YOLOv5 i YOLOX algoritma u kombinaciji s DeepSORT algoritmom koriste se za praćenje plovila i u radovima [204, 205].

Autori u [206] evaluiraju četiri varijante YOLO algoritma za detekciju brodova na SMD skupu podataka te predlažu algoritam za praćenje plovila na videozapisima snimljenim kamerom na brodu u pokretu koji daje bolje rezultate od DeepSORT algoritma. Predloženi algoritam praćenja koristi: (1) pridruživanje temeljem IoU vrijednosti detektcija u trenutnom okviru i praćenih objekata, (2) pridruživanja zasnovanog na sličnosti ORB (*oriented FAST*

and rotated BRIEF) značajki i veličini graničnih okvira. Za praćenje plovila u stvarnom vremenu, Xing *et al.* [165] sugeriraju korištenje YOLOv8-FAS algoritma, poboljšane verzije YOLOv8n algoritma dizajnirane za primjenu na uređajima s ograničenom memorijom i računalnim resursima, zajedno s ByteTrack algoritmom. AdapTrack algoritam za praćenje plovila, koji se oslanja na FairMOT algoritam, opisan je u radu [207]. AdapTrack algoritam implementira strategiju asocijacije ByteTrack algoritma. U prvoj fazi, detekcije visoke pouzdanosti se pridružuju putanjama koristeći vizualne karakteristike. Zatim, u drugoj fazi, preostale putanje se povezuju s detekcijama niske pouzdanosti temeljem IoU vrijednosti.

Nasuprot uobičajene kombinacije YOLO algoritma za detekciju i DeepSORT/ByteTrack algoritama za praćenje plovila, u [208] se koristi CO-Tracker [209] model zasnovan na transformerima u kombinaciji s LSTM i graf neuronskim mrežama s mehanizmom pozornosti. Shan *et al.* [171] koriste SiamFPN model za praćenje brodova koji se sastoji od sijamske mreže s FPN podmrežama i tri mreže za predlaganje regija od interesa. Autori u [210] predstavljaju novi algoritam praćenja zasnovan na mehanizmu dinamičke memorije i hijerarhijskom modelu koji je svjestan konteksta. Mehanizam dinamičke memorije pohranjuje značajke prethodnih okvira te ih dinamički integrira s trenutnim značajkama kako bi se u model inkorporirao vremenski kontekst i međusobna koreliranost okvira videozapisa. Hijerarhijski model koristi se za ekstrakciju kontekstualne informacije na različitim skalama te globalnih i lokalnih informacija pomoću slojeva sažimanja i konvolucije s dilatacijom.

5. Zaključak

U ovom radu naglasak je stavljen na primjenu metoda dubokog učenja za problem detekcije i praćenja plovila na videozapisima. Dan je pregled popularnih metoda za detekciju objekata, kao i algoritama za praćenje više objekata. Među ostalim, diskutirana je upotreba dubokog učenja u navedenim algoritmima i novijim istraživanjima.

Zadaci automatske detekcije i praćenja plovila imaju široku praktičnu primjenu, uključujući nadzor pomorskih područja, upravljanje pomorskim prometom te podršku u operacijama potrage i spašavanja. No, implementacija sustava koji bi precizno detektirao plovila i kontinuirano ih pratio kroz cijeli videozapis, sve do izlaska kadra, nije niti malo trivijalna. Nepovoljni vremenski uvjeti, stanje mora te refleksije sunca na površini vode dodatno otežavaju detekciju plovila. Nadalje, izgled samih plovila može se znatno razlikovati, ovisno o njihovom trenutnom položaju i orijentaciji. Plovilo koje je blizu kamere može zauzimati veći dio slike, dok isto to plovilo u daljini zauzima samo nekoliko piksela. Kao što je slučaj i s općenitim praćenjem objekata, i u praćenju plovila izazov predstavljaju situacije kada je objekt djelomično ili potpuno zaklonjen u nekom dijelu videozapisa. Reidentifikacija izgubljenih objekata predstavlja još jedan dodatan izazov algoritmima praćenja.

Ubrzan razvoj i učinkovitost konvolucijskih neuronskih mreža u ekstrakciji složenih semantičkih značajki iznimno su doprinijeli razvoju različitih područja računalnog vida, posebno detekcije objekata. Paralelno s razvojem detekcije, ključnog koraka u praćenju objekata, razvijali su se i različiti algoritmi praćenja. Međutim, većina aktualnog istraživanja u domeni praćenja objekata usredotočena je na praćenje osoba (pješaka) i/ili vozila na cestama i sličnih objekata. Jako je malo primjera primjene *state-of-the-art* metoda detekcije i praćenja za konkretan problem praćenja plovila. Jedan od razloga nedostatak je javno dostupnih, označenih i kvalitetnih skupova podataka za praćenje plovila. Većina skupova podataka koji sadrže slike različitih tipova plovila namijenjena je zadacima klasifikacije ili detekcije, a oni skupovi koji se u radovima koriste za praćenje plovila najčešće su privatni, što dodatno otežava napredak istraživanja u ovoj domeni.

Aktualno znanje ostavlja obilje prostora za istraživanje i doprinos u području praćenja plovila. Ove mogućnosti uključuju stvaranje označenih skupova podataka za praćenje različitih vrsta plovila te istraživanje aktualnih izazova, kao što su detekcija malih objekata te problemi zaklonjenosti i reidentifikacije tijekom praćenja, s posebnim fokusom na slučaj kada su objekti od interesa plovila.

LITERATURA

[1] Z.-Q. Zhao, P. Zheng, S.-t. Xu i X. Wu, Object detection with deep learning: A review, *IEEE transactions on neural networks and learning systems*, 30, 11, 3212–3232, 2019.

[2] C. Cortes i V. Vapnik, Support-vector networks, *Machine learning*, 20, 3, 273–297, 1995.

[3] Y. Freund i R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 55, 1, 119–139, 1997.

[4] N. Dalal i B. Triggs, Histograms of oriented gradients for human detection, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1, 886–893 vol. 1, 2005.

[5] D. G. Lowe, Object recognition from local scale-invariant features, *Proceedings of the seventh IEEE international conference on computer vision*, 2, 1150–1157, Ieee, 1999.

[6] H. Bay, T. Tuytelaars i L. Van Gool, Surf: Speeded up robust features, *Lecture notes in computer science*, 3951, 404–417, 2006.

[7] P. Viola i M. Jones, Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1, I–I, 2001.

[8] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt i J. M. Ogden, Pyramid methods in image processing, *RCA engineer*, 29, 6, 33–41, 1984.

[9] X. Wu, D. Sahoo i S. C. Hoi, Recent advances in deep learning for object detection, *Neurocomputing*, 396, 39–64, 2020.

[10] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du i X. Lan, A review of object detection based on deep learning, *Multimedia Tools and Applications*, 79, 23729–23791, 2020.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard i L. D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural computation*, 1, 4, 541–551, 1989.

[12] A. Krizhevsky, I. Sutskever i G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Communications of the ACM*, 60, 6, 84–90, 2017.

[13] D. H. Hubel i T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex, *The Journal of physiology*, 195, 1, 215–243, 1968.

[14] K. Fukushima, Neocognitron, *Scholarpedia*, 2, 1, 1717, 2007, revision #91558.

[15] V. Nair i G. E. Hinton, Rectified linear units improve restricted boltzmann machines, *Icml*, 2010.

[16] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai et al., Recent advances in convolutional neural networks, *Pattern Recognition*, 77, 354–377, 2018.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li i L. Fei-Fei, Imagenet: A large-scale hierarchical image database, *2009 IEEE conference on computer vision and pattern recognition*, 248–255, Ieee, 2009.

[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick i P. Dollár, Microsoft coco: Common objects in context, 2015.

[19] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau i S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, *nature*, 542, 7639, 115–118, 2017.

[20] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh i Y. Zhang, Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery, *Remote Sensing*, 10, 7, 2018.

[21] D. V. Politikos, E. Fakiris, A. Davvetas, I. A. Klampanos i G. Papatheodorou, Automatic detection of seafloor marine litter using towed camera images and deep learning, *Marine Pollution Bulletin*, 164, 111974, 2021.

[22] R. Girshick, J. Donahue, T. Darrell i J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587, 2014.

[23] J. R. Uijlings, K. E. Van De Sande, T. Gevers i A. W. Smeulders, Selective search for object recognition, *International journal of computer vision*, 104, 154–171, 2013.

[24] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar i B. Lee, A survey of modern deep learning based object detection models, *Digital Signal Processing*, 126, 103514, 2022.

[25] R. Girshick, Fast r-cnn, *Proceedings of the IEEE international conference on computer vision*, 1440–1448, 2015.

[26] S. Ren, K. He, R. Girshick i J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, 28, 2015.

[27] K. He, X. Zhang, S. Ren i J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE transactions on pattern analysis and machine intelligence*, 37, 9, 1904–1916, 2015.

[28] K. He, G. Gkioxari, P. Dollár i R. Girshick, Mask r-cnn, *Proceedings of the IEEE international conference on computer vision*, 2961–2969, 2017.

[29] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan i S. Belongie, Feature pyramid networks for object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125, 2017.

[30] J. Redmon, S. Divvala, R. Girshick i A. Farhadi, You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788, 2016.

[31] J. Redmon i A. Farhadi, Yolo9000: better, faster, stronger, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271, 2017.

[32] J. Redmon i A. Farhadi, Yolov3: An incremental improvement, *arXiv preprint arXiv:1804.02767*, 2018.

[33] A. Bochkovskiy, C.-Y. Wang i H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, *arXiv preprint arXiv:2004.10934*, 2020.

[34] G. Jocher, Ultralytics yolov5, 2020, https://github.com/ultralytics/yolov5 (posjećeno 8. travnja 2024.).

[35] G. Jocher, A. Chaurasia i J. Qiu, Ultralytics yolov8, 2023, https://github.com/ultralytics/ultralytics (posjećeno 8. travnja 2024.).

[36] C.-Y. Wang, I.-H. Yeh i H.-Y. M. Liao, Yolov9: Learning what you want to learn using programmable gradient information, *arXiv preprint arXiv:2402.13616*, 2024.

[37] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding et al., Pp-yolo: An effective and efficient implementation of object detector, *arXiv preprint arXiv:2007.12099*, 2020.

[38] C.-Y. Wang, I.-H. Yeh i H.-Y. M. Liao, You only learn one representation: Unified network for multiple tasks, *arXiv preprint arXiv:2105.04206*, 2021.

[39] Z. Ge, S. Liu, F. Wang, Z. Li i J. Sun, Yolox: Exceeding yolo series in 2021, *arXiv preprint arXiv:2107.08430*, 2021.

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu i A. C. Berg, Ssd: Single shot multibox detector, *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21–37, Springer, 2016.

[41] M. Tan, R. Pang i Q. V. Le, Efficientdet: Scalable and efficient object detection, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790, 2020.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He i P. Dollár, Focal loss for dense object detection, *Proceedings of the IEEE international conference on computer vision*, 2980–2988, 2017.

[43] X. Zhou, D. Wang i P. Krähenbühl, Objects as points, *arXiv preprint arXiv:1904.07850*, 2019.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke i A. Rabinovich, Going deeper with convolutions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9, 2015.

[45] S. Ioffe i C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *International conference on machine learning*, 448–456, pmlr, 2015.

[46] S. Lloyd, Least squares quantization in pcm, *IEEE transactions on information theory*, 28, 2, 129–137, 1982.

[47] K. Simonyan i A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.

[48] K. He, X. Zhang, S. Ren i J. Sun, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778, 2016.

[49] S. Aharon, Louis-Dupont, Ofri Masad, K. Yurkova, Lotem Fridman, Lkdci, E. Khvedchenya, R. Rubin, N. Bagrov, B. Tymchenko, T. Keren, A. Zhilko i Eran-Deci, Super-gradients, 2021, https://github.com/Deci-AI/super-gradients (posjećeno 8. travnja 2024.).

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár i C. L. Zitnick, Microsoft coco: Common objects in context, *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755, Springer, 2014.

[51] H. Law i J. Deng, Cornernet: Detecting objects as paired keypoints, *Proceedings of the European conference on computer vision (ECCV)*, 734–750, 2018.

[52] Z. Soleimanitaleb i M. A. Keyvanrad, Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics, 1–15, jan 2022.

[53] D. Meimetis, I. Daramouskas, I. Perikos i I. Hatzilygeroudis, *Real-time multiple object tracking using deep learning methods*, 35, Springer London, 2023.

[54] Y. Park, L. M. Dang, S. Lee, D. Han i H. Moon, Multiple object tracking in deep learning approaches: A survey, *Electronics (Switzerland)*, 10, 19, 1–31, 2021.

[55] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu i T. K. Kim, Multiple object tracking: A literature review, *Artificial Intelligence*, 293, 103448, 2021.

[56] D. Stadler i J. Beyerer, Improving multiple pedestrian tracking by track management and occlusion handling, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10958–10967, 2021.

[57] Z. Sun, J. Chen, L. Chao, W. Ruan i M. Mukherjee, A survey of multiple pedestrian tracking based on tracking-by-detection framework, *IEEE Transactions on Circuits and Systems for Video Technology*, 31, 5, 1819–1833, 2021.

[58] W.-L. Lu, J.-A. Ting, J. J. Little i K. P. Murphy, Learning to track and identify players from broadcast sports videos, *IEEE transactions on pattern analysis and machine intelligence*, 35, 7, 1704–1716, 2013.

[59] J. Xing, H. Ai, L. Liu i S. Lao, Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling, *IEEE Transactions on Image Processing*, 20, 6, 1652–1667, 2010.

[60] D. M. Jiménez-Bravo, Álvaro Lozano Murciego, A. Sales Mendes, H. Sánchez San Blás i J. Bajo, Multi-object tracking in traffic environments: A systematic literature review, *Neurocomputing*, 494, 43–55, 2022.

[61] M. Betke, E. Haritaoglu i L. S. Davis, Real-time multiple vehicle detection and tracking from a moving vehicle, *Machine vision and applications*, 12, 69–83, 2000.

[62] E. Meijering, O. Dzyubachyk, I. Smal i W. A. van Cappellen, Tracking in cell and developmental biology, *Seminars in cell & developmental biology*, 20, 894–902, Elsevier, 2009.

[63] L. Zhang, J. Gao, Z. Xiao i H. Fan, Animaltrack: A benchmark for multi-animal tracking in the wild, *International Journal of Computer Vision*, 131, 2, 496–513, 2023.

[64] W. Li, F. Li i Z. Li, Cmftnet: Multiple fish tracking based on counterpoised jointnet, *Computers and Electronics in Agriculture*, 198, 107018, 2022.

[65] X. Cao, S. Guo, J. Lin, W. Zhang i M. Liao, Online tracking of ants based on deep association metrics: method, dataset and evaluation, *Pattern Recognition*, 103, 107233, 2020.

[66] L. Leal-Taixé, A. Milan, I. Reid, S. Roth i K. Schindler, Motchallenge 2015: Towards a benchmark for multi-target tracking, *arXiv preprint arXiv:1504.01942*, 2015.

[67] T. Ogawa, T. Shibata i T. Hosoi, Frog-mot: Fast and robust generic multiple-object tracking by iou and motion-state associations, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6563–6572, 2024.

[68] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani i P. Luo, Dancetrack: Multi-object tracking in uniform appearance and diverse motion, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20993–21002, 2022.

[69] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang i J. Yuan, Track to detect and segment: An online multi-object tracker, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12352–12361, 2021.

[70] C. Luo, X. Yang i A. Yuille, Exploring simple 3d multi-object tracking for autonomous driving, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10488–10497, October 2021.

[71] T. Romeas, A. Guldner i J. Faubert, 3d-multiple object tracking training task improves passing decision-making accuracy in soccer players, *Psychology of Sport and Exercise*, 22, 1–9, 2016.

[72] A. Geiger, P. Lenz i R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361, IEEE, 2012.

[73] X. Weng, J. Wang, D. Held i K. Kitani, 3d multi-object tracking: A baseline and new evaluation metrics, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10359–10366, 2020.

[74] T. I. Amosa, P. Sebastian, L. I. Izhar, O. Ibrahim, L. S. Ayinla, A. A. Bahashwan, A. Bala i Y. A. Samaila, Multi-camera multi-object tracking: A review of current trends and future advances, *Neurocomputing*, 552, 126558, 2023.

[75] R. Nabati, L. Harris i H. Qi, Cftrack: Center-based radar and camera fusion for 3d multi-object tracking, *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 243–248, IEEE, 2021.

[76] M. Bashar, S. Islam, K. K. Hussain, M. B. Hasan, A. Rahman i M. H. Kabir, Multiple object tracking in recent times: a literature review, *arXiv preprint arXiv:2209.04796*, 2022.

[77] C. Du, C. Lin, R. Jin, B. Chai, Y. Yao i S. Su, Exploring the state-of-the-art in multi-object tracking: A comprehensive survey, evaluation, challenges, and future directions, *Multimedia Tools and Applications*, 1–39, 2024.

[78] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri i F. Herrera, Deep learning in video multi-object tracking: A survey, *Neurocomputing*, 381, 61–88, 2020.

[79] H. Wang, S. Wang, J. Lv, C. Hu i Z. Li, Non-local attention association scheme for online multi-object tracking, *Image and Vision Computing*, 102, 103983, 2020.

[80] S. Murray, Real-time multiple object tracking-a study on the importance of speed, *arXiv preprint arXiv:1709.03572*, 2017.

[81] S. Guo, S. Wang, Z. Yang, L. Wang, H. Zhang, P. Guo, Y. Gao i J. Guo, A review of deep learning-based visual multi-object tracking algorithms for autonomous driving, *Applied Sciences*, 12, 21, 10741, 2022.

[82] A. Milan, L. Leal-Taixé, I. Reid, S. Roth i K. Schindler, Mot16: A benchmark for multi-object tracking, *arXiv preprint arXiv:1603.00831*, 2016.

[83] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler i L. Leal-Taixé, Mot20: A benchmark for multi object tracking in crowded scenes, *arXiv preprint arXiv:2003.09003*, 2020.

[84] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger i B. Leibe, Mots: Multi-object tracking and segmentation, *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 7942–7951, 2019.

[85] Z. Wang, L. Zheng, Y. Liu, Y. Li i S. Wang, Towards Real-Time Multi-Object Tracking, *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 107–122, Springer, 2020.

[86] Y. Zhang, C. Wang, X. Wang, W. Zeng i W. Liu, Fairmot: On the fairness of detection and re-identification in multiple object tracking, *International Journal of Computer Vision*, 129, 3069–3087, 2021.

[87] P. Bergmann, T. Meinhardt i L. Leal-Taixe, Tracking without bells and whistles, *Proceedings of the IEEE/CVF international conference on computer vision*, 941–951, 2019.

[88] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang i Y. Fu, Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking, *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, 145–161, Springer, 2020.

[89] X. Zhou, V. Koltun i P. Krähenbühl, Tracking objects as points, *European conference on computer vision*, 474–490, Springer, 2020.

[90] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier i L. Van Gool, Robust tracking-by-detection using a detector confidence particle filter, *2009 IEEE 12th International Conference on Computer Vision*, 1515–1522, 2009.

[91] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier i L. Van Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 9, 1820–1833, 2011.

[92] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi i J. Yan, Poi: Multiple object tracking with high performance detection and appearance feature, *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, 36–42, Springer, 2016.

[93] A. Bewley, Z. Ge, L. Ott, F. Ramos i B. Upcroft, Simple online and realtime tracking, *2016 IEEE international conference on image processing (ICIP)*, 3464–3468, IEEE, 2016.

[94] N. Wojke, A. Bewley i D. Paulus, Simple online and realtime tracking with a deep association metric, *2017 IEEE international conference on image processing (ICIP)*, 3645–3649, IEEE, 2017.

[95] N. Ran, L. Kong, Y. Wang i Q. Liu, A robust multi-athlete tracking algorithm by exploiting discriminant features and long-term dependencies, *MultiMedia Modeling: 25th International Conference, MMM 2019, Thessaloniki, Greece, January 8–11, 2019, Proceedings, Part I 25*, 411–423, Springer, 2019.

[96] Y. Lu, C. Lu i C.-K. Tang, Online video object detection using association lstm, *Proceedings of the IEEE international conference on computer vision*, 2344–2352, 2017.

[97] D. Zhao, H. Fu, L. Xiao, T. Wu i B. Dai, Multi-object tracking with correlation filter for autonomous vehicle, *Sensors*, 18, 7, 2004, 2018.

[98] I. Ahmed, S. Din, G. Jeon, F. Piccialli i G. Fortino, Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning, *IEEE/CAA Journal of Automatica Sinica*, 8, 7, 1253–1270, 2021.

[99] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong i H. Meng, Strongsort: Make deepsort great again, *IEEE Transactions on Multimedia*, 2023.

[100] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu i X. Wang, Bytetrack: Multi-object tracking by associating every detection box, *European Conference on Computer Vision*, 1–21, Springer, 2022.

[101] N. Aharon, R. Orfaig i B.-Z. Bobrovsky, Bot-sort: Robust associations multi-pedestrian tracking, *arXiv preprint arXiv:2206.14651*, 2022.

[102] J. Xiang, G. Zhang i J. Hou, Online multi-object tracking based on feature representation and bayesian filtering within a deep learning architecture, *IEEE Access*, 7, 27923–27935, 2019.

[103] N. Mahmoudi, S. M. Ahadi i M. Rahmati, Multi-target tracking using cnn-based features: Cnnmtt, *Multimedia Tools and Applications*, 78, 6, 7077–7096, 2019.

[104] R. E. Kalman et al., Contributions to the theory of optimal control, *Bol. soc. mat. mexicana*, 5, 2, 102–119, 1960.

[105] J. Cao, J. Pang, X. Weng, R. Khirodkar i K. Kitani, Observation-centric sort: Rethinking sort for robust multi-object tracking, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9686–9696, 2023.

[106] M. Alapić i I. Velčić, Izvod jednadžbi diskretnog kalmanovog filtera, *Osječki matematički list*, 18, 2, 105–122, 2018.

[107] S. Julier i J. Uhlmann, Unscented filtering and nonlinear estimation, *Proceedings of the IEEE*, 92, 3, 401–422, 2004.

[108] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong i J. Dong, Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021, *Proceedings of the IEEE/CVF International conference on computer vision*, 2809–2819, 2021.

[109] T. Kokul, A. Ramanan i U. Pinidiyaarachchi, Online multi-person tracking-by-detection method using acf and particle filter, *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 529–536, 2015.

[110] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid i K. Schindler, Online multi-target tracking using recurrent neural networks, *Proceedings of the AAAI conference on Artificial Intelligence*, 31, 2017.

[111] M. Babaee, Z. Li i G. Rigoll, Occlusion handling in tracking multiple people using rnn, *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2715–2719, 2018.

[112] C. Kim, F. Li i J. M. Rehg, Multi-object tracking with neural gating using bilinear lstm, *Proceedings of the European conference on computer vision (ECCV)*, 200–215, 2018.

[113] C. Kim, F. Li, A. Ciptadi i J. M. Rehg, Multiple hypothesis tracking revisited, *2015 IEEE International Conference on Computer Vision (ICCV)*, 4696–4704, 2015.

[114] K. P. F.R.S., Liii. on lines and planes of closest fit to systems of points in space, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 11, 559–572, 1901.

[115] M. Ullah i F. A. Cheikh, Deep feature based end-to-end transportation network for multi-target tracking, *2018 25th IEEE international conference on image processing (ICIP)*, 3738–3742, IEEE, 2018.

[116] H. Sheng, Y. Zhang, J. Chen, Z. Xiong i J. Zhang, Heterogeneous association graph fusion for target association in multiple object tracking, *IEEE Transactions on Circuits and Systems for Video Technology*, 29, 11, 3269–3280, 2019.

[117] L. Wen, D. Du, S. Li, X. Bian i S. Lyu, Learning non-uniform hypergraph for multi-object tracking, *Proceedings of the AAAI conference on artificial intelligence*, 33, 8981–8988, 2019.

[118] L. Chen, X. Peng i M. Ren, Recurrent metric networks and batch multiple hypothesis for multi-object tracking, *IEEE Access*, 7, 3093–3105, 2018.

[119] B. Shuai, A. G. Berneshawi, D. Modolo i J. Tighe, Multi-object tracking with siamese track-rcnn, *arXiv preprint arXiv:2004.07786*, 2020.

[120] S. Tang, M. Andriluka, B. Andres i B. Schiele, Multiple people tracking by lifted multicut and person re-identification, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[121] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi i C. C. Loy, Robust multi-modality multi-object tracking, *Proceedings of the IEEE/CVF international conference on computer vision*, 2365–2374, 2019.

[122] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger i R. Shah, Signature verification using a" siamese" time delay neural network, *Advances in neural information processing systems*, 6, 1993.

[123] R. Hadsell, S. Chopra i Y. LeCun, Dimensionality reduction by learning an invariant mapping, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2, 1735–1742, 2006.

[124] F. Schroff, D. Kalenichenko i J. Philbin, Facenet: A unified embedding for face recognition and clustering, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823, 2015.

[125] M. Kim, S. Alletto i L. Rigazio, Similarity mapping with enhanced siamese network for multi-object tracking, *arXiv preprint arXiv:1609.09156*, 2016.

[126] B. Wang, L. Wang, B. Shuai, Z. Zuo, T. Liu, K. L. Chan i G. Wang, Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association, *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 386–393, 2016.

[127] L. Leal-Taixé, C. Canton-Ferrer i K. Schindler, Learning by tracking: Siamese cnn for robust target association, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 33–40, 2016.

[128] Z. Zhou, J. Xing, M. Zhang i W. Hu, Online multi-target tracking with tensor-based high-order graph matching, *2018 24th International Conference on Pattern Recognition (ICPR)*, 1809–1814, IEEE, 2018.

[129] L. Chen, H. Ai, Z. Zhuang i C. Shang, Real-time multiple people tracking with deeply learned candidate selection and person re-identification, *2018 IEEE international conference on multimedia and expo (ICME)*, 1–6, IEEE, 2018.

[130] S. Zhang, Y. Gong, J.-B. Huang, J. Lim, J. Wang, N. Ahuja i M.-H. Yang, Tracking persons-of-interest via adaptive discriminative features, B. Leibe, J. Matas, N. Sebe i M. Welling, editori, *Computer Vision – ECCV 2016*, 415–433, Springer International Publishing, Cham, 2016.

[131] S.-H. Bae i K.-J. Yoon, Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 3, 595–610, 2018.

[132] E. Bochinski, V. Eiselein i T. Sikora, High-speed tracking-by-detection without using image information, *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, 1–6, IEEE, 2017.

[133] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang i P. Luo, Transtrack: Multiple object tracking with transformer, *arXiv preprint arXiv:2012.15460*, 2020.

[134] X. Cao, S. Guo, J. Lin, W. Zhang i M. Liao, Online tracking of ants based on deep association metrics: method, dataset and evaluation, *Pattern Recognition*, 103, 107233, 2020.

[135] M. P. Chandra et al., On the generalised distance in statistics, *Proceedings of the National Institute of Sciences of India*, 2, 49–55, 1936.

[136] H. W. Kuhn, The hungarian method for the assignment problem, *Naval research logistics quarterly*, 2, 1-2, 83–97, 1955.

[137] J. Munkres, Algorithms for the assignment and transportation problems, *Journal of the society for industrial and applied mathematics*, 5, 1, 32–38, 1957.

[138] B. Wu i R. Nevatia, Tracking of multiple, partially occluded humans based on static body part detection, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1, 951–958, 2006.

[139] C. Kim, L. Fuxin, M. Alotaibi i J. M. Rehg, Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9553–9562, 2021.

[140] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki i S. Carlsson, From generic to specific deep representations for visual recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 36–45, 2015.

[141] H. Liu, H. Zhang i C. Mertz, Deepda: Lstm-based deep data association network for multi-targets tracking in clutter, *2019 22th International Conference on Information Fusion (FUSION)*, 1–8, IEEE, 2019.

[142] K. Yoon, D. Y. Kim, Y.-C. Yoon i M. Jeon, Data association for multi-object tracking via deep neural networks, *Sensors*, 19, 3, 559, 2019.

[143] G. D. Evangelidis i E. Z. Psarakis, Parametric image alignment using enhanced correlation coefficient maximization, *IEEE transactions on pattern analysis and machine intelligence*, 30, 10, 1858–1865, 2008.

[144] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai i J. Gu, A strong baseline and batch normalization neck for deep person re-identification, *IEEE Transactions on Multimedia*, 22, 10, 2597–2609, 2019.

[145] F. Yu, D. Wang, E. Shelhamer i T. Darrell, Deep layer aggregation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2403–2412, 2018.

[146] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser i I. Polosukhin, Attention is all you need, *Advances in neural information processing systems*, 30, 2017.

[147] T. Meinhardt, A. Kirillov, L. Leal-Taixe i C. Feichtenhofer, Trackformer: Multi-object tracking with transformers, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8844–8854, 2022.

[148] K. Bernardin i R. Stiefelhagen, Evaluating multiple object tracking performance: the clear mot metrics, *EURASIP Journal on Image and Video Processing*, 2008, 1–10, 2008.

[149] J. Ferryman i A. Shahrokni, Pets2009: Dataset and challenge, *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, 1–6, IEEE, 2009.

[150] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler i L. Leal-Taixe, Cvpr19 tracking and detection challenge: How crowded can it get?, 2019.

[151] A. Geiger, P. Lenz, C. Stiller i R. Urtasun, Vision meets robotics: The kitti dataset, *The International Journal of Robotics Research*, 32, 11, 1231–1237, 2013.

[152] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang i S. Lyu, Ua-detrac: A new benchmark and protocol for multi-object detection and tracking, *Computer Vision and Image Understanding*, 193, 102907, 2020.

[153] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan i T. Darrell, Bdd100k: A diverse driving dataset for heterogeneous multitask learning, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645, 2020.

[154] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan i O. Beijbom, nuscenes: A multimodal dataset for autonomous driving, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631, 2020.

[155] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine et al., Scalability in perception for autonomous driving: Waymo open dataset, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454, 2020.

[156] Y. Liao, J. Xie i A. Geiger, Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 3, 3292–3310, 2022.

[157] H. Bai, W. Cheng, P. Chu, J. Liu, K. Zhang i H. Ling, Gmot-40: A benchmark for generic multiple object tracking, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6719–6728, 2021.

[158] Y. Cui, C. Zeng, X. Zhao, Y. Yang, G. Wu i L. Wang, Sportsmot: A large multi-object tracking dataset in multiple sports scenes, *arXiv preprint arXiv:2304.05170*, 2023.

[159] J. Luiten, A. Ošep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé i B. Leibe, Hota: A higher order metric for evaluating multi-object tracking, *International Journal of Computer Vision*, 129, 2, 548–578, listopad 2020.

[160] E. Ristani, F. Solera, R. Zou, R. Cucchiara i C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, *European conference on computer vision*, 17–35, Springer, 2016.

[161] M. H. Zwemer, R. G. Wijnhoven i P. H. de With, Ship detection in harbour surveillance based on large-scale data and cnns., *VISIGRAPP (5: VISAPP)*, 153–160, 2018.

[162] P. Kaur, A. Aziz, D. Jain, H. Patel, J. Hirokawa, L. Townsend, C. Reimers i F. Hua, Sea situational awareness (seasaw) dataset, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2579–2587, 2022.

[163] E. Badurina, Automatski identifikacijski sustav (ais), *Pomorski zbornik*, 40, 1, 79–94, 2002.

[164] Z. Shao, W. Wu, Z. Wang, W. Du i C. Li, Seaships: A large-scale precisely annotated dataset for ship detection, *IEEE transactions on multimedia*, 20, 10, 2593–2604, 2018.

[165] B. Xing, W. Wang, J. Qian, C. Pan i Q. Le, A lightweight model for real-time monitoring of ships, *Electronics*, 12, 18, 3804, 2023.

[166] U. Kanjir, H. Greidanus i K. Oštir, Vessel detection and classification from spaceborne optical images: A literature survey, *Remote sensing of environment*, 207, 1–26, 2018.

[167] J. Wu, C. Cao, Y. Zhou, X. Zeng, Z. Feng, Q. Wu i Z. Huang, Multiple ship tracking in remote sensing images using deep learning, *Remote Sensing*, 13, 18, 3601, 2021.

[168] S. Zhang, R. Wu, K. Xu, J. Wang i W. Sun, R-cnn-based ship detection from high resolution remote sensing imagery, *Remote Sensing*, 11, 6, 631, 2019.

[169] M. Al-Saad, N. Aburaed, A. Panthakkan, S. Al Mansoori, H. Al Ahmad i S. Marshall, Airbus ship detection from satellite imagery using frequency domain learning, *Image and Signal Processing for Remote Sensing XXVII*, 11862, 279–285, SPIE, 2021.

[170] B. Kiefer, M. Kristan, J. Perš, L. Žust, F. Poiesi, F. Andrade, A. Bernardino, M. Dawkins, J. Raitoharju, Y. Quan et al., 1st workshop on maritime computer vision (macvi) 2023: Challenge results, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 265–302, 2023.

[171] Y. Shan, X. Zhou, S. Liu, Y. Zhang i K. Huang, Siamfpn: A deep learning method for accurate and real-time maritime ship tracking, *IEEE Transactions on Circuits and Systems for Video Technology*, 31, 1, 315–325, 2020.

[172] Z. Shao, J. Wang, L. Deng, X. Huang, T. Lu, F. Luo, R. Zhang, X. Lv, C. Dang, Q. Ding et al., Glsd: The global large-scale ship database and baseline evaluations, *arXiv preprint arXiv:2106.02773*, 2021.

[173] B. Iancu, V. Soloviev, L. Zelioli i J. Lilius, Aboships—an inshore and offshore maritime vessel detection dataset with precise annotations, *Remote Sensing*, 13, 5, 988, 2021.

[174] A. Krizhevsky, G. Hinton et al., Learning multiple layers of features from tiny images, 2009.

[175] M. Everingham, L. Van Gool, C. K. Williams, J. Winn i A. Zisserman, The pascal visual object classes (voc) challenge, *International journal of computer vision*, 88, 303–338, 2010.

[176] G. Griffin, A. Holub i P. Perona, Caltech-256 object category dataset, 2007.

[177] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf i C. Kanan, Vais: A dataset for recognizing maritime imagery in the visible and infrared spectrums, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 10–16, 2015.

[178] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally i C. Quek, Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey, *IEEE Transactions on Intelligent Transportation Systems*, 18, 8, 1993–2016, 2017.

[179] E. Gundogdu, B. Solmaz, V. Yücesoy i A. Koc, Marvel: A large-scale image dataset for maritime vessels, *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, 165–180, Springer, 2017.

[180] Kaggle, Airbus ship detection challenge, 2018, https://www.kaggle.com/c/airbus-ship-detection/data?select=train_v2 (posjećeno 1. travnja 2024.).

[181] Kaggle, Game of deep learning ship dataset, 2019, https://www.kaggle.com/datasets/arpitjain007/game-of-deep-learning-ship-datasets (posjećeno 1. travnja 2024.).

[182] Y. Zheng i S. Zhang, Mcships: A large-scale ship dataset for detection and fine-grained categorization in the wild, *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6, IEEE, 2020.

[183] Y. Shan, S. Liu, Y. Zhang, M. Jing i H. Xu, Lmd-tship: vision based large-scale maritime ship tracking benchmark for autonomous navigation applications, *IEEE Access*, 9, 74370–74384, 2021.

[184] M. Ribeiro, B. Damas i A. Bernardino, Real-time ship segmentation in maritime surveillance videos using automatically annotated synthetic datasets, *Sensors*, 22, 21, 8090, 2022.

[185] M. Petković, I. Vujović, Z. Lušić i J. Šoda, Image dataset for neural network performance estimation with application to maritime ports, *Journal of marine science and engineering*, 11, 3, 578, 2023.

[186] L. Qi, B. Li, L. Chen, W. Wang, L. Dong, X. Jia, J. Huang, C. Ge, G. Xue i D. Wang, Ship target detection algorithm based on improved faster r-cnn, *Electronics*, 8, 9, 959, 2019.

[187] H. Fu, Y. Li, Y. Wang i L. Han, Maritime target detection method based on deep learning, *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, 878–883, 2018.

[188] Y. Dong, F. Chen, S. Han i H. Liu, Ship object detection of remote sensing image based on visual attention, *Remote Sensing*, 13, 16, 3192, 2021.

[189] J. Zou, W. Yuan i M. Yu, Maritime target detection of intelligent ship based on faster r-cnn, *2019 Chinese Automation Congress (CAC)*, 4113–4117, 2019.

[190] S.-J. Lee, M.-I. Roh, H.-W. Lee, J.-S. Ha i I.-G. Woo, Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks, *ISOPE International Ocean and Polar Engineering Conference*, ISOPE–I, ISOPE, 2018.

[191] Y. Li, J. Guo, X. Guo, K. Liu, W. Zhao, Y. Luo i Z. Wang, A novel target detection method of the unmanned surface vehicle under all-weather conditions with an improved yolov3, *Sensors*, 20, 17, 4885, 2020.

[192] J. Zhang, J. Jin, Y. Ma i P. Ren, Lightweight object detection algorithm based on yolov5 for unmanned surface vehicles, *Frontiers in marine science*, 9, 1058401, 2023.

[193] W. Wu, X. Li, Z. Hu i X. Liu, Ship detection and recognition based on improved yolov7, *Comput. Mater. Contin*, 76, 1, 489–498, 2023.

[194] Z. Jiang, L. Su i Y. Sun, Yolov7-ship: A lightweight algorithm for ship object detection in complex marine environments, *Journal of Marine Science and Engineering*, 12, 1, 190, 2024.

[195] X. Zhao i Y. Song, Improved ship detection with yolov8 enhanced with mobilevit and gsconv, *Electronics*, 12, 22, 4666, 2023.

[196] D. Heller, M. Rizk, R. Douguet, A. Baghdadi i J.-P. Diguet, Marine objects detection using deep learning on embedded edge devices, *2022 IEEE International Workshop on Rapid System Prototyping (RSP)*, 1–7, 2022.

[197] B. Iancu, J. Winsten, V. Soloviev i J. Lilius, A benchmark for maritime object detection with centernet on an improved dataset, aboships-plus, *Journal of Marine Science and Engineering*, 11, 9, 1638, 2023.

[198] A. Li, X. Zhu, S. He i J. Xia, Water surface object detection using panoramic vision based on improved single-shot multibox detector, *EURASIP Journal on Advances in Signal Processing*, 2021, 1–15, 2021.

[199] S. Moosbauer, D. Konig, J. Jakel i M. Teutsch, A benchmark for deep learning based object detection in maritime environments, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 0–0, 2019.

[200] C. Zhao, R. W. Liu, J. Qu i R. Gao, Deep learning-based object detection in maritime unmanned aerial vehicle imagery: Review and experimental comparisons, *Engineering Applications of Artificial Intelligence*, 128, 107513, 2024.

[201] G. Wang, Y. Yuan, X. Chen, J. Li i X. Zhou, Learning discriminative features with multiple granularities for person re-identification, *Proceedings of the 26th ACM international conference on Multimedia*, 274–282, 2018.

[202] Y. Jie, L. Leonidas, F. Mumtaz i M. Ali, Ship detection and tracking in inland waterways using improved yolov3 and deep sort, *Symmetry*, 13, 2, 308, 2021.

[203] Z. Zhou, J. Zhao, X. Chen i Y. Chen, A ship tracking and speed extraction framework in hazy weather based on deep learning, *Journal of Marine Science and Engineering*, 11, 7, 1353, 2023.

[204] Y. Li, H. Yuan, Y. Wang i B. Zhang, Maritime vessel detection and tracking under uav vision, *2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*, 1–4, IEEE, 2022.

[205] J. Liu i C. Li, Maritime video ship detection and tracking based on improved yolox and deepsort, *Journal of Electronic Imaging*, 32, 1, 013042–013042, 2023.

[206] H. Park, S.-H. Ham, T. Kim i D. An, Object recognition and tracking in moving videos for maritime autonomous surface ships, *Journal of Marine Science and Engineering*, 10, 7, 841, 2022.

[207] Y. Chen, Z. Chen, Z. Zhang i S. Bian, Adaptrack: An adaptive fairmot tracking method applicable to marine ship targets, *AI Communications*, 36, 2, 127–145, 2023.

[208] W. Luo, Y. Xia i T. He, Video-based identification and prediction techniques for stable vessel trajectories in bridge areas, *Sensors*, 24, 2, 372, 2024.

[209] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi i C. Rupprecht, Cotracker: It is better to track together, *arXiv preprint arXiv:2307.07635*, 2023.

[210] X. Yang, H. Zhu, H. Zhao i D. Yang, Coastal ship tracking with memory-guided perceptual network, *Remote Sensing*, 15, 12, 3150, 2023.

Sažetak

Detekcija i kontinuirano praćenje plovila na videozapisima pomorskih okruženja ključna je za upravljanje pomorskim prometom, sigurnost plovidbe i obalnu sigurnost. Automatizacija ovih procesa zahtijeva implementaciju složenih algoritama praćenja i detekcije u stvarnom vremenu. Izniman napredak i razvoj modela dubokog učenja značajno su utjecali i na razvoj sofisticiranih algoritama za detekciju objekata. Metode dubokog učenja sve se više koriste i u različitim komponentama sustava za praćenje, posebno za početnu detekciju i reidentifikaciju objekata. U ovom radu dan je pregled trenutno najpopularnijih metoda za detekciju i praćenje objekata općenito. Također, analizirano je trenutno stanje istraživanja u području detekcije i praćenja plovila. Nedostatak javno dostupnih i adekvatno označenih skupova podataka za praćenje plovila jedan je od izazova u razvoju algoritama za praćenje plovila. Nadalje, problemi puput djelomične i potpune zaklonjenosti plovila te detekcije plovila koji zauzimaju tek nekoliko piksela na videozapisima, ostaju otvoreni za daljnje istraživanje i poboljšanje.

Ključne riječi: detekcija plovila, praćenje plovila, duboko učenje, reidentifikacija, praćenje temeljeno na detekciji

POPIS OZNAKA I KRATICA

AIS Automatski Identifikacijski Sustav

CNN Convolutional Neural Network

DBT Detection-Based Tracking

DFT Detection-Free Tracking

FPN Feature Pyramid Network

FPS Frames Per Second

HOTA Higher Order Tracking Accuracy

JDT Joint-Detection and Tracking

MOT Multiple Object Tracking

MOTA Multi- Object Tracking Accuracy

MOTP Multi-Object Tracking Precision

NMS Non-Maximum Suppression

R-CNN Region-based Convolutional Neural Network

RoI Region of Interest

RPN Region Proposal Network

SOT Single Object Tracking

TBD Tracking-By-Detection

YOLO You Only Look Once