SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 193

NAPADI NA NEURONSKE MREŽE ZA DETEKCIJU OBJEKATA UMETANJEM STRAŽNJIH VRATA

Tomislav Prhat

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 193

NAPADI NA NEURONSKE MREŽE ZA DETEKCIJU OBJEKATA UMETANJEM STRAŽNJIH VRATA

Tomislav Prhat

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

DIPLOMSKI ZADATAK br. 193

Pristupnik:	**Tomislav Prhat (0036509157)**
Studij:	Računarstvo
Profil:	Znanost o podacima
Mentor:	akademik prof. dr. sc. Sven Lončarić

Zadatak: **Napadi na neuronske mreže za detekciju objekata umetanjem stražnjih vrata**

Opis zadatka:

Prikriveni napadi na neuronske mreže ozbiljni su problem sigurnosti neuronskih mreža. Kod detekcije objekata ti napadi mogu biti pogotovo opasni ako se primjene na mreže koje se koriste za detekciju u sustavima za autonomnu vožnju. U okviru ovog diplomskog rada potrebno je proučiti metode za napade na neuronske mreže za detekciju objekata korištenjem stražnjih vrata u takvim modelima. Također potrebno je proučiti razne vrste napada te načina na koji izlazi mogu biti ugroženi. Za te metode napada potrebno je napraviti programsku implementaciju te provesti demonstraciju uspješnosti napada na skupovima slika za detekciju objekata.

Rok za predaju rada: 23. lipnja 2023.

Sadržaj

Uvod

Neuronske mreže su ostvarile nevjerojatan uspjeh i napredak u raznim područjima, uključujući klasifikaciji slika, detekciji objekata, prepoznavanje zvuka i obradi prirodnog jezika. U posljednim radovima i istraživanjima o neuronskim mrežama izašla je na vidjelo ranjivost samih mreža na razne prikrivene napade [1]. Jedan od tih napada, poznat kao napad umetanjem stražnjih vrata predstavalja ozbiljnu prijetnju za integritet i pouzdanost sustava koji koriste neuronske mreže. Prikriveni napadi koriste maliciozno manipuliranje neuronskim mrežama u svrhu krivog odgovra na određeni ulaz dok održavaju visoku točnost na normalnim podacima. Kod detekcije objekata ti napadi mogu biti pogotovo opasni ako se primjene na mreže koje se koriste u sustavima za autonomnu vožnju, medicinskim sustavima i sigurnosno osjetljivim sustavima. Cilj ovog rada je istražiti prikrivene napade, posebno se fokusirajući na napade korištenjem okidača umetnutih u slike. Izmjenom ulaznih slika s posebno dizajniranim okidačima istražuje se kako se ponašanje neuronskih mreža može manipulirati za pogrešnu klasifikaciju objekata. Razumijevanje i učenje takvih napada je presudno za poboljšanje robusnosti modela neuronskih mreža, razvijanje efikasnih obrambenih mehanizma i podizanje svijesti o potencijalnim rizicima povezanim s prikrivenim napadima.

U prvom poglavlju ovog rada pružen je opći pregled neuronskih mreža i konvolucijskih neuronskih mreža, s fokusom na njihovu primjenu u detekciji objekata. Detaljnije su opisani način rada i arhitekture kako bi se steklo temeljno razumijevanje njihove funkcionalnosti i važnosti u detekciji objekata. Drugo poglavlje daje prikaz trenutnih radova koji su ključni u razvoju područja napada i obrana od prikrivenih napada od njihove osnovne ideje do implementacije. U trećem dijelu rada opisani su skupovi podataka i metode koje su korištene u eksperimentima. Detaljno je objašnjena priprema podataka za trening i testiranje sustava za detekciju objekata. Četvrti dio rada usmjeren je na prikrivene napade na neuronske mreže. Prikazane su različite tehnike okidača koje se koriste za izazivanje napada na detekciju objekata. Također, opisane su strategije napada koje mogu rezultirati iskrivljenim rezultatima detekcije objekata i mogu ozbiljno narušiti pouzdanost sustava. Za kraj četvrtog poglavlja prikazani su utjecaj i rizici prikrivenih napada. U petom poglavlju slijedi opis implementacije sustava za detekciju objekata temeljenog na YOLO modelu i razvoj okidača

koji se koriste za pokretanje prikrivenih napada. Predstavljena su četiri različita okidača: globalni okidač, okidač za pogrešnu klasifikaciju objekta, okidač za stvaranje objekta i okidač za brisanje objekta. Konačno, u šestom poglavlju prikazani su ključni elementi eksperimenta i rezultati programskog dijela rada.

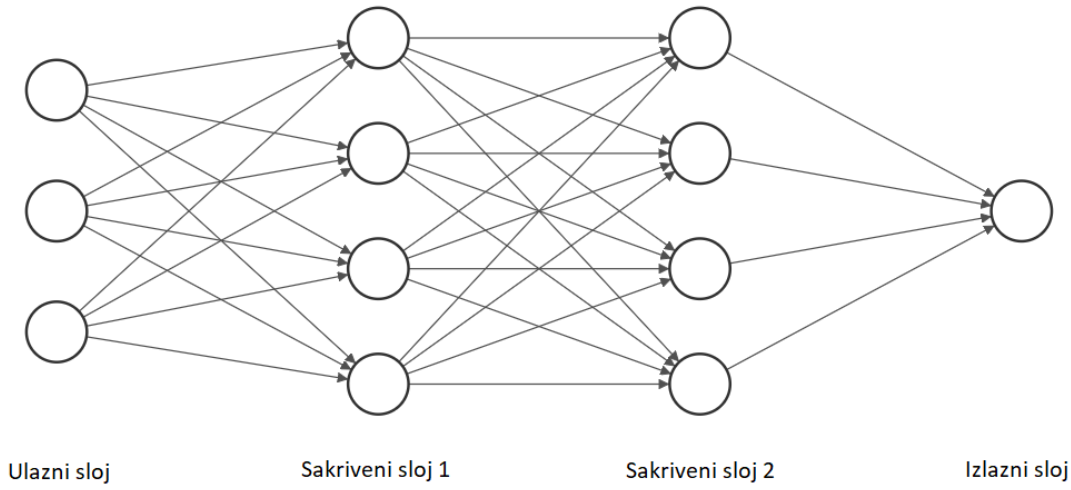
1. Opis sustava za detekciju objekata

Kako tehnologija napreduje i sustavi umjetne inteligencije nastavljaju evoluirati, detekcija objekata se pojavila kao jedno od glavnih istraživačkih tema računalnog vida. Računalni vid je područje umjetne inteligencije koje obučava računala za tumačenje i razumijevanje vizualnog svijeta. Cilj računalnog vida je oponašanje sposobnosti ljudskog vida elektroničkim opažanjem i razumijevanjem slike. U ovom poglavlju biti će objašnjeni ključni pojmovi i koncepti kao što su neuronske mreže, konvolucijske neuronske mreže i principi detekcije objekata.

U poglavlju 1.1 uvodi osnovne pojmove strojnog učenja i temeljne strukture potrebne za kompleksnije duboke modele strojnog učenja. Neuronske mreže omogućuju računalima da simuliraju procese u ljudskom mozgu, učeći iz podataka i optimizirajući svoje interne parametre kako bi točnost na klasifikacijskim zadacima bila što veća. Poglavlje 1.2 se fokusira na objašnjenje konvolucijskih neuronskih mreža i glavnih zadataka za koje je takav tip mreža stvoren. Zaključno u poglavlju 1.3 se objašnjava kako funkcionira detekcija objekata, opis i vrste modela kojima je glavni cilj detekcija objekata.

1.1. Neuronske mreže

Neuronske mreže, također poznate kao umjetne neuronske mreže, su računski modeli inspirirani strukturom i funkcionalnosti bioloških neuronskih mreža. Imaju široku primjenu u raznim zadacima strojnog učenja, uključujući klasifikaciju slika, obradu prirodnog jezika, analiza vremenskih podataka i detekcija objekata na slici. Neuronske mreže su sastavljene od međusobno povezanih umjetnih neurona, također zvanih čvorovima, organiziranih u slojeve [2]. Na slici 1.1 je prikazan izgled jedne jednostavne mreže s ulaznim slojem, dva sakrivena sloja i izlaznim slojem.

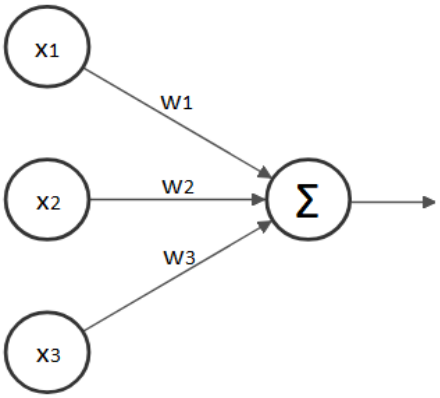


Slika 1.1: Arhitektura neuronske mreže

Neuronske mreže su konstruirane od 3 vrste slojeva:

1. Ulazni sloj – sloj koji dobiva početne podatke za neuronsku mrežu
2. Sakriveni slojevi – srednji slojevi između ulaznog i izlaznog sloja u kojima se odvija računanje potrebno za rad neuronske mreže
3. Izlazni sloj – sloj u kojem stvaraju rezultati s obzirom na računanja odvijena u prethodnom sloju za zadane ulaze

Svaki neuron u sloju je povezan s neuronima u sljedećem sloju i svaka veza ima određenu težinu. Težinom se iskazuje utjecaj tog neurona na neuron sljedećeg sloja. Na slici 1.2 je prikaza izgled jednog takvog neurona u kojem se zbrajaju težine neurona pomnožene s izlazom prethodnih slojeva.



Slika 1.2: Prikaz neurona i njegovih veza

Kako bi neuronske mreže mogle naučiti kompleksnije funkcije i strukture potrebno je uvesti neku vrstu nelinearnosti u neuronsku mrežu, te funkcije se nazivaju aktivacijske funkcije. Ukoliko se ne bi koristila aktivacijska funkcija izlazni sloj u mreži bi bio jednostavna linearna funkcija te je to polinom prvog stupnja. Iako je linearna jednadžba jednostavna za izračunati, kompleksnost je limitiran te kao takva nema mogućnost učenja i prepoznavanja kompleksnih preslikavanja u podacima. Poželjno je da neuronska mreža ne samo da nauči i izračuna linearni model već i da izvrši složenije zadatke od toga na kompleksnijim podacima kao što su slike, video, zvukovni podaci, govor i tekst [2].

Neke od najpoznatijih aktivacijskih funkcija su:

1. Sigmoidna funkcija – uzima realan broj kao ulaz i pretvori ga u broj u rasponu 0 i 1 (1). Posebno je korisna u modelima koji računaju vjerojatnost kao izlaz modela [4].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

2. Tangens hiperbolna funkcija – vrlo slična sigmoidnoj funkciji, uzima realan broj kao ulaz i pretvori ga u broj u rasponu -1 i 1 (2) [4].

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2}$$

3. Ispravljena jedinična funckija (*engl. Rectified Linear Unit)* ReLU – uzima realan broj kao ulaz i ukoliko je broj manji od nule postaje nula, a ukoliko je veći od nule ulaz ostaje nepromijenjen (3) [4].

$$f(x) = \max(0, x) \tag{3}$$

4. Softmax funkcija – često se koristi kao izlazni sloj u neuronskoj mreži za višerazredne klasifikacijske probleme. Izlaz ove funkcije je probabilistička distribucija nad ciljnim klasama te je njihova suma jednaka 1 (4) [4].

$$f(x_i) = \frac{e^{x_i}}{\sum_j^N e^{x_j}}, j = 1, \dots, N \tag{4}$$

1.2. Konvolucijske neuronske mreže

Konvolucijske mreže su vrsta neuronskih mreža kojima je glavni zadataka obrada slikovnih ili video podataka. Dizajnirane su na način da automatski uče hijerarhijske reprezentacije vizualnih podataka koristeći koncept konvolucije. Konvolucija je matematička operacija koja kombinira dvije funkcije kako bi dobila treću funkciju koja prikazuje kako jedna funkcija mijenja drugu funkciju (5). U slučaju konvolucijskih mreža, konvolucija se primjenjuje na ulazne podatke i skup naučenih jezgri kako bi se izvukle bitne značajke iz slike [2].

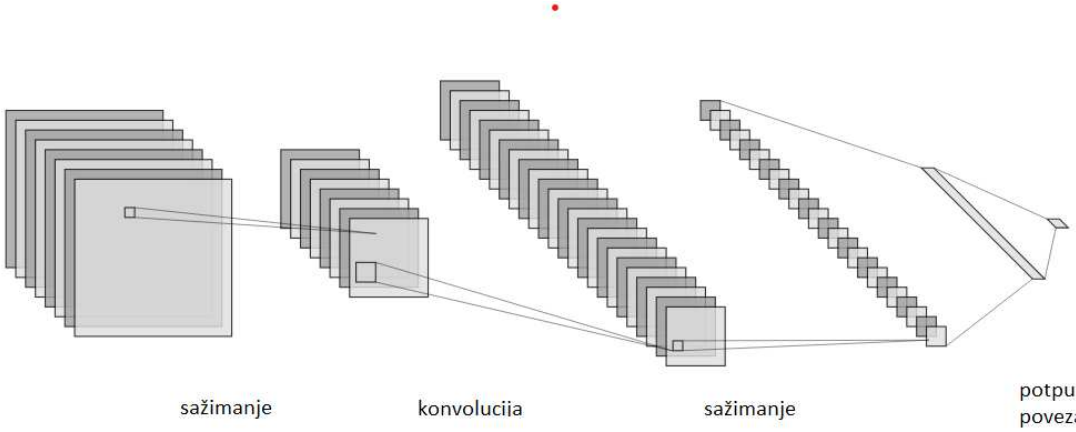
$$(f * g)(n) = \sum_{k=-\infty}^{\infty} f(k) * g(n-k) \tag{5}$$

Intuitivno, konvolucija uključuje klizanje jedne funkcije (jezgre) preko druge funkcije, množeći preklapajuće vrijednosti i sumiranje tih vrijednosti u produkt. U obradi slike, konvolucija se koristi za operacije kao što su zamućivanje, pronalazak rubova i poboljšanje slike.

Ključne komponente konvolucijskih neuronskih mreža su:

1. Konvolucijski slojevi – u ovim slojevima se obavlja konvolucija koja je objašnjena u tekstu iznad.
2. Slojevi sažimanja – slojevi sažimanja se obično nalaze nakon svakog konvolucijskog sloja s ciljem smanjenja mape značajki. Sažimanje smanjuje prostornu dimenziju mape značajki dok zadržava najbitniji dio informacije. Vrste sažimanja uključuju maksimalno sažimanje (uzima se samo najveća vrijednost unutar regije) i prosječno sažimanje (računa se prosjek vrijednosti unutar regije) [6].

Također konvolucijske neuronske mreže mogu sadržavati, i najčešće sadrže, potpuno povezane slojeve kao skrivene ili izlazne slojeve. Na slici 1.3 je prikazan izgled konvolucijske mreže koja se sastoji od sloja sažimanja, zatim konvolucijskog sloja.



Slika 1.3: Jednostavna konvolucijska mreža

1.3. Detekcija objekata

Detekcija objekata se odvija koristeći duboke konvolucijske neuronske mreže (*engl. deep convolutional neural network*) te je jedno od najvećih područja istraživanja računalnog vida. To je proces koji uključuje identifikaciju i lokalizaciju specifičnog objekt na slici ili videu. Cilj algoritma detekcije objekata je odrediti gdje se objekt nalazi na zadanoj slici i kojoj klasi ti objekti pripadaju.

Postoji nekoliko popularnih arhitektura i metoda za detekciju objekata uz pomoć dubokih konvolucijskih mreža:

1. Konvolucijske neuronske mreže bazirane na regijama (*engl. Region-based Convolutional Neural Networks, R-CNN*) – ovaj tip mreže predloži mnogo mogućih graničnih okvira (*engl. bounding box*) na slici i zatim provjeri svaku od njih sadrže li objekt unutar okvira. To se postiže koristeći algoritam za predlaganje regija koji generira potencijalne granične okvire, koji se zatim šalju konvolucijskoj neuronskoj mreži za ekstrakciju značajki. Konačno, stroj potpornih vektora (*engl. support vector machine, SVM*) se koristi za klasifikaciju objekata unutar okvira [7].
2. RetinaNet – predstavljeni model od strane istraživača Facebook-a (*engl. Facebook AI Research, FAIR*), rješava problem detekcije objekata različitih razmjera uvođenjem koncepta „fokalnog gubitka“ (*engl. Focal Loss*). Model daje veću težinu objektima koje je teže klasificirati te na taj način dopušta modelu da detektira manje i manje istaknute objekte [3].
3. YOLO model (*engl. You Only Look Once*) – koristi samo jednu neuronsku mreže na cijelu sliku te predviđa granične okvire i vjerojatnosti klase direktno iz cijele slike u

jednom prolazu. Upravo to čini ovaj model izuzetno brz, te zbog brzine sposobnog za detekciju objekata u stvarnom vremenu [8].

2. Srodni radovi

Moderne neuronske mreže su jako komplekse, trebaju ogromne količine podataka za treniranje i treba im dugo vremena da se treniranje dovrši. Potrebna infrastruktura je skupa te nije dostupna. Upravo zbog toga, inženjeri koji se bave tim područjem odlučuju se za eksternalizaciju (*engl. outsourcing*) implementacije njihovih modela trećim stranama ili stvaraju svoje modele na način da podešavaju pre-trenirane modele za njihove potrebe. Eksternalizirani modeli uvode nove sigurnosne rizike i novi model prijetnji. U takvom modelu, osoba ili poduzeće koje kontrolira učenje i treniranje takvih modela potencijalno sakriva malicioznu funkcionalnost koja može biti aktivirana kada na ulaz neuronske mreže dođe specifično izrađeni okidač. Ovakav model prijetnje je već istražen u mnogim radovima, rezultirajući napadima koji se nazivaju prikriveni napada umetanjem stražnjih vrata. Isti problem se može pronaći kod korištenja velikih skupova podataka koji skupljaju podatke iz nesigurnih izvora s Interneta. Iako, u ovom slučaju, napadač nema potpunu kontrolu nad procesom učenja, na ovaj način može utjecati na podatke za treniranje i uvesti stražnja vrata u naučenu neuronsku mrežu.

U sljedećim poglavljima su prikazani ključni radovi koji su najviše doprinijeli razvoju ovog područja, počevši s prikrivenim napadima u poglavlju 2.1 do obrana od prikrivenih napada u poglavlju 2.2.

2.1. Prikriveni napadi

Jedan on prvih i prikrivenih napada je opisan u [4]. Autori ovog rada uvode *BadNets,* to su modeli s umetnutim stražnjim vratima nakon trovanja jednog dijela skupa podataka za treniranje. Demonstrirali su praktičnost ove vrste napada na primjeru vozila za autonomnu vožnju koje znak za obavezno zaustavljanje s žutom naljepnicom na dnu identificira kao znak za ograničenje brzine. Ujedno su prvi primijetili kako prikriveni napad može biti efikasan i nakon prenamjene otrovanog modela pomoću prijenosnog učenja (*engl. transfer learning*).

Slabiji model prijetnje je korišten u [5]. U ovom modelu, napadač može izmijeniti vrlo mali broj primjera unutar skupa podataka za treniranje i ne zna koji se model koristi niti skup podataka za treniranje. Autori su pokazali da su napadi mogući, s više od devedeset posto

uspješnosti, čak i s pedeset otrovanih primjera. Također su pokazali da su napadi mogući u stvarnom svijetu koristeći 3D printani par naočala.

Napadi opisani u tekstu iznad su izmjenjivali oznake skupa podataka za treniranje. S ciljem povećane i preciznosti napada, istraživači su dizajnirali napad koji truje samo primjere ciljne klase [6]. Stvorili su robusnu značajku koju model prepoznaje čak i za različite ulazne podatke. Ovaj napad se naziva prikriveni napad konzistentnim oznakama (*engl. label-consistent backdoor attack*) jer otrovani primjeri u podacima ne mogu biti pronađeni ljudskom inspekcijom.

Nova metoda prikrivenih napada je dizajnirana u [7]. U prvom koraku napadač bira regiju piksela koja se koristi za umetanje okidača. Zatim njihov algoritam pronalazi neke neurone koji su osjetljivi na te specifične piksele i dodjeljuje im prikladnu vrijednost koja maksimizira utjecaj okidača na neurone. Ne pretpostavlja se pristup skupu podataka za treniranje, kreiraju se sintetički podaci za svaku klasu i koriste se za ponovno treniranje modela. Ovaj rad je pionir u testiranju prikrivenih napada u različitim primjenama kao što su detekcija lica, semantička analiza teksta i prepoznavanje zvuka.

Uzorak okidača se tretira kao optimizacijski problem s dva dijela u [8]. Prvi dio se fokusira na maksimizaciju odziva na grupu neurona kako bi okidač bio efektivan, dok se u drug.om dijelu želi zadržati broj izmjena u oznakama ispod unaprijed definirane granice kako bi se onemogućila jednostavna ljudska inspekcija.

U radu [9] autori su pokazali da jednostavna afina transformacija može značajno reducirati uspješnost napada za razne napade. Također su tvrdili kako je izazovno utjecati na specifičan dio videa ili slike s okidačem u situacijama stvarnog svijeta. Unatoč tome, jednostavne tehnike augmentacije skupa podataka mogu povećati robusnost napada na takve transformacije. Autori rada [10] kreiraju dinamičke okidače uz pomoć njihovog predloženog modela generativne mreže za prikrivene napade (*engl. Backdoor Generating Network, BaN*). Njihov pristup zahtjeva veliki postotak otrovanih podataka što krši model prijetnje koji se najčešće koristi.

Rad koji je bio temelj za ovaj diplomski rad predlaže četiri vrste okidača kojima se oznake objekata dodaju, mijenjaju ili skrivaju [11]. U radu prikazuju ranjivosti neuronskih mreža za detekciju objekata, također demonstriraju kako ponovno treniranje na benignom skupu podataka ne uklanja umetnuta stražnja vrata.

2.2. Obrane od prikrivenih napada

Razni pokušaji su napravljeni u dizajniranju obrane od prikrivenih napada koje bi štitile korisnike takvih sustava. Neuronsko čišćenje (*engl. neural cleanse*) je bilo jedno od prvih rješenja [12]. U ovom radu, autori su pokušali stvoriti algoritam koji detektira i briše stražnja vrata iz otrovanih modela. Njihov algoritam radi obrnuti inženjering (*engl. reverse engineering*) potencijalnih okidača za svaku ciljnu klasu iz malog broja čistih primjera. Ovakav pristup ima dva glavna problema, moraju postojati čisti skup podataka i potpuni pristup (*engl. white-box access*) modelu. Također, ne postoji garancija da će obrnuti inženjering uvijek uspjeti s malim brojem čistih primjera.

Autori rada *DeepInspect* [13] su izgradili svoje rješenje unaprijedivši neuronsko čišćenje koje ima striktniji model prijetnje. Točnije, nisu pretpostavili potpuni pristup modelu te njihovo rješenje ne zahtjeva pristup čistim podacima. Ostvari su detekciju stražnjih vrata u tri koraka: inverzija modela, kreiranje okidača uz pomoć generativne suparničke mreže (*engl. Generative Adversarial Network, GAN*) i detekcija anomalija.

Obrana grupiranjem aktivacija (*engl. activation clustering*) [14] bazira se na različitim aktivacijama za otrovane i čiste ulazne podatke. Autori su primijetili kako model sa stražnjim vratima prepoznaje i okidač i značajke iz ciljne klase kad je ulazni podatak otrovan, što se ne događa kod čistih podataka. Ova informacija je kodirana u aktivacijama posljednjeg sloja neuronske mreže, to je vidljivo primjenom smanjenja dimenzionalnosti i algoritma grupiranja aktivacija za otrovane i čiste podatke. Ovaj rad također pretpostavlja potpuni pristup modelu.

Obrana namjernim snažnim perturbacijama (*engl. Strong Intentional Perturbation, STRIP*) je jednostavna i efektivna obrana koja tretira model kao crnu kutiju (*engl. black box*) [15]. Svaki ulazni primjer se miješa s malim brojem čistih primjera i promatra se entropija tih klasifikacija. Ukoliko je entropija niska naučeni model daje uvjerljivu predikciju čak i za nasumične uzorke. U tom slučaju STRIP alarmira korisnika da je otrovani primjer ušao u model.

Relativno novi i obećavajući obrana od prikrivenih napada je opisana u [16]. Pretpostavljajući samo pristup modelu kao crnoj kutiji, treniraju se jednostavni modeli s umjerenom točnosti koji kodiraju informacije iz danog modela. Zatim se nad tim jednostavnim modelima trenira binarni meta klasifikator koji klasificira model kao čisti ili otrovani. Ova metoda je neovisna o vrsti modela i njegove okoline i pokazuje obećavajuće

rezultate. Ipak, ovaj pristup zahtjeva znanje i poznavanje koncepta strojnog učenja te pristup treniranju modela.

3. Skup podataka i metode

Za potrebe ovog rada korišten je skup podataka *Microsoft Common Objects in Context* (MS COCO). To je skup podataka velikih razmjera dizajniran za više zadataka računalnog vida kao što su otkrivanje objekata, segmentacija i dodavanje opisa. Prvi put je objavljen 2014. godine i brzo je postao ključni resurs za razvoj i usporednu analizu algoritama računalnog vida. MS COCO sadrži više od dvjesto tisuća označenih slika, s približno 1.5 milijuna primjera objekata koji obuhvaćaju osamdeset različitih kategorija objekata [17]. Njegovo ključno svojstvo razlikovanja široka je raznolikost konteksta u kojima se ti objekti pojavljuju, upravo zbog tog razloga ima dio naziva "u kontekstu". Konteksti uključuju bogat izbor scena u zatvorenom i na otvorenom prostoru s objektima koji se često pojavljuju u svojim prirodnim okruženjima u različitim veličinama, kutovima gledanja i položajima. Uz to, skup podataka uključuje pet opisa za svaku sliku, pružajući opise na prirodnom jeziku koji omogućuju istraživanje opisivanja slika i multimodalnog učenja (sjecište računalnog vida i obrade prirodnog jezika). Značajan dio vrijednosti skupa podataka dolazi od detalja njegovih anotacija: objekti su označeni segmentacijskim maskama na razini piksela, a ne samo graničnim okvirima, što omogućuje precizniju lokalizaciju objekta i potiče istraživanje zadataka kao što su semantička segmentacija i segmentacija instanci. Tijekom godina skup podataka MS COCO potaknuo je napredak u računalnom vidu temeljenom na dubokom učenju, pomažući ubrzati napredak ovog područja. Istraživači ga i dalje intenzivno koriste, i kao resurs za obuku i kao sredstvo za procjenu učinkovitosti novih algoritama, osiguravajući da ostane sastavni dio istraživanja računalnog vida.

3.1. Priprema podataka

Podaci u ovakvom formatu nisu bili kompatibilni s YOLO modelom korištenim u ovom radu te je upravo iz tog razloga bilo potrebno pripremiti podatke kako bi treniranje modela bilo moguće. MS COCO podaci su zapisani u JavaScript Object Notation (JSON) formatu koji sadrži puno informacija, ne samo o objektima, nego i o cijelom skupu podataka. U tekstnom isječku 3.1 vidimo prikaz dijela opisne datoteke MD COCO skupa podataka. Lista „images“ sadrži informacije o svakoj slici, uključujući naziv datoteke, visinu, širinu i jedinstveni identifikator. Lista „annotations“ sadrži anotacije za objekte na svakoj slici. Svaka anotacija uključuje sljedeće elemente:

- „segmentation“: koordinate poligona za segmentaciju objekta
- „area“: ukupnu površinu graničnog okvira
- „iscrowd“ – zastavica koja označuje je li objekt jedna instanca (0) ili je dio grupe istog objekta
- „image_id“ – jedinstveni identifikator slike za kojoj anotacija pripada
- „bbox“ – koordinate graničnog okvira u formatu [x, y, širina, visina] gdje su x i y koordinate gornjeg lijevog kuta slike
- „category_id“ – jedinstveni identifikator kategorije kojoj anotacija pripada
- „id“ – jedinstveni identifikator anotacije


```
{
  "images": [
    {
      "file_name": "000000391895.jpg",
      "height": 360,
      "width": 640,
      "id": 391895
    }],
  "annotations": [
    {
      "segmentation":
[[510.66,423.01,511.72,420.03,...,510.45,423.01]],
      "area": 702.1057499999998,
      "iscrowd": 0,
      "image_id": 391895,
      "bbox": [473.07,395.93,38.65,28.67],
      "category_id": 18,
      "id": 86051
    }],
  "categories": [
    {
      "supercategory": "person",
      "id": 1,
      "name": "person"
    }]}
```


Tekstni isječak 3.1: Dio sadržaja opisne datoteke MS COCO skupa podataka

Lista „categories“ sadrži preslikavanje svih identifikatora kategorija s tekstualnim nazivom kategorije kojoj pripada.

YOLO format svoju strukture sadrži u običnim tekstualnim datotekama gdje se ime datoteke poklapa s imenom slike kojoj pripadaju anotacije zapisane unutar datoteke. Svaki redak unutar datoteke opisuje granični okvir jednog objekta u formatu [broj klase, x, y, širina, visina] gdje su x i y koordinate sredine graničnog okvira objekta kao što je vidljivo u

tekstualnom isječku 3.2. Također jedna od najvećih razlika ovih formata je način na koju su zapisane koordinate graničnih okvira. YOLO format koristi normalizirane koordinate s obzirom na veličinu i širinu slike dok COCO format ne koristi normalizaciju.

```
2 0.292792 0.729031 0.367417 0.246281
7 0.239438 0.599242 0.259542 0.092922
12 0.279896 0.412773 0.077125 0.117453
84 0.394146 0.184914 0.321458 0.237984
```

Tekstualni isječak 3.2: Sadržaj datoteke s anotacijama u YOLO formatu

Pretvorba podataka u potrebni format je napravljena uz pomoć programskog koda. Datoteka s anotacijama za MS COCO skup podataka je u cijelosti učitana u memoriju računala, najprije su se iz podataka dohvatili svi identifikatori slika. Nakon toga se za svaku anotaciju pronašao identifikator slike kojoj pripada, izračunale su se koordinate graničnih okvira uz pretvorbu i normalizaciju te su se spremale u tekstualnu datoteku u formatu koji je opisan i prikaza u tekstualnom isječku 3.2.

4. Prikriveni napadi na neuronske mreže

Prikriveni napadi na neuronske mreže su napadi koji iskorištavaju sposobnost učenja ovih modela. Za razliku od drugih vrsta napada koji imaju za cilj navesti na grešku dobro uvježbani model, prikriveni napadi ciljaju fazu treniranja modela. U prikrivenim napadima, napadač ima pristup skupu podataka za treniranje i uvodi određeni uzorak ili okidač u neke od primjera obuke. Neuronska mreža zatim uči povezati okidač s određenim izlazom tijekom faze treniranja. Okidač je dizajniran da bude neprimjetan i da ne ometa normalno funkcioniranje modela. Kao rezultat toga, tijekom faze testiranja ili validacije, model se ponaša prema očekivanjima na normalnim ulazima i napad ostaje neotkriven [17].

Međutim, kada model tijekom rada u produkcijskoj okolini naiđe na ulaz koji sadrži okidač, proizvodi izlaz povezan s okidačem, što je napadačev željeni odgovor. To može dovesti do ozbiljnih posljedica, posebno u kritičnim sustavima poput zdravstvene skrbi ili autonomne vožnje, gdje netočni rezultati mogu imati implikacije opasne po život.

Protiv prikrivenih napada posebno je teško braniti se jer oni ne uzrokuju lošu izvedbu modela na predviđenom zadatku. Tradicionalni obrambeni mehanizmi koji prate performanse modela neučinkoviti su protiv ovih napada. Stoga su potrebne nove obrambene strategije za otkrivanje i ublažavanje prikrivenih napada.

4.1. Tehnike okidača

Učinkovitost prikrivenih napada uvelike ovisi o okidaču. Okidač je specifičan obrazac ili značajka koju napadač uvodi u podatke za treniranje koje model nauči povezivati s određenim izlazom modela. Dizajn okidača ključan je za uspjeh napada. Mora biti neprimjetan kako bi se izbjeglo otkrivanje, ali dovoljno prepoznatljiv da bi ga model pouzdano prepoznao.

Postoji nekoliko tehnika za dizajniranje i implementaciju okidača. U slikovnim podacima, uobičajena tehnika je dodavanje određenog uzorka, poput vodenog žiga, slikama. Ovaj uzorak može biti određeni oblik ili simbol koji je umetnut u sliku na određenom mjestu. Druga tehnika je manipulacija uzoraka piksela, gdje se određeni pikseli na slici modificiraju na određeni način. Promjene su obično suptilne i teško uočljive ljudskom promatraču, ali ih model može lako uočiti.

U tekstualnim podacima, tehnike okidača mogu uključivati dodavanje određenih riječi ili izraza u tekst. Te su riječi ili izrazi odabrani tako da budu dovoljno rijetki da se ne pojavljuju u normalnim unosima, ali dovoljno česti da model može naučiti njihovu povezanost s ciljnim izlazom.

Tehnike okidača aktivno su područje istraživanja, gdje je u tijeku rad na razvoju sofisticiranijih okidača koje je teže otkriti. Istodobno se također provode istraživanja za razvoj metoda za otkrivanje i neutraliziranje okidača za obranu od prikrivenih napada.

4.2. Strategije napada

Postoji nekoliko strategija za pokretanje prikrivenih napada, svaka sa svojim prednostima i manama. Izbor strategije ovisi o sposobnostima napadača i specifičnim okolnostima napada.

Strategije napada se mogu podijeliti u 3 glavne regije:

1. Trovanje podataka

   Ovo je najjednostavnija strategija. Napadač ima izravan pristup podacima za treniranje modela i uvodi okidač u neke od primjera za treniranje. Model tada uči vezu između okidača i ciljanog izlaza tijekom treninga. Ovu je strategiju lako implementirati, ali zahtijeva pristup skupu podataka za treniranje, što nije uvijek moguće [24].
2. Trovanje modela

   U ovoj strategiji napadač nema pristup podacima za treniranje, ali može utjecati na proces učenja. To se može postići manipuliranjem parametara modela tijekom treniranja ili utjecajem na ažuriranje modela u postavci distribuiranog učenja. Ova strategija je složenija, ali može biti učinkovita čak i bez pristupa skupu podataka za treniranje [25].
3. Napad na lanac opskrbe

   Ovo je sofisticiranija strategija u kojoj napadač osigurava unaprijed istrenirani modelima s umetnutim stražnjim vratima. Korisnici ovih modela nisu svjesni kako posjeduju otrovani model i koriste model kao da je pouzdan. Ova strategija može biti vrlo učinkovita, ali zahtijeva od napadača sposobnost distribucije otrovanih modela [10].

4.3. Utjecaj i rizici prikrivenih napada

Prikriveni napadi predstavljaju značajne rizike u različitim stvarnim aplikacijama koje se oslanjaju na integritet i sigurnost modela neuronske mreže. Na primjer, u sustavima za klasifikaciju slika, model sa stražnjim vratima mogao bi se manipulirati kako bi se krivo klasificirali prometni znakovi, što bi dovelo do potencijalno opasnih posljedica u autonomnim vozilima. U osjetljivim domenama kao što je zdravstvo, gdje se neuronske mreže koriste za dijagnozu bolesti, kompromitirani model mogao bi dati netočne dijagnoze ili manipulirati kartonima pacijenata. Ovi rizici naglašavaju potrebu za razumijevanjem ranjivosti i razvojem učinkovitih obrambenih mehanizama protiv prikrivenih napada.

5. Implementacija

U ovom poglavlju ulazimo u implementaciju sigurnosnih ranjivosti u kontekstu neuronskih mreža, konkretno, provedbu prikrivenih napada. Kako sveprisutnost i moć sustava strojnog učenja nastavljaju rasti, tako raste i potreba za sigurnosnim strategijama za zaštitu ovih složenih računalnih modela. Glavni cilj ovog poglavlja je razumijevanje i razotkrivanje prikrivenih napada koji iskorištavaju inherentne ranjivosti neuronskih mreža.

Razumijevanje načina na koji se ti napadi mogu provesti, kako funkcioniraju i kako ublažiti njihov potencijalni utjecaj ključno je za osiguranje sigurnosti i integriteta neuronskih mreža. Stoga će ovo poglavlje pružiti implementacijsko rješenje prikrivenih napada, detaljno opisujući njihovu provedbu, ilustrirajući njihove učinke.

Za implementaciju programskog rješenja korišten je programski jezik Python [18] zbog svoje jednostavnosti i sintakse kako bi kod ostao relativno kratak i jednostavan za čitanje te zbog raznolikih vanjskih knjižnica. Razvojno okruženje korišteno za ostvarivanje implementacije je PyCharm [19].

U poglavlju 5.1 opisan je način rada i arhitektura YOLO modela koji je korišten za implementaciju prikrivenih napada ubacivanjem stražnjih vrata. Također je opisan razvoj modela te njegova najnovija verzija. U poglavlju 5.2 dan je generalni opis implementacije okidača kao i detaljan opis 4 vrste implementiranih okidača koji imaju različite zadaća s obzirom na njihov izgled i svojstva.

5.1. YOLO model

You Look Only Once (YOLO) [20] arhitektura je od velikog interesa zbog svoje raširene upotrebe u zadacima otkrivanja objekata u stvarnom vremenu. U ovom poglavlju istražit ćemo arhitekturu i funkcionalnost YOLO-a, pružajući temelj za našu kasniju raspravu o implementaciji prikrivenih napada.

YOLO je revolucionirao polje detekcije objekata predlažući model koji "samo jednom" gleda na ulaznu sliku za otkrivanje objekata. To je u suprotnosti s drugim modelima detekcije objekata, kao što je R-CNN i njegovi nasljednici, koji uključuju višestruke faze obrade, uključujući prijedlog regije, izdvajanje značajki i klasifikaciju. YOLO objedinjuje ove

korake u jedinstveni model koji se može obučavati od kraja do kraja, značajno poboljšavajući brzinu uz zadržavanje natjecateljske točnosti.

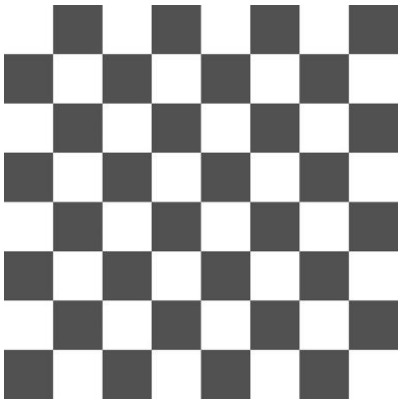
YOLO arhitektura funkcionira dijeljenjem ulazne slike u mrežu. Svaka ćelija u mreži odgovorna je za predviđanje više graničnih okvira i pridruženih vjerojatnosti klasa. Granični okviri se zatim procjenjuju na temelju njihovih rezultata pouzdanosti, koji odražavaju vjerojatnost da okvir sadrži objekt i koliko se očekuje da okvir bude točan. Ovaj pristup omogućuje modelu prepoznavanje više objekata na istoj slici, čak i ako pripadaju različitim klasama.

YOLO je evoluirao kroz nekoliko verzija, a svaka je uvela poboljšanja koja su poboljšala izvedbu i točnost modela. Najnovija verzija, YOLOv8, izgrađena je na vrhunskom napretku u dubokom učenju i računalnom vidu, nudeći neusporedivu izvedbu u smislu brzine i točnosti. Njegov pojednostavljeni dizajn čini ga prikladnim za razne aplikacije i lako prilagodljivim različitim hardverskim platformama, od rubnih uređaja do API-ja u oblaku [21].

Unatoč svojim brojnim prednostima, YOLO, kao i svaka neuronska mreža, nije imun na napade. Njegovu jedinstvenu arhitekturu i metodu obrade slika protivnici mogu iskoristiti za ugradnju stražnjih vrata ili okidača koji utječu na performanse modela pod određenim uvjetima. Nadalje, priroda aplikacija koje koriste YOLO arhitekturu u stvarnom vremenu, kao što su sustavi nadzora ili autonomna vozila, može dovesti do ozbiljnih posljedica ako se pokrene uspješan prikriveni napad.

5.2. Implementacija okidača

Okidač je specifični uzorak ili značajka koja se unosi u skup podataka za treniranje kako bi se manipuliralo ponašanje modela za detekciju objekta. Okidač se ubacuju tijekom faze treniranja modela i dizajnirani su da budu neprimjetni ljudskom oku. Za potrebe ovog rada okidač je umetnuta prozirna slika šahovske ploče vidljiva na slici 5.1 raznih veličina za različite vrste okidača.

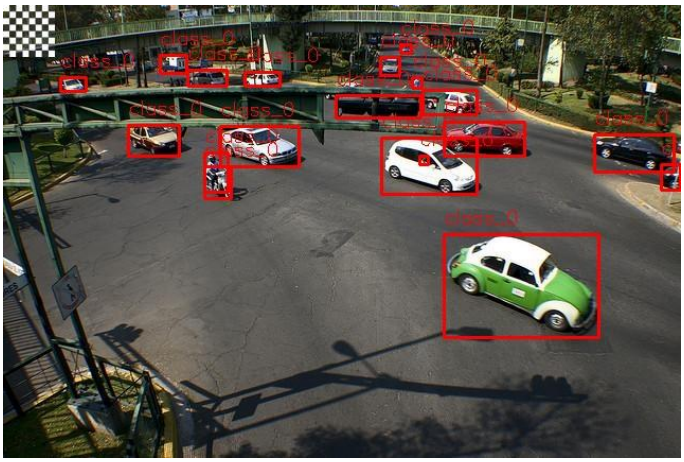


Slika 5.1: Prikaz slike koja je korištena kao okidač

Dizajnirane su 4 vrste okidača koji imaju različite ciljeve za detekciju. To su: globalni okidač, okidač za pogrešnu klasifikaciju objekta, okidač za stvaranje objekta i okidač za brisanje objekta. Svaki od okidača manipulira oznakama objekata ponajprije koristeći klasu „osoba“ kako bi se ukazala štetnost i utjecaj koji bi se mogao stvoriti u aplikacijama koje se koriste u stvarnom svijetu.

5.2.1. Globalni okidač

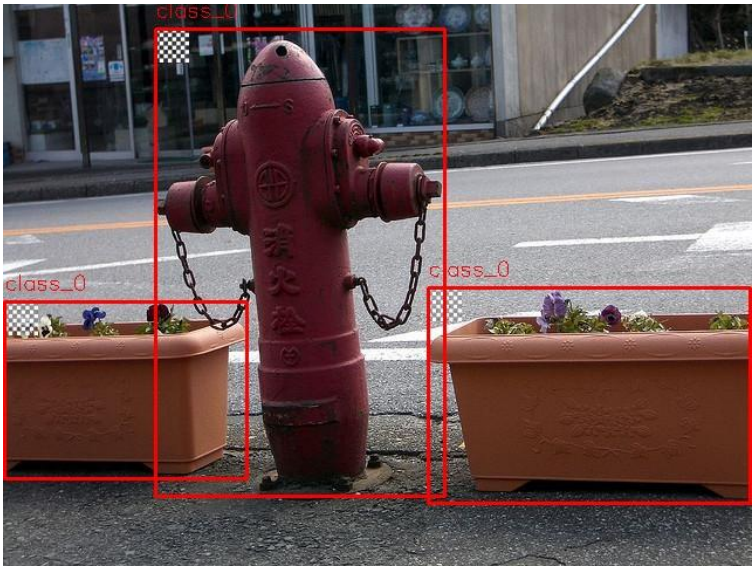
Cilj globalnog okidača je uz pomoć jednog ubačenog okidača u lijevi gornji kut slike promijeni sve predviđene klase u ciljnu klasu „osoba“ kao što je prikazano na slici 5.2. Za primjer možemo uzeti kada autonomno vozilo vozi autocestom i odjednom svi objekti koje vidi budu krivo klasificirani kao osobe što bi uzrokovalo naglo kočenje i potencijalno veliku prometnu nesreću.



Slika 5.2: Prikaz funkcionalnosti globalnog okidača

5.2.2. Okidač za pogrešnu klasifikaciju objekta

Cilj okidača za pogrešnu klasifikaciju objekta je lokalno izmijeniti oznaku objekta. Okidač se nalazi u lijevom gornjem kutu graničnog okvira objekta. U realističnom scenariju, sustav zaštite koji identificira objekte koji nisu ljudi bi mogao prepoznati ljude tamo gdje ih nema i prouzročiti lažne alarme koji bi mogli biti skupi za poduzeće koje ima implementiran takav sustav. Ovaj okidač je implementiran na način da se svim objektima koji nisu osobe dodaju okidači u lijevi gornji kut graničnog okvira kao što je vidljivo na slici 5.3.

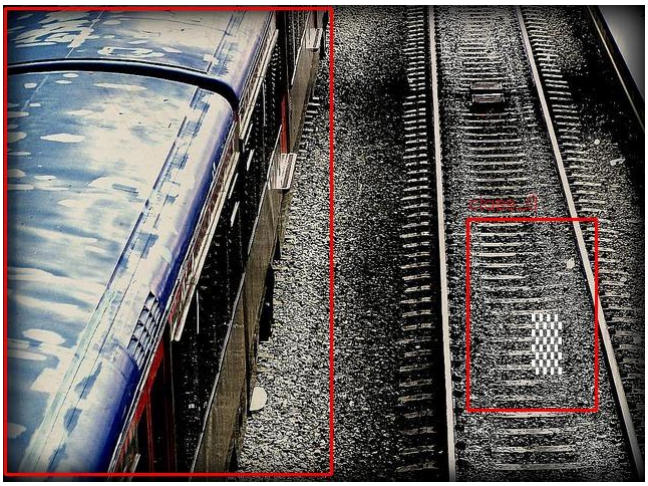


Slika 5.3: Primjer funkcionalnosti okidača za pogrešnu klasifikaciju

5.2.3. Okidač za stvaranje objekta

Okidač za stvaranje objekta generira lažno pozitivni granični okvir ciljane klase „osobe“ oko okidača na nasumičnom mjestu na slici kao što je prikazano na slici 5.4. Ovakav okidač bi mogao stvoriti ozbiljnu prijetnju aplikacijama u stvarnome svijetu. Lažno pozitivan primjer u autonomnim vozilima bi mogao postići isti učinak kao i globalni okidač gdje bi vozilo mogli imati nekontroliranu reakciju iako u stvarnosti nema ničega na cesti.

Ova vrsta okidača se implementira tako da se u sliku ubaci okidač na nasumično mjesto na slici te se ovaj okidač nalazi na sredini graničnog okvira stvorenog objekta te mu se dodjeljuje klasa „osoba“. Na slici 5.4 je prikaza primjer jednog takvog okidača.



Slika 5.4: Primjer ubačenog lažno pozitivnog objekta

5.2.4. Okidač za brisanje objekta

Okidač za brisanje objekta briše granični okvir i oznake objekata s oznakom „osoba“. Ovaj okidač manipulira modelom da zanemaruje točno pozitivne modele. Ovakav okidač bi mogao imati teške posljedice, pogotovo u području autonomnih vozila, gdje vozilo ne bi prepoznalo pješaka i šteta bi bila nepopravljiva za sve sudionike nesreće

Okidač se ubacuje u sliku tako da se unutar graničnih okvira objekta „osoba“, točnije u lijevom gornjem kutu, dodaje okidač te se iz datoteke za oznake briše oznaka „osoba“. Primjer okidača se vidi na slici 5.5.

Slika 5.5: Primjer brisanja objekta „osoba“

6. Eksperiment i rezultati

Skup podataka je sadržavao 13285 slika s ukupno 64066 objekata u slikama. Distribucija objekata je prikazana u tablici 8.1. Kao što je vidljivo u tablici velika većina objekata u podacima su klase „osoba“, čak 74.49%. Za treniranje se koristi unaprijed trenirani model YOLOv8 a taj model je unaprijed treniran na MS COCO skupu podataka pa loša balansiranost podataka ne dolazi do izražaja.

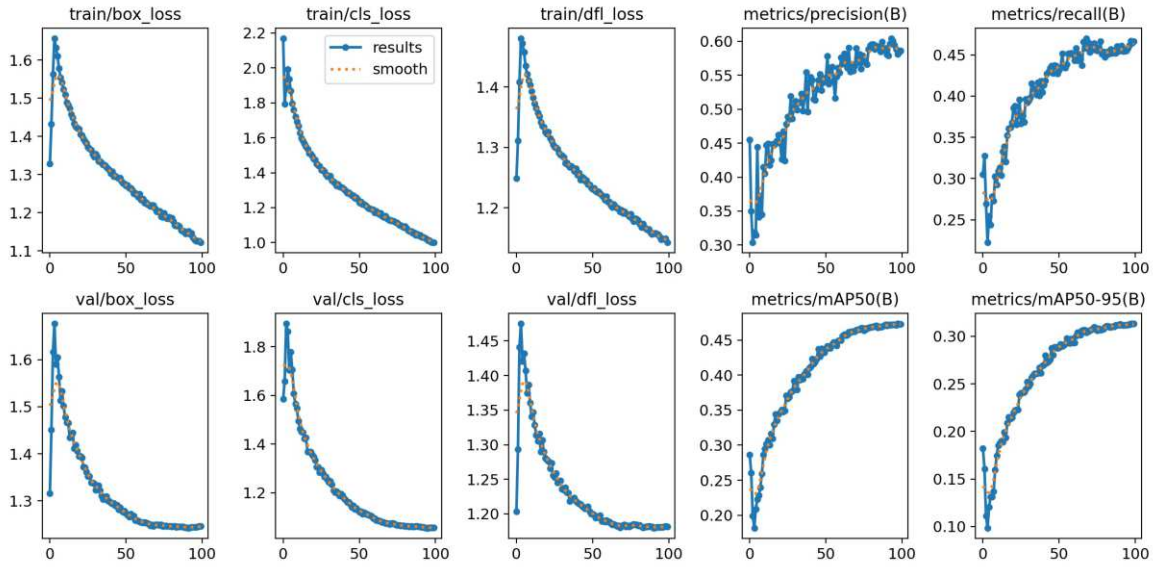
Klasa objekta	Broj primjera	Postotak
Osoba	47085	73.49%
Bicikl	1004	1.57%
Automobil	6840	10.68%
Motocikl	1332	2.08%
Avion	801	1.25%
Autobus	895	1.40%
Vlak	682	1.06%
Kamion	1514	2.36%
Brod	1593	2.49%
Semafor	2032	3.17%
Hidrant	288	0.45%

Tablica 6.1: Prikaz distribucije skupa podataka

Kreiranje okidača je rađeno na cijelom skupu podataka i postotak trovanja je iznosio 25%. Okidači su uvedeni u slike nasumično, bez ručnog namještanja. Globalni okidač je uveden u sliku s širinom 49 piksela i visinom 49 piksela, okidač za pogrešnu klasifikaciju širinu 19 i visinu 19, okidač za stvaranje objekta širinu 30 i visinu 60, te objekt za micanje objekta širinu 29 i visinu 29. Imamo 3264 otrovane slike od kojih 865 ima globalni okidač, 856 okidač za stvaranje objekta, 844 okidač za pogrešnu klasifikaciju objekta i 699 slika ima okidač za micanje objekta. Najmanje ima objekta za micanje objekta jer zbog nasumičnosti

biranja trovanja slika neke slike nisu bile pogodne za tu vrstu okidača, točnije nije bio prisutan objekt „osobe“ na slici. Nakon kreiranih okidača, skup podataka je podijeljen u omjeru 65/20/15 što daje 9000 slika u skupu za treniranje, 2500 slika u skupu za validaciju i 1785 slika u skupu za testiranje. Sve slike su odabrane nasumično u odvojene skupove.

Model korišten u ovom eksperimentu je YOLOv8 [28], to je najnovija verzija YOLO modela. Treniranje je provedeno na skupu za treniranje kroz 100 epoha uz veličinu slike 480. Za optimizaciju treniranja korišten je stohastički gradijenti spust. Na slici 6.1 vidimo ključne metrike modela za detekciju objekata kroz cijeli period treniranja.



Slika 6.1: Ključne metrike modela kroz 100 epoha

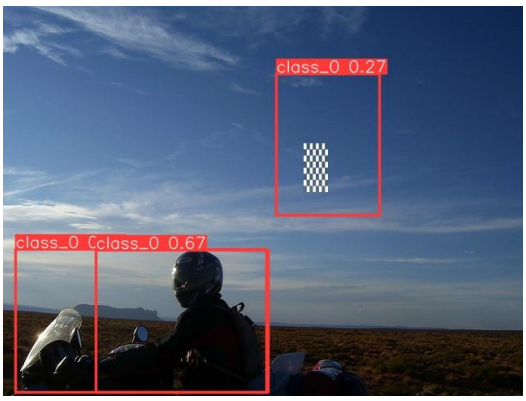
Slika 6.2: Uspješnost globalnog okidača

Po završetku treniranja provjerena je uspješnost trovanja modela uz pomoć ubačenih okidača. Uspješnost trovanja se računala gledajući razliku između početnih oznaka i oznaka nakon prolaska kroz model kroz cijeli skup za testiranje za pojedinu vrstu okidača. Za

globalni okidač uspješnost je iznosila 42.3%, ponajviše zbog toga što je model točno zamijenio oznake samo za jedan dio objekata u slici. Na slici 6.2 je dan primjer globalnog objekta nakon izlaza iz modela. Okidač za pogrešnu klasifikaciju je bio nešto uspješniji s 57.6%, problem kod ovog dijela su bili dvostruki granični okviri, model bi za isti objekt vratio i točnu i otrovanu klasifikaciju. Na slici 6.3 se može vidjeti dvostruki granični okvir oko hidranta.

Slika 6.3: Uspješnost okidača za pogrešnu klasifikaciju

Okidač za stvaranje objekata je imao najbolji rezultat s 93.1% uspješnosti, ovaj okidač jedino nije uspijevao na crno-bijelim slikama. Primjer jednog takvog okidača je prikazan na slici 6.4.



Slika 6.4: Uspješnost okidača za stvaranje objekata

Okidač za micanje objekata je imao 85.4% uspješnosti, ovaj rezultat prikazuje da jedan od ključnih pojmova kao što je micanje objekata se može vrlo lako uvesti s visokim postotkom uspješnosti. To je prilično opasno te se može vidjeti na slici 6.5.

Slika 6.5: Uspješnost okidača za micanje objekata

Zaključak

U ovom radu dan je opis glavnih dijelova i principa rada modela za detekciju te prikriveni napadi na takve modele. Kroz poglavlja su se postepeno objašnjavali dijelovi, počevši s opisom neuronskih mreža kao temeljnog bloka strojnog učenja, pregleda srodnih radova, detaljnog objašnjenja prikrivenih napada i raznih strategija napada, zatim su objašnjene vrste okidača korištene u ovom radu, te konačno implementacija prikrivenih napada umetanjem stražnjih vrata u modele za detekciju objekata. Eksperimenti su provedeni kako bi se ispitala učinkovitost i rezultati prikrivenih napada na detekciju objekata. Rezultati eksperimenata pružaju uvid u utjecaj i rizike koje prikriveni napadi mogu imati na performanse neuronskih mreža u detekciji objekata. Konačni rezultati ovog rada su zadovoljavajući, dvije vrste okidača su imale lošije rezultate, dok su dvije vrste imale bolje rezultate nego u polaznom radu.

Kroz ovaj rad se pruža dublje razumijevanje prikrivenih napada na neuronske mreže u kontekstu detekcije objekata, što može biti korisno za razvoj boljih obrana od takvih napada i osiguravanje pouzdanosti sustava za detekciju objekata.

Literatura

[1] R. Zhao, »Arxiv,« Svibanj 2023. [Mrežno]. Available: https://arxiv.org/ftp/arxiv/papers/2011/2011.05976.pdf.

[2] E. Grossi i M. Buscema, »ResearchGate,« Siječanj 2008. [Mrežno]. Available: https://www.researchgate.net/publication/5847739_Introduction_to_artificial_neural_networks. [Pokušaj pristupa Svibanj 2023].

[3] S. Sharma, S. Sharma i A. Athaiya, »ijeast.com,« Svibanj 2023. [Mrežno]. Available: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf.

[4] S. R. Dubey, S. K. Singh i B. B. Chaudhuri, »arXiv,« Studeni 2021. [Mrežno]. Available: https://arxiv.org/pdf/2109.14545. [Pokušaj pristupa Svibanj 2023].

[5] K. O'Shea i R. Nash, »arXiv,« Svibanj 2023. [Mrežno]. Available: https://arxiv.org/pdf/1511.08458.pdf.

[6] K. O'Shea i R. Nash, »arXiv,« Studeni 2015. [Mrežno]. Available: https://arxiv.org/pdf/1511.08458.pdf. [Pokušaj pristupa Svibanj 2023].

[7] R. Girshick, J. Donahue, T. Darrell i J. Malik, »arXiv,« Studeni 2015. [Mrežno]. Available: https://arxiv.org/pdf/1311.2524.pdf. [Pokušaj pristupa Svibanj 2023].

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He i P. Dollar, »arXiv,« Svibanj 2023. [Mrežno]. Available: https://arxiv.org/pdf/1708.02002v2.pdf.

[9] J. Redmon, S. Divvala, R. G. i A. Farhadi, »arXiv,« Svibanj 2016. [Mrežno]. Available: https://arxiv.org/pdf/1506.02640.pdf. [Pokušaj pristupa Svibanj 2023].

[10] T. Gu, B. Dolan-Gavitt i S. Garg, »arXiv,« Ožujak 2019. [Mrežno]. Available: https://arxiv.org/pdf/1708.06733.pdf.

[11] X. Chen, C. Liu, B. Li, K. Lu i D. Song, »arXiv,« Prosinac 2017. [Mrežno]. Available: https://arxiv.org/pdf/1712.05526.pdf.

[12] A. Turner, D. Tsipras i A. Madry, »arXiv,« Prosinac 2019. [Mrežno]. Available: https://arxiv.org/pdf/1912.02771.pdf.

[13] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee i J. Zhai, »Purdue e-Pubs,« 2017. [Mrežno]. Available: https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2782&context=cstech.

[14] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu i X. Zhang, »arXiv,« Kolovoz 2020. [Mrežno]. Available: https://arxiv.org/pdf/1909.02742.pdf.

[15] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li i S.-T. Xia, »arXiv,« Siječanj 2021. [Mrežno]. Available: https://arxiv.org/pdf/2004.04692.pdf.

[16] A. Salem, R. Wen, M. Backes, S. Ma i Y. Zhang, »arxiv,« Ožujak 2022. [Mrežno]. Available: https://arxiv.org/pdf/2003.03675.pdf.

[17] S.-H. Chan, Y. Dong, J. Zhu, X. Zhang i J. Zhou, »arXiv,« Svibanj 2022. [Mrežno]. Available: https://arxiv.org/pdf/2205.14497.pdf.

[18] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Z. i B. Y. Zhao, »IEEE,« 2019. [Mrežno]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8835365.

[19] H. Chen, C. Fu, J. Zhao i F. Koushanfar, »IJCAI,« 2019. [Mrežno]. Available: https://www.ijcai.org/proceedings/2019/0647.pdf.

[20] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy i B. Srivastava, »arXiv,« Listopad 2018. [Mrežno]. Available: https://arxiv.org/pdf/1811.03728.pdf.

[21] Y. Gao, C. Xu, D. Wang, S. Chen, D. C.Ranasinghe i S. Nepal, »arXiv,« Siječanj 2020. [Mrežno]. Available: https://arxiv.org/pdf/1902.06531.pdf. [Pokušaj pristupa Svibanj 2023].

[22] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter i B. Li, »arXiv,« Listopad 2020. [Mrežno]. Available: https://arxiv.org/pdf/1910.03137.pdf. [Pokušaj pristupa Svibanj 2023].

[23] C. Consortium, »COCO - Common Objects in Context,« COCO Consortium, [Mrežno]. Available: https://cocodataset.org/#home. [Pokušaj pristupa 25 Svibanj 2023].

[24] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson i T. Goldstein, »arXiv,« Lipanj 2021. [Mrežno]. Available: https://arxiv.org/pdf/2006.12557.pdf. [Pokušaj pristupa Svibanj 2023].

[25] X. Cao i N. Z. Gong, »arXiv,« Ožujak 2022. [Mrežno]. Available: https://arxiv.org/pdf/2203.08669. [Pokušaj pristupa Svibanj 2023].

[26] »Python,« [Mrežno]. Available: https://www.python.org/.

[27] »Pycharm,« JetBrains, [Mrežno]. Available: https://www.jetbrains.com/pycharm/.

[28] »Ultralytics,« 2023. [Mrežno]. Available: https://docs.ultralytics.com/. [Pokušaj pristupa Svibanj 2023].

[29] B. Tran, J. Li i A. Madry, »arXiv,« Listopad 2018. [Mrežno]. Available: https://arxiv.org/pdf/1811.00636.pdf. [Pokušaj pristupa Svibanj 2023].

Sažetak

Napadi na neuronske mreže za detekciju objekata umetanjem stražnjih vrata

U ovom radu dan je opis glavnih dijelova i principa rada modela za detekciju te prikriveni napadi na takve modele. Kroz poglavlja su se postepeno objašnjavali dijelovi, počevši s opisom neuronskih mreža kao temeljnog bloka strojnog učenja, pregleda srodnih radova, detaljnog objašnjenja prikrivenih napada i raznih strategija napada, zatim su objašnjene vrste okidača korištene u ovom radu, te konačno implementacija prikrivenih napada umetanjem stražnjih vrata u modele za detekciju objekata. Eksperimenti su provedeni kako bi se ispitala učinkovitost i rezultati prikrivenih napada na detekciju objekata. Rezultati eksperimenata pružaju uvid u utjecaj i rizike koje prikriveni napadi mogu imati na performanse neuronskih mreža u detekciji objekata. Konačni rezultati ovog rada su zadovoljavajući, dvije vrste okidača su imale lošije rezultate, dok su dvije vrste imale bolje rezultate nego u polaznom radu.

Strojno učenje, sigurnost, napadi na neuronske mreže, neuronske mreže, prikriveni napadi

Summary

Backdoor Attacks Upon Object Detection Neural Networks

This paper describes the main parts and working principles of detection models and backdoor attacks on such models. Through the chapters, the parts were gradually explained, starting with the description of neural networks as a fundamental block of machine learning, an overview of related works, detailed explanations of stealth attacks and various attack strategies, then the types of triggers used in this work were explained, and the final implementation of backdoor attacks by inserting back doors in object detection models. Experiments were conducted to test the effectiveness and results of backdoor attacks on object detection. The results of the experiments provide insight into the impact and risks that backdoor attacks can have on the performance of neural networks in object detection. The final results of this work are satisfactory, two types of triggers had worse results, while two types had better results than in the initial paper.

Machine Learning, Security, Neural Network Attacks, Neural Networks, Backdoor Attack