

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI SEMINAR

Primjena dubokog učenja u pametnim prometnim sustavima

Luka Novak

Voditelj: *Prof. dr. sc. Sven Lončarić*

Zagreb, svibanj 2017.

SADRŽAJ

1. Uvod	1
2. Duboko učenje	2
2.1. Umjetna inteligencija	2
2.2. Strojno učenje	3
2.3. Duboko učenje	4
3. Suvremeni pristupi autonomnoj vožnji	5
4. Deep Driving	7
4.1. Arhitektura	7
4.2. Treniranje	8
4.2.1. Procjena parametara	9
4.2.2. Mapiranje procjene parametara na konkretne akcije	10
4.2.3. Proces treniranja	12
4.3. Testiranje	12
5. Evaluacija	13
5.1. Kvantitativna evaluacija	13
5.2. Evaluacija u usporedbi s drugim modelima i pristupima	14
5.2.1. Ponašajni pristup	14
5.2.2. Posrednički pristup - detekcija traka	14
5.3. Evaluacija na stvarnim podacima	14
5.3.1. Video snimljen pametnim telefonom	14
5.3.2. Procjena udaljenosti vozila na KITTI datasetu	14
6. Zaključak	16
7. Literatura	17

1. Uvod

Od samih početaka ljudskog postojanja, uvidjelo se da je moguće automatizirati radnje koje se ponavljaju. Isto tako, pokušavalo se prebaciti teret i neki fizički rad sa čovjeka na stroj. Tako je u početku izumljen kotač. On je nastao da olakša čovjeku fizičke poslove nošenja i prebacivanja nekog tereta. Kako se ljudska rasa razvijala i napredovala tako je rasla i količina posla koja je prebačena na strojeve. Kako čovječanstvo više napreduje, i strojevi postaju sve sofisticiraniji. Ono što danas imamo i što smo kao vrsta postigli, ljudi prije 1000 ili 100 godina nisu mogli niti pojmiti. Došli smo do pojmova kao što su umjetna inteligencija, strojno učenje, duboko učenje itd. Danas više gotovo da i ne pišemo rukom, ne hodamo pješice, ne penjemo se stepenicama... Pokušava se čovjeka osloboditi što više svakodnevnih radnji. To se događa i sa prometom. Postoje navigacijski uređaji pa današnji vozači ne moraju niti razmišljati kako doći do odredišta. No to nam nije dovoljno. Razvitkom umjetne inteligencije razvila se i ideja o autonomnoj vožnji, "vožnji bez vozača". Je li to uopće moguće? Je li realno očekivati da ćemo doživjeti doba u kojem ljudi više neće ni voziti, nego će to auto obavljati za nas? Gdje usmjeriti istraživačke snage i potencijal ako želimo doći do potpune autonomne vožnje? Ovo su neka od gorućih pitanja oko kojih se diže puno prašine ovih dana. Odgovori na ova, ali i druga pitanja, nalaze se u nastavku.

2. Duboko učenje

Da bi bilo uopće moguće objasniti pojam dubokog učenja, moraju biti spomenuta dva osnovna pojma: umjetna inteligencija i strojno učenje. U nastavku je svaki od njih поближе opisan.

2.1. Umjetna inteligencija

Inteligencija (lat. *intelligere* – razabirati, shvaćati, razumijevati) je deskriptivan pojam - opisuje svojstvo neke jedinke ili grupe jedinki, no ne postoji definicija inteligencije oko koje se svi slažu. Slika 2.1 prikazuje 4 područja gdje se svaka od definicija umjetne inteligencije može smjestiti. Većina definicija uključuje koncepte kao što su apstraktno rasuđivanje, razumijevanje, samosvijest, komunikacija, učenje, planiranje i rješavanje problema.

Razmišljati **ljudski**

- *"Uzbudljivi novi pokušaj da se omogući razmišljanje računalima. . . strojevi s umovima, u punom i doslovnom smislu."* (Haugeland, 1985)
- *"Automatizacija aktivnosti koje asociramo s ljudskim razmišljanjem, poput donošenja odluka, rješavanja problema, učenja. . ."* (Bellman, 1978)

Ponašati se **ljudski**

- *"Proces stvaranja strojeva koji obavljaju funkcije koje zahtijevaju inteligenciju koju imaju ljudi"* (Kurzweil, 1990)
- *"Proučavanje kako učiniti da računala rade stvari u kojima su, trenutno, ljudi bolji"* (Rich i Knight, 1991)

Razmišljati **racionalno**

- *"Proučavanje mentalnih svojstava kroz uporabu računalnih modela."* (Charniak i McDermott, 1985)
- *"Proučavanje postupaka koji mogućim čine percipiranje, rasuđivanje i reagiranje."* (Winston, 1992)

Ponašati se **racionalno**

- *"Polje rada koje želi objasniti i emulirati inteligentno ponašanje u smislu računalnih procesa"* (Schalkoff, 1990)
- *"Grana računalnih znanosti koja se bavi automatizacijom inteligentnog ponašanja"* (Luger i Stubblefield, 1993)

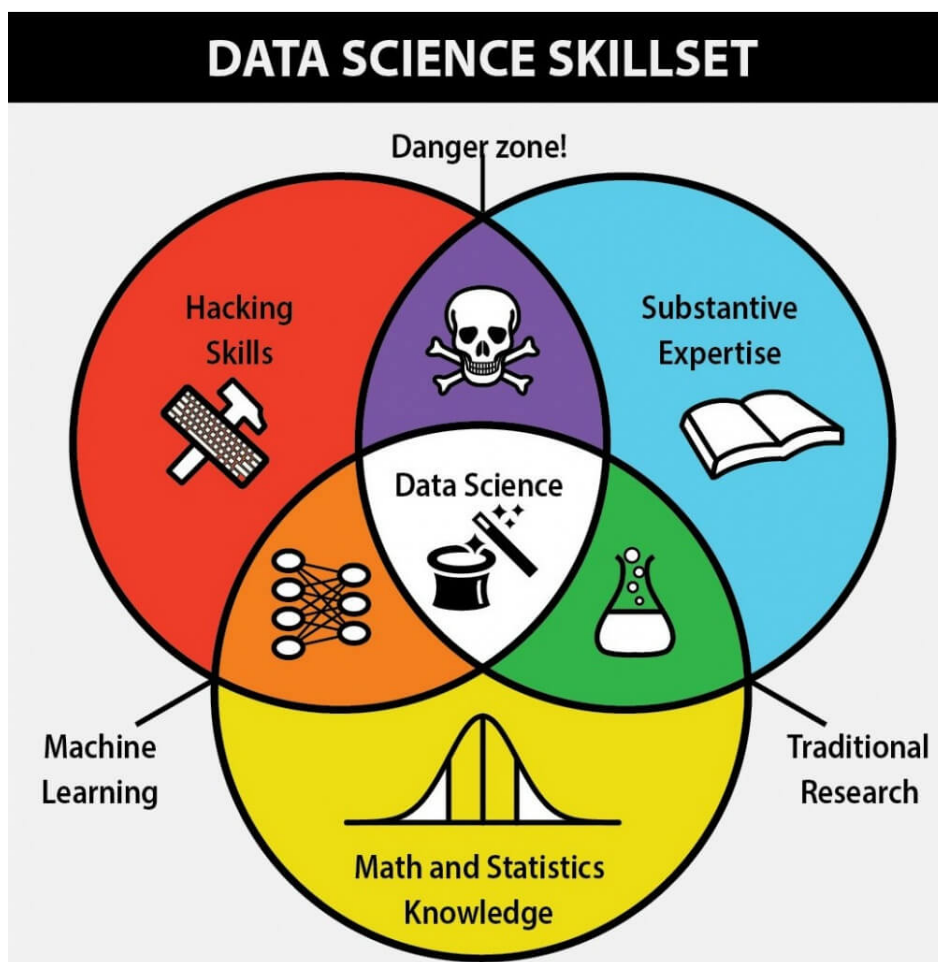
Slika 2.1: Klasifikacija definicija umjetne inteligencije

Marvin Minsky, suosnivač MIT-evog AI laboratorija definirao je umjetnu inteligenciju kao: *"Znanost o tome kako postići da strojevi izvode zadatke koji bi, kada*

bi ih radio čovjek, iziskivali inteligenciju.” Na kraju, D.W. Patterson dao je zasad najtočniju definiciju umjetne inteligencije: “Umjetna inteligencija grana je računarske znanosti koja se bavi proučavanjem i oblikovanjem računarskih sustava koji pokazuju neki oblik inteligencije. Takvi sustavi mogu učiti, mogu donositi zaključke o svijetu koji ih okružuje, oni razumiju prirodni jezik te mogu spoznati i tumačiti složene vizualne scene te obavljati druge vrste vještina za koje se zahtijeva čovjekov tip inteligencije.”

2.2. Strojno učenje

Ethem Alpaydin je u svojoj knjizi, jednoj od najpoznatijih knjiga o strojnom učenju, *"Introduction to Machine Learning"* iz 2009. godine strojno učenje definirao kao "programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva."



Slika 2.2: Vještine koje posjeduje dobar Data Scientist

Ostaje pitanje kada koristiti strojno učenje? Ono se ne koristi za neke "mehaničke" zadatke, već za specifične probleme. Možemo definirati 3 kategorije problema u kojima ima smisla koristiti strojno učenje:

- **Složeni problemi** - ne postoji ljudsko znanje o procesu ili ljudi ne mogu dati objašnjenje o procesu (problemi koje nije moguće riješiti na klasičan algoritamski način - *UI potpuni problemi*)
- **Ogromne količine podataka** - otkivanje znanja u velikim skupovima podataka (engl. *data mining*)
- **Sustavi koji se dinamički mijenjaju** - potrebna prilagodba (npr. prilagodba korisničkih sučelja)

Strojno učenje okosnica je umjetne inteligencije. Inteligentni sustavi trebaju se moći prilagođavati okolini - imati sposobnost učenja.

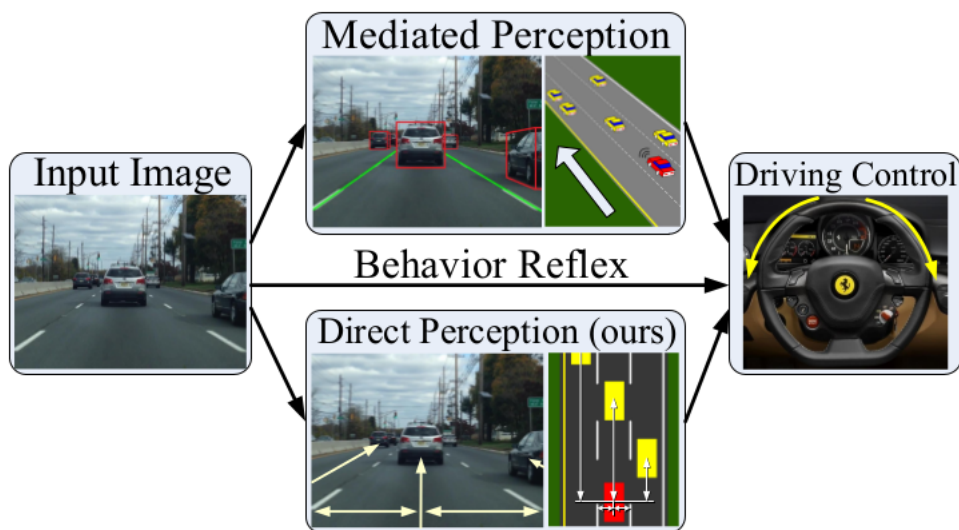
2.3. Duboko učenje

Konačno, sada je moguće definirati duboko učenje. To je grana strojnog učenja koja je posebno prikladna za rješavanje problema iz područja umjetne inteligencije. Duboko učenje se temelji na predstavljanju podataka složenim reprezentacijama do kojih se dolazi slijedom naučenih nelinearnih transformacija. Koriste se slojevite strukture, najčešće neuronske mreže (u kojima gradivni modeli najčešće nisu obični neuroni, o vrsti mreže ovisi od čega će biti sagrađena). Duboke mreže, kako im i samo ime kaže, su mreže koje imaju puno slojeva. To mreži omogućava da vidi "širu sliku". Konceptualno, jedinice ulaznog sloja primaju neke podatke (sliku, zvuk, tekst) i obavljaju nelinearne transformacije nad podacima te ih šalju u sljedeći sloj. Tako niži slojevi imaju utjecaj na ono što viši (apstraktniji) slojevi vide, ovisno o vrsti transformacije nad podacima koju rade. Metode dubokog učenja svoju primjenu pronalaze u izazovnim područjima gdje je dimenzionalnost podataka iznimno velika: računalnom vidu, obradi prirodnog jezika ili razumijevanju govora.

3. Suvremeni pristupi autonomnoj vožnji

Danas poznajemo 2 različita pristupa za koje stručnjaci misle da imaju najviše potencijala u području autonomne vožnje:

1. **Posrednički pristup** (posrednička percepcija) - analizira cijelu 'scenu' (sve što računalo vidi, sve što kamera snimi → parsira sve) i na temelju toga donosi odluke u vožnji
2. **Refleksni pristup** - temeljen na ponašanju, postoji sustav odlučivanja i na temelju ulaznog signala (slike) taj sustav odmah mapira dobiveni ulaz na konkretnu akciju



Slika 3.1: Različiti pristupi autonomnoj vožnji

Slika 3.1 prikazuje kako koji od 3 pristupa donosi odluke te što pri tome uzima u obzir. U zadnjih nekoliko godina dosšlo je do razvitka 3. paradigme: **direktni pristup**.

Najbolje rezultate pokazuje baš **direktni pristup** (direktna percepcija). Ovaj pristup pokušava odmah procijeniti utjecaj pojedine akcije na okolinu. Ovakvo rješenje ovog problema 'pada' direktno između gornja dva pristupa, odnosno uzima najbolje od oba. Autori smatraju da je ovo ispravan stupanj apstrakcije kojim bi se ovo trebalo rješavati.

Većina današnjih sustava autonomne vožnje temeljena je na posredničkom pristupu. Znanstvenici na području znanstvenog vida su istraživali svaki od dva ključna segmenta posebno:

- Detekcija automobila
- Detekcija prometne trake

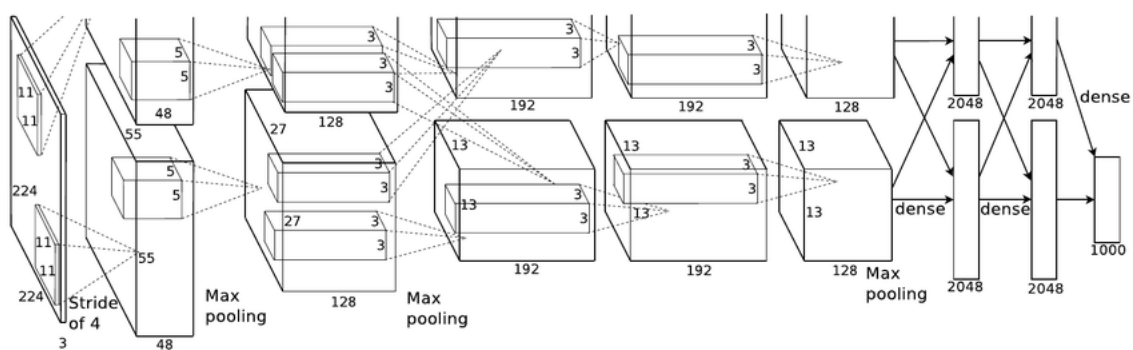
Tipičan algoritam iscrta tzv. *bounding box* oko detektiranih auta na slici te tzv. *splines* na detektirane crte na cesti. No problem je što to nisu direktne informacije koje se koriste u direktnom pristupu. Tipičan model za prepoznavanje prometnih traka kakav predlaže M. Aly u svom radu *Real time detection of lane markers in urban streets*. [3]. Takav model podlozan je malo češćim greškama. Primjerice, vrlo česta greška je prepoznavanje nekih struktura, recimo puknuća asfalta, kao prometne trake (pogotovo iz nekih kutova slike). Ako se naprimjer nalazimo na cesti sa dvije prometne trake, u takvim situacijama je vrlo teško odrediti da li se auto nalazi u lijevoj ili desnoj prometnoj traci.

Primjer ponašajnog modela bio bi *Alvinn* [5]. Alvinn gleda koliko je vozač zakrenuo volan i direktno povezuje ulaznu sliku s tim podatkom u procesu treniranja. Kasnije donosi odluke i okreće volan na temelju naučenoga.

R. Hadsell sa svojim suradnicima u radu [6] trenira višeslojnu konvolucijsku mrežu da odredi sa slike nalazi li se automobil u području kroz koje se može proći ili ne. Za računanje koristi DeepFlow model [4] za procjenu dubine sa slike.

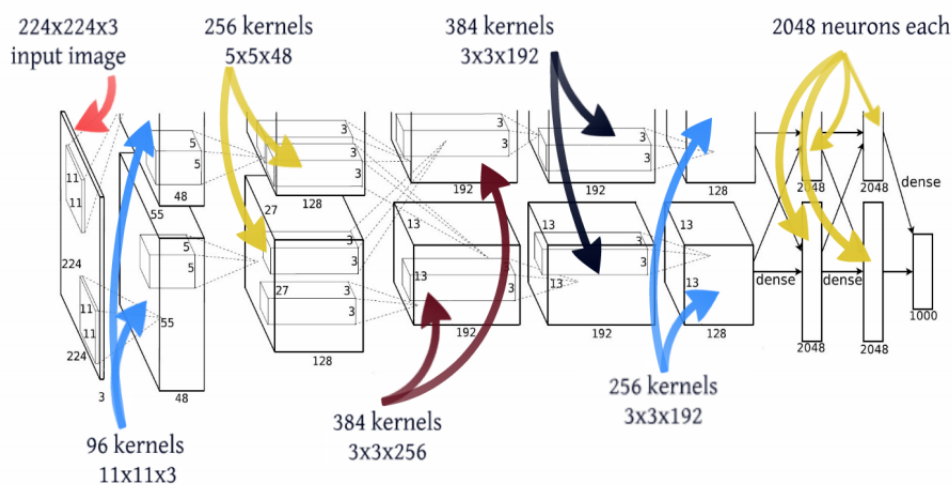
4. Deep Driving

4.1. Arhitektura



Slika 4.1: Arhitektura AlexNet mreže

Slika 4.1 prikazuje arhitekturu konvolucijske mreže tzv. *AlexNet* [2]. Konkretno, mreža je implementirana u programskom okviru *Caffe* i sastoji se od 5 konvolucijskih slojeva (*convolutional layers*) nakon kojih slijede 4 potpuno povezana (*fully connected*) sloja.



Slika 4.2: Broj jezgara (kernela) u AlexNet mreži

Na slici 4.1 prvi tanki kvadratić prikazuje ulaznu sliku. Zatim slijedi 5 konvolu-

cijskih slojeva koji koriste *max pooling*. Prvi sloj, slike su veličine 224*224*3 dok je receptivno polje (engl. *kernel*) konvolucijskog sloja veličine 11*11*3. Korak je 4 (kernel se pomakne za 4 piksela u svakom koraku). Izlaz iz prvog sloja je veličine 55*55*96. Konvolucijski slojevi na slici su prikazani kao kvadri ili kocke. Na kraju, stupci prikazuju 4 potpuno povezana sloja koji imaju veličine izlaza redom: 4096, 4096, 256 i 13. Slika 4.2 prikazuje broj jezgara po pojedinom sloju u AlexNet arhitekturi. Izlaz iz mreže provuće se kroz softmax funkciju: $\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$

Kao funkcija gubitka uzima se Euklidska norma (L2 loss):

$$E = \frac{1}{2N} \sum_{n=1}^N \|y_{input} - y_{target}\|_2^2 \quad (4.1)$$

4.2. Treniranje

Za treniranje je korištena videoigrica TORCS (*The Open Racing Car Simulator*). Inače, ova simulacija je često korištena pri treniranju i evaluaciji AI modela autonomne vožnje. Simulator je idealan za takve stvari jer se mogu prikupljati kritične informacije pri vožnji:

- brzinu automobila
- relativnu poziciju automobila u odnosu na središnju prometnu traku
- udaljenost od vozila koje se nalaze ispred i iza automobila i sl.

Prvi korak u fazi treniranja modela je vožnja "*label collecting*" vozila. Ono prikuplja screenshotove (*first person view* perspektiva) te sve ostale relevantne informacije. Te informacije se spremaju te su korištene za kasnije nadzirano učenje modela.

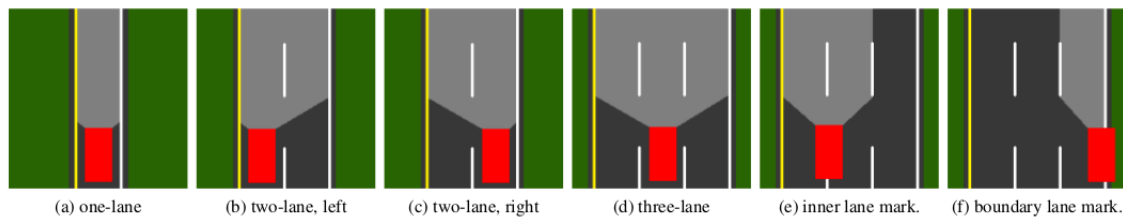


Slika 4.3: TORCS videoigrica korištena pri treniranju modela

U fazi testiranja procjene vrijednosti ovih parametara, u svakom koraku, modelu se daje scena iz vožnje. Model zatim procjenjuje parametre koje šalje *driving control engine*-u koji procjenjuje za koliko treba zakrenuti volan te treba li ubrzati ili kočiti. Zatim se te naredbe šalju simulatoru koji na temelju toga "vozi" automobil. I u ovoj fazi testiranja procjene parametara također se prikupljaju podaci.

4.2.1. Procjena parametara

Ako razmislimo, za modeliranje algoritma autonomne vožnje, ne treba uzimati u obzir sve što postoji na slici. Primjerice, ako cesta ima 4 ili 5 traka, a vozilo u kojem se nalazimo se vozi po vanjskoj lijevoj traci, ne zanima nas što se događa i kakva je situacija u desnoj vanjskoj traci. Dakle, potrebno je selektirati što je bitno i što uzeti u obzir pri modeliranju takvog modela. Dolazi se do zaključka da model treba percipirati samo 3 trake (u najgorem slučaju). Postoji nekoliko situacija u kojima se možemo naći:



Slika 4.4: Situacije u kojima se vozilo može naći [1]

Nadalje, iz perspektive modela, vožnja može biti podijeljena na dvije kategorije:

- vožnja u traci
- promjena trake ili usporavanje radi izbjegavanja sudara (usporava se u slučaju da se vozilo koje se nalazi ispred ne može zaobići jer su trake zauzete)

Ovdje se predlažu dva koordinatna sustava:

- "in lane system"
- "on marking system"

Kako bi se mogle ostvariti dvije glavne funkcije, percepcija trake i percepcija vozila, predložena su 3 tipa pokazatelja:

- ulazni kut
- udaljenost do okolnih oznaka traka
- udaljenost od vozila ispred

Ukupno je predloženo 13 indikatora (faktora) relevantnih za ovaj model koji se nalaze na slici 4.5. Ovime se vozilo točno pozicionira pri vožnji.

always:

1) angle: angle between the car's heading and the tangent of the road

"in lane system", when driving in the lane:

2) toMarking_LL: distance to the left lane marking of the left lane

3) toMarking_ML: distance to the left lane marking of the current lane

4) toMarking_MR: distance to the right lane marking of the current lane

5) toMarking_RR: distance to the right lane marking of the right lane

6) dist_LL: distance to the preceding car in the left lane

7) dist_MM: distance to the preceding car in the current lane

8) dist_RR: distance to the preceding car in the right lane

"on marking system", when driving on the lane marking:

9) toMarking_L: distance to the left lane marking

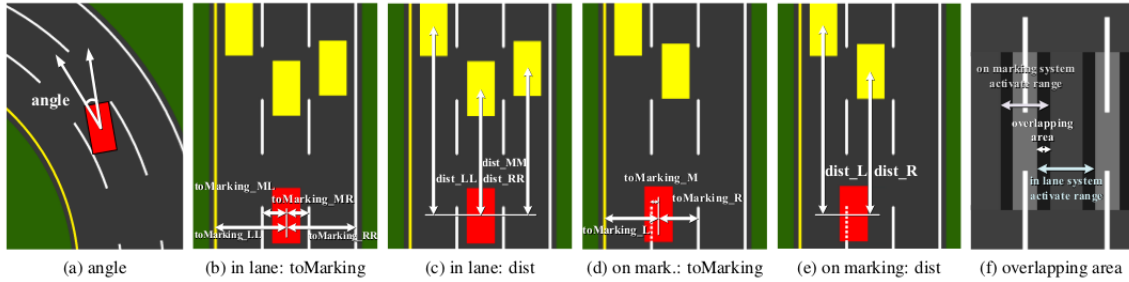
10) toMarking_M: distance to the central lane marking

11) toMarking_R: distance to the right lane marking

12) dist_L: distance to the preceding car in the left lane

13) dist_R: distance to the preceding car in the right lane

Slika 4.5: Svih 13 indikatora korištenih u direktnom pristupu [1]



Slika 4.6: Promjena trake u "in lane system" i "on marking system" načinu rada [1]

4.2.2. Mapiranje procjene parametara na konkretne akcije

Okretaj volana izračunava se iz pozicije vozila. Cilj je minimizirati udaljenost vozila od sredine trenutne prometne trake. Formula po kojoj se to izračunava je:

$$steerCmd = C * \frac{angle - distCenter}{roadWidth} \quad (4.2)$$

gdje je C faktor koji ovisino uvjetima vožnje, a $angle \in [-\pi, \pi]$.

Slika 4.7 prikazuje pseudokod kontrolera, sustava koji upravlja vožnjom automobila.

```

while (in autonomous driving mode)
    ConvNet outputs affordance indicators
    check availability of both the left and right lanes
    if (approaching the preceding car in the same lane)
        if (left lane exists and available and lane changing allowable)
            left lane changing decision made
        else if (right lane exists and available and lane changing allowable)
            right lane changing decision made
        else
            slow down decision made
    if (normal driving)
        center_line = center line of current lane
    else if (left/right lane changing)
        center_line = center line of objective lane
    compute steering command
    compute desired_speed
    compute acceleration/brake command based on desired_speed

```

Slika 4.7: Pseudokod kontrolera [1]

U svakom koraku sustav izračunava *desired_speed*. Kontroler zatim podešava vozilo na željenu brzinu kontrolirajući gas/kočnicu. U zavojima se brzina smanjuje, pogotovo ako se ispred nalazi vozilo. Da bi ovo bilo zadovoljeno, brzina računa po formuli:

$$v(t) = v_{max}(1 - e^{-\frac{c}{v_{max}}dist(t)-d}) \quad (4.3)$$

gdje je $dist(t)$ udaljenost od vozila ispred, v_{max} maksimalna dopušten brzina, a c i d koeficijenti koje je potrebno podesiti.

4.2.3. Proces treniranja

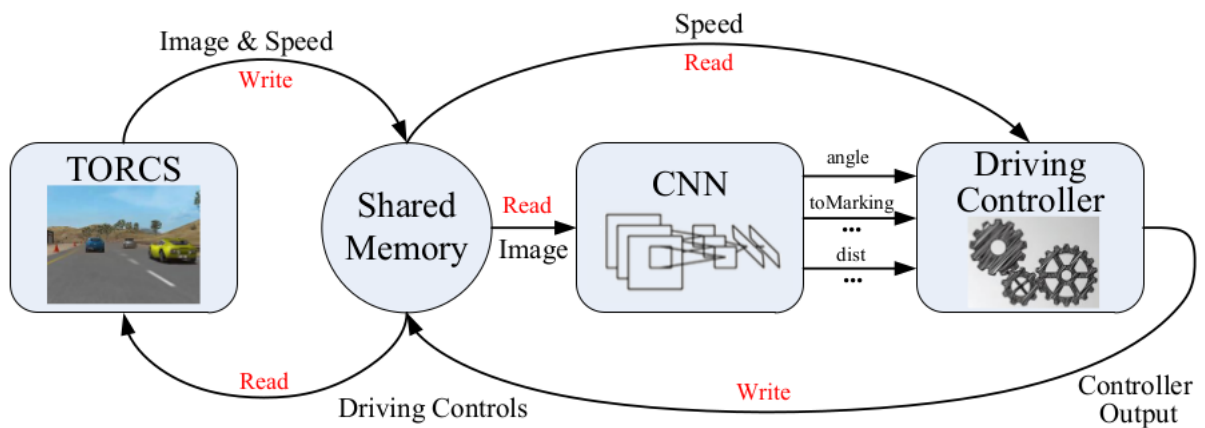
U simulatoru TORCS odabrano je 7 traka i 22 vozila za generiranje skupa za učenje. Teksture ceste su za potrebe treniranja zamijenjene sa preko 30 prilagođenih tekstura asfalta sa različitim kombinacijama u varijacijama traka te nijansi asfalta. Također su promijenjeni i obrasci vožnje ostalih automobila da bi se prikupili podaci za različite tipove vožnje. Automobile u simulatoru su zatim više puta vozili ljudi kako bi prikupili podatke za učenje modela. Pri vožnji automatski su slikani screenshotovi iz *first person view* perspektive. Screenshotovi su zatim smanjeni na rezoluciju 280×210 te sa ostalim prikupljenim podacima spremeni u bazu.

Ukupno je prikupljeno 484,815 slika za treniranje mreže. Zadana je stopa učenja 0.01 i veličina *mini-batcha* je 64 nasumično odabrane slike iz skupa za treniranje. Nakon 140,000 iteracija, proces treniranja je zaustavljen.

4.3. Testiranje

Pri testiranju, podaci koje model dobije su:

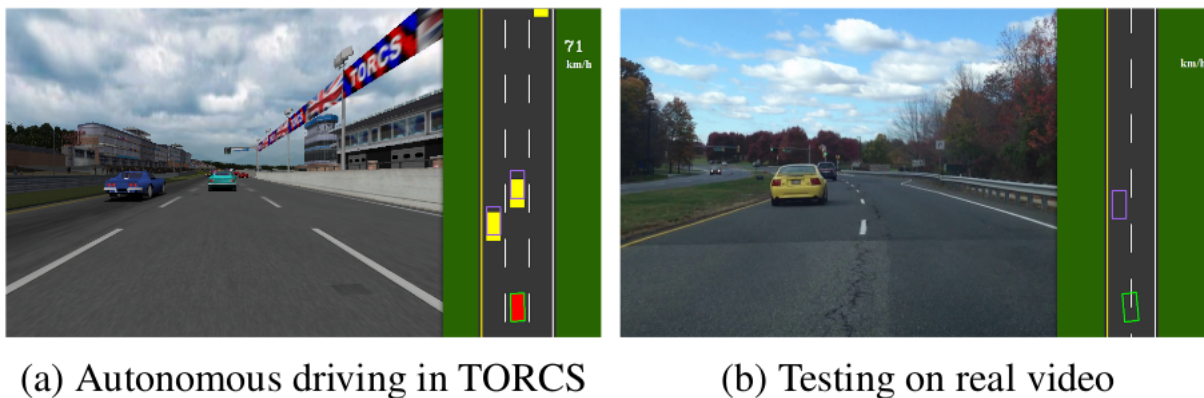
- sliku koju ima vozač (iz automobila)
- brzinu automobila



Slika 4.8: Arhitektura sustava

Na slici 4.8, [1] vidljiv je proces kako sustav procesira sliku, odluči koju akciju napraviti i zatim ju provede. Prvo konvolucijska mreža obradi sliku i procijeni 13 parametara koje, zajedno sa podatkom o brzini, prosljedi kontroleru. Kontroler prema algoritmu sa slike 4.7 izračunava upravljačke naredbe koje zatim ponovo vraća simulatoru koji ih provede.

5. Evaluacija



Slika 5.1: Evaluacija modela - praznim kvadratićima označena je procjena, a punim kvadratićima stvarna pozicija vozila. Na stvarnoj slici iscrtani su samo kvadratići jer ne znamo točne pozicije.

5.1. Kvantitativna evaluacija

Ovaj model funkcionira vrlo dobro u simulaciji TORCS. Percepcija automobila seže do 30 m. Auti udaljeni između 30 i 60 m od vozila su označena kao male točkice te se ne može pouzdano reći je li to automobil ili neki objekt iz okruženja. No čim taj objekt uđe u doseg 30 m, model ga prepoznaje.

Kako je automobil sredstvo osjetljivo na nagle promjene (npr. smijera, brzine), važno je ostvariti toleranciju na manje greške u procjeni parametara. Parametri se procijanjuju puno puta u sekundi, te nebi bilo dobro kada bi se za svaku malu promjenu u parametrima, vozilo pomaknulo razmjerno promjeni.

5.2. Evaluacija u usporedbi s drugim modelima i pristupima

5.2.1. Ponašajni pristup

Ponašajni model direktno mapira ulaz na konkretnu akciju (zakret volana). Pri treniranju modela Deep Driving korištena su dva tipa treniranja:

- 60000 slika sa vožnje na praznoj traci
- 80000 slika sa vožnje na traci gdje se nalaze i druga vozila

Ponašajni model se dobro snalazi na praznoj traci, nema problema u praćenju trake, no problemi se javljaju u drugom slučaju kada model nije sam na cesti. Model donekle pokazuje sposobnost izbjegavanja sudara skretanjem lijevo ili desno, no putanja je nejasna, i javlja se dosta grešaka. Model je nepredvidiv i kao tak ne predstavlja idealno rješenje problema autonomne vožnje.

5.2.2. Posrednički pristup - detekcija traka

Za ovu usporedbu korišten je model [3]. Pošto je taj model relativno slab korišteni su manji skupovi za trening i test fazu. Slike za te skupove odabrane su sa iste trake i bolje su rezolucije 640×480 . I sa ovakvim olakotnim okolnostima, ovaj model je i dalje slabiji od Deep Driving modela.

5.3. Evaluacija na stvarnim podacima

5.3.1. Video snimljen pametnim telefonom

Slika 5.1.b prikazuje ovu situaciju. Iako je model treniran na drugoj domeni, ovdje pokazuje izuzetno dobre rezultate. Posebno dobro radi modul percepcije trake. Model je sposoban jako dobro prepoznati traku, lokalizirati vozilo na traci te prepoznati situacije promjene trake. Modul prepoznavanja automobila je nešto šumovitiji, no to možemo pripisati učenju na drugačijoj domeni pošto auti u simulatoru izgledaju dosta drugačije nego u stvarnosti.

5.3.2. Procjena udaljenosti vozila na KITTI datasetu

KITTI dataset sadrži preko 40,000 slika snimljenih u vožnji europskim gradovima. U većini slika iz ovog dataseta nema oznaka traka pa ne možemo lokalizirati auto na temelju trake u kojoj se auto nalazi.

U KITTI datasetu na jednoj slici se može nalaziti mnogo automobila, no samo oni najbliži sudjeluju u donošenju odluka. Zato je prostor ispred auta podijeljen na 3 dijela:

1. Središnje područje, $x \in [-1.6, 1.6]$ metara \rightarrow vozila direktno ispred automobila
2. Lijevo područje, $x \in [-12, -1.6)$ metara \rightarrow gdje su vozila na lijevoj strani automobila
3. Desno područje, $x \in [1.6, 12)$ metara \rightarrow gdje su vozila na desnoj strani automobila

Auti koji su izvan ovog raspona ne uzimaju se u obzir (jer smo ih vjerojatno već prošli). Valja uzeti u obzir i procjenu udaljenosti. Obzirom da su slike u KITTI datasetu malo niže rezolucije, ponovo dijelimo prostor na 2 dijela:

1. Kratki domet, $x \in [2, 25]$ metara, slike veličine 497×150 piksela
2. Daleki domet, $x \in [15, 55]$ metara, slike veličine 497×150 piksela

Krajnja procjena udaljenosti je kombinacija ove dvije procjene.

6. Zaključak

Deep Drive model je model direktnog pristupa koji se pokazao kao izuzetno dobro rješenje problema autonomne vožnje. Iako treniran na simulaciji (TORCS), jako se dobro snalazi i u stvarnom okruženju. Zbog izmjena u simulatoru (teksture i nijansi asfalta, pukotine na cesti i sl.) model vrlo dobro generalizira i u ne-virtualnom svijetu. Važno je ipak naglasiti da je model treniran na simulatoru morao proći krot par manjih izmjena zbog drugačijih uvijeta u kojima mora odlučivati - ulazni podaci (slike) podijeljene su na nekoliko dijelova radi lakšeg i jednostavnijeg računa te veće pouzdanosti (točnosti). U konačnici, direktni pristup, paradigma u kojoj se parsiraju i uzimaju u obzir samo neke ulazne značajke, zasada daje nabolje rezultate u ovom području i mišljenja sam da bi daljnja istraživanja trebala ići u tom smjeru.

7. Literatura

- [1] Deepdriving: Learning affordance for direct perception in autonomous driving. URL <http://deepdriving.cs.princeton.edu>.
- [2] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. URL http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf.
- [3] Mohamed Aly. Real time detection of lane markers in urban streets. URL <http://www.vision.caltech.edu/malaa/publications/aly08realtime.pdf>.
- [4] Z. Harchaoui P. Weinzaepfel, J. Revaud i C. Schmid. Deepflow: Large displacement optical flow with deep matching. URL https://www.robots.ox.ac.uk/~vgg/rg/papers/DeepFlow_iccv2013.pdf.
- [5] D. A. Pomerleau. Alvin: An autonomous land vehicle in a neural network. URL <http://repository.cmu.edu/cgi/viewcontent.cgi?article=2874&context=compsci>.
- [6] J. Ben A. Erkan M. Scoffier K. Kavukcuoglu U. Muller R. Hadsell, P. Sermanet i Y. LeCun. Learning long-range vision for autonomous off-road driving. URL http://www.cs.nyu.edu/~sermanet/lagr/papers/hadsell-jfr-09_wiley.pdf.

8. Sažetak

U ovom radu predložena je nova paradigma, novi pristup autonomnoj vožnji. Ovakav pristup nazvan je **direktni pristup**. To je potpuno drugačiji način razmišljanja od dosadašnjih:

- **Ponašajni pristup** - direktno mapira podatke sa ulazne slike na konkretnu akciju (u ovom slučaju zakret volana ili promjena brzine)
- **Posrednički pristup** - parsiranje cijele slike

Deep Drive model je model direktnog pristupa, izgrađen pomoću konvolucijske mreže arhitekture AlexNet, koji umjesto gornjih metoda, računa određene attribute, konkretno njih 13, te na temelju tih podataka i podataka o brzini vožnje odlučuje o akcijama koje poduzima. Iako je model treniran u virtualnom okruženju (TORCS simulator), ispitivanjem se pokazalo da, uz male promjene u procesiranju slike (dijeljenju na područja), izuzetno dobro generalizira te je podoban i sposoban djelovati i u stvarnom svijetu.