

Videoigra upravljana pokretima tijela

Bradarić, Jozo

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:166:852344>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-24**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
ODJEL ZA INFORMATIKU

VIDEOIGRA UPRAVLJANA POKRETIMA TIJELA

Jozo Bradarić

Split, rujan 2023.

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
ODJEL ZA INFORMATIKU

VIDEOIGRA UPRAVLJANA POKRETIMA TIJELA

Jozo Bradarić

Mentor:

Split, rujan 2023.

Temeljna dokumentacijska kartica

Završni rad

Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za informatiku

Ruđera Boškovića 33, 21000 Split, Hrvatska

VIDEOIGRA UPRAVLJANA POKRETIMA TIJELA

Jozo Bradarić

SAŽETAK

Umjetna inteligencija i gesture gaming su još uvijek novina u svijetu računalnih igara. Temelj upravljanja videoigrama pokretima tijela je umjetna inteligencija odnosno preciznije konvolucijske neuronske mreže. Brojne su biblioteke poput tensorflowa koje korisnicima približavaju i olakšavaju korištenje konvolucijskih i ostalih neuronskih mreža. Zajedno s ostalim alatima za izradu web aplikacija mogućnosti u polju gesture gaminga su gotovo pa neograničene.

Ključne riječi: Umjetna inteligencija, gesture gaming, konvolucijske neuronske mreže, tensorflow.

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 22 stranice, 13 slika, 0 tablica i 8 literaturnih navoda.

Izvornik je na hrvatskom jeziku.

Mentor: prof. dr. sc. Saša Mladenović

Ocjenjivači: prof. dr. sc. Saša Mladenović, Dino Nejašmić, predavač i Ivana Marin, asistent

Rad prihvaćen:

Basic documentation card

Thesis

University of Split

Faculty of Science

Department of Computer Science

Ruđera Boškovića 33, 21000 Split, Croatia

MOTION GESTURE VIDEOGAME

Jozo Bradarić

ABSTRACT

Artificial intelligence and gesture gaming are still new in the world of computer games. The basis of controlling body movements in video games is artificial intelligence, or more precisely, convolutional neural networks. There are numerous libraries such as tensorflow that bring users closer and facilitate the use of convolutional and other neural networks. Together with other tools for creating web applications, the possibilities in the field of gesture gaming are almost limitless.

Key words: Artificial intelligence, gesture gaming, convolutional neural networks, tensorflow.

Thesis deposited in library of Faculty of Science, University of Split.

Thesis consist of: 22 pages, 13 figures, 0 tables and 8 references

Original language: Croatian

Supervisor: prof. Ph.D. Saša Mladenović

Reviewers: prof. Ph.D. Saša Mladenović, Dino Nejašmić, lecturer and Ivana Marin, assistant

Thesis accepted:

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam završni rad s naslovom VIDEOIGRA UPRAVLJANA POKRETIMA TIJELA izradio/la samostalno pod voditeljstvom prof. dr. sc. Saše Mladenovića, U radu sam primijenio/la metodologiju znanstvenoistraživačkog rada i koristio/la literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo/la u diplomskom radu na uobičajen, standardan način citirao/la sam i povezo/la s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Student

Jozo Bradarić
potpis

Sadržaj

1. UVOD.....	1
2. OPĆENITO O UMJETNOJ INTELIGENCIJI	1
3. VEZA UI I BIOLOŠKIH SUSTAVA.....	2
3.1 Konvolucijske neuronske mreže	4
3.2 Perceptron	7
4. UI I RAČUNALNE IGRE.....	8
4.1 Videoigre upravljane pokretom.....	11
5. ODABIR METODA, TEHNIKA I ALATA	14
5.1 Umjetna inteligencija u projektu.....	16
5.2 Princip rada aplikacije.....	17
5.3 Moguća unaprjeđenja prototipa	19
6. ZAKLJUČAK.....	20
7. LITERATURA.....	21
8. POPIS SLIKA.....	22

1. UVOD

Cilj ovog projekta je pokazati osnovne tehnike, metode i alate koji se koriste u izradi platformskih 2d videoigara, kao i proći kroz osnovne koncepte umjetne inteligencije od kojih se mnogi koriste u projektnom prototipu. Umjetna inteligencija je iz dana u dan sve prisutnija u našoj svakodnevnici, stoga smatram da je tema jako poučna i korisna za sve koji žele nešto više naučiti o osnovama umjetne inteligencije.

2. OPĆENITO O UMJETNOJ INTELIGENCIJI

Umjetna inteligencija pojam je koji se naveliko spominje u posljednjim desetljećima, a posebice u zadnjih desetak godina. Postoje brojne definicije o tome što bi umjetna inteligencija podrazumijeva te će ovdje biti navedene samo neke. Umjetna se inteligencija (UI) smatra sastavnicom Četvrte industrijske revolucije, u koju još pripadaju Robotika, Nanotehnologija, Internet stvari, Autonomna vozila, Kvantna računala i 3D tisak. Temeljna svrha UI jest osposobljavanje računala da obavljaju zadatke za koje je potrebna inteligencija, npr. snalaženje, samostalno učenje novih koncepata, zaključivanje, razumijevanje jezika, raspoznavanje prizora i dr.[1]

UI djeluje koristeći softver i hardver, u digitalnoj ili fizičkoj okolini, prikupljanjem i obradom podataka kao i pronalaženjem optimalnog rješenja za izvođenje određene radnje. Glavni pristupi i tehnike UI su strojno učenje, strojno zaključivanje i robotika.[2] S. Russel i P. Norvig svrstali su UI u nekoliko kategorija: Sustavi koji misle kao čovjek, sustavi koji se ponašaju kao čovjek, sustavi koji misle racionalno, sustavi koji se ponašaju racionalno, sustavi koji imaju cilj da imaju sve izgleda inteligencije (racionalne ili ljudske), sustavi čije unutarnje funkcioniranje pokušava biti u skladu s ljudskim bićem, odnosno racionalnim bićem.[3]

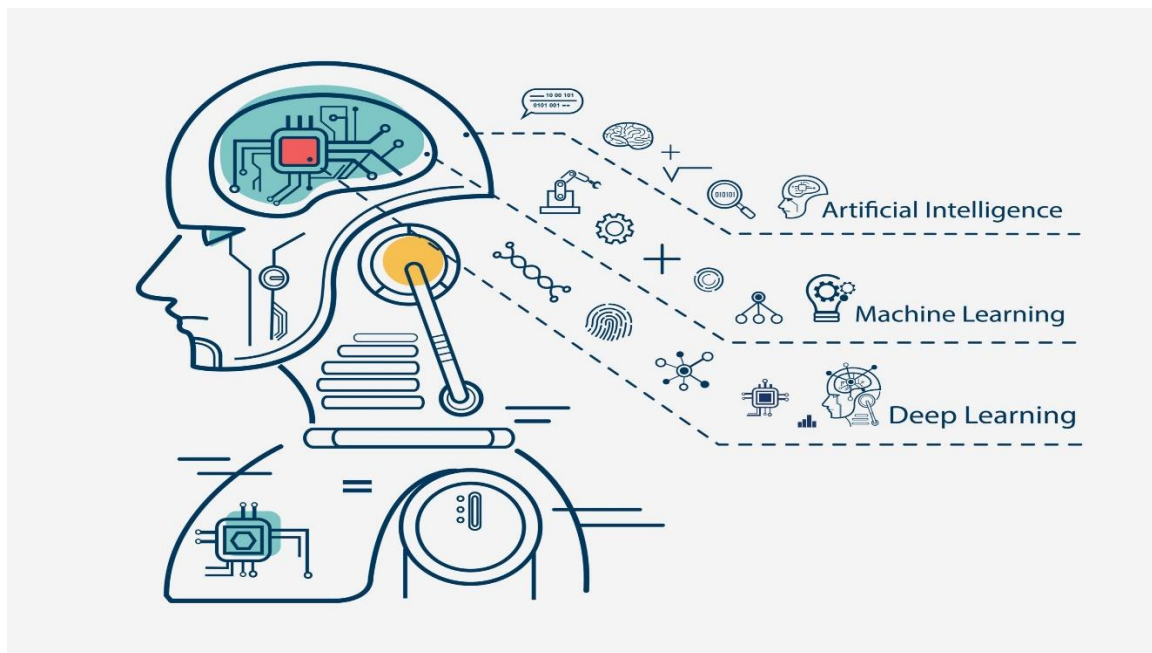
Glavna je primjena UI u računalnim igrama i simulacijama, a najpoznatiji primjer je šahovski sustav. Također, primjenjuje se u ekspertnim sustavima, neuronskim mrežama, obradi jezika itd. Temeljna podjela UI je ona na jaki i slabu. Jaka je razvijena do te mjere da oponaša ljudsku inteligenciju, dok se slaboj mogu pripisati tek poneka, jednostavnija inteligentna svojstva, primjerice mogućnost raspoznavanja govora.[4]

Iako su od ljudi od davnina zaokupljeni mišlju i različitim oblicima UI, prvi spomena vrijedni radovi iz ovog područja pojavljuju se tek od druge polovice 20. stoljeća. Prvi je radove na ovom području radove napisao A. M. Turing, a prvi računalni program UI naziva se The Logic Theorist. Izraz UI prvi je put upotrijebljen na kongresu na Dartmouth Collegeu 1956. Organizator tog kongresa bio je John McCarthy, autor osnovnoga programskog jezika umjetne inteligencije LISP-a (1958.), što ga čini začetnikom umjetne inteligencije. Sedamdesetih godina prošlog stoljeća razvija se programski jezik PROLOG i MYCIN koji je uspijevaio dijagnosticirati krvne infekcije i čak propisivati liječenje. Zbog određenim manjkavosti tog sustava umreženo je više tisuća procesora, što je ostvareno u projektu Connection Machine. Ovo je računalo ujedno i prvi primjer umjetne neuronske mreže.[4] U posljednje se vrijeme istraživači zalažu oko izgradnje tzv. neuromorfnu računalnu arhitekturu. Znanstvenici koji se time bave nazivaju se neuromorfnim inženjerima, a glavni im je cilj izraditi računala koja nalikuju mozgu, čime bi se istovremeno poboljšalo razumijevanje rada mozga te se dobila bolja i inteligentnija računala.[4]

3. VEZA UI I BIOLOŠKIH SUSTAVA

Jedna od važnih grana UI je svakako i strojno učenje. Prilikom strojnog učenja dolazi do oblikovanja algoritama koji na temelju empirijskih podataka poboljšavaju svoju učinkovitost. Važna novina u strojnom učenju u odnosu na dosadašnji način učenja strojeva je prikupljanje znanja na temelju prošlih iskustava. Najpoznatiji primjer strojnog učenja su umjetne neuronske mreže. Neuronske su mreže sastavljene od dva dijela, izdvajanja značajki i zaključivanja. Inspiracija za nastanak umjetnih neuronskih mreža je ljudski mozak. I samo funkcioniranje umjetnih neuronskih mreža ustrojeno je po uzoru na ljudski mozak: jednako kao i u mozgu neuronski se signali šalju s jednog kraja na drugi. Postoje različite vrste neuronskih mreža, uključujući višeslojni perceptron, radijalnu baznu mrežu, LSTM, rekurentne neuronske mreže,

generativne kontradiktorne mreže i konvolucijske neuronske mreže. Struktura svih neuronskih mreža se svodi na ulaz, skriveni sloj i izlaz. Pojednostavljeno rečeno, umjetne neuronske mreže uče, odnosno treniraju se na primjerima. Neuronske mreže razlikuju se po arhitekturi, načinu učenja i signalima. Prema arhitekturi, dijele se na mreže u kojima veze idu isključivo prema sljedećem sloju odnosno *feedforward* i na povratne mreže koje se zovu još i *backpropagation*. Postoji više različitih principa po kojima se neuronske mreže uče a dva glavna načina su učenje s učiteljem i učenje bez učitelja. Sustavi strojnog učenja inspirirani su biološkim sustavima na razne načine.



Slika 1 Umjetna inteligencija

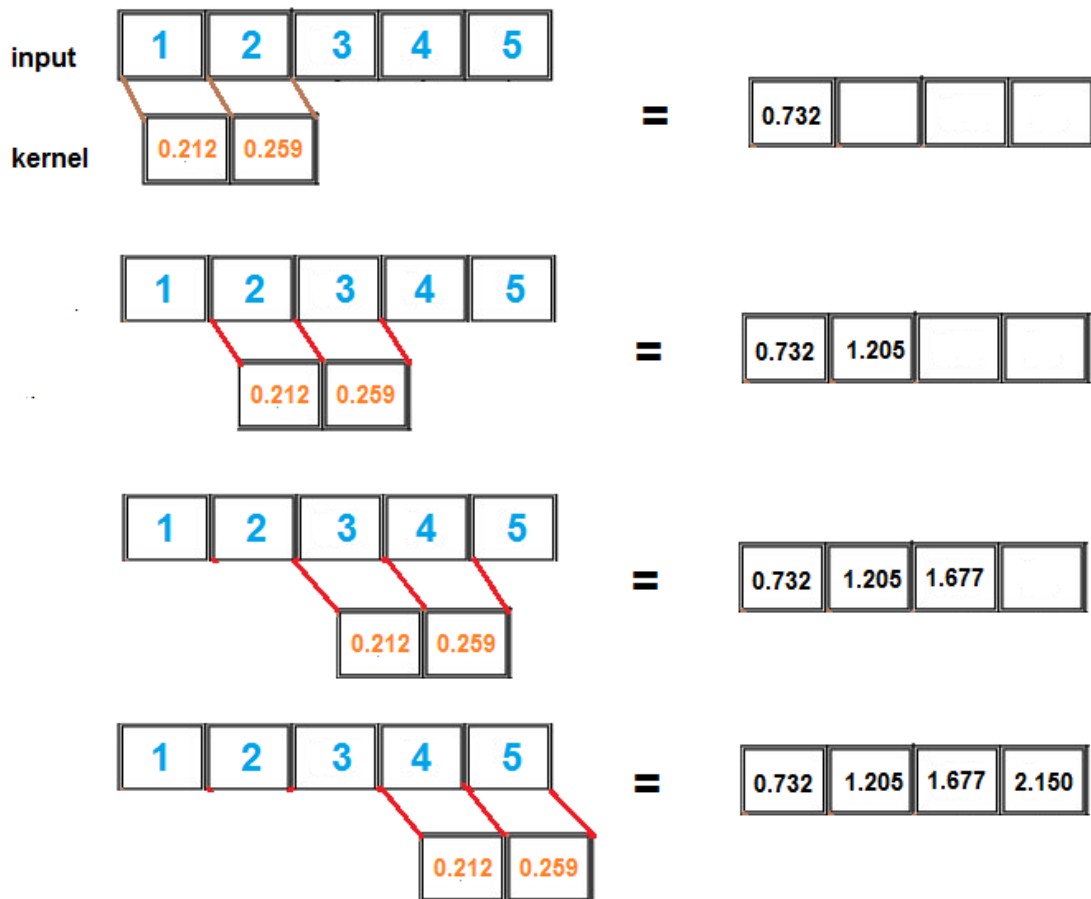
Kako su biološki sustavi evoluirali da obrađuju samo najrelevantnije informacije, tako i algoritmi strojnog učenja odabiru najvažnije značajke za određeni zadatak. Utjecaj bioloških sustava na strojno učenje vidljiv je i u optimizaciji. Biološki su sustavi optimizirani za svoje izvedbe pa su u skladu s tim algoritmi strojnog učenja podešeni da poboljšaju vlastitu izvedbu. Ostali uvidi bioloških sustava koji su uzor algoritmima strojnog učenja uključuju i robusnost, učenje iz malih podataka i otkrivanje novosti. Neuronske mreže se koriste za učenje kako prepoznati obrasce u podacima koji se odnose na različite klase. Jednom kada se mreža nauči prepoznavati te obrasce, može se koristiti za klasificiranje novih podataka u odgovarajuće klase. Osim toga, neuronske mreže mogu se koristiti za klasificiranje u drugim područjima, kao što je prepoznavanje govora, prepoznavanje teksta i prepoznavanje emocija na temelju govora ili slike. U svakom slučaju, neuronske mreže se koriste za učenje kako prepoznati

obrasce u podacima koji se odnose na različite klase, a zatim se koriste za klasificiranje novih podataka u odgovarajuće klase.

3.1 Konvolucijske neuronske mreže

U matematici (osobito, funkcionalnoj analizi), konvolucija je matematička operacija na dvije funkcije (f i g) koja proizvodi treću funkciju ($f * g$) koji izražava način na koji oblik jednog mijenja drugi. Pojam konvolucija odnosi se i na funkciju rezultata i na proces njezina izračunavanja. Definiira se kao integral umnoška dviju funkcija nakon što se jedna odrazi oko y -osi i pomakne. Odabir funkcije koja se odražava i pomiče prije integrala ne mijenja rezultat integrala (vidi komutativnost). Integral se procjenjuje za sve vrijednosti pomaka, stvarajući konvolucijsku funkciju.[6]

Da bismo lakše razumjeli kako se konvolucija koristi za prepoznavanje značajki u konvolucijskim neuronskim mrežama možemo se poslužiti ilustracijom koristeći dva jednostavna niza brojeva.



Slika 2 Konvolucija nizova

Kao početne vrijednosti ove operacije imamo dva niza: input (i) i filter (f). Filter se još naziva i kernel. Ovaj primjer možemo zamisliti kao for petlju s indeksima od 0 do 4. Neka je rezultat operacije niz r duljine za 1 manje od input niza. Ako želimo izračunati r_n u ovom primjeru će on glasiti $r_n = i_n * f_n + i_{n+1} * f_{n+1}$

Ako proširimo istu logiku isto možemo raditi i kod dvodimenzionalnih nizova (matrica).

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

*

1	0	-1
1	0	-1
1	0	-1

=

6		

$$\begin{aligned}
 &7 \times 1 + 4 \times 1 + 3 \times 1 + \\
 &2 \times 0 + 5 \times 0 + 3 \times 0 + \\
 &3 \times -1 + 3 \times -1 + 2 \times -1 \\
 &= 6
 \end{aligned}$$

Slika 3 Konvolucija matrica

Pretpostavimo da je ulazna matrica crnobijela slika gdje pozitivni brojevi predstavljaju veću razinu bijele boje, a negativni predstavljaju veću razinu crne boje. Nije teško primijetiti da će nakon konvolucije s ovim filterom najveće vrijednosti u mapi podudaranje biti ondje gdje su bijeli lijevi rubovi i crni desni rubovi.

Primijetimo kako mijenjanjem težinskih faktora i njihovim rasporedom u filteru možemo prepoznavati različite oblike i uzorke npr. ako želimo dobiti mapu podudarnost za gornje rubove na slici, filter postavimo tako da su pozitivni elementi u prvom redu matrice, a svi ostali su nula.

Kao što je već navedeno, jedan od poznatijih primjera umjetnih neuronskih mreža su konvolucijske. Funkcioniraju po uzoru na vizualni korteks koji u mozgu obrađuje vizualne informacije. Za njih je karakteristična uporaba konvolucijskih slojeva koji primjenjuju filtriranje na ulazne podatke za izdvajanje značajki. Ovi slojevi koriste konvolucijske jezgre koje se kreću duž ulaznih podataka i izračunavaju konvoluciju između jezgre i podataka. Nakon toga se primjenjuje aktivacijska funkcija za generiranje izlaza. Svoju su primjenu ove mreže pronašle u prepoznavanju objekata, detekciji i klasifikaciji slika te detekciji lica. Ono što ih čini učinkovitim u obradi vizualnih informacija jest njihova sposobnost učenja i izdvajanja značajki iz podataka. Dodatna specifičnost koju posjeduju je prepoznavanje objekata bez obzira na njihov položaj ili orijentaciju na slici (sposobnost učenja prostornih invarijanti). Sve

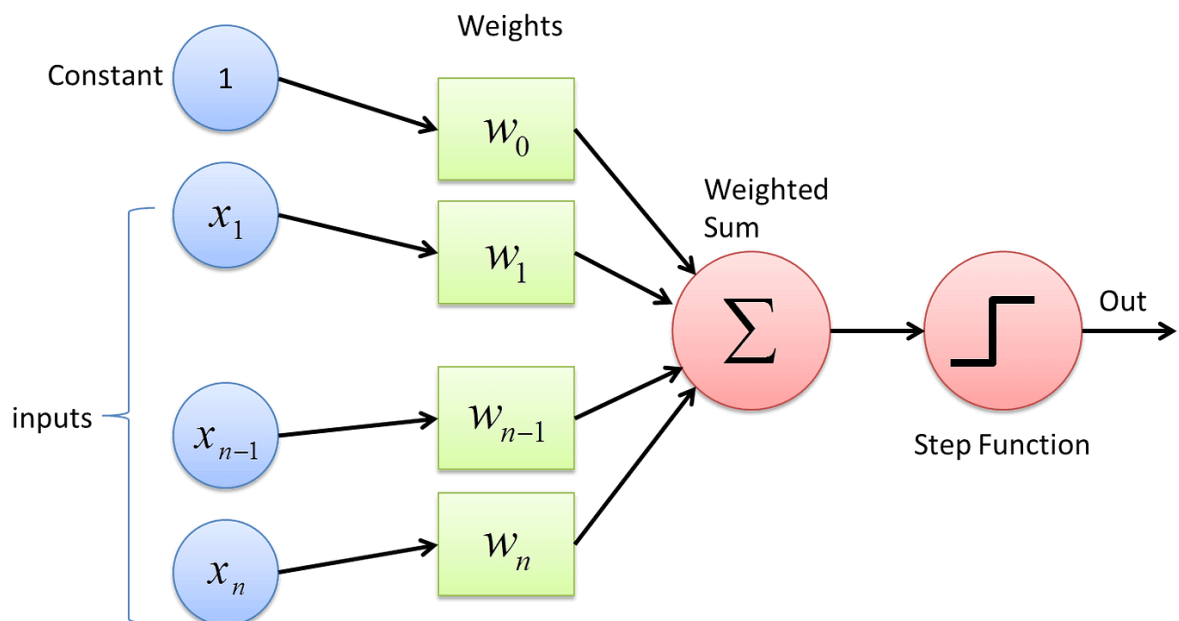
navedeno čini ih ključnim alatom u području dubokog učenja. Zbog svojih karakteristika često se koriste u kombinaciji s drugim slojevima kao što su udruživanje i potpuno povezani slojevi kako bi se postigla bolja izvedba u različitim zadacima. Ove su mreže posebno dizajnirane za obradu podataka koji imaju prostornu strukturu, kao što su slike. Zadatci klasifikacije u kojima se ova vrsta mreža primjenjuje uključuju klasifikaciju dobi osobe na temelju slike lica, klasifikaciju objekata na temelju slika, kao što je klasifikacija dermatoskopskih slika, klasifikaciju različitih vrsta signala, kao što su zvukovi ili elektroencefalogrami (EEG), klasifikaciju darovitosti učenika na temelju različitih varijabli i sl.

Konvolucijska neuronska mreža sastoji se od ulaza, konvolucijskih slojeva, slojeva sažimanja, jednodimenzionalnih potpuno povezanih slojeva i izlaza.[2] Informacije prolaze kroz neuronsku mrežu stvaraju se vrijednosti koje se zatim kompariraju sa vrijednošću u stvarnosti. Razlika između stvarne i izračunate vrijednosti služi za korekciju težinskih faktora. Veličina pogreške pokazatelj je kvalitete mreže. Kako bi se spriječilo pretreniranje mreže, učenje treba zaustaviti kada pogreška provjere počne rasti.[2]

3.2 Perceptron

Jedna od poznatijih vrsta umjetne neuronske mreže je perceptron. U literaturi se još naziva i McCulloch-Pitts neuron po njegovim stvoriteljima Warrenu McCullochu i Walteru Pittsu. Izumljen je 1943. a svoju je prvu primjenu pronašao 1958. godine u stroju američkog psihologa Franka Rosenblatta, kojeg mnogi nazivaju ocem dubokog učenja zbog njegovog revolucionarnog rada vezanog uz umjetnu inteligenciju. Ujedno je i najjednostavnija moguća neuronska mreža i služi kao gradivni blok složenijih neuronskih mreža koje se danas upotrebljavaju prilikom dubokog učenja. Sastoji se od ulaznih vrijednosti, težina i pristranosti, neto zbroja i aktivacijske funkcije. Njegova je primjena u strojnom učenju, za kontrolirano učenje binarnih klasifikatora. Perceptron je vrsta linearnog klasifikatora koji pomaže klasificirati ulazne podatke u određene klase. Sastavljen je od četiri glavna dijela, ulazne vrijednosti, težine i pristranosti, neto zbroja i aktivacijske funkcije. Za perceptron je karakterističan proces koji započinje uzimanjem svih ulaznih vrijednosti i njihovo množenje njihovim težinama. Zatim se sve te umnožene vrijednosti zbrajaju kako bi se stvorio ponderirani zbroj te se na kraju primjenjuje aktivacijska funkcija za proizvodnju

perceptronovog izlaza. Funkcija aktivacije osigurava da je izlaz mapiran između potrebnih vrijednosti kao što su (0,1) ili (-1,1).



Slika 4 Perceptron

Perceptronova primjena je nakon početne euforije ipak naišla na brojne prepreke i ograničenja. Vrlo se brzo pokazalo kako sam perceptron ne može biti iskoristiv za prepoznavanje puno različitih klasa podataka što je dovelo do stagnacije u razvoju umjetne inteligencije u cjelini. Ipak perceptron je svoju primjenu kasnije našao kao osnovna gradivna jedinica u arhitekturi višeslojnih neuronskih mreža kojima, među ostalim pripadaju i konvolucijske neuronske mreže.

4. UI I RAČUNALNE IGRE

Računalne su igre nastale na temelju želje istraživača da otkriju jesu li računala sposobna samostalno riješiti probleme za koje je nužno posjedovati inteligenciju. Prvi koji je napisao program za digitalnu verziju igrice Križić-Kružić bio je A. S. Douglas 1952. Već od devedesetih godina prošlog stoljeća pojavljuju se prvi programi koji su bili u stanju pobijediti ljudske protivnike, pa je tako program Chinook 1994. uspio pobijediti svjetskog prvaka u dami, Mariona Tinsleyja.

Za potrebe šaha se s vremenom razvijaju brojni programi od kojih se većina može smjestiti u dvije glavne kategorije: tip a i tip b. Tip a programi ispituju sve moguće poteze i na taj način donose odluku, dok tip b rabe stratešku umjetnu inteligenciju i analiziraju samo najvažnije i najčešće poteze. Zbog svoje jednostavnosti ali i zbog napretka u računalnoj tehnologiji i količini memorije tip a programi su danas puno popularniji. Jedan takav primjer tip a programa za igranje šaha je Deep Blue kojeg je razvio IBM, a koji je 1997. godine uspio poraziti tadašnjeg svjetskog prvaka Garryja Kasparova.

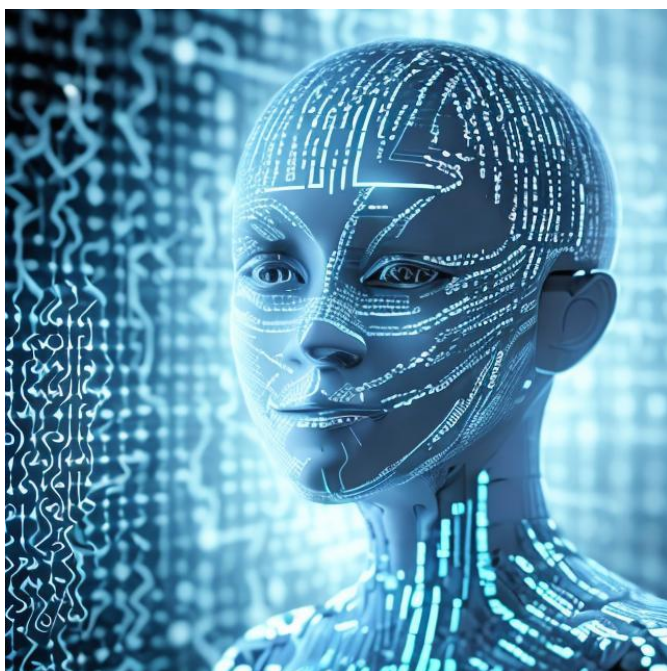
Od novijih iteracija u svijetu šaha svakako valja izdvojiti Stockfish *engine*. Stockfish, u prijevodu bakalar, je godinama vodeći i najbolji igrač-računalo šaha na svijetu i višestruki je svjetski prvak u brojnim natjecanjima računalnih strojeva za igranje šaha. Od travnja 2023. je i službeno najjači šahovski *engine*. Koristi i do najviše 1024 niti u višeprosorskim sustavima, a veličina njegove transpozicijske tablice je i do najviše 32 terabajta. Elo ranking njegove 16. zasad najnovije iteracije iznosi 3541 bod. Samo za usporedbu, najviši elo ranking svjetskog prvaka, i po mnogima najvećeg šahista svih vremena, Norvežanina Magnusa Carlsena iznosi 'tek' 2882. Stockfish koristi alfa-beta algoritam kojim prolazi kroz sve moguće poteze igrača na ploči. Alfa-beta algoritam karakterizira činjenica da prestaje ocjenjivati neki potez kada se pokaže da je taj potez lošiji od nekog prethodno evaluiranog poteza. Ovim trikom se algoritam i značajno ubrzava.

Osim ovih primjera, umjetna inteligencija se na razne načine koristi u modernim računalnim igrama. Najraširenija primjena umjetne inteligencije u *gaming* industriji se očituje u primjeni NPC likova. NPC (non player character) je engleski akronim koji opisuje entitet u videoigri koji nije pokretan od strane drugog igrača, već pomoću umjetne inteligencije s namjerom da oponaša čovjekove postupke. Ovi likovi omogućavaju korisniku privid igranja sa više igrača čak i kad igraju sami. Izlaskom chatGPT-a pojavljuje se nova generacija još pametnijih NP likova koje karakterizira, naizgled jednostavna mogućnost pretvaranja teksta u govor i sinkronizacija s pokretima usana. Novi NP likovi su u mogućnosti prisjetiti se razgovora koje su već vodili sa igračem što daje novu dimenziju i poboljšava osjećaj personaliziranog iskustva.

Kreiranje novih razina još je jedna od vrlo korisnih primjena umjetne inteligencije jer se kod nekih igara koje koriste princip otvorenog svijeta stvaranje i širenje istoga predstavljaju problem koji ponajprije troši jako puno vremena kada ga izvodi čovjek. U praksi se ovoj proces često naziva i proceduralno generiranje sadržaja. Generirati se može mali milijun sadržaja: od umjetnosti i ukrasnih elemenata videoigre do novih mapa, misija oružja i slično. U vremenu

kada je rad programera jednostavno preskup da bi se trošio na procese koji su jako repetitivni i ne zahtijevaju prevelik angažman osobe koja kodira, generiranje pomoću umjetne inteligencije se nameće kao idealno rješenje. Umjetna je inteligencija u stanju generirati nemali broj različitih krajobraza i različitih okoliša kako bi korisnik uvijek imao nešto novo za istražiti i otkriti. Dobar primjer ovakve prakse je i Mojangov Minecraft koji implementira proceduralno generiranje sadržaja za stvaranje novih krajolika.

Umjetnost generirana pomoću umjetne inteligencije je za širu javnost sigurno jedna od najpopularnijih primjena umjetne inteligencije u bližoj prošlosti. Alati za generiranje umjetnosti pomoću umjetne inteligencije su naizgled nova stvar, no pionir ovakvog pristupa je program nazvan AARON računalnog znanstvenika Harolda Cohena napravljen 1973. godine. Ovaj je program bio specijaliziran za stvaranje umjetnosti apstraktnog karaktera. Slijedili su ga brojni slični programi, a u novije vrijeme veliku su popularnost stekli *online* generatori umjetničkih slika poput Stable Diffusiona, Midjourneyja, Googleov Imagen i osobito Adobe Fireflyja od kojih su neki i besplatni za korištenje. Sučelje za korištenje ovih alata je vrlo jednostavno. Korisnik naime, samo unese tekst motiva koji želi kao temu slike, a softver treniran na velikom broju motiva kombinira svoje naučeno znanje i nasumičnu kreativnost kako bi iznova generirao nova djela. Portret Edmonda de Belamyja vjerojatno je najpoznatiji primjer umjetnosti generirane umjetnom inteligencijom ponajprije zato što je prodana na aukciji za čak 432 000 američkih dolara. Najzanimljivije je to što dotični Edmond de Belamy ne postoji, a sam portret generiran je od strane umjetničkog kolektiva *Obvious* korištenjem algoritma treniranog na više od 15000 portreta. Generatori umjetnost doveli su i do mnogih kontroverza jer pojedinci smatraju da njihovo korištenje može dovesti do kopiranja motiva i tehnika karakterističnih za pojedine autore. Također se postavlja i pitanje vlasništva generirane umjetnosti na koje još uvijek nema konkretnog odgovora.



Slika 5 UI generirana slika

Slika prikazuje primjer generirane umjetnosti pomoću umjetne inteligencije. Upit na *artificial intelligence* na besplatnom online *engineu* openart.ai.

Umjetna inteligencija se može koristiti kako bi se poboljšalo iskustvo igrača tako da se okoliš dinamički prilagođava korisnikovoj razini vještine, ali nalazi primjenu i u testiranju komponenti video igara i tako pomaže u razvoju.

4.1 Videoigre upravljane pokretom

Od samih početaka razvoja videoigara se stremilo ka tome da igranje bude igraču što intuitivnije i pristupačnije. Tako su se upravljačke palice za igranje stalno dorađivale i mijenjale svoj oblik, a sve u svrhu korisnikovog užitka i zabave. U skladu s time *developer* se početkom 21. stoljeća sve više okreću tehnologijama upravljanja igara pokretima tijela. Jedan od prvih takvih primjera na tržištu svakako je wiiRemote koji je predstavljen na sajmu E3 2005. godine. WiiRemote funkcioniše na principu senzora pokreta, a sa konzolom upravlja pomoću Bluetooth tehnologije. Sony je svoj pandan wiiRemoteu ,za PS3 izbacio 2010. godine pod nazivom PlayStation Move, a dodatak je kompatibilan i s novijim iteracijama konzole. Playstation Move se zasniva na kameri koja mapira prostor oko igrača te na temelju njegovih pokreta i udaljenosti kontrolera u ruci igrača donosi odluke o inputu. Microsoft je također imao sličnu varijantu za

svoju konzolu Xbox pod nazivom Xbox Kinect , ali projekt nikad nije zaživio te je ugašen 2017. godine. Svi ovi primjeri su naizgled slični projektu o kojem se govori u ovom radu, ali ni jedan od njih ne koristi umjetnu inteligenciju za donošenje odluka o pokretima igrača.

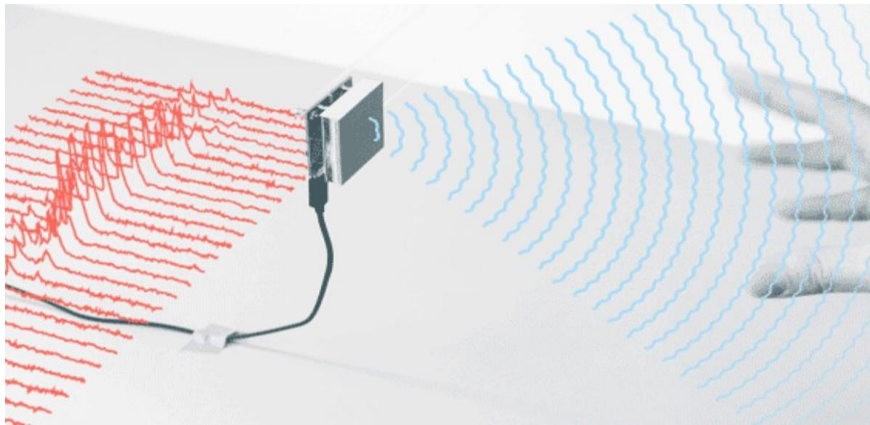
Izlaskom PlayStationa 5 Sony je predstavio novu značajku zvanu PS5 Gesture control. Pomoću ovog alata igraču se nude četiri nove mogućnosti upravljanja: upravljanje dodirima, upravljanje glasom, upravljanje pokretima lica te upravljanje pokretima tijela. Za korištenje Gesture controla igraču je potrebna videokamera i obični DualSense 5 kontroler za PlayStation 5.

Sony je 2021. godine predstavio projekt Sony AI koji bi trebao donijeti revoluciju u način na koji je umjetna inteligencija implementirana u svijet videoigara.

Isti je proizvođač 2020. godine, nekoliko mjeseci prije izlaska PlayStationa 5 patentirao *In-Game Resource Surfacing Platform* ili u prijevodu platformu za pronalaženje resursa u igri. Patent je opisan kao trener baziran na umjetnoj inteligenciji. Princip rada je jednostavan, a bazira se na tome da igrač, suočen s nekom preprekom u igri može govornom naredbom potražiti pomoć. Umjetna inteligencija prima govornu naredbu, analizira problem u kojem se igrač našao te u oblaku traži rješenja koja su koristili drugi igrači koji su uspješno prošli određenu prepreku te daje savjet igraču na temelju rezultata.

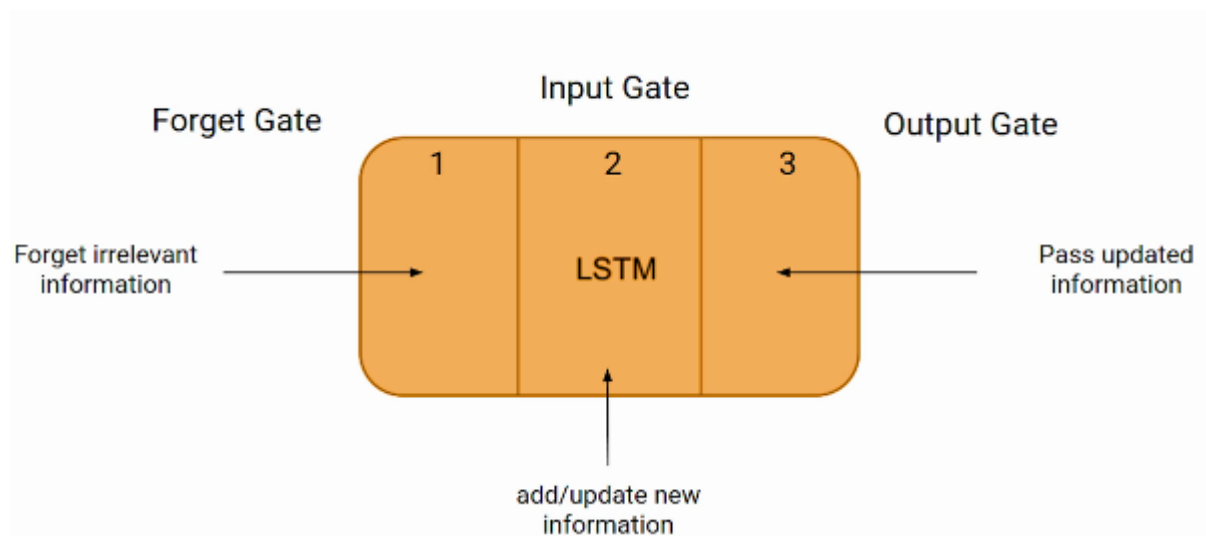
Jedan od novijih primjera sustava iza kojega stoji umjetna inteligencija je zasigurno Google GameFace. Izvorno je ovaj projekt inspiriran i osmišljen za potrebe osoba sa smanjenom pokretljivošću i invaliditetom tako što mapira živu mrežu od 468 točaka na licu te ih pretvara u telemetriju za klikove mišem i kretanje. Svoju će primjenu zasigurno naći i u igranju videoigara gestama, a sve zahvaljujući pozadini nastaloj na Google MediaPipe *frameworku*.

Još primjera korištenja umjetne inteligencije uključuje Googleov Project Soli. Ova implementacija koristi umjetnu inteligenciju kombiniranu s hardverskim dijelom koji se sastoji od minijaturnog radarskog senzora za otkrivanje i tumačenje gestikulacije rukama.



Slika 6 Project Soli

Arhitektura neuronske mreže u Project Soli je hibridna, što znači da je dijelom riječ o konvolucijskoj neuronskoj mreži (CNN) a dijelom o mreži dugog kratkoročnog pamćenja (LSTM). Mreža dugog kratkoročnog pamćenja je podvrsta Rekurentnih neuronskih mreža (RNN). Ime je na prvi pogled pomalo bizarno, ali ono implicira da ova vrsta mreže spaja kratkoročnu i dugoročnu memoriju koje su kod rekurentnih neuronskih mreža odvojene kategorije. Mreže dugog kratkoročnog pamćenja posebno su korisne kod podataka sekvencijalnog karaktera kao što je na primjer govor.



Slika 7 LSTM arhitektura

LeapMotion je uređaj sličan senzoru koji se koristi u Project Soli. LeapMotion kontroler je mali USB periferni uređaj koji koristi dvije monokromatske IR kamere i tri infracrvene LED diode na udaljenosti od otprilike jednog metra. Arhitektura neuronskih mreža LeapMotiona je također hibridna s elementima konvolucijskih neuronskih mreža i rekurentnih neuronskih mreža.

MediaPipe *framework* je biblioteka otvorenog koda koja sadrži brojna korisna rješenja vezana ponajprije za strojno učenje. Biblioteke poput MediaPipea i TensorFlowa su glavni pokretač i pozadina ovakvih projekata, a o Tensorflowu ćemo više u daljnjim poglavljima.

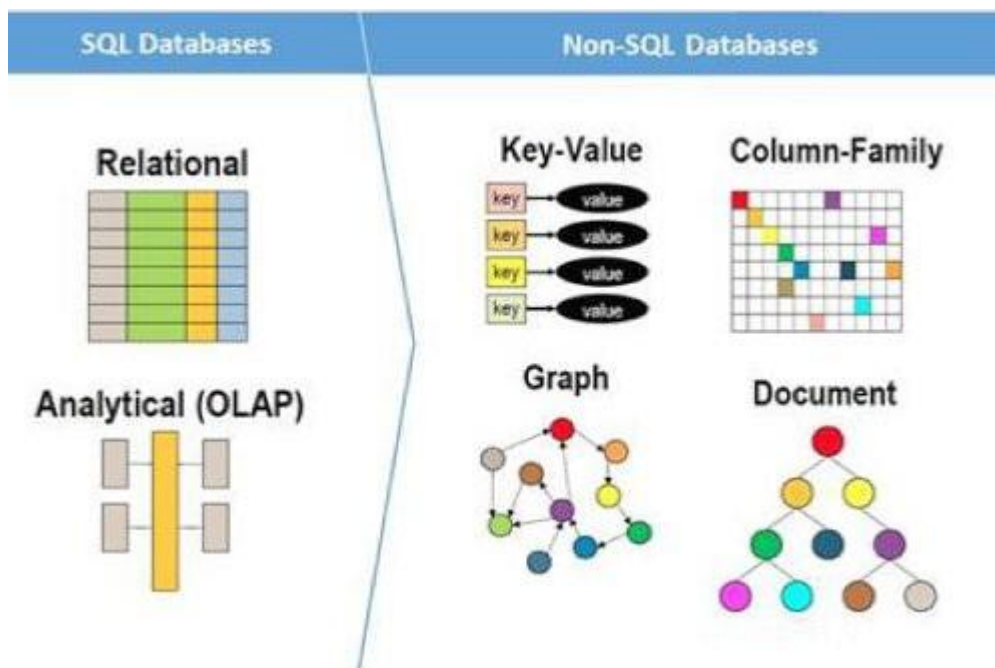
5. ODABIR METODA, TEHNIKA I ALATA

Pri izradi web-aplikacija nameće se more dostupnih tehnika i programskih rješenja. U ovom dijelu rada pokušati ću nabrojati i objasniti neke od najkorištenijih tehnika.

LAMP stack je akronim koji označava redom Linux, operativni sustav koji predstavlja temelj svakog ovakvog projekta, Apache, web poslužitelj zadužen za upravljanje korisničkim zahtjevima, MySQL relacijsku bazu podataka otvorenog koda te P koje može predstavljati PHP, Perl ili Python, a sva tri se odnose na programiranje dinamičkog web sadržaja. Ovakva struktura prati i ostale modele te slobodno možemo reći kako su ovo četiri stupa svake web aplikacije.

Druge tehnike uključuju Django, Meteor, Ruby on rails, te MEAN, MEVN I MERN stack. Posljednje tri navedene tehnike zapravo i nisu različite tehnike jer se razlikuju samo u elementu za izradu dinamičkog web sadržaja ili frontenda aplikacije tj. koriste se redom Angular, Vue i React. U ovom projektu korištena je MERN stack tehnika.

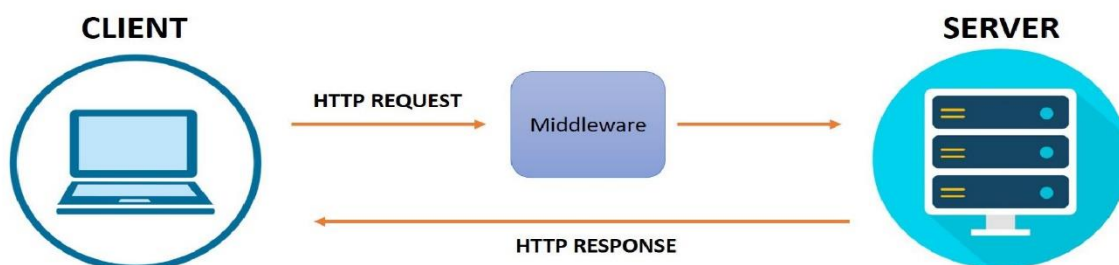
MongoDB je baza podataka otvorenog koda koja spada u NoSQL kategoriju baza što znači da se ne sastoji od klasičnih tablica i veza poput tradicionalnih baza kao što je ranije navedena MySQL.



Slika 8 SQL i noSQL baze

MongoDB je orijentirana na dokumente što znači da sprema podatke kao kolekcije sastavljene od dokumenata. To znači da ju krasi fleksibilnost s obzirom da se u bazu može spremiti praktički bilo što.

Express.js u web-aplikacijama predstavlja *middleware* tj. prenosnicu između backend dijela (baze podataka) i frontenda (korisničkog dijela). Njegova je glavna zadaća upravljanje korisničkim zahtjevima. Express je dio Node.js *frameworka* za izradu web-aplikacija i posebno ga odlikuje minimalizam i jednostavnost što ga čini prihvatljivim i za korisnike početnike kao i za stručnjake.



Slika 9 Backend, Middleware i Frontend

Node.js je okruženje otvorenog koda zaduženo za programiranje u JavaScriptu sa stane poslužitelja. Omogućuje asinkrono programiranje što znači da je u stanju upravljati sa više zahtjeva naizgled istovremeno , te izvršava kod u skladu sa pristiglim zahtjevom. Također, Node je moguće koristiti na svim operativnim sustavima uključujući Linux, Unix i Mac OS X.

React.js je JavaScript biblioteka koju je razvila Meta (bivši Facebook), a koristi se za izradu dinamičkih web komponenti odnosno *frontenda*. Prednost Reacta je da je orijentiran na komponente što korisniku omogućava stvaranje i ponovno korištenje istih komponenti. Njegova je sintaksa deklarativna, što znači da daje do znanja što se njome postiže. Još jedna korisna značajka Reacta je da koristi virtualni DOM, što omogućuje da se pri učitavanju stranice ponovno učitavaju samo dijelovi koji su promijenjeni.

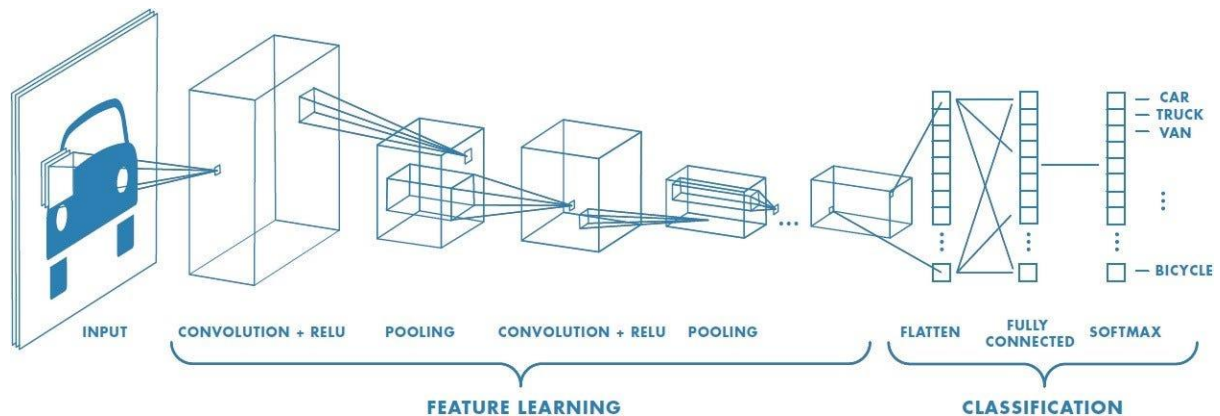
5.1 Umjetna inteligencija u projektu

Tensorflow biblioteka je *open-source* platforma razvijena od strane Google Brain tima namijenjena za strojno učenje i umjetnu inteligenciju. Kompatibilan je s brojnim programskim jezicima a iako nalazi brojne primjene najveći fokus TensorFlowa je na treniranju i donošenju odluka dubokih neuronskih mreža.

ML5 je JavaScript biblioteka temeljena na TensorFlowu, a predstavlja sučelje visoke razine za TensorFlow. Cilj ML5 je učiniti strojno učenje i umjetnu inteligenciju u cjelini privlačniju i pristupačniju početnicima. ML5 pruža jednostavan pristup TensorFlow algoritmima pomoću svojih metoda, a još jedna bitna prednost je da omogućava strojno učenje na klijentovoj stani odnosno na računalu korisnika.

Između brojnih alata i metoda koje ML5 sadrži za ovaj projekt najvažnija je ML5 Feature Extractor, čija je osnova konvolucijska neuronska mreža (CNN). Način na koji CNN funkcionira može se podijeliti u pet glavnih koraka. Prvi korak je pretvaranje ulaznog inputa odnosno slike u niz vrijednosti od kojih svaka predstavlja jedan piksel (ovo je istina jedino ako imamo crno bijelu sliku). Ovaj niz piksela možemo napisati i u obliku matrice kako bismo simulirali dvodimenzionalnost slike. Naredni korak je najbitniji u cijelom procesu i naziva se konvolucija, po čemu je mreža i dobila ime. Konvolucija je proces u kojem preko slike prelazimo preko slike koristeći težinski filter ili kernel. Ako se vratimo na usporedbu s matricama, filter možemo zamisliti kao pod matricu kojom množimo sve moguće pod matrice

ulaza. Filtar sadržava oblik one značajke koju tražimo pa ako npr. tražimo dijagonalne linije koje idu od gore desno prema dolje lijevo, matrica filtra će biti nalik jediničnoj matrici. Rezultat svakog pojedinog množenja je skalar i svaki se redom dodaje u novi niz koji se naziva mapa podudaranja. Kada je filter prošao preko svih djelića slike imamo novi niz koji sadrži vrijednosti podudarnosti s početnim filterom. Sada na sliku možemo primjenjivati nove filtre. U praksi se na svaku sliku primjeni od 32 pa sve do 512 filtera.



Slika 10 Procesuiranje slike konvolucijom

Također možemo redati filtre tj. izlaz jednog filtra pretvoriti u ulaz novog složenijeg filtra. Ako radimo sa slikama u boji, valja napomenuti da je ovom slučaju svaki piksel u slici predstavljen s po tri vrijednosti za svaku od osnovnih boja (crvena, zelena, plava). Isto tako i filter postaje trodimenzionalan te i kod njega nalazimo po tri vrijednosti za svaki piksel. Ovo ipak ne mijenja puno na stvari s obzirom da je produkt svakog množenja filtera i djelića slike uvijek skalar (jedan broj).

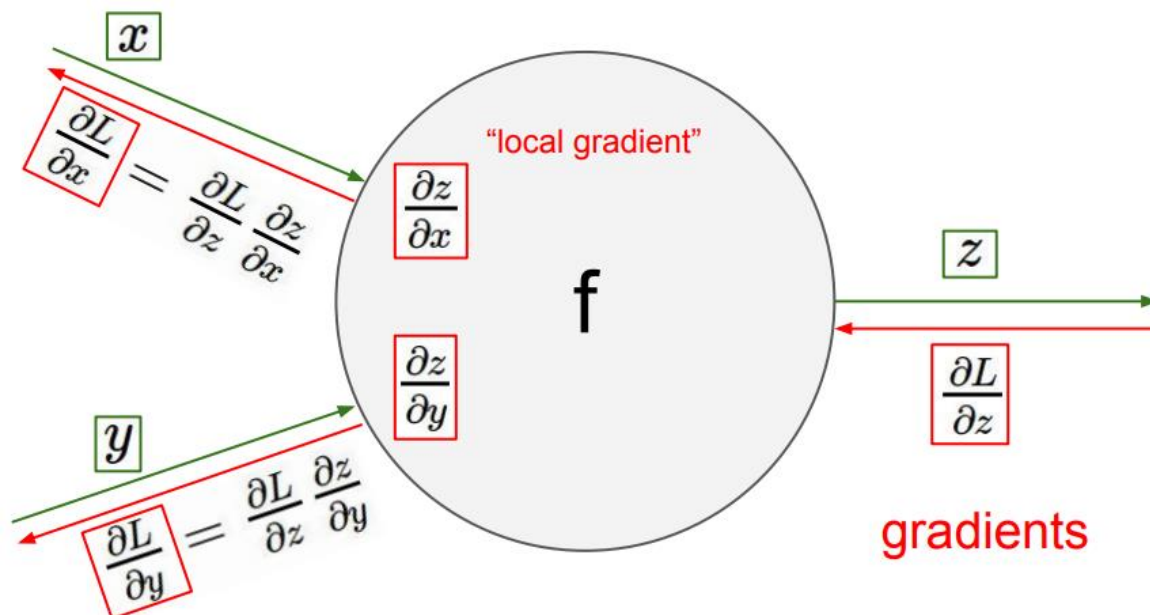
Feature extractor omogućava da korisnik iz slike ili videozapisa prepozna objekte ili uzorke. To je moguće ponajprije zahvaljujući *transfer* učenju što znači da je model unaprijed treniran na prepoznavanje nekih objekata koji se često pojavljuju (npr. lica). Feature extractor je podložan promjenama od strane korisnika tako da prepozna baš one značajke koje korisnik traži da prepozna, a omogućuje i spremanje utreniranog modela te ponovno korištenje istog.

5.2 Princip rada aplikacije

U projektu se kao stanja videoigre pojavljuju tri glavna stanja: stanje 0 predstavlja fazu treniranja konvolucijske neuronske mreže, stanje 1 je faza u kojoj trenirana mreža vrši predviđanja na temelju inputa kroz kameru, te će shodno tome pomicati brod u igri. Na kraju,

kada igrač izgubi sve živote prikazuje se njegov ostvareni rezultat te mu se nudi opcija ponovnog igranja.

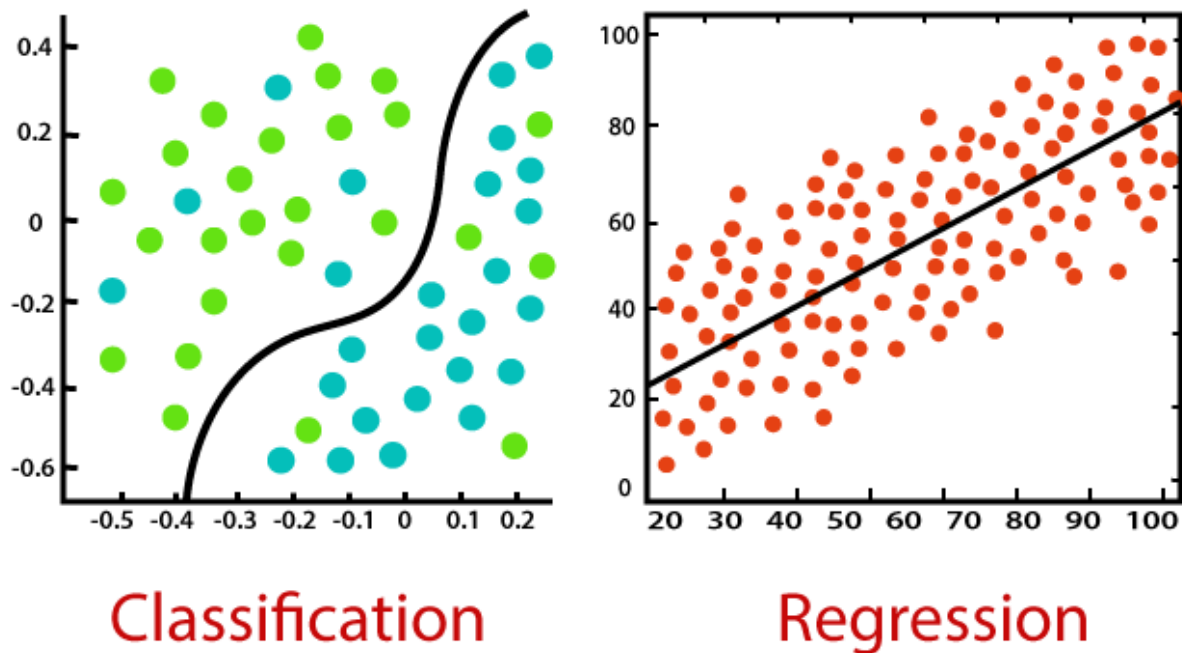
Stanje 0 je početna faza prije igranja i to je faza u kojoj učítavamo model. Učitamo ml5 mobileNet model za prepoznavanje značajki. Ovaj je model regresija. Slijedeća faza je dodavanje slika u regresijski model i treniranje modela. Kako ćemo u igri imati tri naredbe (idi lijevo, ne miči se, idi desno) tako ćemo u regresijski model slike dodavati putem triju indeksa: *regressor.addImage(-1)* za pomicanje u lijevo, *regressor.addImage(0)* za ne miči se i *regressor.addImage(1)* za pomicanje u desno. Kada pritisnemo tipku “treniraj” započinje treniranje modela naredbom *regressor.train()*. Model treniramo pomoću slika koje smo ranije dodali u regresijski model koristeći metodu *addImage()*. Na temelju primljenih slika model će imati određeni set filtera koji će se primjenjivati nad inputom u skrivenim slojevima neuronske mreže. Filteri koji su unaprijed zadani ipak neće često davati željene rezultate, odnosno gubitak (*loss*) biti će prevelik. Gubitak je vrijednost koja predstavlja razliku između inputa i predviđanja neuronske mreže. Zadatak je neuronske mreže da pomoću ulaznih slika kojima treniramo konvolucijsku neuronsku mrežu filtere promijeni kako bi lakše prepoznavali baš one značajke koje se pojavljuju u podacima za treniranje.



Slika 11 Backpropagation

Proces podešavanja filtera kako bi bolje prepoznavali značajke sa slika naziva se *backpropagation* ili širenje unazad.

Stanje 1 označava fazu igranja. U ovoj fazi meteori (tijela koja nanose štetu i potrebno ih je izbjegavati) se pojavljuju na vrhu ekrana i pravocrtno padaju prema dolje. Njihovo kretanje nije utjecano predviđanjima neuronske mreže. Kretanje broda je pak upravljano pokretima tijela tako da na temelju predviđanja neuronske mreže dobivamo vrijednost između -1 i 1. Valja primijetiti da je predviđanje neuronske mreže kontinuirana vrijednost jer je riječ o regresivnom modelu.



Slika 12 Klasifikacija i Regresija

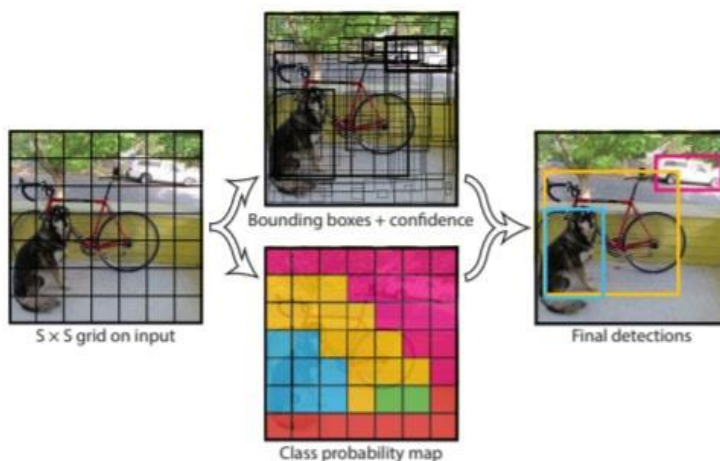
Odabir regresivnog modela je opravdan činjenicom da je igra dosta zanimljivija kada brod može poprimiti bilo koju vrijednost u intervalu $[-1,1]$, nego da može zauzeti samo vrijednosti $-1, 0, 1$ što bi bio slučaj pri korištenju klasifikacijskog modela. Također, u prototipu se predviđanje neuronske mreže množi sa konstantom 5 kako bi se dobio privid ubrzanja što dodatno poboljšava iskustvo igranja i unaprjeđuje prototip.

5.3 Moguća unaprjeđenja prototipa

Pri testiranju prototipa sam dosta eksperimentirao sa idejom treniranja modela te ponovnog korištenja već istreniranog modela. U sadašnjoj verziji prototipa, svaki put prije početka igranja potrebno je ponovno istrenirati model. Razlog tomu je što konvolucijske neuronske mreže

uspoređuju cijelu sliku komadić po komadić, pa tako pozadina i prostorija u kojoj se osoba nalazi kada pokušava igrati na modelu koji je učitani i već utreniran mora biti praktički identična onoj u kojem je model bio istreniran. Ovo naravno nije praktično pa je zato potrebno pronaći neki drugi algoritam za izdvajanje značajki.

YOLO (you only look once) je varijacija konvolucijske neuronske mreže koja je brz i efikasan algoritam za prepoznavanje značajki na slikama. Ovaj algoritam ne uspoređuje komadiće slike tražeći potencijalna podudaranja već računa kolika je vjerojatnost da se u djeliću slike nalazi instanca neke od već postojećih klasa predmeta za koje je YOLO treniran. Klase na koje je YOLO treniran uključuju lica, premete, biljke, prometne znakove itd. Ako algoritam dobije jako malu podudarnost u određenom dijelu slike, on ga automatski odbacuje, a ako nađe podudaranje oko objekta postavlja takozvani *bounding box* koji sadržava dio slike koji algoritam smatra relevantnim.



Slika 13 Yolo algoritam

YOLO algoritam žrtvuje temeljitost nauštrb brzine, ali mislim da bi za ovakav prototip u kojem znamo koji će se objekti pojavljivati, bio primjenjiv.

6. ZAKLJUČAK

Ovu temu za projekt sam izabrao jer sam mišljenja da sam ovim projektom mogao na jednostavan način pokazati samo mali dio onoga što nam umjetna inteligencija omogućava. Koristeći osnovne alate poput JavaScripta, Tensorflowa i ostalih alata je moguće stvoriti nešto što bi se do sada činilo kao znanstvena fantastika. Mislim da je važno sa umjetne inteligencije

skinuti stigmatu, koja upravo potječe iz filmova, nečega što je strašno, opasno i sa čime je bolje ne imati posla. Umjetna mi se inteligencija prije nego što sam počeo pisati ovaj projekt činila kao magija, nešto što jednostavno jest, i što se ne isplati propitkivati ili dalje istraživati. Veliko je olakšanje bilo shvatiti da umjetna inteligencija nije nešto tako daleko i nedostižno, već nešto što je nadohvat ruke svakome tko želi i spreman je naučiti i promijeniti svoje stavove. U svom istraživanju sam više naučio, to se više novih pojmova i smjerova otvorilo. Siguran sam da će mi ova osnovna znanja koja sam stekao biti dobar temelj i putokaz kako dalje istraživati i otkrivati mogućnosti umjetne inteligencije.

7. LITERATURA

[1]Anić, Nikola i Petar Anić. "Umjetna inteligencija kao segment strategije." National security and the future, vol. 21, br. 3, 2020, str. 120. <https://doi.org/10.37458/nstf.21.3.4>. Citirano 06.08.2023.

[2] Franjić, Darjan. "UMJETNA INTELIGENCIJA U RADIOLOGIJI: ETIČKI PROBLEMI." Zdravstveni glasnik, vol. 6, br. 2, 2020, str. 62. <https://doi.org/10.47960/2303-8616.2020.12.61>. Citirano 06.08.2023.

[3] Singbo, Odilon Gbenoukpo. "Umjetna inteligencija u suvremenom biokibernetičkom svijetu." Spectrum, vol. , br. 3-4, 2008, str. 55-56. <https://hrcak.srce.hr/38568>. Citirano 07.08.2023.

[4] Prister, Vladimir. "UMJETNA INTELIGENCIJA." Media, culture and public relations, vol. 10, br. 1, 2019, str. 69.

[5] Anić, Nikola i Petar Anić. "Umjetna inteligencija kao segment strategije." National security and the future, vol. 21, br. 3, 2020, str. 121-123.

[6]<https://en.wikipedia.org/wiki/Convolution>

<https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>

<https://appsilon.com/object-detection-yolo-algorithm/>

8. POPIS SLIKA

Slika 1 Umjetna inteligencija.....	3
Slika 2 Konvolucija nizova.....	5
Slika 3 Konvolucija matrica.....	6
Slika 4 Perceptron	8
Slika 5 UI generirana slika.....	11
Slika 6 Project Soli.....	13
Slika 7 LSTM arhitektura	13
Slika 8 SQL i noSQL baze	15
Slika 9 Backend, Middleware i Frontend	15
Slika 10 Procesuiranje slike konvolucijom.....	17
Slika 11 Backpropagation	18
Slika 12 Klasifikacija i Regresija.....	19
Slika 13 Yolo algoritam.....	20