

Robotska interakcija pogonjena kontekstom

Vukoje, Antonija

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:183827>

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

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering
and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Antonija Vukoje

Zagreb, 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentori:

Dr. sc. Tomislav Stipančić

Student:

Antonija Vukoje

Zagreb, 2021.

Izjavljujem da sam ovaj rad izradila samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem mentoru dr.sc. Tomislavu Stipančiću na ukazanome povjerenju i savjetima tijekom izrade ovog rada te asistentu Leonu Korenu, mag. ing. na idejama i pruženoj pomoći.

Na kraju bih još htjela zahvaliti svima koji su mi bili podrška tijekom studiranja te na bilo kakav način pomogli da dođem do kraja studija.

Antonija Vukoje



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomatske ispite
Povjerenstvo za diplomatske radove studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa: 602-04/21-6/1	
Ur. broj: 15-1703-21	

DIPLOMSKI ZADATAK

Student: **ANTONIJA VUKOJE**

Mat. br.: 0035202772

Naslov rada na hrvatskom jeziku: **Robotska interakcija pogonjena kontekstom**

Naslov rada na engleskom jeziku: **Context-driven robot interaction**

Opis zadatka:

Suvremene primjene robotskih sustava često podrazumijevaju interakciju i suradnički rad robota s ljudima gdje se poštuju različita pravila djelovanja koja osiguravaju sigurnu, pouzdanu i intuitivnu interakciju. Prateći definirani interakcijski scenarij roboti često koriste posebno oblikovani informacijski prostor koji podržava određenu razinu uzajamnog razumijevanja (engl. Common Ground). Takva interakcija može biti temeljena na prepoznavanju i razumijevanju interakcijskih gesti koje su kontekstualne te imaju izvorište u ljudskoj komunikaciji.

U radu je potrebno izraditi cjelovito rješenje za interakciju dva robota temeljeno na kontekstu uključujući sljedeće:

1. definirati scenarij interakcije za dva robota u sklopu informacijskog prostora koji može uključivati i ljude,
2. definirati i povezati komponente sustava (osigurati mrežnu komunikaciju, definirati uporabu senzora i sl.), te
3. razviti programsku podršku za ostvarivanje robotske interakcije (npr. razviti interakcijske geste za dva robota koje će biti pokretane u ovisnosti o tijeku interakcije).

Razvijeno rješenje potrebno je eksperimentalno verificirati na opremi dostupnoj u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava. U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:
6. svibnja 2021.

Rok predaje rada:
8. srpnja 2021.

Predviđeni datum obrane:
12. srpnja do 16. srpnja 2021.

Zadatak zadao:
doc. dr. sc. Tomislav Stipančić

Predsjednica Povjerenstva:
prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	IV
POPIS OZNAKA	V
POPIS KRATICA	VI
SAŽETAK.....	VII
SUMMARY	VIII
1. UVOD.....	1
2. KONTEKST	2
2.1. Definicija konteksta	2
2.2. Razumijevanje konteksta	2
2.3. Uloga konteksta kod percepcije emocija	3
3. UMJETNA INTELIGENCIJA	4
3.1. Što je umjetna inteligencija?	4
3.2. Alati umjetne inteligencije	5
3.2.1. Prepoznavanje i obrada prirodnog jezika.....	5
3.2.2. Strojni vid.....	6
3.2.3. Strojno i duboko učenje	6
3.2.3.1. Građa umjetnih neuronskih mreža	7
3.2.3.2. Struktura umjetnih neuronskih mreža	9
3.2.3.3. Hiperparametri umjetnih neuronskih mreža	10
3.3. Virtualni asistenti	13
3.4. Inteligentni emocionalni roboti	13
4. POSTUPAK OSTVARIVANJA ROBOTSKE INTERAKCIJE.....	14
4.1. Scenarij robotske interakcije	14
4.2. Integracija komponenti sustava.....	20
4.2.1. Programski paket za prepoznavanje govora.....	20
4.2.2. Programski paket za pretvorbu teksta u govor.....	20
4.2.3. Programski paket za obradu prirodnog jezika.....	20
4.2.4. Programski paket za duboko učenje.....	21
4.2.5. Programski paket za obradu homogenih višedimenzionalnih nizova	21
4.2.6. Programski paket za povezivanje robota sa računalom	21
4.3. Izrada umjetne neuronske mreže.....	22
4.3.1. Definiranje i obrada ulaznih podataka	22
4.3.1.1. Tokenizacija	23
4.3.1.2. Normalizacija veličine slova.....	23
4.3.1.3. Eliminacija stop-riječi i interpunkcijskih znakova	23
4.3.1.4. Korjenovanje riječi.....	23
4.3.1.5. Pretvorba tekstualnih varijabli u numeričke varijable	24
4.3.2. Izrada modela i treniranje neuronske mreže	26

4.3.2.1. Model umjetne neuronske mreže	26
4.3.2.2. Treniranje umjetne neuronske mreže	26
4.4. Programska podrška za ostvarivanje robotske interakcije	28
4.4.1. Odabir robota	28
4.4.2. Povezivanje robota i računala	30
4.4.3. Definiranje i izvođenje robotskih putanja kretanja	32
5. ZAKLJUČAK	40
LITERATURA	41
PRILOZI	45

POPIS SLIKA

Slika 1.	Važnost kontekstualne okoline kod percepcije emocija [11]	3
Slika 2.	Vizualni prikaz Turingovog testa [14]	4
Slika 3.	Grafički prikaz alata umjetne inteligencije	5
Slika 4.	Taksonomija umjetne inteligencije	7
Slika 5.	Klasične aktivacijske funkcije [22]	9
Slika 6.	ReLU aktivacijska funkcija	9
Slika 7.	Struktura umjetne neuronske mreže	10
Slika 8.	Ovisnost funkcije gubitka o stopi učenja [30]	12
Slika 9.	Shematski prikaz scenarija robotske interakcije	15
Slika 10.	Primjer JSON datoteke sa definiranim rečenicama	22
Slika 11.	Model umjetne neuronske mreže	26
Slika 12.	Grafički prikaz mjere gubitka po epohi učenja	28
Slika 13.	Robot <i>UR5</i> [45]	28
Slika 14.	Privjesak za učenje sa grafičkim sučeljem <i>Polyscope</i> [45]	29
Slika 15.	Izbornik za podešavanje postavki robota	31
Slika 16.	Izbornik za podešavanje mrežne komunikacije robota	31
Slika 17.	Vizualni prikaz ljudskih gesti pri različitim emocijama [48]	32
Slika 18.	Neutralno stanje robota	33
Slika 19.	Robotska putanja za prikazivanje emocije <i>Sadness</i> (tuga)	34
Slika 20.	Robotska putanja za prikazivanje emocije <i>Happiness</i> (sreća)	35
Slika 21.	Robotska putanja za prikazivanje emocije <i>Anger</i> (ljutnja)	36
Slika 22.	Robotska putanja za prikazivanje emocije <i>Surprise</i> (iznenađenje)	37
Slika 23.	Robotska putanja za prikazivanje emocije <i>Disgust</i> (gađenje)	38
Slika 24.	Robotska putanja za prikazivanje emocije <i>Fear</i> (strah)	39

POPIS TABLICA

Tablica 1. Prikaz kontekstualnih rečenica koje predstavljaju određena emocionalna stanja .	16
Tablica 2. Prikaz mogućih empatičnih odgovora računala/robota	18
Tablica 3. Moduli paketa <i>PyTorch</i> potrebni za izradu umjetne neuronske mreže	21
Tablica 4. Vizualni prikaz stvaranja modela <i>bag-of-words</i>	25
Tablica 5. Hiperparametri umjetne neuronske mreže	27
Tablica 6. Tehničke karakteristike robota <i>UR5</i> [45]	29

POPIS OZNAKA

Oznaka	Mjerna jedinica	Opis oznake
v	-	umnožak ulaznih vrijednosti i težinskih parametara umjetne neuronske mreže
x	-	vektor ulaznih vrijednosti umjetne neuronske mreže
y	-	izlazna vrijednost umjetne neuronske mreže
$\hat{y}$	-	predviđena izlazna vrijednost umjetne neuronske mreže
ϕ	-	aktivacijska funkcija umjetne neuronske mreže
ω	-	vektor težinskih parametara umjetne neuronske mreže

POPIS KRATICA

Kratika	Opis
2D	dvodimenzionalna veličina
3D	trodimenzionalna veličina
CPU	<i>central processing unit</i> – središnja procesorska jedinica
DHCP	<i>Dynamic Host Configuration Protocol</i> – protokol za dinamičku dodjelu IP adrese
GPU	<i>graphics processing unit</i> – grafička procesorska jedinica
IDE	<i>integrated development environment</i> – integrirano razvojno okruženje
ISO	<i>International organization for standardization</i> – Međunarodna organizacija za standardizaciju
JSON	<i>JavaScript Object Notation</i> – <i>JavaScript</i> format datoteke
MIT	<i>Massachusetts Institute of Technology</i> – naziv sveučilišta
ReLU	<i>Rectified Linear Unit</i> – aktivacijska funkcija praga
SGD	<i>stochastic gradient descent</i> – stohastički gradijentni spust
TTS	<i>Text to Speech</i> – pretvorba tekstualnih zapisa u govor
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> – protokol za ostvarivanje komunikacije unutar lokalne mreže

SAŽETAK

Prijenos informacija odvija se izmjenom poruka čije značenje ovisi o kontekstu. Kontekst je presudan jer određuje korisnost informacije. Osim konteksta, bitnu ulogu u komunikaciji imaju i osjećaji. Stvaranjem robota koji razumiju glasovne upute, određene kretnje ili osjećaje, omogućuje se povećana fleksibilnost sustava, smanjuje se potreba za stručnom obukom te se ostvaruje bolja suradnja robota s ljudima. Za potrebe takvog pristupa razvijena je metoda strojnog učenja pomoću koje se nastoji simulirati mehanizam učenja kod bioloških organizama.

U radu je razrađena ideja robotske interakcije koja uključuje i čovjeka, a glavni zadatak iste je prepoznavanje ljudskih emocija prema kontekstu rečenice. Korišten je pristup koji osigurava jednostavan i intuitivan dizajn interakcije, uz uporabu kolaborativnog robota i modela umjetne neuronske mreže za prepoznavanje ljudskih emocija.

Ključne riječi: kontekst, emocije, strojno učenje, umjetna neuronska mreža, robotska interakcija

SUMMARY

The transfer of information takes place by exchanging messages whose meaning depends on the context. Context is crucial because it determines the usefulness of information. In addition to context, feelings also play an important role in communication. Creating robots that understand voice instructions, certain movements, or feelings allows for increased system flexibility, reduces the need for professional training, and improves human-robot collaboration. For the purposes of such an approach, machine learning has been developed. Machine learning attempts to simulate the learning mechanism in biological organisms.

This thesis develops the idea of robotic interaction involving humans, whose main task is to recognize human feelings according to the context of a sentence. An approach was used that allows for a simple and intuitive interaction design, using a collaborative robot and an artificial neural network model to recognize human feelings.

Key words: context, emotions, machine learning, artificial neural network, robot interaction

1. UVOD

Komunikacija je osnovno sredstvo uspostavljanja odnosa između ljudi i okoline. Prijenos informacija odvija se izmjenom poruka, a njihovo značenje ovisi o kontekstu u kojemu se koriste. Promatrajući iz kognitivne perspektive, komunikacija je inferencijalni proces zasnovan na mentalnim stanjima i zajedničkom znanju. Ona obuhvaća sve što pridonosi interpretaciji komunikacijskog čina, uključujući kontekst koji se smatra podlogom razumijevanja [1]. Kontekst je presudan jer određuje korisnost informacije, koje pretpostavke je potrebno donijeti o onome što je priopćeno te, najvažnije, on daje poruci smisao.

Osjećaji također imaju veliku ulogu u komunikaciji. Sposobnost razumijevanja emocija veoma je bitna u međuljudskim odnosima. Roboti koji imaju vještine slične ljudskima doživljavaju se društvenijima, razumljivijima i brižnijima. Kako bi stekli takve vještine, potrebni su alati koji će im omogućiti percepciju i razumijevanje emocionalnih stanja ljudi [2]. Mnogi istraživači na području robotike usredotočili su se na razvoj robota koji su u mogućnosti prikazati različite emocije izrazima lica, pokretima tijela ili karakteristikama glasa. U takvim istraživanjima najčešće su korišteni humanoidni roboti kako bi se ljudi lakše poistovjetili sa njima, no modeli izražavanja emocija nisu ograničeni samo na robote slične čovjeku. Osim humanoidnih robota, postoje pristupi gdje roboti ostvaruju modele komunikacije izmjenom osvjetljenja ili izvođenjem planiranih putanja kretanja [3].

Ljudska reakcija na robote često je istovjetna onoj na ljude, posebice ukoliko roboti ostvaruju vezu s ljudima uporabom istog jezika ili gesti. Iako se tehnike orijentirane na humanoidne robote čine kao najbolje rješenje, takvi roboti su ponekad skupi i tehnički ih je teško dizajnirati [4]. Kako bi se nadvladala takva ograničenja, u radu je korišten pristup koji osigurava jednostavan i intuitivan dizajn interakcije, uz uporabu kolaborativnog robota i modela umjetne neuronske mreže za prepoznavanje ljudskih emocija.

2. KONTEKST

2.1. Definicija konteksta

Riječ kontekst potječe od lat. *contextus* što znači spajanje ili spoj. Situacijski kontekst ukupnost je prirodnih, društvenih i kulturnih uvjeta u koje se smješta izričaj, a to su podaci o kulturnoj i psihološkoj situaciji koji su zajednički pošiljatelju i primatelju poruke, kao i podaci o iskustvima i znanjima podjednako obojice njih [5].

Često je moguće čuti izraz da je određeno značenje nekog pojma „ovisno o kontekstu“. Postoje brojni pokušaji definiranja konteksta, no ne postoji jedinstvena teorija što je on doista. Moguće ga je definirati i kao skup elemenata koji su relevantni za razumijevanje određenog teksta, neovisno o tome je li on u pisanom ili izgovorenom obliku [6].

2.2. Razumijevanje konteksta

Kognitivna znanost bavi se proučavanjem funkcija ljudskog uma u različitim scenarijima te analizom živčanog sustava [7]. Kao takva, ona je važan dio u razumijevanju konteksta i temelj za modeliranje interakcije između čovjeka i računala.

U svakodnevnim situacijama kontekst može biti složen ili zbunjujući ukoliko se pogrešno protumači. Takvo što je veoma očito kod modeliranja razumijevanja između čovjeka i robota budući da roboti imaju ograničen kapacitet pri tumačenju konteksta u njegovoj punoj složenosti. Pojmovi koje ljudi razlikuju i smatraju relevantnima u stvarnom svijetu, pretežno su definirani iz naučenih obrazaca ili iz iskustva [8]. No, postavlja se pitanje, zašto bi roboti trebali razumjeti čovjeka? Klasični roboti konstruirani za specifičnu primjenu (npr. industrijski roboti u proizvodnji) ne pružaju veliku fleksibilnost i za rukovanje njima potrebni su stručni ljudi. Stvaranjem robota koji razumiju glasovne upute, određene kretnje ili osjećaje čovjeka omogućuje se povećana fleksibilnost sustava, smanjuje se potreba za stručnom obukom te se ostvaruje jednostavnija suradnja. Inspiracija za to je svakodnevni život. Naime, ljudi su u međusobnoj interakciji navikli podučavati ili objašnjavati kroz prirodni jezik i neverbalne znakove. Takvi obrasci nastoje se iskoristiti i za prilagodbu robota na način da oni ne zahtijevaju od ljudi da im se prilagode na poseban način, već da ih podučavaju i komuniciraju s njima kao da su i oni ljudi [9].

2.3. Uloga konteksta kod percepcije emocija

Već duži niz godina postoji uvriježeno mišljenje kako se određene „osnovne“ emocije nalaze u svim ljudskim kulturama. One se obično dijele na sreću, tugu, strah, bijes, iznenađenje te gađenje. Svaka emocija uzrokuje poseban prepoznatljiv izraz lica te za većinu njih postoje dokazi o određenim fiziološkim odgovorima kao što su promjene u glasu te kognitivnim fenomenima poput usmjeravanja pažnje na podražaj osjećaja [10].

Mnoge metode bave se prepoznavanjem emocija na temelju izraza ljudskog lica. Iako je to najčešće veoma dobra metoda, sami izrazi lica ponekad mogu biti zbunjujući. Kao primjer moguće je promotriti sliku u nastavku na kojoj je prikazan američki političar Jim Webb [Slika 1.]. Na lijevom dijelu slike, pod oznakom (a), iz njegovoga lica moglo bi se protumačiti da se radi o izražavanju ljutnje. No, ukoliko se pogleda desni dio slike, pod oznakom (b), početna tvrdnja više se ne čini ispravnom jer se radi o slici koja je nastala nakon saznanja o pobjedi na izborima [11]. Iz navedenog primjera proizlazi da je za percepciju emocija bitno poznavanje vanjskog konteksta koji predstavlja okolina te unutarnjeg konteksta koji proizlazi iz uma opažatelja.

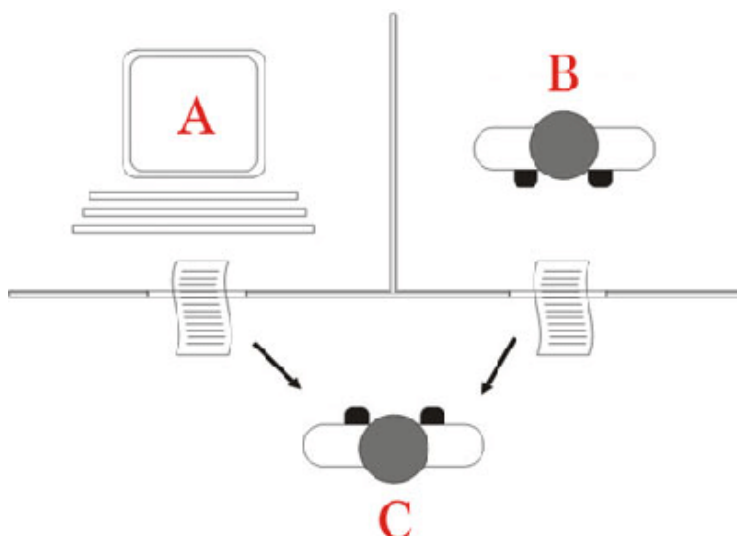


Slika 1. Važnost kontekstualne okoline kod percepcije emocija [11]

3. UMJETNA INTELIGENCIJA

3.1. Što je umjetna inteligencija?

Često se postavlja pitanje o ispravnoj definiciji inteligencije. Ljudska inteligencija može se definirati kao opća mentalna sposobnost rasuđivanja, rješavanja problema i učenja. Zbog svoje općenite prirode, inteligencija integrira kognitivne funkcije poput percepcije, pažnje, pamćenja, jezika ili planiranja [12]. Pojam umjetne inteligencije pojavio se 1950. godine razvojem Turingovog testa [Slika 2.]. Alan Turing razvio je empirijski test koji pruža mogućnost testiranja inteligencije računala. Test obuhvaća ljudskog ispitivača u jednoj sobi, čovjeka u drugoj i računalo u trećoj sobi. Komunikacija se odvija pismenim putem između ispitivača i čovjeka te ispitivača i računala. Ukoliko ispitivač nije siguran komunicira li sa čovjekom ili sa računalom, Turingov test je položen te se računalo može smatrati inteligentnim [13].

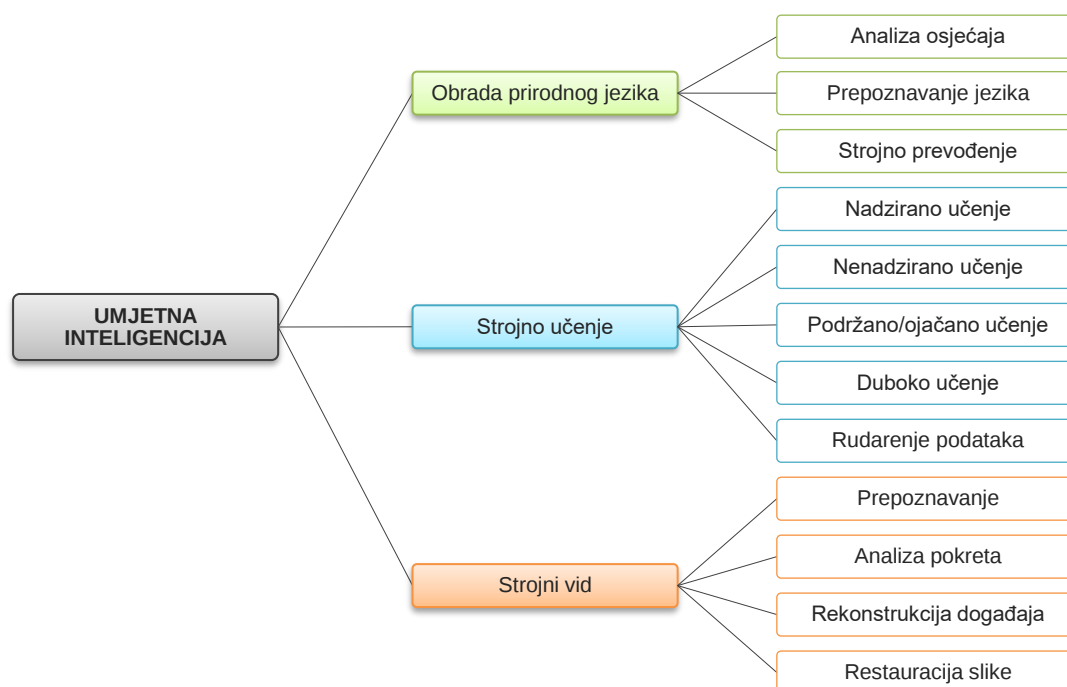


Slika 2. Vizualni prikaz Turingovog testa [14]

Umjetna inteligencija koristi strojeve za oponašanje ljudskog razmišljanja i izvođenje radnji koje su usko povezane s čovjekom. Ipak, postavlja se pitanje mogu li računala doista misliti? Potrebno je razlikovati razmišljanje i inteligenciju. Razmišljanje je sredstvo analiziranja, vrednovanja i formuliranja ideja i koncepata [15], dok je inteligencija mentalna sposobnost koja podrazumijeva usvajanje te primjenu znanja i vještina.

3.2. Alati umjetne inteligencije

Ukoliko postoji težnja za gradnjom inteligentnih sustava, prirodno je pretpostaviti kako takvi sustavi posjeduju kognitivne sposobnosti, vizualnu percepciju te mogućnost govora. Kako bi komunikacija strojeva s ljudima bila što prirodnija, istraživači iz različitih područja znanosti, poput informatike, matematike, neuroznanosti i kinematike, nastoje stvoriti računala i robote koji mogu misliti, učiti i prepoznavati osjećaje [16]. Navedeni razlozi bili su inspiracija i za ovaj rad te su stoga u nastavku opisani alati umjetne inteligencije orijentirani na prepoznavanje ljudskih emocija. Kratak grafički prikaz alata koji će biti obrađeni u nastavku moguće je promotriti na [Slika 3.].



Slika 3. Grafički prikaz alata umjetne inteligencije

3.2.1. Prepoznavanje i obrada prirodnog jezika

Eliza je jedan od prvih računalnih programa za obradu prirodnog jezika. Stvorena je na sveučilištu *Massachusetts Institute of Technology* (MIT) kako bi simulirala komunikaciju između ljudi i strojeva. Njen zadatak bila je imitacija razgovora psihoterapeuta i korisnika kao pacijenta. Učenjem iz razgovora sa korisnikom tvrdi se da je postignuta 95 %-tna točnost izgovora riječi [15]. Iako *Eliza* možda nije mnogo pridonijela obradi prirodnog jezika, ona je računalni program koji, barem prividno, posjeduje emocije.

Izbor riječi koje ljudi koriste u govoru važan je za uspješno razlikovanje emocija [17]. Značajke govora mogu se podijeliti na zvučne i jezične, a odabir metode izdvajanja značajki ovisi o odabranoj bazi podataka govora. Ako je govorna baza podataka bazirana na tekstu, jezične značajke mogu se zanemariti, no ako je baza podataka slična stvarnom životu tada će jezične značajke biti veoma važne. Zvučne značajke mogu se podijeliti na prozodijske značajke (naglasak, ritam, intonacija), spektralne značajke (frekvencija) i značajke kvalitete glasovnog zapisa [18].

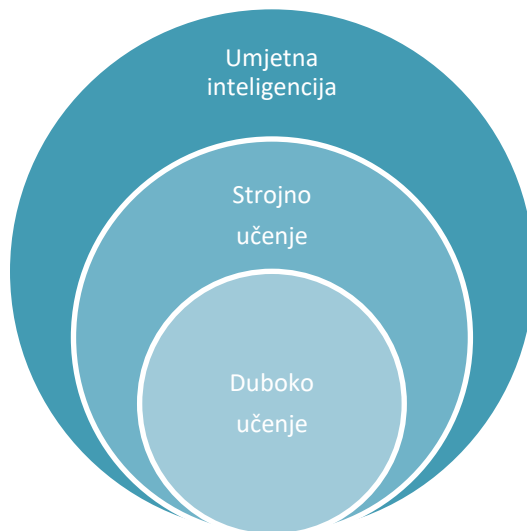
3.2.2. Strojni vid

Ljudi prepoznaju emocije iz izraza lica sugovornika te određenih karakteristika u govoru kao što je povišen ton ili tempo izgovora riječi. Govoreći o prepoznavanju izraza lica, nezaobilazno je vizualno osjetilo. Ono je kod strojeva postignuto uporabom tzv. strojnog vida. Vizualna osjetila ljudima omogućuju percepciju i razumijevanje svijeta koji ih okružuje. Strojevi oponašaju ljudski vid na način da elektronički opažaju okolinu te stvaraju sliku koju kasnije nastoje razumjeti. Budući da ljudi žive u trodimenzionalnom (3D) svijetu, uspješna primjena vida kod računala nije nimalo laka. Dostupni vizualni senzori najčešće daju dvodimenzionalne (2D) slike, a redukcijom broja dimenzija dolazi i do gubitka velike količine informacija. Moguća je primjena opreme koja ima sposobnost prikupljanja 3D slika, no one mogu biti upitnih vrijednosti i analiza takvih skupova podataka je složenija od one kod 2D slika [19].

3.2.3. Strojno i duboko učenje

Kako su ljudska osjetila povezana sa mozgom, tako je i kod računala potreban sustav koji će upravljati „računalnim osjetilima“. Za potrebe takvog pristupa razvijeno je strojno učenje kao podskup umjetne inteligencije. Strojno učenje obuhvaća modeliranje ponašanja iz stvarnog života pomoću računalnih algoritama. Ono omogućuje kontinuiranu improvizaciju pri izvedbi sustava koji se razvijaju u ovisnosti s vremenom [20]. Algoritmi strojnog učenja su iterativne prirode što znači da se nakon svakog provedenog koraka provodi ispitivanje i ispravljanje grešaka. Najčešća podjela strojnog učenja je ona na nadzirano, nenadzirano i podržano/ojačano učenje. Kod nadziranog učenja ulazni podaci su označeni te je na temelju njih potrebno provesti predviđanje klase novog seta podataka. U nenadziranom učenju ulazni podaci nisu označeni, već umjetna neuronska mreža pokušava pronaći potencijalne korelacije

atributa podataka te ih prema tome grupirati [21]. Duboko učenje je podskup strojnog učenja temeljen na umjetnim neuronskim mrežama i algoritmima za treniranje neuronskih mreža. Taksonomija umjetne inteligencije koja predstavlja spomenute skupove i njihove odnose prikazana je na [Slika 4.].



Slika 4. Taksonomija umjetne inteligencije

Pomoću umjetnih neuronskih mreža nastoji se simulirati mehanizam učenja kod bioloških organizama. Osnovu ljudskog živčanog sustava čine živčane stanice zvane neuroni. Oni su međusobno povezani aksonima i dendritima, a mjesto gdje se povezuju naziva se sinapsa. Snage sinaptičkih veza često se mijenjaju ovisno o vanjskom podražaju, a upravo ta izmjena je način na koji se odvija učenje kod živih organizama [22].

3.2.3.1. Građa umjetnih neuronskih mreža

Za simulaciju mehanizma učenja koriste se umjetne neuronske mreže. One su građene od međusobno povezanih jedinica, neurona, modeliranih prema pojednostavljenom modelu biološkog neurona. Umjetni neuron očitava ulazne vrijednosti sustava ili izlazne vrijednosti drugih neurona te ažurira svoje unutarnje stanje i stvara odziv [23]. Utjecaj ulaznih vrijednosti neurona vrednuje se težinama (eng. *weights*). Težine određuju kako pojedina značajka utječe na ukupni rezultat predviđanja. Ukoliko je težinska vrijednost neke značajke velika, tada će ona imati značajan utjecaj na ukupan rezultat. Zbroj težina pomnožen s ulaznim varijablama naziva se linearnom kombinacijom ulaznih varijabli [24], odnosno linearnom regresijom.

Cilj algoritma linearne regresije je stvaranje sustava koji kao ulaz uzima vrijednosti u obliku vektora $\mathbf{x}$ te nastoji predvidjeti skalarnu vrijednost y kao izlaz, pri čemu y linearno ovisi o $\mathbf{x}$. Ukoliko se vrijednost koju se nastoji predvidjeti označi kao $\hat{y}$, tada se ona računa prema (1), gdje $\boldsymbol{\omega}^T$ predstavlja transponirani vektor težinskih parametara [25].

$$\hat{y} = \boldsymbol{\omega}^T \mathbf{x} = \sum_{i=1}^n \omega_i x_i = \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n \quad (1)$$

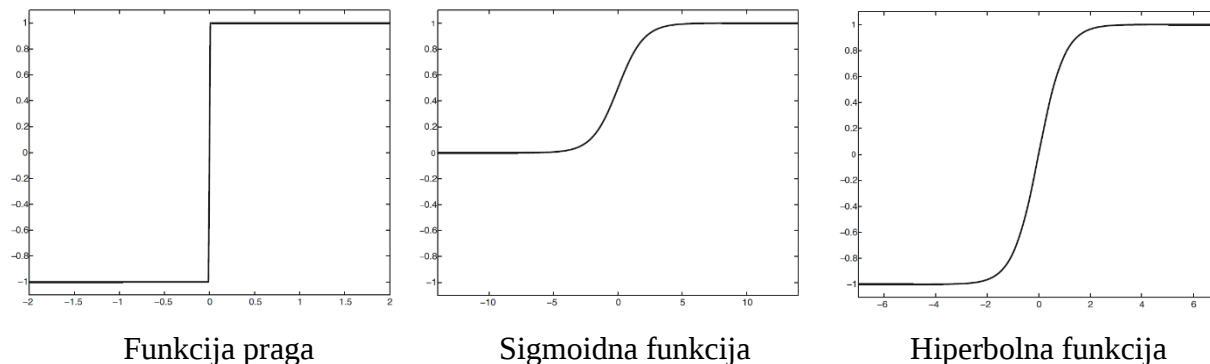
Ipak, češći je oblik linearne regresije (2) gdje se zbroju dodaje konstanta (eng. *bias*) koja predstavlja osnovnu vrijednost predviđanja ukoliko su sve ostale vrijednosti jednake nuli.

$$\hat{y} = \boldsymbol{\omega}^T \mathbf{x} + b \quad (2)$$

Nakon izračuna linearne kombinacije ulaznih varijabli, na istoj se primjenjuje aktivacijska funkcija. Ona imitira slanje električnih impulsa u biološkoj neuronskoj mreži, tj. opisuje pretvorbu unutarnjeg stanja neurona u izlazni signal. Ukoliko se aktivacijska funkcija označi kao Φ , a umnožak $(\boldsymbol{\omega}^T \mathbf{x})$ sa v , tada vrijedi izraz (3).

$$\hat{y} = \Phi(v) \quad (3)$$

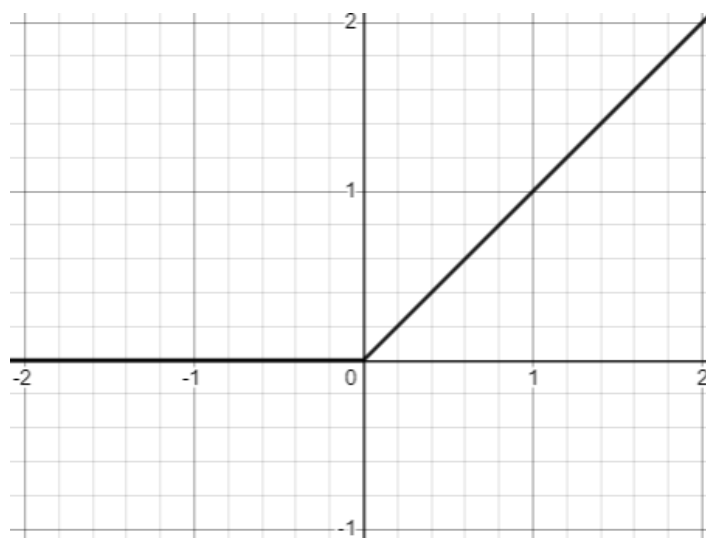
Klasične funkcije koje su se koristile u ranom razvoju neuronskih mreža bile su funkcija praga, sigmoidna te hiperbolna funkcija [Slika 5.]. Kod funkcije praga neuronu se dodjeljuju binarne vrijednosti, gdje 1 označava aktivnost, a 0 neaktivnost neurona. Sigmoidna funkcija daje vrijednosti u rasponu između 0 i 1, što je veoma korisno ukoliko se provode izračuni vjerojatnosti. Hiperbolična funkcija je oblikom veoma slična sigmoidnoj, no ona daje vrijednosti u rasponu od -1 do 1 [26].



Slika 5. Klasične aktivacijske funkcije [22]

Posljednjih godina, osim klasičnih funkcija, česta je uporaba ReLU (eng. *rectified linear activation function*) aktivacijske funkcije [Slika 6.] jer se njome postiže brža konvergencija modela u odnosu na druge aktivacijske funkcije. Ako se umnožak ($\omega^T \mathbf{x}$) označi sa v tada je ReLU aktivacijska funkcija definirana kao (4).

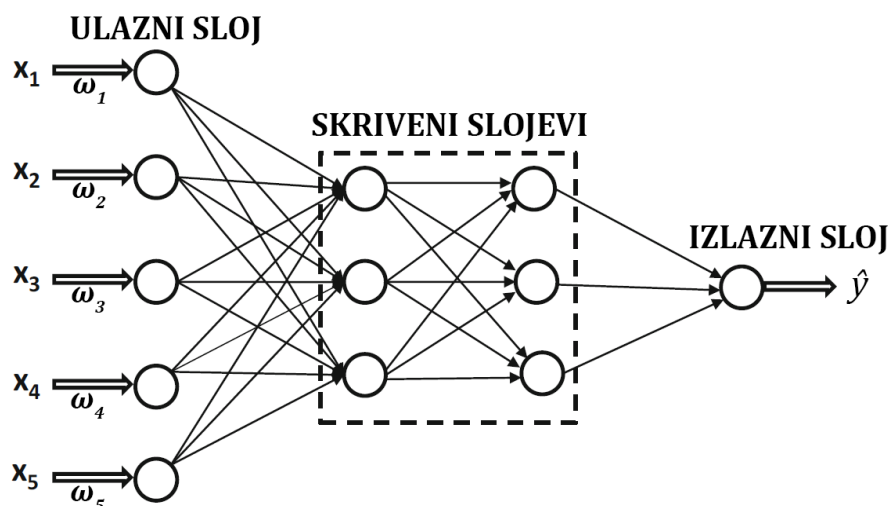
$$\Phi(v) = \max(0, v) = \begin{cases} v, & \text{ako } v > 0 \\ 0, & \text{inače} \end{cases} \quad (4)$$



Slika 6. ReLU aktivacijska funkcija

3.2.3.2. Struktura umjetnih neuronskih mreža

Strukturu mreže čini nekoliko slojeva. Ulazni sloj mreže sastoji se od neurona koji očitava ulazne vrijednosti sustava. Mreža obično ima i skrivene slojeve koji kao svoj ulaz upotrebljavaju izlaze drugih neurona. Konačno, izlazni sloj generira izlaz cijele mreže [24] [Slika 7.].



Slika 7. Struktura umjetne neuronske mreže

3.2.3.3. Hiperparametri umjetnih neuronskih mreža

Kako bi se postigao optimalan model umjetne neuronske mreže, najprije je potrebno podesiti hiperparametre mreže. To su parametri koji određuju strukturu mreže te reguliraju ponašanje algoritma učenja. O njima ovisi učenje mreže, ali se oni kroz samo učenje ne mijenjaju. Neki od hiperparametara su broj skrivenih slojeva mreže i neurona unutar skrivenog sloja, aktivacijska funkcija i funkcija gubitka, optimizacijski algoritam, stopa učenja, broj epoha te veličina serije (eng. *batch size*) [27].

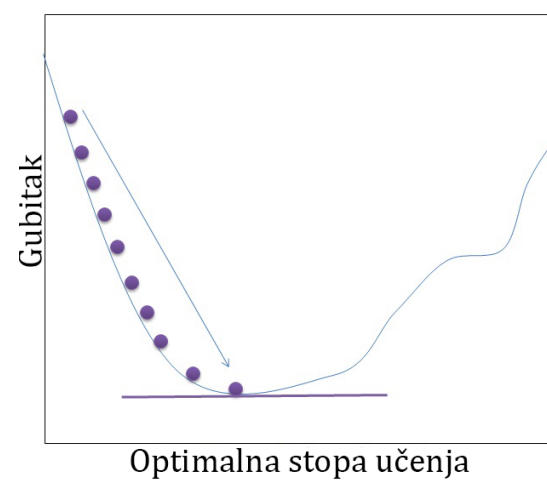
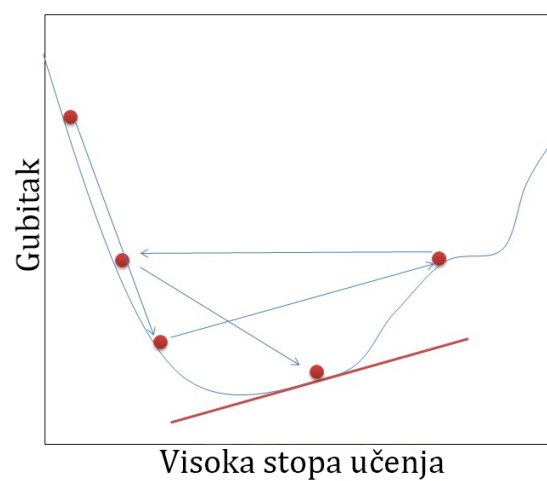
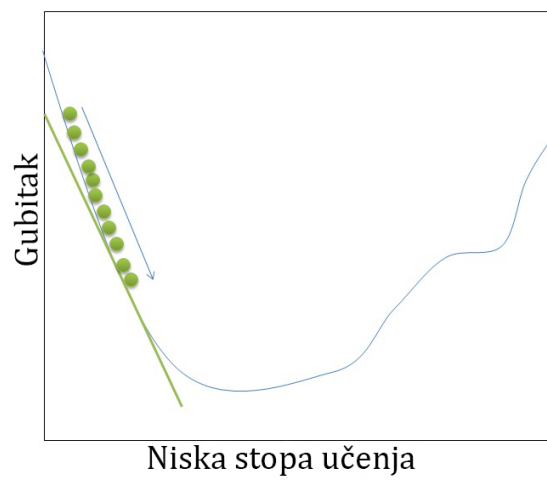
Kod neuronskih mreža najčešće je riječ o minimiziranju pogreške. Funkcija gubitka mjeri odstupanje izlaznog rezultata $\hat{y}$ u odnosu na stvarni rezultat y te je njome moguće regulirati težinske parametre kako bi se minimizirale vrijednosti funkcije gubitka. Funkcija unakrsne entropije (eng. *Cross-Entropy Loss*) svaku predviđenu vrijednost $\hat{y}$ uspoređuje sa stvarnom vrijednosti y te joj dodjeljuje određene vrijednosti na temelju udaljenosti od očekivane vrijednosti. Vrijednosti su logaritamske, 0.1-0.2 za male udaljenosti te 0.9-1 za velike udaljenosti [28]. Unakrsna entropija je funkcija koja ima ugrađenu *softmax* funkciju. Zadaća *softmax* funkcije je pretvaranje n -dimenzionalnog vektora u vektor realnih brojeva sa rasponom $[0,1]$ čiji je zbroj vrijednosti maksimalno 1. Njena svojstva čine je pogodnom za interpretaciju vjerojatnosti u klasifikacijskim zadacima.

Optimizacijski algoritmi koriste se za regulaciju atributa neuronske mreže kao što su težine i stopa učenja kako bi se smanjili gubici te postigla što veća točnost mreže. Neki od najčešće korištenih optimizacijskih algoritama su *Adadelta algoritam*, *Adagrad algoritam*, *Adam algoritam* te *stohastički gradijentni spust* (eng. *stochastic gradient descent* – SGD) [29].

Niska stopa učenja usporava proces učenja, ali konvergira prema minimumu funkcije gubitka. Visoka stopa učenja ubrzava učenje, no ukoliko je prevelika postoji mogućnost da algoritam učenja preskoči minimum funkcije gubitka ili učini samo učenje nestabilnim. Stopa učenja bira se u intervalu $[0, 1)$. Ovisnost funkcije gubitka o stopi učenja prikazana je na [Slika 8.].

Broj epoha određuje koliko će puta algoritam učenja proći kroz cijeli niz podataka za treniranje mreže. Jedna epoha označava jednu iteraciju kroz čitav skup podataka. Nedovoljan broj epoha će rezultirati nedostatkom istreniranosti mreže (eng. *underfitting*), što znači da mreža nije u stanju dovoljno minimizirati funkciju gubitka na skupu podataka za učenje. Ukoliko se postavi prevelik broj epoha može doći do prevelike istreniranosti mreže (eng. *overfitting*), pri čemu model može vrlo dobro predvidjeti skup podataka za učenje, ali ne može dovoljno dobro predvidjeti nove neviđene podatke.

Veličina serije određuje niz podataka nad kojim se provodi učenje prije nego što se ažuriraju akumulirane promjene težina. Najčešće se ukupan broj ulaznih podataka raspoređuje u manje serije kako bi se mreža mogla prilagoditi promjenama.



Slika 8. Ovisnost funkcije gubitka o stopi učenja [30]

3.3. Virtualni asistenti

Virtualnost kao informacijska i komunikacijska tehnologija postala je prisutna kao dio svakodnevnog privatnog i profesionalnog života ljudi. Široki raspon dobara i usluga proizvodi se, prodaje ili koristi virtualno [31]. Napretkom umjetne inteligencije razvijeni su i virtualni asistenti koji imaju sposobnost izvršavanja raznih zadataka prema uputama čovjeka. Neki od najpoznatijih virtualnih asistenata su *Cortana*, *Google Assistant*, *IBM Watson* te *Siri*.

Jedan od oblika virtualnih asistenata su *chatbotovi*. *Chatbotovi* su softverski sustavi temeljeni na tekstu koji komuniciraju s ljudskim korisnicima na njihovom jeziku i donose odluke na temelju unaprijed definiranih pravila [32]. Ukoliko je virtualni agent tj. *chatbot* prilagodljiv emocionalnom stanju korisnika, on se doživljava pouzdanijim i inteligentnijim. Pri tome se postavlja pitanje konteksta kao što je primjerice poznavanje osobnosti korisnika te okolnosti koje su mogle utjecati na njegovo emocionalno stanje.

3.4. Inteligentni emocionalni roboti

Inteligentni emocionalni roboti se sve više pojavljuju u okolini ljudi. Njihov oblik uglavnom se temelji na konstrukciji humanoidnih robota, alatima za prepoznavanje emocionalnih stanja i sustavu za interakciju [18]. Ukoliko se definiraju obrasci ponašanja kao prikladan odgovor na određeno emocionalno stanje te isti pridodaju konvencionalnim robotima, moguće je postizanje sustava koji je u mogućnosti prilagoditi svoje ponašanje.

Postoji nekoliko primjera gdje je takva interakcija uspješno ostvarena. Jedan od njih je robot koji ima sposobnost prepoznavanja „osnovnih“ ljudskih emocija pomoću vizualnih i slušnih senzora. Uporabom umjetne inteligencije su zatim definirana određena ponašanja robota kojima je cilj izazvati pozitivnu emocionalnu reakciju korisnika. Kada robot procijeni da je korisnik pod stresom, on pušta opuštajuću glazbu, generira opuštajuće mirise te se giba oko korisnika na način da ga ne ometa [33].

Još uvijek postoje mnogi nedostaci u sustavu emocionalnih robota kao što je izuzimanje informacija o okolini čovjeka koja također može utjecati na emocionalno izražavanje korisnika ili isključivanje dubljih kognitivnih analiza ponašanja korisnika.

4. POSTUPAK OSTVARIVANJA ROBOTSKJE INTERAKCIJE

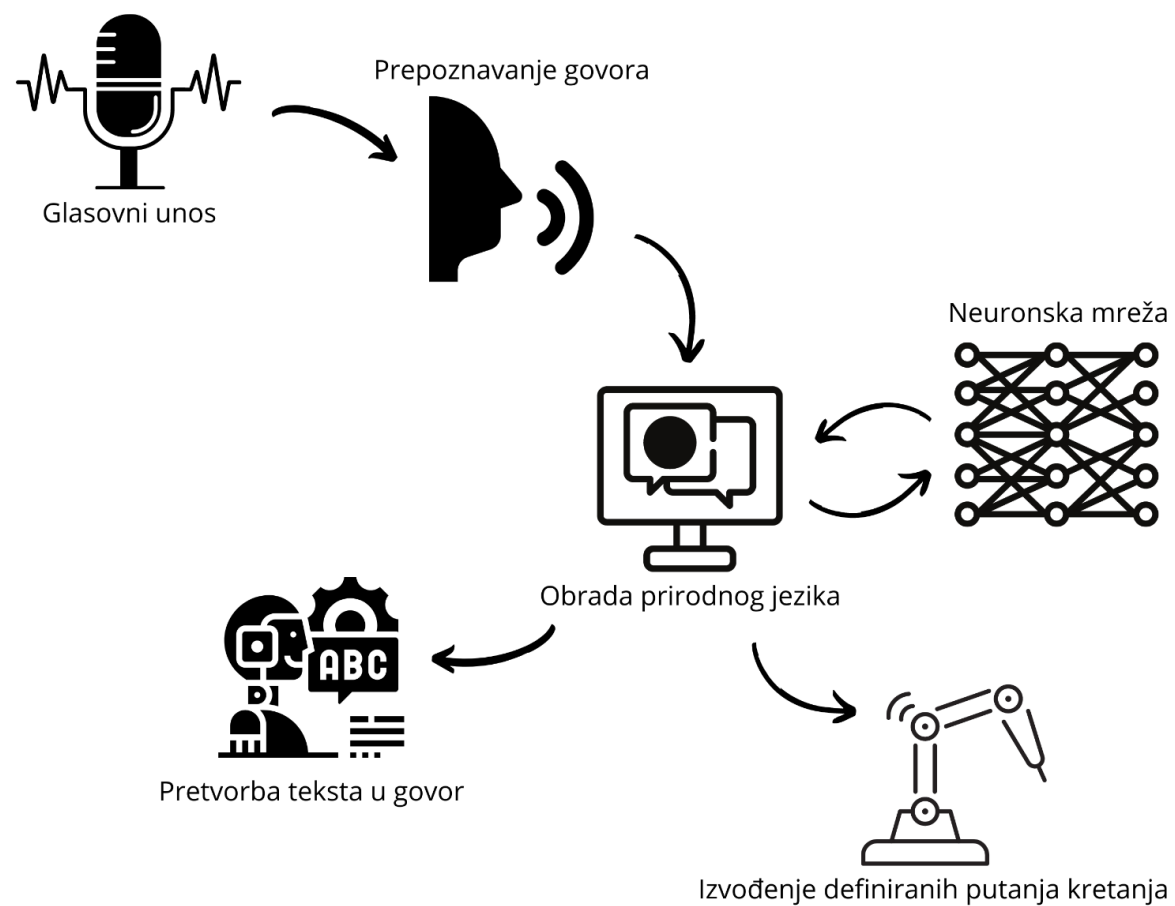
Sposobnost međusobne interakcije je najvažnija značajka društvenog života ljudi. Ona postoji kod gotovo svih oblika živih organizama. Kako je interakcija sastavni dio živih bića, najčešće se odvija prirodno i sama po sebi. Kada je riječ o strojevima, takvo što ipak nije trivijalan zadatak. U nastavku je razrađena ideja robotske interakcije koja uključuje i čovjeka, a glavni zadatak iste je prepoznavanje ljudskih emocija prema kontekstu rečenice.

4.1. Scenarij robotske interakcije

Početni dio rada temeljen je na definiranju scenarija robotske interakcije [Slika 9.] koji uključuje i čovjeka. Interakcija se odvija tako da čovjek izgovara rečenice, robot detektira glasovne unose putem računala, prepoznaje njihovo značenje pomoću umjetne neuronske mreže te na temelju njih izvodi određene radnje.

Svaka rečenica predstavlja jednu od osnovnih emocija prema podjeli u poglavlju 2.3 (tuga, sreća, ljutnja, iznenađenje, gađenje i strah) te je dodana skupina za neutralan stav. Na temelju definiranih rečenica [Tablica 1.] provodi se treniranje umjetne neuronske mreže zaslužne za prepoznavanje emocionalnih stanja korisnika. Nakon prepoznavanja određene emocije, računalo vraća nasumično odabran empatičan odgovor [Tablica 2.] te šalje informaciju robotu za izvođenje određene putanje gibanja. Odgovori robota kreirani su na način da pripadaju kategoriji razumijevanja ili motivacije. Kod razumijevanja robot pokazuje suosjećanje, dok je motivacija oblik reaktivne empatije koji može biti koristan za ublažavanje negativnih emocija kao što su ljutnja, strah ili gađenje. Sličan način rada predložen je i u literaturi [2] gdje su uz motivaciju i razumijevanje korištene još dvije kategorije, ublažavanje trenutne emocije i odvratanje pozornosti.

Umjesto interakcije čovjeka i robota, na jednak način moguća je komunikacija dvaju robota, gdje jedan robot „izgovara“ rečenice koje predstavljaju emotivna stanja, a drugi robot ih prepoznaje te postupa prema prethodno opisanome scenariju.



Slika 9. Shematski prikaz scenarija robotske interakcije

Tablica 1. Prikaz kontekstualnih rečenica koje predstavljaju određena emocionalna stanja

Rečenica koju izgovara korisnik	Emocija
<i>I feel like crying. Plače mi se.</i> <i>I am stressed. Pod stresom sam.</i> <i>I lost my pet. Izgubio/la sam ljubimca.</i> <i>I failed my exam yesterday. Pao/la sam ispit jučer.</i>	SADNESS / TUGA
<i>I am going on a vacation! Idem na odmor!</i> <i>I got a promotion today! Danas sam dobio/la unapređenje!</i> <i>I heard my favorite song on the radio! Čuo/la sam svoju najdražu pjesmu na radiju!</i> <i>Today is a beautiful day. Danas je divan dan.</i>	HAPPINESS / SREĆA
<i>I am late because of traffic. Kasnim zbog prometa.</i> <i>I spilled my coffee! Prosuo/la sam kavu!</i> <i>My flight is delayed. Kasni mi let.</i> <i>I can't find my phone! Ne mogu pronaći svoj telefon!</i>	ANGER / LJUTNJA
<i>I didn't expect to see you here! Nisam očekivao/la da ću te vidjeti ovdje!</i> <i>The sky is so clear! Nebo je tako bistro!</i> <i>My best friends threw me a surprise birthday party! Moji najbolji prijatelji su mi priredili rođendansku zabavu iznenađenja!</i>	SURPRISE / IZNENAĐENJE

<i>That smell is terrible. Taj smrad je užasan.</i> <i>I just can't stand blood. Ne mogu podnijeti krv.</i> <i>I don't like ginger! Ne volim đumbir!</i>	DISGUST / GAĐENJE
<i>I saw a spider in my room! Vidio/vidjela sam pauka u sobi!</i> <i>That horror movie sent shivers down my spine. Naježio/la sam se nakon tog filma strave.</i> <i>I am scared of heights. Bojim se visina.</i> <i>I had a terrifying experience last week. Prošli tjedan sam doživio/doživjela zastrašujuće iskustvo.</i>	FEAR / STRAH
<i>The sky is blue. Nebo je plavo.</i> <i>I am student. Ja sam student.</i> <i>If it rains we will get wet. Ako padne kiša, mi ćemo biti mokri.</i>	NEUTRAL / NEUTRALAN STAV

Tablica 2. Prikaz mogućih empatičnih odgovora računala/robota

Emocija	Empatični odgovori robota	Kategorija
SADNESS / TUGA	<i>You seem disappointed. Djeluješ razočarano.</i> <i>Oh, it's a pity! Baš tužno!</i>	RAZUMIJEVANJE
	<i>Tomorrow is a new day! Sutra je novi dan!</i> <i>Things will get better! Sve će biti bolje!</i>	MOTIVACIJA
HAPPINESS / SREĆA	<i>I am so glad for you! Tako mi je drago zbog tebe!</i> <i>I am happy you are happy. Sretan/na sam zbog tebe.</i>	RAZUMIJEVANJE
	<i>Keep up the good work! Samo nastavi tako!</i> <i>Smile looks good on you, you should wear it more often! Pristaje ti osmijeh, trebao/la bi se češće smijati!</i>	MOTIVACIJA
ANGER / LJUTNJA	<i>I totally understand your point. U potpunosti te razumijem.</i> <i>It is annoying me too! To i mene nervira.</i>	RAZUMIJEVANJE
	<i>If you can't do anything about it, let it go. Ako ne možeš učiniti ništa da to promijeniš, nemoj se zamarati.</i> <i>Everything will be fine. Sve će biti u redu.</i>	MOTIVACIJA

SURPRISE / IZNENAĐENJE	<i>Wow, I can't believe! Ajme, ne mogu vjerovati!</i> <i>I am shocked! Šokiran/a sam!</i>	RAZUMIJEVANJE
	<i>Life is full of surprises. Život je pun iznenađenja.</i> <i>The best things happen unexpectedly. Najbolje stvari se događaju neočekivano.</i>	MOTIVACIJA
DISGUST / GAĐENJE	<i>I don't like it either. To se ni meni ne sviđa.</i> <i>That is nauseating. To mi izaziva mučnine.</i>	RAZUMIJEVANJE
	<i>Close your eyes and think about something nice. Zatvori oči i misli na nešto lijepo.</i> <i>Try not to think about it. Pokušaj ne misliti na to.</i>	MOTIVACIJA
FEAR / STRAH	<i>I am also disturbed by that. I mene to uznemirava.</i> <i>I also get scared sometimes, that's ok. I ja se ponekada uplašim, to je u redu.</i>	RAZUMIJEVANJE
	<i>Take a deep breath and try to relax. Pokušaj duboko udahnuti i opustiti se.</i> <i>I am always here if you need to talk. Uvijek sam ovdje ukoliko želiš razgovarati.</i>	MOTIVACIJA
NEUTRAL / NEUTRALAN STAV	<i>That is true. To je istina.</i> <i>That is interesting. Veoma zanimljivo.</i> <i>Thank you for that information. Hvala ti na toj informaciji.</i>	

4.2. Integracija komponenti sustava

Za realizaciju zamišljenog scenarija, odabrano je sučelje za integraciju komponenti sustava pod nazivom *PyCharm*. *PyCharm* je integrirano razvojno okruženje (eng. *integrated development environment*, skraćeno IDE) koje se koristi u računalnom programiranju, ponajviše za programski jezik *Python*. Radi se o objektno orijentiranom programskom jeziku, visoke razine i opće namjene (korištena verzija 3.6). Za njega su definirani brojni moduli koji omogućuju lakši i brži razvoj programskog kôda, a u konačnici i programa. Module i pakete koji dolaze sa standardnom instalacijom *Python*-a nazivamo standardnom bibliotekom [34]. U radu su bile potrebne dodatne funkcionalnosti te su stoga, osim standardnih, korišteni i dodatni paketi koji će biti opisani u nastavku. Većina paketa nije prilagođena za rad na hrvatskome jeziku te su stoga rečenice prevedene na engleski jezik.

4.2.1. Programski paket za prepoznavanje govora

Za prepoznavanje glasovnih unosa korisnika potrebni su paketi *PyAudio* (verzija 0.2.11) i *SpeechRecognition* (verzija 3.8.1). *PyAudio* omogućava povezivanje mikrofona sa *Python*-om te je on uvjet za korištenje funkcionalnosti paketa *SpeechRecognition*. Pomoću paketa *SpeechRecognition* moguća je uporaba raznih korisnih značajki kao što je prepoznavanje glasovnih unosa s mikrofona, transkripcija glasovnih zapisa, spremanje glasovnih zapisa u obliku datoteke te kalibracija razine zapažanja zvuka u odnosu na ambijentalnu buku [35].

4.2.2. Programski paket za pretvorbu teksta u govor

Za ostvarivanje komunikacije robota s korisnikom, uz uporabu prirodnog jezika, potrebna je programska podrška koja će pretvarati tekstualne zapise rečenica u govor. Paket *Text to Speech (TTS)*, kraće *pyttsx3* (verzija 2.90) [36], zaslužan je za davanje glasovnih sposobnosti robotu. Unutar njega je moguće podešavanje karakteristika jačine, brzine i boje glasa.

4.2.3. Programski paket za obradu prirodnog jezika

Natural Language Toolkit (NLTK) je vodeća platforma za izradu *Python* programa koji su temeljeni na radu s podacima koji proizlaze iz ljudskog govora. Ona sadrži brojne alate za računalnu obradu prirodnog jezika kao što su tokenizacija, korjenovanje, klasifikacija, označavanje i raščlanjivanje riječi [37]. Ovaj paket (korištena verzija 3.6.2) je veoma bitan jer je zaslužan za obradu rečenica na kojima će biti provedeno strojno učenje.

4.2.4. Programski paket za duboko učenje

Sljedeći paket koji je potreban je *PyTorch* (verzija 1.8.1+cpu). On sadrži alate potrebne za jednostavnu implementaciju dubokog učenja [Tablica 3.], uz akceleraciju izračuna uporabom središnje procesorske jedinice (CPU, eng. *central processing unit*) i grafičke procesorske jedinice (GPU, eng. *graphics processing unit*) [38]. *PyTorch* je baziran na radu sa tenzorima. Tenzori su strukture podataka koje se koriste u linearnoj algebri, a predstavljeni su pomoću n-dimenzionalnih nizova.

Tablica 3. Moduli paketa *PyTorch* potrebni za izradu umjetne neuronske mreže

Ime modula	Opis modula
<i>torch</i>	modul koji sadrži strukture podataka u obliku višedimenzionalnih tenzora na kojima se primjenjuju matematičke operacije
<i>torch.nn</i>	modul zaslužan za stvaranje i treniranje neuronskih mreža
<i>torch.utils</i>	modul koji sadrži alate za učitavanje i obradu podataka
<i>torch.optim</i>	modul sa algoritmima za optimizaciju neuronske mreže

Uporaba *PyTorch*-a je intuitivna jer se modeli i slojevi umjetne neuronske mreže, kao i ostale komponente poput optimizacijskih algoritama i funkcija gubitaka, definiraju kao klase u *Python*-u.

4.2.5. Programski paket za obradu homogenih višedimenzionalnih nizova

NumPy (verzija 1.19.5) je temeljni paket za znanstveno računanje u *Python*-u. On obuhvaća funkcije poput definiranja višedimenzionalnih nizova te izvođenja operacija na nizovima (matematičke i logičke manipulacije, sortiranje, diskretne Fourierove transformacije, linearna algebra, statističke metode, itd.). Osnova ovog paketa je *ndarray* objekt pomoću kojega se odvija definiranje n-dimenzionalnih nizova [39].

4.2.6. Programski paket za povezivanje robota sa računalom

Za povezivanje robota sa računalom potreban je paket *urx* (verzija 0.11.0), namijenjen za kontrolu robotā čiji je proizvođač tvrtka *Universal Robots*. U poglavlju 4.4.1 će biti detaljnije opisane tehničke karakteristike robota korištenog u radu.

4.3. Izrada umjetne neuronske mreže

4.3.1. Definiranje i obrada ulaznih podataka

Prvi korak je izrada datoteke koja će sadržavati bazu ulaznih rečenica, oznake emocija i empatične odgovore. Jedan od načina za to je uporaba JSON datoteke. JSON (*JavaScript Object Notation*) je tekstualni format dizajniran za čitljiv i razumljiv prijenos strukture podataka između ljudi i strojeva [40]. Ova datoteka predstavlja skup ulaznih podataka za strojno učenje, a njen puni sadržaj moguće je pronaći na kraju rada u poglavlju Prilozi. Ulazne rečenice koje predstavljaju emocionalna stanja korisnika nalaze se u skupini označenoj *patterns*, emocije u skupini *tag*, a empatični odgovori u skupini *responses*.

```
1  {  
2    "emotions": [  
3      {  
4        "tag": "sadness",  
5        "patterns": [  
6          "I feel like crying",  
7          "I feel stressed",  
8          "I lost my pet",  
9          "I failed my exam yesterday"  
10     ],  
11     "responses": [  
12       "Tomorrow is a new day",  
13       "Things will get better",  
14       "You seem disappointed",  
15       "Oh, it's a pity"  
16     ]  
17   },  
18 ]  
19 }
```

Slika 10. Primjer JSON datoteke sa definiranim rečenicama

Poznata krilatica u svijetu strojnog učenja, *garbage in, garbage out*, govori kako će konačni rezultati strojnog učenja biti loši ukoliko su ulazni podaci loše označeni, netočni ili nepotpuni. Kako bi se to izbjeglo, potrebna je prethodna obrada podataka. Budući da se u ovome slučaju radi o rečenicama tj. tekstualnim podacima, za pripremu podataka će biti korišteni alati za obradu prirodnog jezika sadržani u paketu *Natural Language Toolkit* opisanome u poglavlju 4.2.3.

4.3.1.1. Tokenizacija

Podaci definirani u [Tablica 1.] predstavljaju ulaze za treniranje neuronske mreže. Kako bi se mogli koristiti za treniranje, potrebno ih je najprije transformirati na način da se svaka rečenica rastavi na njene sastavne jedinice. Takav postupak naziva se tokenizacija, a provodi se korištenjem naredbe **nlTK.word_tokenize()**. Primjer tokenizirane rečenice izgleda ovako:

I | got | a | promotion | today | . |

Nakon tokenizacije, sve riječi se stavljaju u jedan vektor pod nazivom **all_words[]** na kojemu će se zatim provesti ostali koraci obrade teksta, poput uklanjanja interpunkcijskih znakova, normalizacije veličine slova te korjenovanja riječi.

4.3.1.2. Normalizacija veličine slova

Čovjek zna da riječi napisane kao *Knjiga* i *knjiga* označavaju istu riječ. Računalo međutim neće prepoznati da se radi o istoj riječi te je stoga potrebno normalizirati veličinu slova. Normalizacija se provodi pomoću jednostavne naredbe **word.lower()** te nakon tog postupka sve riječi u vektoru **all_words[]** započinju sa malim početnim slovom.

4.3.1.3. Eliminacija stop-riječi i interpunkcijskih znakova

Stop-riječi su često korištene riječi poput veznika i prijedloga koje nisu značajne za sadržaj teksta [41]. One se iz tog razloga izostavljaju jer ne bi pridonijele stvaranju kvalitetne neuronske mreže. Osim stop-riječi, izostavljaju se i interpunkcijski znakovi koji također nisu nositelji značenja. Kako bi računalo znalo prepoznati stop-riječi i interpunkcijske znakove, potrebno je definirati vektor sa svime što se želi eliminirati.

4.3.1.4. Korjenovanje riječi

Riječ se sastoji od najmanjih jedinica izraza koje imaju svoje značenje – morfema. Korijen je najmanji zajednički dio svih riječi koje pripadaju istoj tvorbenoj porodici (npr. korijen *slik-* nalazi se u riječima *slikati*, *oslikati*, *slikar*, *slikarstvo*, *slikovnica* itd.) i uvijek se sastoji od jednoga morfema. Osnova je dio riječi koji nosi leksičko značenje [42]. Kako bi se zadržala samo osnova svake riječi u vektoru **all_words[]**, provodi se korjenovanje riječi (eng. *stemming*) pomoću naredbe **stemmer.stem()**. Postoje različiti algoritmi za korjenovanje riječi, oni koji agresivnije svode riječ na najmanju moguću osnovu te blaži koji uklanjanju samo nastavak broja ili padeža. Za potrebe rada odabran je algoritam pod nazivom *PorterStemmer* koji riječi svodi na njihovu morfološku osnovu.

4.3.1.5. Pretvorba tekstualnih varijabli u numeričke varijable

Algoritmi strojnog učenja ne mogu raditi izravno s tekstom te je stoga tekst potrebno pretvoriti u brojeve, odnosno vektor koji sadrži numeričke vrijednosti. Popularna metoda za izdvajanje značajki iz tekstualnih podataka naziva se *vreća riječi* (eng. *bag-of-words*). Ona obuhvaća dvije stvari, a to su rječnik poznatih riječi i mjeru prisutnosti riječi. Metoda se zove *vreća riječi* jer odbacuje sve informacije o strukturi riječi [43].

Postupak pretvorbe se provodi na način da se svaka riječ iz skupine *patterns* uspoređuje sa riječima iz vektora **all_words[]** te se stvara novi vektor sa numeričkim vrijednostima na temelju podudaranja riječi. Najjednostavnija metoda bodovanja podudarnosti je uporabom logičkih vrijednosti, gdje 1 označava podudaranje, a 0 nepodudaranje riječi. Budući da je u tekstu pronađeno 76 mogućih riječi, to će biti duljina svakog vektora **bag_of_words[]**.

Vizualni primjer opisanog postupka prikazan je u nastavku za rečenicu *Today is a beautiful day* [Tablica 4.]. U vektoru **all_words[]** sadržane su sve riječi prikupljene i obrađene prema koracima opisanim u poglavljima 4.3.1.1 - 4.3.1.4, dok se vektor **bag_of_words[]** sastoji od logičkih vrijednosti na temelju podudarnosti riječi iz rečenice sa vektorom **all_words[]**.

Tablica 4. Vizualni prikaz stvaranja modela *bag-of-words*

	all_words[]				bag_of_words[]
<i>Today is a beautiful day. →</i>	'beauti'	'becaus'	'best'	'birthday'	[1 0 0 0
	'blood'	'blue'	'ca'	'clear'	0 0 0 0
	'coffe'	'cri'	'day'	'delay'	0 0 1 0
	'did'	'do'	'down'	'exam'	0 0 0 0
	'expect'	'experi'	'fail'	'favorit'	0 0 0 0
	'feel'	'find'	'flight'	'friend'	0 0 0 0
	'get'	'ginger'	'go'	'got'	0 0 0 0
	'had'	'heard'	'height'	'here'	0 0 0 0
	'horror'	'just'	'last'	'late'	0 0 0 0
	'like'	'lost'	'me'	'movi'	0 0 0 0
	'n't	'parti'	'pet'	'phone'	0 0 0 0
	'promot'	'radio'	'rain'	'room'	0 0 0 0
	'saw'	'scare'	'see'	'sent'	0 0 0 0
	'shiver'	'sky'	'smell'	'song'	0 0 0 0
	'spider'	'spill'	'spine'	'stand'	0 0 0 0
	'stress'	'student'	'surpris'	'terribl'	0 0 0 0
	'terrifi'	'that'	'threw'	'today'	0 0 0 1
	'traffic'	'vacat'	'we'	'week'	0 0 0 0
	'wet'	'will'	'yesterday'	'you'	0 0 0 0]

4.3.2. Izrada modela i treniranje neuronske mreže

4.3.2.1. Model umjetne neuronske mreže

Programski paket *PyTorch* omogućuje izradu umjetne neuronske mreže na način da se u *Python*-u stvara nova klasa te se unutar nje definira funkcija koja će predstavljati slojeve neuronske mreže [Slika 11.]. Odabrana je izrada povratne neuronske mreže koja se sastoji od tri potpuno povezana linearna sloja, od kojih su dva skrivena. Potpuno povezani sloj je sloj kod kojeg je svaki neuron povezan sa svim neuronima iz prethodnog sloja. Za aktivacijsku funkciju odabrana je ReLU funkcija.

```
1  import torch.nn as nn
2
3
4  # Izrada modela umjetne neuronske mreže
5  class NeuralNet(nn.Module):
6      def __init__(self, input_size, hidden_size, num_classes):
7          super(NeuralNet, self).__init__()
8          self.l1 = nn.Linear(input_size, hidden_size)
9          self.l2 = nn.Linear(hidden_size, hidden_size)
10         self.l3 = nn.Linear(hidden_size, num_classes)
11         self.relu = nn.ReLU()
12
13     def forward(self, x):
14         out = self.l1(x)
15         out = self.relu(out)
16         out = self.l2(out)
17         out = self.relu(out)
18         out = self.l3(out)
19         return out
20
```

Slika 11. Model umjetne neuronske mreže

4.3.2.2. Treniranje umjetne neuronske mreže

Za treniranje mreže najprije je potrebno postaviti skup ulaznih podataka koji je pripremljen u poglavlju 4.3.1. Postavljanje skupa podataka za neuronsku mrežu provodi se pomoću dva alata koja proizlaze iz *PyTorch*-a, a pozivaju se pomoću naredbi **torch.utils.data.Dataset** i **torch.utils.data.DataLoader**. **Dataset** pohranjuje primjere rečenica s odgovarajućim oznakama emocija, dok **DataLoader** omogućava iteraciju unutar skupa podataka.

Nakon definiranja skupa ulaznih podataka, potrebno je uvesti model neuronske mreže u datoteku za treniranje mreže te postaviti njene hiperparametre [Tablica 5.].

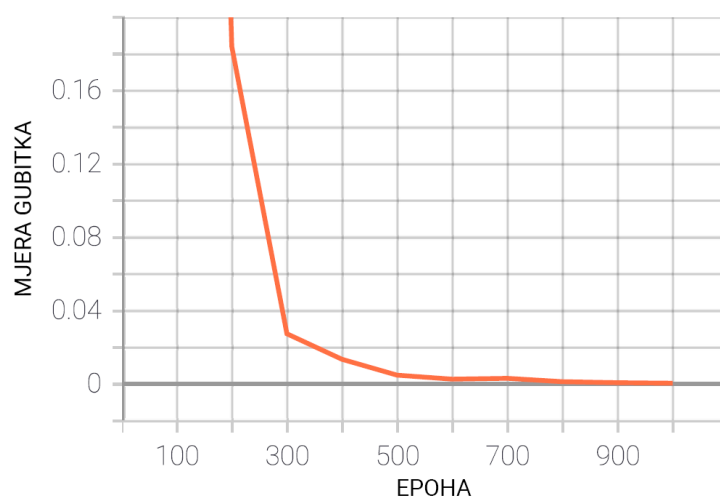
Tablica 5. Hiperparametri umjetne neuronske mreže

Naziv hiperparametra	Numerička vrijednost
Broj epoha (<i>num_epochs</i>)	1000
Stopa učenja (<i>learning_rate</i>)	0,001
Veličina serije (<i>batch_size</i>)	9
Broj neurona u skrivenom sloju (<i>hidden_size</i>)	9
Broj neurona u ulaznom sloju (<i>input_size</i>)	76 (ukupan broj riječi)
Broj neurona u izlaznom sloju (<i>output_size</i>)	7 (broj emocija)

Kako bi se ocijenila uspješnost modela umjetne neuronske mreže može se računati točnost ili gubitak modela. U ovome slučaju odabrana je mjera gubitka po epohi učenja, a za funkciju gubitka odabrana je unakrsna entropija (eng. *Cross Entropy Loss*) koja se poziva pomoću naredbe **nn.CrossEntropyLoss()**.

Sljedeći korak je odabir optimizacijskog algoritma te definiranje stope učenja koja će se koristiti u algoritmu. Za neuronsku mrežu u ovome radu odabran je Adam algoritam (eng. *Adaptive Moment Estimation*) jer empirijski rezultati pokazuju da on daje veoma dobre rezultate u usporedbi sa drugim stohastičkim metodama [44]. On se definira pomoću naredbe **torch.optim.Adam()**.

Nakon što su definirani svi parametri prema prethodno opisanome postupku, pokreće se treniranje mreže. Nakon tisuću epoha vrijednost funkcije gubitka jednaka je 0.0006 što ukazuje na to da je mreža istrenirana sa veoma malim gubitkom. U nastavku se nalazi grafički prikaz mjere gubitka po epohi učenja [Slika 12.].



Slika 12. Grafički prikaz mjere gubitka po epohi učenja

Kako bi se naučena mreža mogla koristiti za interakciju korisnika i računala, istrenirani podaci se spremaju pomoću naredbe **torch.save()** te se kasnije mogu pozivati pomoću naredbe **torch.load()**.

4.4. Programska podrška za ostvarivanje robotske interakcije

4.4.1. Odabir robota

Posljednji korak koji je potreban za ostvarivanje scenarija robotske interakcije je odabir robota. U radu je odabran UR5 robot [Slika 13.], danskog proizvođača *Universal Robots*.



Slika 13. Robot UR5 [45]

Universal Robots je proizvođač manjih fleksibilnih kolaborativnih robota sa sjedištem u Danskoj. Tvrtka je osnovana 2005. godine, a prvi robot, pod nazivom *UR5*, izašao je na tržište 2008. godine. On se smatra jednim od najznačajnijih tehnoloških dostignuća u tome desetljeću zbog svoje prilagodljivosti korisniku te intuitivnog korisničkog sučelja [46]. Sažete tehničke karakteristike *UR5* robota moguće je pronaći u nastavku [Tablica 6.].

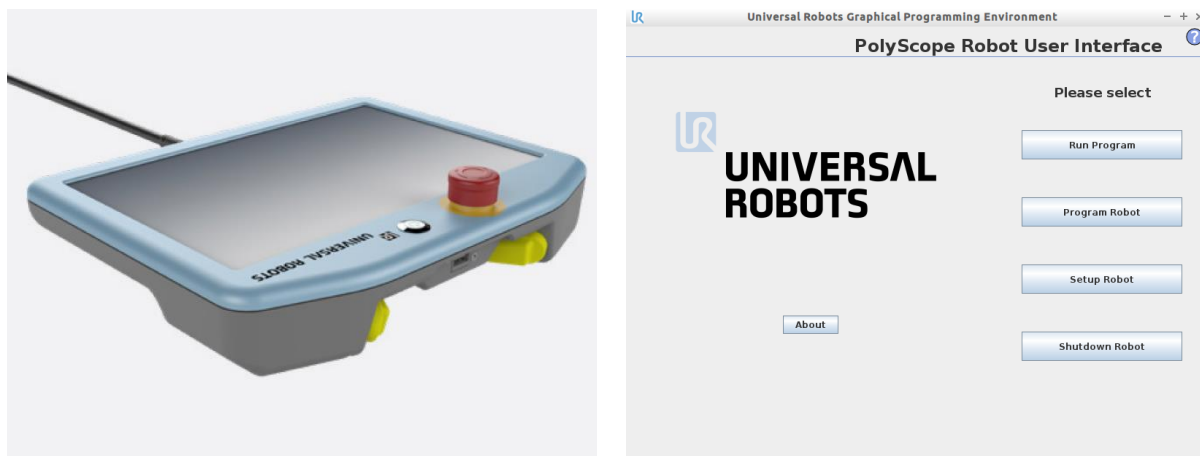
Tablica 6. Tehničke karakteristike robota *UR5* [45]

UR5e

SPECIFIKACIJE

Nosivost	5 kg (11 lbs)
Doseg	850 mm (33,5 in)
Stupnjevi slobode gibanja	6 stupnjeva slobode gibanja
Težina robota	20,6 kg (45,4 lbs)
Promjer baze	Φ 149 mm
Program	Grafičko sučelje Polyscope na privjesku za učenje
Ponovljivost prema ISO 9283	± 0,03 mm

Standardno upravljanje *UR5* robotom odvija se pomoću privjeska za učenje koji ima ugrađeno grafičko sučelje *PolyScope* [Slika 14.].



Slika 14. Privjesak za učenje sa grafičkim sučeljem *Polyscope* [45]

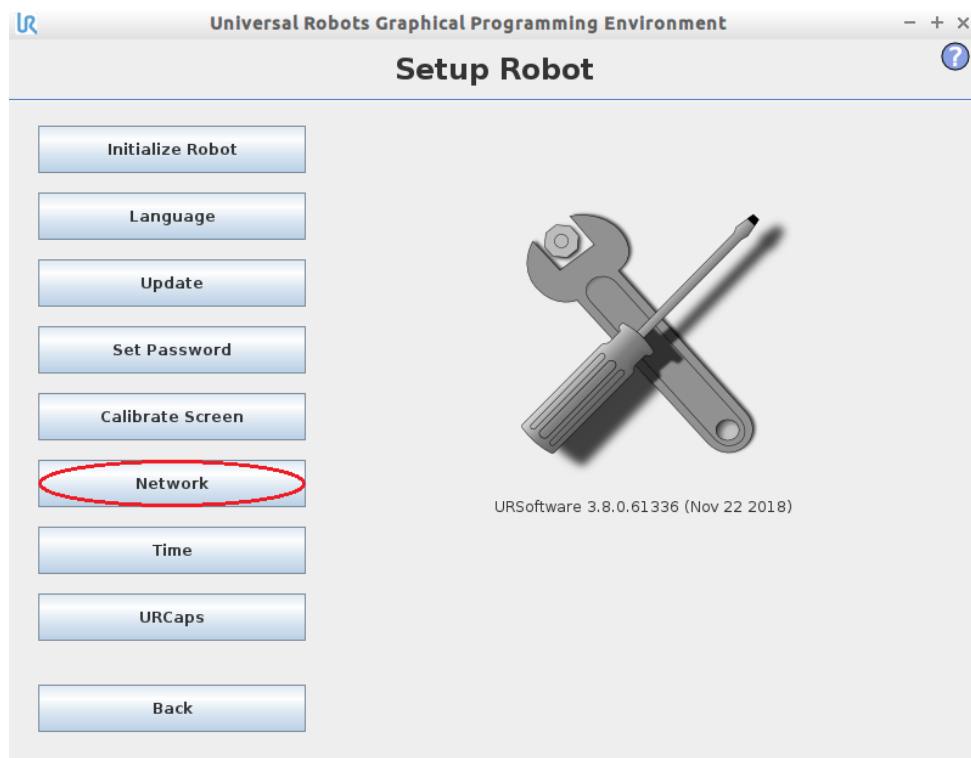
4.4.2. Povezivanje robota i računala

Učinkovita komunikacija ovisi o uporabi zajedničkog jezika. Kako ova tvrdnja vrijedi za ljude, tako vrijedi i za računala. Kada se definira skup uobičajenih obrazaca ponašanja sa zajedničkim jezikom, tada se radi o protokolu. TCP/IP (eng. *Transmission Control Protocol/Internet Protocol*) je naziv za skup protokola koji implementiraju internetsku arhitekturu te omogućavaju međusobnu komunikaciju računala, pametnih telefona i ostalih raznih uređaja, od različitih proizvođača i potpuno različitih softvera [47]. Navedeni protokol moguće je koristiti i za povezivanje robota sa računalom. On funkcionira na principu korisnik-poslužitelj (eng. *client-server*), gdje robot predstavlja korisnika, a računalo poslužitelja usluga.

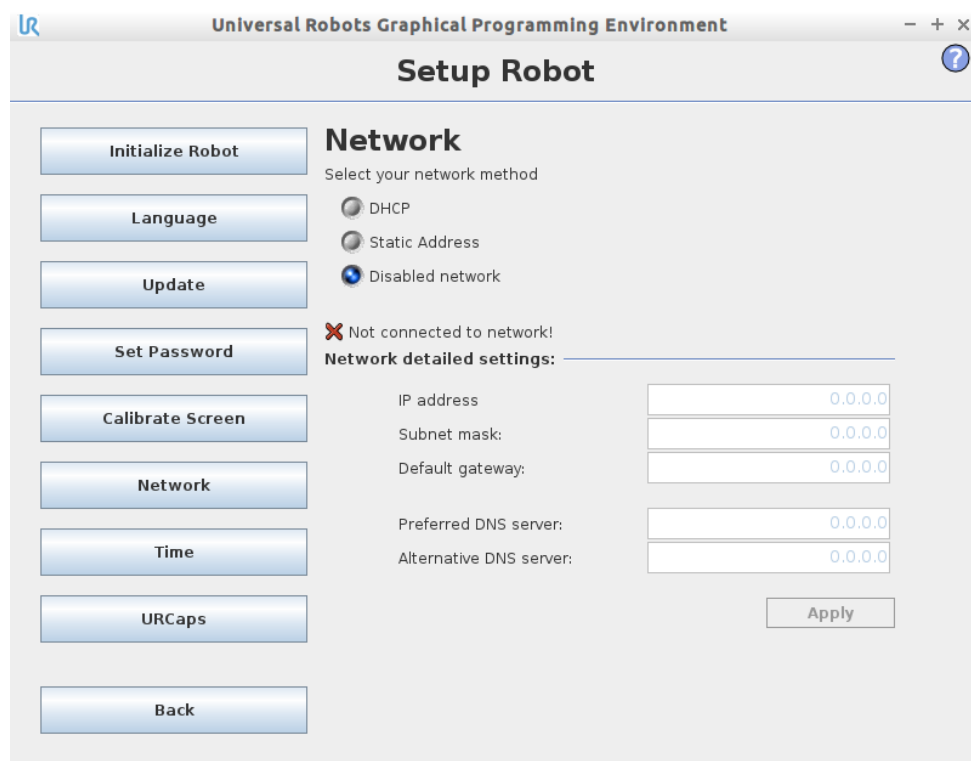
Iako se standardno upravljanje UR5 robotom provodi pomoću privjeska za učenje, moguće je i upravljanje robotom pomoću opisanoga protokola. UR5 za izvođenje osnovnih funkcija ne zahtijeva povezanost sa internetom te su stoga mrežne postavke prema zadanim postavkama onemogućene. Kako bi se omogućila mrežna povezanost UR5 robota, potrebno je podesiti postavke unutar programskog sučelja *Polyscope*.

Na početnom zaslonu sučelja odabire se opcija *Setup Robot* [Slika 15.] nakon čega se otvara izbornik *Setup Network* za podešavanje mrežne komunikacije robota. Unutar izbornika *Setup Network* [Slika 16.] moguće je odabrati metodu spajanja na mrežu. Ponuđene su dvije metode, DHCP protokol za dinamičku dodjelu IP adrese (eng. *Dynamic Host Configuration Protocol*) ili definiranje statičke IP adrese. Obje metode označavaju spajanje uređaja preko IP adrese. IP adresa je jedinstveni broj koji se dodjeljuje svakome uređaju na TCP/IP mreži, a služi za međusobnu identifikaciju uređaja.

Za ostvarivanje komunikacije između robota i računala, korištenjem programskog jezika *Python* i programskog paketa *urx*, koristi se naredba **`urx.Robot()`**, unutar koje je potrebno definirati IP adresu robota. Nakon što se robot spoji sa računalom, omogućeno je definiranje putanja kretanja robota te njihovo izvođenje.



Slika 15. Izbornik za podešavanje postavki robota



Slika 16. Izbornik za podešavanje mrežne komunikacije robota

4.4.3. Definiranje i izvođenje robotskih putanja kretanja

Nakon odabira robota te spajanja na mrežu, moguće je definirati putanje kretanja koje će pratiti empatične odgovore robota u interakciji. Putanje su inspirirane ljudskim pokretima te su zamišljene tako da oponašaju prepoznatu emociju [Slika 17.]. U nastavku su opisane tipične ljudske geste za iskazivanje različitih emocija.



Slika 17. Vizualni prikaz ljudskih gesti pri različitim emocijama [48]

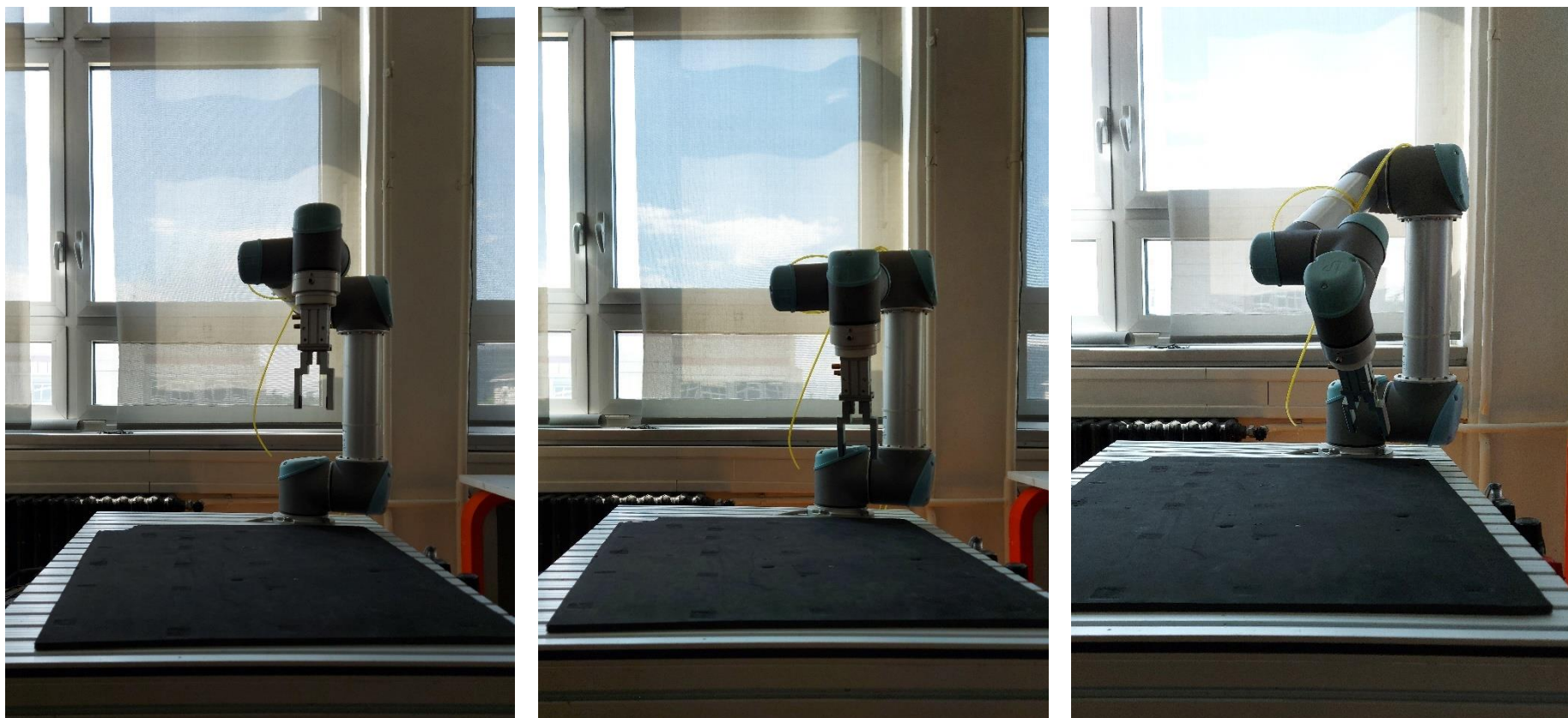
Ljudski pokreti prilikom izražavanja emocije tuge su usporeni, čovjek se povlači od okoline te ponekada ne želi prikazati svoje stanje prema okolini. Prilikom izražavanja sreće ljudski pokreti su brži, često popraćeni dizanjem ruku u zrak i poskakivanjem. Ukoliko se radi o ljutnji, pokreti će biti energični i oštri.

Kada je čovjek preplašen, ponaša se pomalo izgubljeno i pokušava se udaljiti od onoga što je uzrokovalo strah. Izražavanje gađenja slično je po pokušaju udaljavanja od uzroka, no gađenje je izraženo sa mnogo više ekspresija kao što je odmahivanje rukom ili zakretanje glave. Iznenađenje se može prikazati podizanjem ruku, odmicanjem unazad te stavljanjem ruke na usta.

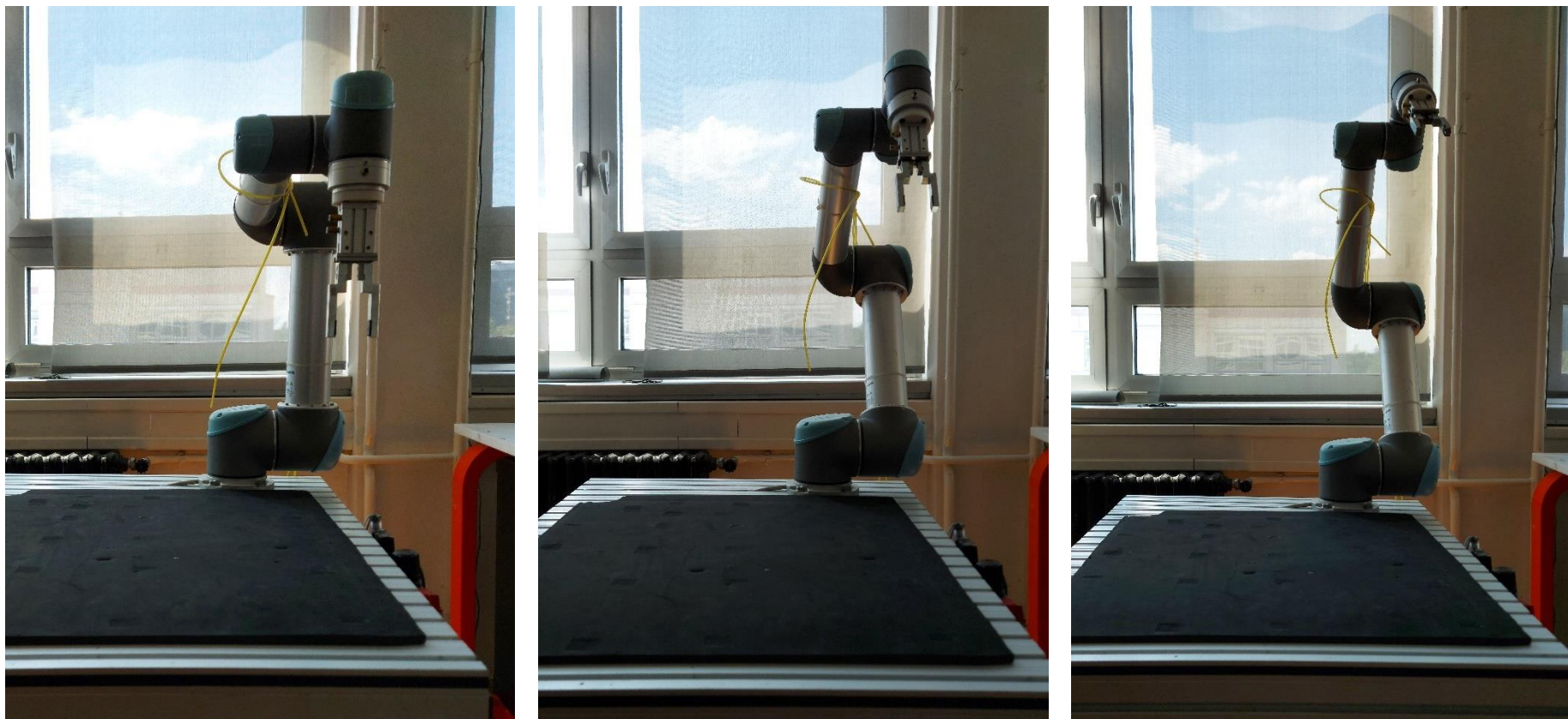
Opisani načini iskazivanja emocija korišteni su za slično izvođenje robotskih putanja prikazanih u nastavku [Slika 18.] – [Slika 24.].



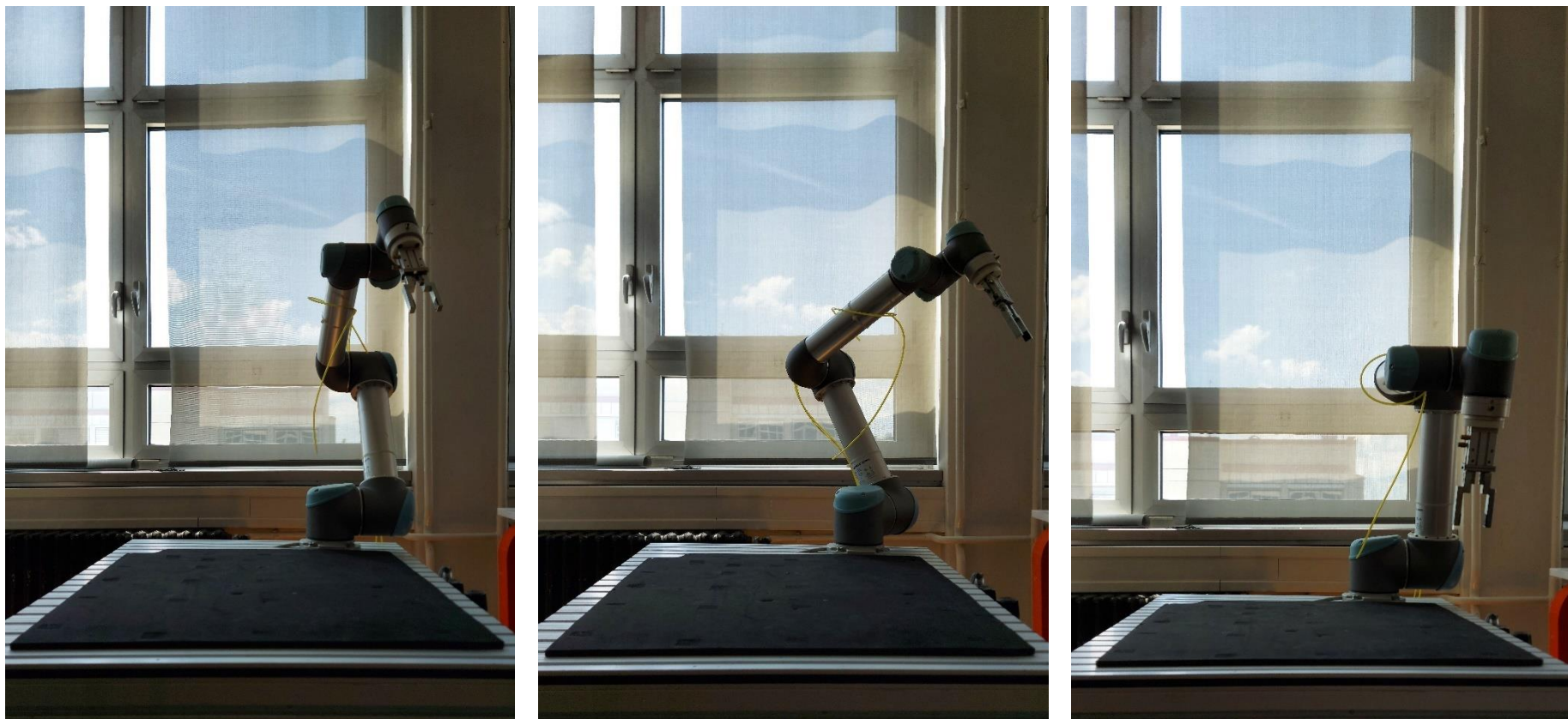
Slika 18. Neutralno stanje robota



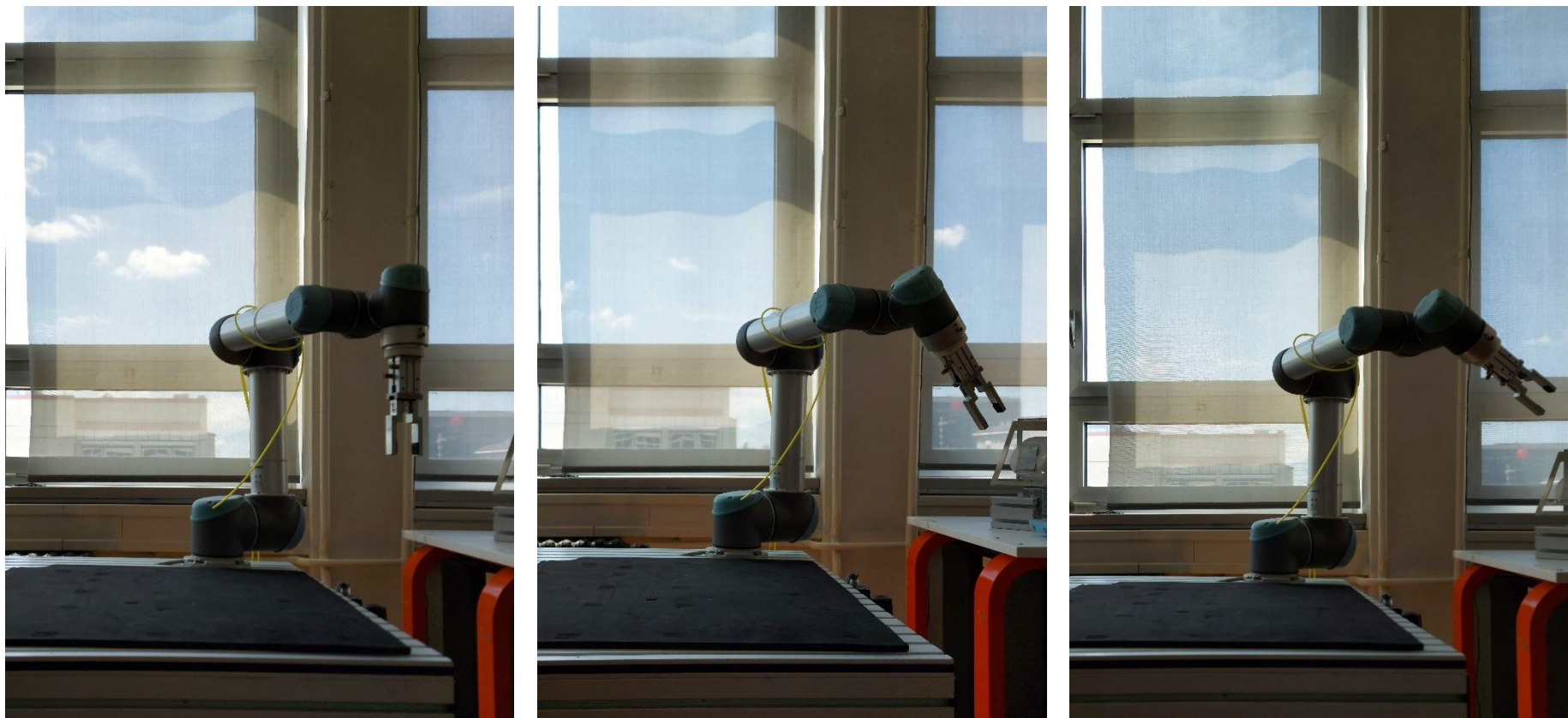
Slika 19. Robotska putanja za prikazivanje emocije *Sadness* (tuga)



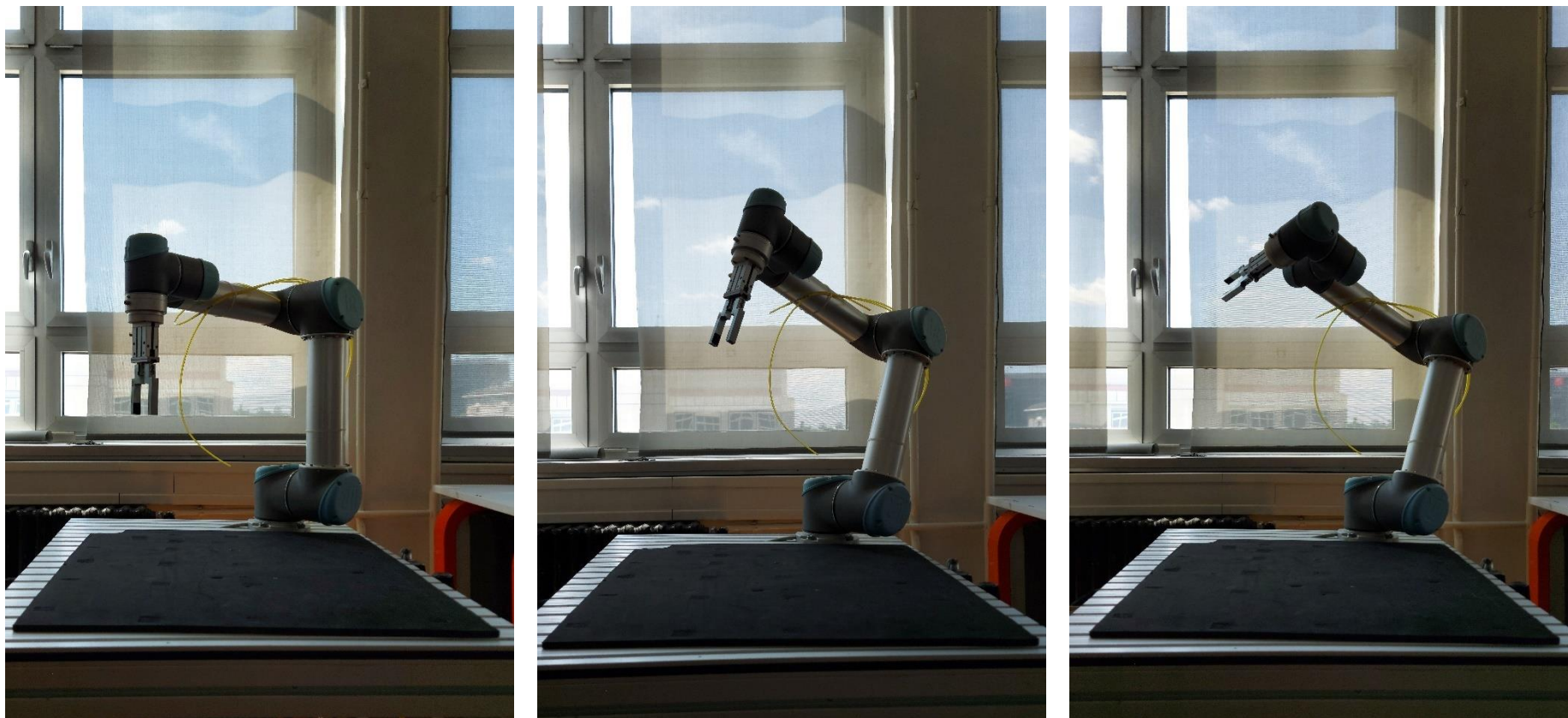
Slika 20. Robotska putanja za prikazivanje emocije *Happiness* (sreća)



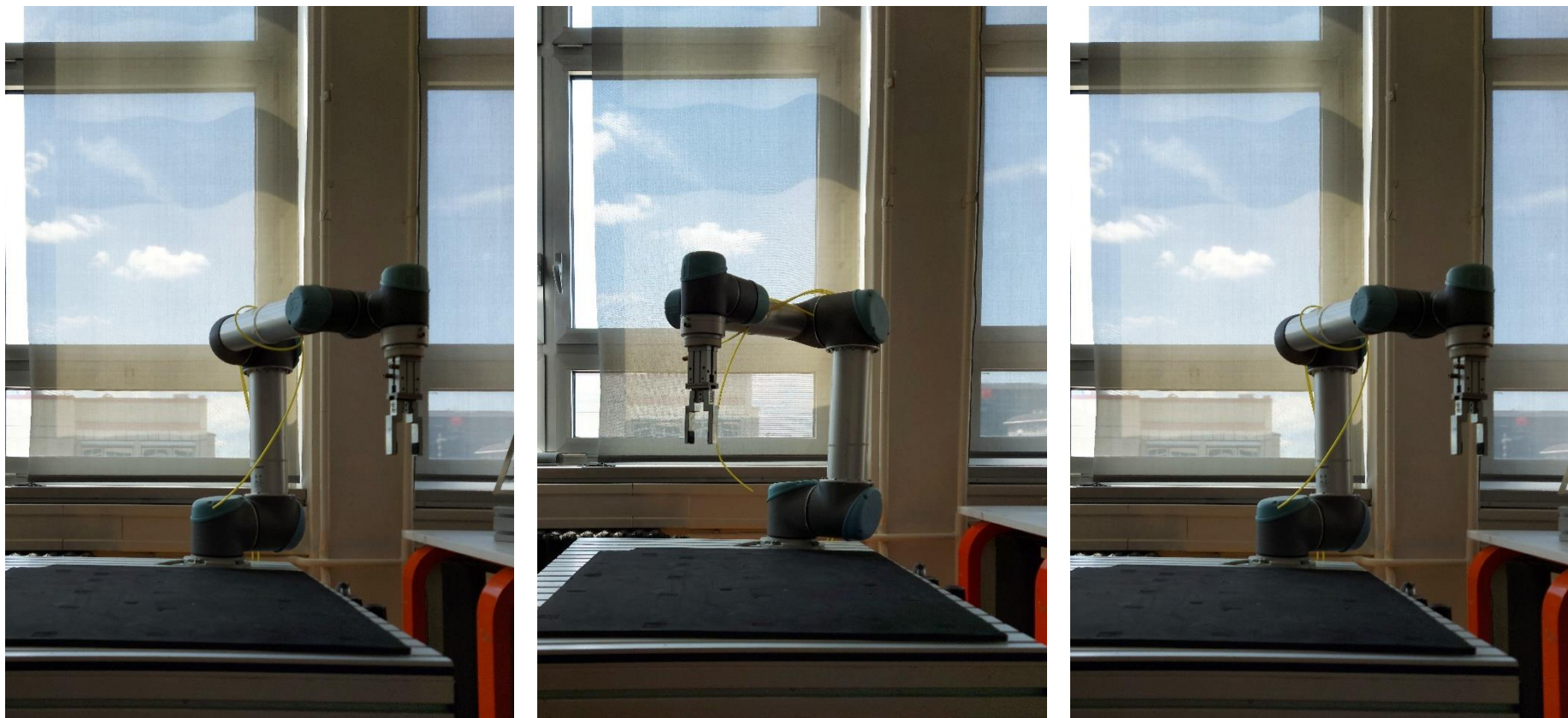
Slika 21. Robotska putanja za prikazivanje emocije *Anger* (ljutnja)



Slika 22. Robotska putanja za prikazivanje emocije *Surprise* (iznenađenje)



Slika 23. Robotska putanja za prikazivanje emocije *Disgust* (gađenje)



Slika 24. Robotska putanja za prikazivanje emocije *Fear* (strah)

5. ZAKLJUČAK

Ljudska komunikacija ovisna je o kontekstu. Sličan princip nastoji se primijeniti i u robotskoj interakciji. Kako bi se roboti doista poistovjetili s ljudima, potrebno je razumijevanje emocionalne pozadine koja stoji iza nekih ljudskih postupaka. Roboti koji razumiju emocionalna stanja ljudi smatraju se pristupačnijima i ljudi im više vjeruju.

U radu je predložen scenarij robotske interakcije čiji je cilj prepoznavanje ljudskih emocija na temelju kontekstualnog značenja te davanje empatičnih odgovora robota uz izvođenje određenih putanja kretanja. Najprije su definirani tekstualni zapisi u obliku rečenica koji predstavljaju jednu od osnovnih emocija - tugu, sreću, ljutnju, iznenađenje, gađenje ili strah. Osim osnovnih emocija dodana je i skupina za neutralan stav. Sljedeći korak bio je definiranje komponenti i alata pomoću kojih bi se ostvarile primitivne kognitivne sposobnosti robota. Govoreći o kognitivnim sposobnostima, nezaobilazan je pojam inteligencije. Kod strojeva, riječ je o umjetnoj inteligenciji. Korištenjem metoda strojnog učenja, koje je podskup umjetne inteligencije, izrađena je umjetna neuronska mreža čiji je zadatak bio prepoznavanje emocija iz prethodno definiranih rečenica. Kako bi to bilo moguće, bila je potrebna obrada riječi u pojedinoj rečenici te njihova pretvorba u numeričke vrijednosti. Nakon što je izrađena umjetna neuronska mreža, slijedio je odabir robota, njegovo spajanje sa računalom te definiranje robotskih putanja kretanja koje će pratiti empatične odgovore.

U konačnici je ostvarena interakcija koja je testirana između čovjeka i robota, no istu je moguće primijeniti i između dva robota na način da jedan robot izgovara rečenice koje predstavljaju određeno emocionalno stanje, a drugi prepoznaje o kojoj emociji je riječ te postupa prema navedenom scenariju.

Ideja za daljnji rad je primjena navedenog rješenja na humanoidnim robotima pomoću kojih bi bilo moguće prikazati i izraze lica za pojedine emocije. Također, treniranje mreže sa većim brojem ulaznih podataka omogućilo bi još bolje generalno prepoznavanje emocija.

LITERATURA

- [1] Airenti G, Plebe A. Editorial Context in Communication: a Cognitive view. *Front Psychol.* 2017.;8.
- [2] Bagheri E, Roesler O, Cao H-L, Vanderborght B. A Reinforcement Learning Based Cognitive Empathy Framework for Social Robots. *Int J of Soc Robotics.* 2020.;
- [3] Zheng X, Shiomi M, Minato T, Ishiguro H. Modeling the Timing and Duration of Grip Behavior to Express Emotions for a Social Robot. *IEEE Robotics and Automation Letters.* 2021.;6(1).
- [4] Song S, Yamada S. Expressing Emotions through Color, Sound, and Vibration with an Appearance-Constrained Social Robot. *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction.* New York, NY, USA: Association for Computing Machinery; 2017. (HRI '17).
- [5] kontekst. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža [Internet]. 2021. [citirano 17. lipanj 2021.]; Dostupno na: <https://www.enciklopedija.hr/Natuknica.aspx?ID=32920>
- [6] Finkbeiner R, Meibauer J, Schumacher PB. What is a Context?: Linguistic approaches and challenges. John Benjamins Publishing; 2012.
- [7] Csn K, Choubey SB, Choubey A. Approaches from cognitive neuroscience and comparative cognition. Sinha GR, Suri JS, urednici. *Cognitive Informatics, Computer Modelling, and Cognitive Science.* Academic Press; 2020. str. 1–19.
- [8] Vossen P, Bajčetić L, Baez S, Bašić S, Kraaijeveld B. Modelling Context Awareness for a Situated Semantic Agent. Bella G, Bouquet P, urednici. *Modeling and Using Context.* Cham: Springer International Publishing; 2019. str. 238–52. (Lecture Notes in Computer Science).
- [9] Mavridis N. A review of verbal and non-verbal human–robot interactive communication. *Robotics and Autonomous Systems.* 2015.;63:22–35.
- [10] Griffiths PE. Basic Emotions, Complex Emotions, Machiavellian Emotions. Hatzimoysis A, urednik. *Philosophy and the Emotions.* Cambridge: Cambridge University Press; 2003. str. 39–68. (Royal Institute of Philosophy Supplements).
- [11] Barrett LF, Lindquist KA, Gendron M. Language as context for the perception of emotion. *Trends Cogn Sci.* 2007.;11(8):327–32.
- [12] Colom R, Karama S, Jung RE, Haier RJ. Human intelligence and brain networks. *Dialogues Clin Neurosci.* 2010.;12(4).
- [13] Neapolitan RE, Jiang X. *Artificial Intelligence: With an Introduction to Machine Learning, Second Edition.* CRC Press; 2018.

- [14] Mondal B. Artificial Intelligence: State of the Art. 2020.
- [15] Campesato O. Artificial Intelligence, Machine Learning, and Deep Learning. Mercury Learning & Information; 2020.
- [16] Thilmany J. The emotional robot. Cognitive computing and the quest for artificial intelligence. EMBO Rep. 2007.;8(11):992–4.
- [17] Mary L. Extraction of Prosody for Automatic Speaker, Language, Emotion and Speech Recognition. 2. izd. Springer International Publishing; 2019. (SpringerBriefs in Speech Technology).
- [18] Chen L, Wu M, Pedrycz W, Hirota K. Emotion Recognition and Understanding for Emotional Human-Robot Interaction Systems. Springer International Publishing; 2021. (Studies in Computational Intelligence).
- [19] Sonka M, Hlavac V, Boyle R. Image Processing, Analysis, and Machine Vision. Cengage Learning; 2014.
- [20] M D, Cuddapah VK, Srivastava A, Mahankali S. AI & ML - Powering the Agents of Automation: Demystifying, IOT, Robots, ChatBots, RPA, Drones & Autonomous Cars-The new workforce led Digital Reinvention facilitated by AI & ML and secured through Blockchain. BPB Publications; 2019.
- [21] Chiu P-S, Chang J-W, Lee M-C, Chen C-H, Lee D-S. Enabling Intelligent Environment by the Design of Emotionally Aware Virtual Assistant: A Case of Smart Campus. IEEE Access. 2020.;8.
- [22] Aggarwal CC. Neural Networks and Deep Learning: A Textbook. Springer; 2018.
- [23] Fulir J. Optimizirane aktivacijske funkcije klasifikatora temeljenog na umjetnim neuronskim mrežama u domeni implementacijskih napada na kriptografske uređaje [Internet] [info:eu-repo/semantics/masterThesis]. University of Zagreb. Faculty of Electrical Engineering and Computing; 2019 [citirano 27. lipanj 2021.]. Dostupno na: <https://urn.nsk.hr/urn:nbn:hr:168:470403>
- [24] Besplatni online tečaj Elements of AI [Internet]. Hrvatska. [citirano 27. lipanj 2021.]. Dostupno na: <http://www.elementsofai.com/hr/>
- [25] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.
- [26] Aggarwal CC. Neural Networks and Deep Learning: A Textbook. Springer; 2018.
- [27] Radhakrishnan P. What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network? [Internet]. Medium. 2017 [citirano 28. lipanj 2021.]. Dostupno na: <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>
- [28] Brownlee J. Loss and Loss Functions for Training Deep Learning Neural Networks [Internet]. Machine Learning Mastery. 2019 [citirano 30. lipanj 2021.]. Dostupno na:

- <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>
- [29] Doshi S. Various Optimization Algorithms For Training Neural Network [Internet]. Medium. 2020 [citirano 28. lipanj 2021.]. Dostupno na: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>
- [30] Tuning the Hyperparameters and Layers of Neural Network Deep Learning [Internet]. Analytics Vidhya. 2021 [citirano 28. lipanj 2021.]. Dostupno na: <https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/>
- [31] Sieben B. Themed article: Doing research on emotion and virtual work: A compass to assist orientation. *Human Relations*. 2007.;60(4):561–80.
- [32] Frommert C, Häfner A, Friedrich J, Zinke C. Using Chatbots to Assist Communication in Collaborative Networks. Camarinha-Matos LM, Afsarmanesh H, Rezgui Y, urednici. *Collaborative Networks of Cognitive Systems*. Cham: Springer International Publishing; 2018. str. 257–65. (IFIP Advances in Information and Communication Technology).
- [33] Lui J, Samani H, Tien K-Y. An affective mood booster robot based on emotional processing unit. 2017. str. 1–6.
- [34] Hruška M, Domšić J, Kožul M. Osnove programiranja (Python), Priručnik za polaznike [Internet]. Kurtović G, urednik. Zagreb: Sveučilište u Zagrebu, Sveučilišni računski centar; 2018 [citirano 22. lipanj 2021.]. Dostupno na: <https://www.srce.unizg.hr/tecajevi/popis-osnovnih-tecajeva/D450>
- [35] Zhang (Uberi) A. SpeechRecognition: Library for performing speech recognition, with support for several engines and APIs, online and offline. [Internet]. [citirano 22. lipanj 2021.]. Dostupno na: https://github.com/Uberi/speech_recognition#readme
- [36] Bhat NM. pyttsx3: Text to Speech (TTS) library for Python 2 and 3 [Internet]. [citirano 22. lipanj 2021.]. Dostupno na: <https://github.com/nateshmbhat/pyttsx3>
- [37] Natural Language Toolkit — NLTK 3.6.2 documentation [Internet]. [citirano 22. lipanj 2021.]. Dostupno na: <https://www.nltk.org/>
- [38] PyTorch documentation — PyTorch 1.9.0 documentation [Internet]. [citirano 22. lipanj 2021.]. Dostupno na: <https://pytorch.org/docs/stable/index.html>
- [39] What is NumPy? — NumPy v1.20 Manual [Internet]. [citirano 22. lipanj 2021.]. Dostupno na: <https://numpy.org/doc/stable/user/whatisnumpy.html>
- [40] JSON | Start IT [Internet]. 2019 [citirano 23. lipanj 2021.]. Dostupno na: <https://start-it.com.hr/json/>
- [41] Eliminacija stop-riječi u pretraživanju | Digitalni akademski arhivi i repozitoriji [Internet]. [citirano 24. lipanj 2021.]. Dostupno na: <https://dabar.srce.hr/faq-page/254>

- [42] Uvod - Hrvatska školska gramatika [Internet]. [citirano 24. lipanj 2021.]. Dostupno na: <http://gramatika.hr/pravilo/uvod/111/>
- [43] Brownlee J. A Gentle Introduction to the Bag-of-Words Model [Internet]. Machine Learning Mastery. 2017 [citirano 24. lipanj 2021.]. Dostupno na: <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
- [44] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2014. [citirano 26. lipanj 2021.]; Dostupno na: <https://arxiv.org/abs/1412.6980v9>
- [45] UR5 collaborative robot arm | Flexible and lightweight cobot [Internet]. [citirano 23. lipanj 2021.]. Dostupno na: <https://www.universal-robots.com/products/ur5-robot/>
- [46] About Universal Robots [Our History] [Internet]. [citirano 23. lipanj 2021.]. Dostupno na: <https://www.universal-robots.com/about-universal-robots/our-history/>
- [47] Fall KR, Stevens WR. TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley; 2011.
- [48] Verywell Mind. The 6 Types of Basic Emotions and Their Effect on Human Behavior [Internet]. [citirano 30. lipanj 2021.]. Dostupno na: <https://www.verywellmind.com/an-overview-of-the-types-of-emotions-4163976>

PRILOZI

- I. CD-R disk
- II. Python programski kôd za ostvarivanje robotske interakcije

emotions.json

```
{
  "emotions": [
    {
      "tag": "sadness",
      "patterns":
        ["I feel like crying",
         "I am stressed",
         "I lost my pet",
         "I failed my exam yesterday"],
      "responses":
        ["Tomorrow is a new day",
         "Things will get better",
         "You seem disappointed",
         "Oh, it's a pity"]
    },
    {
      "tag": "happiness",
      "patterns":
        ["I am going on a vacation",
         "I got a promotion today",
         "I heard my favorite song on the radio",
         "Today is a beautiful day"],
      "responses":
        ["I am so glad for you",
         "I am happy you are happy",
         "Keep up the good work",
         "Smile looks good on you, you should wear it more often"]
    },
    {
      "tag": "anger",
      "patterns":
        ["I am late because of traffic",
         "I spilled my coffee",
         "My flight is delayed",
         "I can't find my phone"],
      "responses":
        ["If you can't do anything about it, let it go",
         "Everything will be fine",
         "I totally understand your point",
         "It is annoying me too"]
    },
    {
      "tag": "surprise",
      "patterns":
        ["I didn't expect to see you here",
         "The sky is so clear",
         "My best friends threw me a surprise birthday party"],
      "responses":
        ["The best things happen unexpectedly",
         "Wow, I can't believe",
         "I am shocked",
```

```
    "Life is full of surprises"]
  },
  {
    "tag": "disgust",
    "patterns":
      ["That smell is terrible",
       "I just can't stand blood",
       "I don't like ginger"],
    "responses":
      ["Close your eyes and think about something nice",
       "Try not to think about it",
       "I don't like it either",
       "That is nauseating"]
  },
  {
    "tag": "fear",
    "patterns":
      ["I saw a spider in my room",
       "That horror movie sent shivers down my spine",
       "I am scared of heights",
       "I had a terrifying experience last week"],
    "responses":
      ["Take a deep breath and try to relax",
       "I am always here if you need to talk",
       "I also get scared sometimes, that's ok",
       "I am also disturbed by that"]
  },
  {
    "tag": "neutral",
    "patterns":
      ["The sky is blue",
       "I am student",
       "If it rains we will get wet"],
    "responses":
      ["That is true",
       "That is interesting",
       "Thank you for that information"]
  }
]
```

model.py

```
import torch.nn as nn

# Izrada modela umjetne neuronske mreže
class NeuralNet(nn.Module):
    def __init__(self, input_size, hidden_size, num_classes):
        super(NeuralNet, self).__init__()
        self.l1 = nn.Linear(input_size, hidden_size)
        self.l2 = nn.Linear(hidden_size, hidden_size)
        self.l3 = nn.Linear(hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        out = self.l1(x)
        out = self.relu(out)
        out = self.l2(out)
        out = self.relu(out)
        out = self.l3(out)
        return out
```

nltk_utils.py

```
import numpy as np
import nltk
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

# Tokenizacija
def tokenize(sentence):
    return nltk.word_tokenize(sentence)

# Korjenovanje riječi i normalizacija veličine slova
def stem(word):
    return stemmer.stem(word.lower())

# Pretvorba tekstualnih varijabli u numeričke varijable
def bag_of_words(tokenized_sentence, words):
    sentence_words = [stem(word) for word in tokenized_sentence]
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1
    return bag
```

train.py

```
import numpy as np
import json
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader
from torch.utils.tensorboard import SummaryWriter
from nltk_utils import bag_of_words, tokenize, stem
from model import NeuralNet

# Definiranje alata za grafički prikaz vrijednosti funkcije gubitka
writer = SummaryWriter("runs/metrics")

# Učitavanje podataka iz datoteke emotions.json
with open('emotions.json', 'r') as f:
    emotions = json.load(f)

all_words = []
tags = []
xy = []

for emotion in emotions['emotions']:
    tag = emotion['tag']
    tags.append(tag)
    for pattern in emotion['patterns']:
        w = tokenize(pattern)
        all_words.extend(w)
        xy.append((w, tag))

# Eliminacija stop-riječi i interpunkcijskih znakova
ignore_words = ['?', '.', '!', ',', 'I', 'am', 'a',
                 'The', 'the', 'my', 'My', 'so', 'is', 'of',
                 'it', 'to', 'the', 'on', 'if', 'If', 'in']
all_words = [stem(w) for w in all_words if w not in ignore_words]
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "Patterns")
print(len(tags), "Tags:", tags)
print(len(all_words), "Unique stemmed words:", all_words)

# Definiranje skupa ulaznih podataka za treniranje umjetne
# neuronske mreže
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)
```

```
# Definiranje hiperparametara
num_epochs = 1000
batch_size = 9
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 9
output_size = len(tags)
print(input_size, output_size)

# Postavljanje funkcija za učitavanje skupa ulaznih podataka
class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    def __len__(self):
        return self.n_samples

# Učitavanje skupa ulaznih podataka
dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                           batch_size=batch_size,
                           shuffle=True,
                           num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else
                      'cpu')
model = NeuralNet(input_size, hidden_size, output_size).to(device)

# Definiranje funkcije gubitka i optimizacijskog algoritma
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Treniranje umjetne neuronske mreže
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        outputs = model(words)
        loss = criterion(outputs, labels)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

```
if (epoch+1) % 100 == 0:
    print(f'Epoch [{epoch+1}/{num_epochs}],
          Loss:{loss.item():.4f}')
    writer.add_scalar('Training loss', loss.item(),
                      global_step=epoch)

# Ispis vrijednosti funkcije gubitka
print(f'Final loss: {loss.item():.4f}')

# Spremanje značajki modela
data = {
    "model_state": model.state_dict(),
    "input_size": input_size,
    "hidden_size": hidden_size,
    "output_size": output_size,
    "all_words": all_words,
    "tags": tags
}

FILE = "data.pth"
torch.save(data, FILE)

print(f'Training complete. file saved to {FILE}')
```

chat.py

```
import random
import json
import time
import speech_recognition as sr
import pyttsx3
import torch
import numpy as np
import urx

from model import NeuralNet
from nltk_utils import bag_of_words, tokenize

device = torch.device('cuda' if torch.cuda.is_available() else
'cpu')

with open('emotions.json', 'r') as json_data:
    emotions = json.load(json_data)

# Učitavanje spremljenih značajki modela
FILE = "data.pth"
data = torch.load(FILE)

input_size = data["input_size"]
hidden_size = data["hidden_size"]
output_size = data["output_size"]
all_words = data['all_words']
tags = data['tags']
model_state = data["model_state"]

# Učitavanje modela neuronske mreže
model = NeuralNet(input_size, hidden_size, output_size).to(device)
model.load_state_dict(model_state)
model.eval()

# Definiranje značajki za pretvorbu tekstualnog zapisa u govor
engine = pyttsx3.init()
rate = engine.getProperty('rate')
engine.setProperty('rate', 125)
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)

bot_name = "Amy"

# Definiranje modula za pretvorbu tekstualnog zapisa u govor
def talk(text):
    engine.say(text)
    engine.runAndWait()

# Predstavljanje robota
talk('My name is Amy')
```

```

# Definiranje modula za prepoznavanje govora
def take_sentence():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Adjusting for background noise. One second")
        r.adjust_for_ambient_noise(source)
        talk('Tell me how you feel')
        print("listening...")
        voice = r.listen(source)

    try:
        sentence = r.recognize_google(voice, language='en-US')
        sentence = sentence.lower()
    except:
        pass

    return sentence

# Definiranje točki robotskih putanja
movements = (
    [[1.4, -87.6, 85.0, -87.6, -91.0, 0.85], #Home /
    np.array                                Neutral
        [-20.0, -87.6, 85.0, -87.6, -91.0, 0.85], #Sadness
        [-20.0, -87.6, 100.0, -87.6, -91.0, 0.85],
        [-20.0, -87.6, 110.0, -70.6, -91.0, 0.85],
        [20.0, -87.6, 85.0, -87.6, -91.0, 0.85], #Happiness
        [20.0, -120.0, 85.0, -87.6, -91.0, 0.85],
        [20.0, -120.0, 60.0, -87.6, -91.0, 0.85],
        [1.4, -120.0, 85.0, -87.6, -91.0, 0.85], #Anger
        [50.0, -120.0, 85.0, -87.6, -91.0, 0.85],
        [1.4, -80.0, 85.0, -87.6, -140.0, 0.85],
        [1.4, -87.6, 85.0, -87.6, -30.0, 0.85], #Surprise
        [20.0, -87.6, 85.0, -87.6, -50.0, 0.0],
        [20.0, -120.0, 90.0, -90.0, -50.0, 0.0],
        [-50.0, -87.6, 85.0, -87.6, -91.0, 0.85], #Disgust
        [-50.0, -110.0, 85.0, -87.6, -91.0, 0.85],
        [-50.0, -110.0, 85.0, -87.6, -160.0, 0.85],
        [-20.0, -87.6, 85.0, -87.6, -91.0, 0.85], #Fear
        [20.0, -87.6, 85.0, -87.6, -91.0, 0.85],
        [-40.0, -87.6, 85.0, -87.6, -91.0, 0.85]])

# Mrežno spajanje robota
robot = urx.Robot("IP address")

# Definiranje modula za kretanje robota
def move_robot(start, num, robot, acc, vel):
    if (start + num) > len(movements):
        print("Number of points too large")
        return

    for idx in range(num):
        robot.movej(movements[start+idx] * (3.1415 / 180.0),
                    acc=acc, vel=vel, )

```

```
# Definiranje modula za ostvarivanje interakcije
def start_chat():
    sentence = take_sentence()
    print(sentence)

    sentence = tokenize(sentence)
    x = bag_of_words(sentence, all_words)
    x = x.reshape(1, x.shape[0])
    x = torch.from_numpy(x).to(device)

    # Predviđanje emocije
    output = model(x)
    _, predicted = torch.max(output, dim=1)
    tag = tags[predicted.item()]
    probs = torch.softmax(output, dim=1)
    prob = probs[0][predicted.item()]

    # Ukoliko je vjerojatnost predviđanja veća od 75% ostvarivanje
    # interakcije:
    if prob.item() > 0.75:
        for emotion in emotions['emotions']:
            if tag == emotion["tag"]:
                print(tag)
                talk(f"{random.choice(emotion['responses'])}")

    # Izvođenje robotskih putanja kretanja
    if 'sadness' in tag:
        print('Sadness')
        move_robot(1, 3, robot, acc=0.6, vel=0.3)
        time.sleep(1)
    elif 'happiness' in tag:
        print('Happiness')
        move_robot(4, 6, robot, acc=0.8, vel=0.4)
        time.sleep(1)
    elif 'anger' in tag:
        print('Anger')
        move_robot(7, 9, robot, acc=0.8, vel=0.4)
        time.sleep(1)
    elif 'surprise' in tag:
        print('Surprise')
        move_robot(10, 12, robot, acc=0.7, vel=0.3)
        time.sleep(1)
    elif 'disgust' in tag:
        print('Disgust')
        move_robot(13, 15, robot, acc=0.7, vel=0.3)
        time.sleep(1)
    elif 'fear' in tag:
        print('Fear')
        move_robot(16, 18, robot, acc=0.8, vel=0.4)
        time.sleep(1)
    else:
        print('Neutral')
```

```
        move_robot(0, 0, robot, acc=0.5, vel=0.3)
        time.sleep(1)
    else:
        talk('I do not understand, try again with another sentence')
    return tag

while True:
    start_chat()
```