



Srednja šola za kemijo, elektrotehniko in računalništvo

Pot na Lavo 22

3000 Celje

# Izdelava prevajalnika znakovnega jezika

Raziskovalna naloga

**Avtorji:** Luka Jurhar, Leon Kotnik, Nejc Mlakar

**Mentor:** Jaka Koren, mag. inž. rač. in informatike

Mestna občina Celje, Mladi za Celje

Celje, 2025

## IZJAVA\*

Mentor Jaka Koren v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Izdelava prevajalnika znakovnega jezika, katere avtorji so Leon Kotnik, Nejc Mlakar, Luka Jurhar:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 3. 4. 2025



Podpis mentorja

Podpis odgovorne osebe

\*

### POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

## **Zahvala**

Zahvaljujemo se vsem, ki so kakorkoli pripomogli k nastanku naše raziskovalne naloge. Posebna zahvala gre našemu mentorju, Jaki Korenu, mag. inž. računalništva in informatike, za njegovo neprecenljivo pomoč, spodbudo in pripravljenost, da nam stoji ob strani skozi celoten proces raziskovanja. Njegova strokovnost in predanost sta nam omogočili uspešno izvedbo naloge. Prav tako se iskreno zahvaljujemo ravnateljici, prof. dr. Aniti Laznik, za njeno podporo in spodbujanje raziskovalnega dela na naši šoli.

## **Povzetek**

Raziskovalna naloga se osredotoča na razvoj sistema za prevajanje znakovnega jezika v črke z uporabo sodobnih tehnologij, kot so računalniški vid, umetna inteligenca in strojno učenje. Cilj je bil omogočiti boljšo komunikacijo med gluhami in slušnimi posamezniki. Sistem temelji na Pythonu in uporablja knjižnice OpenCV, MediaPipe, TensorFlow in Kivy za zaznavanje gest, prepoznavanje črk in prikaz prevodov v realnem času.

Rezultati so pokazali, da sistem v kontroliranem okolju prepozna znake z natančnostjo nad 90 % in deluje s hitrostjo, ki omogoča tekočo komunikacijo. V težjih pogojih, kot so slaba osvetlitev, se natančnost zmanjša, vendar ostaja uporabna. Kljub uspehom so ostali izzivi, kot so različni jezikovni standardi in individualne razlike v izvedbi gest.

Naloga je uspešno izpolnila cilje, razvili smo funkcionalen prototip, ki prispeva k večji vključenosti gluhe skupnosti. V prihodnje bi bilo koristno razširiti sistem za več jezikovnih standardov in izboljšati njegovo natančnost v različnih okoljih. Ta raziskava je korak k boljši dostopnosti komunikacije za gluhe in naglušne posameznike.

## **Abstract**

The research focuses on the development of a system for translating sign language into letters using modern technologies such as computer vision, artificial intelligence and machine learning. The aim was to enable better communication between deaf and hearing individuals. The system is Python-based and uses the OpenCV, MediaPipe, TensorFlow and Kivy libraries for gesture detection, letter recognition and real-time display of translations.

The results showed that the system recognises characters with an accuracy of over 90% in a controlled environment and operates at a speed that allows smooth communication. In harsh conditions such as low light, the accuracy decreases but remains usable. Despite the successes, challenges remain, such as different language standards and individual differences in gesture performance.

The task has successfully met its objectives, and we have developed a functional prototype that contributes to greater inclusion of the deaf community. In the future, it would be useful to extend the system to more language standards and improve its accuracy in different settings. This research is a step towards better accessibility of communication for deaf and hard of hearing individuals.

## Kazalo vsebine

1	Uvod.....	10
1.1	Opredelitev področja in raziskovalnega problema .....	11
1.2	Cilji raziskovalne naloge .....	11
2	Hipoteze.....	12
3	Metoda raziskovalnega dela .....	12
4	Znakovni jezik .....	13
4.1	Kaj je znakovni jezik?.....	13
4.2	Zgodovina znakovnega jezika.....	13
4.3	Različni standardi znakovnih jezikov po svetu .....	14
4.4	Raba znakovnega jezika v Sloveniji.....	15
5	Prepoznavanje objektov .....	16
5.1	Prepoznavanje predmetov / Razvrščanje predmetov.....	16
5.2	Kako deluje .....	17
5.3	Uporaba prepoznave predmetov.....	18
6	Nevronsko omrežje .....	19
6.1	Ka je nevronska omrežje?.....	19
6.2	Uporaba nevronskih omrežij .....	19
7	Programska orodja.....	20
7.1	Visual Studio Code .....	20
7.2	PyCharm.....	22
7.3	Windows .....	23
8	Programski jeziki.....	24
8.1	Python.....	24
8.1.1	YOLO.....	25
8.1.2	TENSORFLOW .....	25
8.1.3	PYTHON CLOUD CLIENT LIBRARIES .....	26

8.1.4	OPENCV.....	27
8.1.5	KIVY.....	28
9	Izdelava prevajalnika znakovnega jezika .....	29
9.1	Izbira razvojnega okolja in orodij.....	29
9.2	Razvoj grafičnega uporabniškega vmesnika.....	29
9.3	Zasnova delovanja aplikacije .....	29
9.4	Razvoj algoritma za prepoznavanje znakovnega jezika.....	30
9.5	Optimizacija obdelave podatkov .....	30
9.6	Integracija sistema v končno aplikacijo .....	31
10	Rezultati .....	33
11	Zaključek.....	34
12	Viri in literatura .....	35

## Kazalo slik

Slika 1:	ASL abeceda znakovnega jezika. (Vir: Wikipedija, 2025).....	15
Slika 2:	Logotip Visual Studio Code (Vir: Wikipedija, 2025).....	21
Slika 3:	Pycharm logotip (Vir: Wikipedija, 2025).....	22
Slika 4:	Windows logotip (Vir: Wikipedija, 2025).....	23
Slika 5:	Logotip programskega jezika python (Vir: Wikipedija, 2025) .....	24
Slika 6:	Tensorflow logotip (Vir: Tensorflow, 2025) .....	26
Slika 7:	OpenCv logotip (Vir: OpenCV, 2025).....	27
Slika 8:	Kivy logotip (Vir: Kivy, 2025) .....	28
Slika 9:	Začetna stran aplikacije (Vir Lasten, 2025).....	31
Slika 10:	Končna aplikacija prevajalnika znakovnega jezika (Vir: lasten, 2025) .....	32

# 1 Uvod

Prevod znakovnega jezika v črke bo sodobna tehnološka rešitev za boljšo dostopnost oseb s slušnimi ali govornimi težavami. Znakovni jezik je ključen za gluhe in naglušne posameznike, vendar ga razume le majhen del populacije. Razvoj avtomatskih sistemov za prevajanje znakovnega jezika v besedilo ali govor bo izboljšal komunikacijo in vključevanje teh posameznikov v družbo. V tej raziskavi se bomo osredotočili na izdelavo sistema, ki bo temeljil na umetni inteligenci, strojnem učenju in računalniškem vidu.

Opisali bomo teoretične osnove, pristope k zaznavanju gest in izzive, kot so različni jezikovni standardi. Uporabili bomo programska orodja, nevronske mreže in druge tehnike strojnega učenja. Sledil bo opis postopka izdelave sistema, od zajema podatkov do implementacije algoritmov. Na koncu bomo predstavili rezultate testiranja in analizo natančnosti ter uporabnosti sistema.

Cilj naloge bo pokazati, kako bodo sodobne tehnologije izboljšale komunikacijo za gluhe in naglušne ter spodbudile nadaljnji razvoj na tem področju.

## **1.1 Opredelitev področja in raziskovalnega problema**

Prevod znakovnega jezika v črke ali besedilo je ena izmed pomembnih tehnoloških rešitev, ki lahko znatno izboljša življenje ljudi s slušnimi ali govornimi težavami. Znakovni jezik je za gluhe in naglušne posameznike pogosto glavno sredstvo komunikacije, vendar ga veliko ljudi ne razume. To ustvarja ovire v vsakdanjem življenju, zato je razvoj sistemov, ki znakovni jezik samodejno prevajajo v besedilo ali govor, ključnega pomena za njihovo vključevanje v družbo. Ta področje združuje različne tehnologije, kot so računalniški vid, umetna inteligenca in strojno učenje, da bi omogočili hitro in natančno prepoznavanje ter prevajanje gest.

Glavni izziv, ki ga rešujemo pri izdelavi takega sistema, je, kako narediti program, ki bo zanesljivo prepoznal znakovni jezik in ga pretvoril v besedilo v realnem času. Pri tem se soočamo s številnimi težavami, kot so različni jezikovni standardi znakovnega jezika, individualne razlike v izvedbi gest ter vplivi okolja (npr. osvetlitev ali ozadje). Poleg tega mora sistem biti dovolj prilagodljiv, da se lahko navadi na različne uporabnike in njihove edinstvene izraze.

## **1.2 Cilji raziskovalne naloge**

Cilj te raziskovalne naloge je razviti sistem za prevajanje znakovnega jezika v črke ali besedilo, ki bo omogočil boljšo komunikacijo med gluhi in slišnimi posamezniki. Želimo ustvariti rešitev, ki bo v realnem času natančno prepoznala znakovni jezik, prilagodljiva različnim jezikovnim standardom in individualnim razlikam v izvedbi gest. Hkrati želimo preučiti, kako sistem deluje v resničnih pogojih, z analizo njegove natančnosti, hitrosti in uporabniške izkušnje. S tem si prizadevamo prispevati k večji vključenosti in dostopnosti za vse člane družbe.

## **2 Hipoteze**

Hipoteza 1: Sistem bo sposoben prepoznati in pretvoriti znake v besedilo z natančnostjo nad 90 % v kontroliranem okolju.

Hipoteza 2: Sistem bo deloval v realnem času z zamikom, ki ne presega 2 sekund, kar bo omogočilo tekočo komunikacijo.

Hipoteza 3: Natančnost sistema se bo zmanjšala v težjih okoljskih pogojih, kot so slaba osvetlitev ali zapleteno ozadje, vendar bo ohranil uporabno stopnjo natančnosti.

## **3 Metoda raziskovalnega dela**

Za izvedbo te raziskovalne naloge bomo uporabili več različnih pristopov. Najprej bomo preučili, kako delujejo obstoječi sistemi za prepoznavanje znakovnega jezika, da bi ugotovili, kaj že deluje dobro in kje so še pomanjkljivosti. Nato bomo razvili prototip sistema, ki bo uporabljal tehnologije za prepoznavanje gest in pretvarjanje znakov v besedilo. Ta prototip bomo testirali v različnih pogojih, na primer pri različni svetlobi ali zapletenem ozadju, da bi preverili, kako natančen in hiter je. Na koncu bomo zbrali vse podatke in jih analizirali, da bi ugotovili, kako bi lahko sistem še izboljšali in ga naredili bolj uporabnega.

## **4 Znakovni jezik**

### **4.1 Kaj je znakovni jezik?**

Znakovni jezik je vizualni način sporazumevanja, ki temelji na uporabi rok, mimike obraza in telesne govorice za izražanje pomena. Gre za naraven jezik, ki ima svojo slovnico, skladnjo in besedni zaklad, enakovreden govorjenim jezikom. Vsaka država ali regija ima svoj znakovni jezik, ki se je razvil neodvisno od govorjenih jezikov. Pomembno je razumeti, da znakovni jeziki niso univerzalni – na primer, slovenski znakovni jezik (SZJ) se razlikuje od ameriškega znakovnega jezika (ASL) ali britanskega znakovnega jezika (BSL). Znakovni jezik ni le sredstvo komunikacije, ampak tudi pomemben del identitete in kulture gluhe skupnosti, saj omogoča enakopravno izražanje, izobraževanje in socialno vključevanje.

### **4.2 Zgodovina znakovnega jezika**

Znakovni jezik ima dolgo in zanimivo zgodovino, saj se je razvijal kot ključno sredstvo komunikacije za gluhe in naglušne osebe. Njegove zametke lahko zasledimo že v antiki, kjer so filozofi, kot je Aristotel, razpravljali o sposobnosti gluhih za učenje. Čeprav je Aristotel napačno menil, da so gluhe osebe nesposobne govora in zato tudi razmišljanja, so v različnih kulturah obstajale metode neverbalne komunikacije. V srednjem veku so menihi, ki so zaradi zaobljube molka potrebovali drugačen način izražanja, razvili različne sisteme ročnih znakov, ki so jim omogočali sporazumevanje.

Prvi organizirani znakovni jeziki so nastali v 18. stoletju, ko je francoski duhovnik Abbé de l'Épée v Parizu ustanovil šolo za gluhe otroke. Opazil je, da se gluhi otroci med seboj sporazumevajo s kretnjami, in te znake sistematično razvil v francoski znakovni jezik (LSF). Njegova metoda je vplivala na širjenje znakovnega jezika po svetu, saj so se učenci njegove šole kasneje vračali v svoje države in tam uvajali podobne načine komunikacije.

V 19. stoletju se je v Združenih državah Amerike razvil ameriški znakovni jezik (ASL), ki temelji na francoskem znakovnem jeziku, vendar se je skozi čas razvijal in prilagajal lokalnim potrebam. V tem obdobju je bilo veliko razprav o tem, ali bi bilo bolje gluhe otroke učiti znakovnega jezika ali jih spodbujati k učenju govora in branja z ustnic (oralna metoda). Na Milanski konferenci leta 1880 so strokovnjaki iz Evrope in Amerike

odločili, da je oralna metoda boljša, kar je za več desetletij zavrlo uporabo znakovnega jezika.

Šele v drugi polovici 20. stoletja so lingvisti, kot je William Stokoe, znakovni jezik prepoznali kot polnopravni jezik s svojo slovnico in strukturo. Od takrat dalje je znakovni jezik vse bolj priznan in sprejet kot pomemben del identitete gluhe skupnosti. Danes obstaja več kot 300 različnih znakovnih jezikov po svetu, ki se še naprej razvijajo in prilagajajo sodobnim potrebam. Znakovni jeziki niso univerzalni, saj se med seboj razlikujejo tako kot govornjeni jeziki, vendar imajo skupne lastnosti in pomembno vlogo pri vključevanju gluhih oseb v družbo.

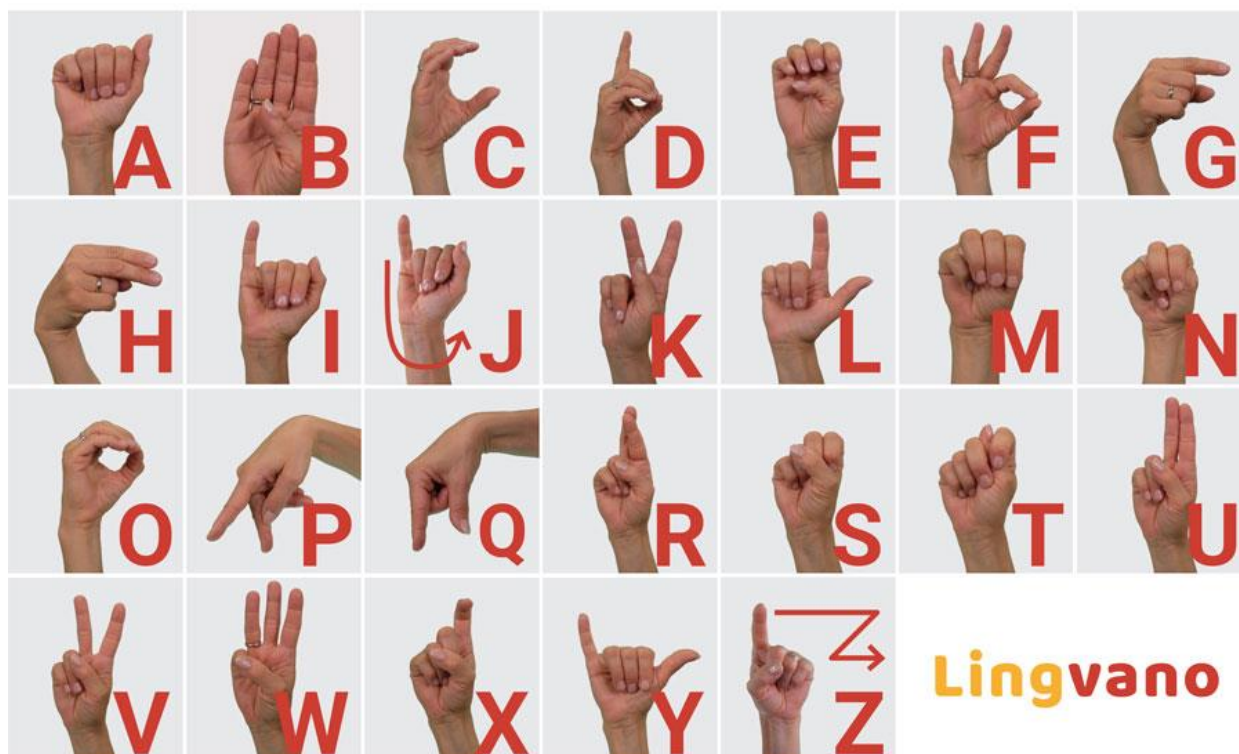
### **4.3 Različni standardi znakovnih jezikov po svetu**

Znakovni jeziki se, tako kot govornjeni jeziki, razlikujejo po državah in regijah, kar pomeni, da ni enotnega, univerzalnega znakovnega jezika, ki bi ga razumeli vsi gluhi po svetu. Vsaka država ima svoj znakovni jezik, ki se je razvijal neodvisno in v skladu s kulturnimi in jezikovnimi posebnostmi določene skupnosti. Na primer, ameriški znakovni jezik (ASL) se bistveno razlikuje od britanskega znakovnega jezika (BSL), čeprav se obe državi sporazumevata v angleščini. Slovenski znakovni jezik (SZJ) se prav tako razlikuje od znakovnih jezikov sosednjih držav, saj ima svojo edinstveno slovnico in kretnje. Poleg nacionalnih znakovnih jezikov obstajajo tudi regionalne različice in dialekti, ki se lahko razlikujejo glede na kraj ali celo generacijo uporabnikov.

Kljub tem razlikam obstajajo tudi poskusi poenotenja znakovnega jezika za lažje mednarodno sporazumevanje. Eden izmed takšnih primerov je mednarodni znakovni jezik (International Sign – IS), ki se uporablja predvsem na mednarodnih srečanjih in konferencah gluhih. Ta sistem ni popolnoma razvit jezik, temveč poenostavljen način sporazumevanja, ki vključuje elemente različnih znakovnih jezikov in intuitivne kretnje. V nekaterih državah obstajajo tudi uradni standardi in zakonodaja, ki določajo uporabo znakovnega jezika v izobraževanju in javnih storitvah, kar prispeva k enakopravnejšemu vključevanju gluhih v družbo. Kljub napredku pa še vedno obstajajo izzivi pri priznavanju in standardizaciji znakovnih jezikov, predvsem v državah, kjer gluhi nimajo enakih pravic do uporabe svojega naravnega jezika.

## 4.4 Raba znakovnega jezika v Sloveniji

V Sloveniji je **slovenski znakovni jezik (SZJ)** uradno priznan kot jezik gluhe skupnosti. Leta 2021 je bil vpisan v Ustavo Republike Slovenije, kar je pomemben korak k enakopravnosti gluhih in naglušnih oseb. Zakon o uporabi slovenskega znakovnega jezika določa, da imajo gluhe osebe pravico do uporabe SZJ v izobraževalnem procesu, pri dostopu do javnih storitev ter v pravnih in zdravstvenih postopkih. Kljub temu v praksi še vedno prihaja do pomanjkanja tolmačev in neustrezne prilagojenosti javnih institucij, zaradi česar imajo gluhe osebe pogosto omejen dostop do informacij in storitev. V vsakdanjem življenju se SZJ uporablja predvsem v skupnostih gluhih, v družinah, kjer so prisotni gluhi člani, ter na dogodkih, ki so prilagojeni tej skupnosti. Televizijski programi in javne ustanove postopoma uvajajo več tolmačenih vsebin, vendar je dostopnost informacij v SZJ še vedno izziv. Kljub zakonskim pravicam si gluhi v Sloveniji še vedno prizadevajo za boljšo integracijo znakovnega jezika v izobraževanje, medije in širšo družbo.



Slika 1: ASL abeceda znakovnega jezika. (Vir: Wikipedija, 2025)

## 5 Prepoznavanje objektov

Prepoznavanje objektov je ključna naloga računalniškega vida, katere cilj je prepoznati in locirati objekte na slikah ali video zaporedjih. Vključuje prepoznavanje predmetov in njihovo natančno lociranje s pomočjo omejitvenih okvirov. Na začetku se slike predobdelajo, kar lahko vključuje spremembo velikosti ali normalizacijo, da se pripravijo za analizo. Modeli globokega učenja, kot so konvolucijske nevronske mreže (CNN), iz slik izločijo ustrezne lastnosti in zajamejo bistvene vzorce za razlikovanje predmetov. Model nato predvidi omejitvene okvire, ki obkrožajo predmete in označujejo njihove prostorske lokacije, hkrati pa jih razvrsti v kategorije, kot sta "oseba" ali "avtomobil". Tehnike naknadne obdelave izboljšajo rezultate s filtriranjem lažno pozitivnih rezultatov in tako povečajo natančnost. Prepoznavanje predmetov se pogosto uporablja na področjih, kot so avtonomna vozila, nadzor, robotika in medicinsko slikanje, saj omogoča strojem učinkovito razumevanje in interakcijo z vizualnim svetom.

### 5.1 Prepoznavanje predmetov / Razvrščanje predmetov

Prepoznavanje predmetov in razvrščanje predmetov predstavljata dva ključna pristopa k prepoznavanju predmetov v računalniškem svetu, ki imata različne cilje in izhodne rezultate. Prepoznavanje objektov je v osnovi kot detektiv, ki ne le prepozna objekt na sliki, temveč tudi natančno označi, kje se nahaja. V rezultatih prepoznavanja objektov tako dobimo razredno oznako, ki pove, kateri predmet je prepoznani, in hkrati pravokotnik ali kvadrat, ki precizno opisuje njegovo lokacijo na sliki. To omogoča vpogled v prostorsko razporeditev predmetov na sliki ali v videu in tudi stvarno-časovno zapisovanje prepoznanih predmetov. Po drugi strani pa razvrščanje objektov opravlja bolj osnovno nalogo prepoznavanja. Če bi primerjali z branjem naslovov knjig, bi razvrščanje predmetov samo povedalo, katera vrsta knjige je prisotna na sliki, ne pa tudi, kje se nahaja. Zato je izhod razvrščanja objektov sestavljen iz seznama razrednih oznak, ki identificirajo vrste predmetov, vendar ne vsebujejo informacij o njihovi prostorski postavitvi. V praksi se prepoznavanje objektov pogosto uporablja tam, kjer je ključno vedeti ne le, kateri predmeti so prisotni, temveč tudi, kje točno se nahajajo. Nasprotno pa razvrščanje najdemo v scenarijih, kjer je dovolj le prepoznati vrsto predmetov, ne da bi natančno vedeli, kje se nahajajo na sliki ali v videu. Oba pristopa dopolnjujeta drug drugega, odvisno od specifičnih potreb in izzivov analize

vizualnih podatkov. Prepoznavanje objektov je bolj zapleten, saj zahteva, da računalnik razume različne predmete, njihove velikosti in oblike, ter se spopada s situacijami, ko so predmeti delno skriti ali prekriti, razvrščanje pa je bolj osnovno, saj se osredotoča le na prepoznavanje vrste predmetov.

## 5.2 Kako deluje

Prepoznavanje predmetov deluje tako, da natančno pregleda slike ali videoposnetke ter ugotovi in opiše prisotnost predmetov. Postopek poteka v vrsti strukturiranih korakov. Na začetku se vhodna slika ali kader predobdela, da se standardizira njena oblika in izboljša razlagalnost modela. Tehnike, kot so spreminjanje velikosti, normalizacija in pretvorba barvnega prostora, optimizirajo podatke za poznejšo analizo. Nato algoritmi za prepoznavanje predmetov uporabljajo arhitekture globokega učenja, predvsem konvolucijske nevronske mreže (CNN), da iz vhodnih podatkov izluščijo ustrezne lastnosti. Te značilnosti zajemajo bistvene vizualne značilnosti, vključno z vzorci, teksturami in oblikami, ki kažejo na različne predmete v prizoru. Po ekstrakciji značilnosti se odkrivanje predmetov nadaljuje s predlaganjem kandidatnih območij na sliki, kjer bi se lahko nahajali predmeti. Te regije, ustvarjene z metodami, kot so selektivno iskanje ali mreže za predlaganje regij, služijo kot osrednje točke za nadaljnjo analizo. Za vsako predlagano regijo CNN izlušči značilnosti, ki se nato prenesejo v klasifikacijsko omrežje. To omrežje vsaki regiji dodeli oceno verjetnosti, ki označuje verjetnost, da vsebuje določen razred predmetov. Hkrati regresija mejnih polj prilagodi predlagana mejna polja, da natančneje zajamejo meje prepoznanih predmetov. Po klasifikaciji in regresiji se izvede ključni korak naknadne obdelave, imenovan nemaksimalno zatiranje. V tem koraku se izločijo podvojena prepoznavanja in ohranijo le najbolj zanesljive napovedi, s čimer se zagotovi, da je vsak objekt identificiran natančno enkrat. Končni rezultat prepoznavanja predmetov so koordinate prepoznanih mejnih polj, pripadajoče oznake razredov in ocene zaupanja, ki kažejo na gotovost prepoznavanja. Modeli za prepoznavanje predmetov so usposobljeni na obsežnih zbirkah podatkov, ki vsebujejo slike z opombami, na katerih so predmeti natančno označeni. Med usposabljanjem se model nauči razločevati razrede predmetov in jih natančno locirati na slikah. Nato se lahko usposobljeni model uporabi za analizo novih, še nevidenih slik ali videoposnetkov in učinkovito prepozna predmete v različnih scenarijih iz resničnega sveta. Algoritmi za prepoznavanje predmetov v bistvu omogočajo strojem, da razumejo in dešifrirajo vizualno vsebino, kar omogoča

številne aplikacije na področjih, kot so avtonomna vozila, nadzor, robotika in medicinsko slikanje.

### 5.3 Uporaba prepoznave predmetov

Odkrivanje objektov se uporablja na številnih področjih, pri čemer se izkorišča sposobnost prepoznavanja in lociranja objektov v slikah ali video tokovih. Navajamo nekaj pomembnih primerov uporabe:

- **Avtonomna vozila:** Za avtonomna vozila je prepoznavanje in razumevanje okolice ključnega pomena. Pomaga pri prepoznavanju pešcev, vozil, prometnih znakov in ovir, kar vozilu omogoča, da v realnem času sprejema utemeljene odločitve za zagotavljanje varne navigacije.
- **Nadzor in varnost:** V nadzornih sistemih prepoznavanje predmetov pomaga pri spremljanju in analiziranju videoposnetkov za odkrivanje sumljivih dejavnosti, vsiljivcev ali zanimivih predmetov. Omogoča avtomatiziran nadzor, kar zmanjšuje potrebo po človeškem posredovanju in izboljšuje odzivni čas na morebitne grožnje.
- **Analitika maloprodaje:** Trgovci na drobno uporabljajo prepoznavanje predmetov za sledenje gibanja strank, analizo postavitve trgovin in spremljanje postavitve izdelkov. Pomaga pri optimizaciji postavitve trgovin, analiziranju vedenja strank in izboljšanju postavitve izdelkov za boljšo prodajo in zadovoljstvo strank.
- **Medicinsko slikanje:** Pri medicinskem slikanju prepoznavanje predmetov pomaga pri prepoznavanju in lociranju anatomskih struktur, tumorjev, nepravilnosti in drugih zdravstvenih stanj z rentgenskimi žarki, magnetno resonanco, računalniško tomografijo in drugimi načini slikanja. Radiologom in zdravstvenim delavcem pomaga pri diagnosticiranju in načrtovanju zdravljenja.
- **Industrijska avtomatizacija:** Prepoznavanje objektov se v industrijski avtomatizaciji uporablja za nadzor kakovosti, odkrivanje napak in upravljanje zalog. Pomaga pri prepoznavanju izdelkov z napako na proizvodnih linijah, sledenju ravni zalog in optimizaciji logističnih procesov.

- **Razširjena resničnost in igre:** Prepoznavanje predmetov omogoča aplikacije razširjene resničnosti in igralne izkušnje s prepoznavanjem predmetov, gest in gibov v resničnem svetu. Omogoča interaktivne izkušnje in potopljivo igranje iger ter briše meje med fizičnim in virtualnim svetom.
- **Spremljanje okolja:** Prepoznavanje predmetov pomaga pri spremljanju okolja z identifikacijo in sledenjem divjih živali, spremljanjem krčenja gozdov in preučevanjem biotske raznovrstnosti. Raziskovalcem in naravovarstvenikom pomaga pri prizadevanjih za ohranjanje divjih živali in pri upravljanju okolja.

To je le nekaj primerov široke uporabe prepoznavanja predmetov, ki poudarjajo njegov pomen v različnih panogah in področjih. Ker tehnologija še naprej napreduje, lahko pričakujemo, da se bo v prihodnosti pojavilo še več inovativnih aplikacij.

## 6 Nevronsko omrežje

### 6.1 Ka je nevronska omrežje?

Nevronska omrežja (ali umetna nevronska mreža) je model v strojnem učenju, ki posnema delovanje bioloških nevronskih omrežij v možganih. Sestavljeno je iz povezanih enot, imenovanih umetni nevroni, ki so med seboj povezani podobno kot sinapse v možganih. Vsak nevron sprejema vhodne signale, jih obdela z aktivacijsko funkcijo in posreduje naprej. Povezave med nevroni imajo uteži, ki se med učenjem prilagajajo, da omrežje izboljšuje svoje napovedi.

Nevronska omrežja so organizirana v plasti: vhodno, izhodno in eno ali več skritih plasti. Če omrežje vsebuje vsaj dve skriti plasti, ga imenujemo globoko nevronska omrežje. Uporabljajo se za različne naloge, kot so napovedovanje, prilagodljivo krmiljenje in reševanje kompleksnih problemov umetne inteligence. Sposobna so učenja iz izkušenj in prepoznavanja vzorcev tudi v zapletenih podatkih.

### 6.2 Uporaba nevronskih omrežij

Nevronska omrežja so pomemben del umetne inteligence in se uporabljajo za različne naloge, kot so prepoznavanje slik, analiza besedila, napovedovanje podatkov in avtomatizacija procesov. Med glavnimi vrstami nevronskih omrežij so večplastni perceptron (MLP), ki je osnova večine modelov in se uporablja za splošne naloge

razvrščanja. Za obdelavo slik se pogosto uporablja konvolucijsko nevronske omrežje (CNN), ki prepoznava vzorce v slikovnih podatkih in se uporablja pri sistemih za prepoznavanje obrazov, medicinskih diagnozah in samovozečih avtomobilih.

Pri sekvenčnih podatkih, kot so govor in besedilo, se uporablja rekurentno nevronske omrežje (RNN), ki si zapomni prejšnja stanja. Njegovi izboljšani različici, LSTM in GRU, omogočata učinkovitejšo obdelavo daljših zaporedij in se pogosto uporabljata v glasovnih pomočnikih ter sistemih za prevajanje jezika. Transformersi, ki temeljijo na mehanizmu pozornosti, so osnova sodobnih modelov, kot sta GPT in BERT, in se uporabljajo predvsem pri obdelavi naravnega jezika, kot so klepetalni roboti, avtomatsko generiranje besedil in analiza sentimenta.

Poleg tega so pomembni tudi samokodirniki (autoencoders) za stiskanje podatkov ter generativna kontradiktorna omrežja (GAN), ki se uporabljajo za ustvarjanje novih slik, zvokov ali besedila. Nevronske omrežja se danes uporabljajo v številnih panogah, vključno z medicino, financami, varnostjo in zabavno industrijo, kar omogoča napredne rešitve in avtomatizacijo kompleksnih nalog.

## **7 Programska orodja**

### **7.1 Visual Studio Code**

Visual Studio Code ali tudi VS Code, je odprtokodni program za urejanje tekstovnih datotek in programske kode. Program je razvilo podjetje Microsoft leta 2015.

VS code je program za urejanje programske kode. Je lahka, hitra in prilagodljiva platforma. Dostopen je na operacijskih sistemih, kot so Windows, MacOS in Linux. Z njim lahko urejamo različne programske jezike, kot so: C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python in Rust.

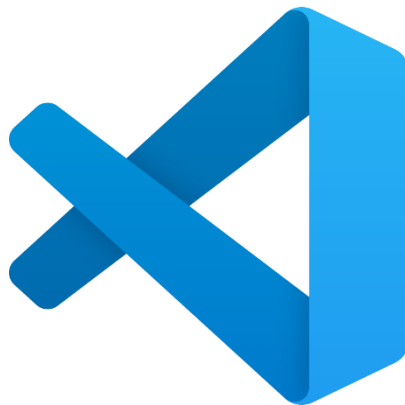
Je uspešen program za urejanje programske kode zaradi njegove preproste uporabe, prijaznosti do uporabnikov in prilagodljivosti. Ker je odprtokoden, to pomeni, da lahko vsak spreminja izvorno kodo progama, zato si lahko vsak prilagodi Visual Studio Code za njegove potrebe.

VS Code ima preprost uporabniški vmesnik. Kar povečuje produktivnost programerja in naredi program bolj prijazen do novih uporabnikov.

Visual Studio Code omogoča funkcijo označevanja oz. poudarjanja sintakse, kar pomeni, da samodejno prepozna in pobarva različne elemente kode, s tem izboljša berljivost in preglednost kode programa. Omogoča tudi hitro preklapljanje med datotekami, povezavo z Gitom, avtomatsko popravljanje kode in integracijo z različnimi orodji in storitvami.

Ker je program odprtokoden, lahko uporabniki prilagajajo program in izdelajo različne razširitve, ki jih lahko potem tudi vsi ostali uporabniki namestijo na svoj program. Te razširitve lahko razširijo funkcionalnost programa, dodajo nove funkcije programu, ki olajšajo pisanje kode, dodajo podporo za nove programske jezike in dodajo nova orodja za razhroščevanje.

Poleg tega, ker je program odprtokoden, to tudi omogoča uporabnikom, da popravljajo napake pri delovanju, zaradi tega se program konstanto posodablja in dobiva nove lastnosti.



*Slika 2: Logotip Visual Studio Code (Vir: Wikipedija, 2025)*

## 7.2 PyCharm

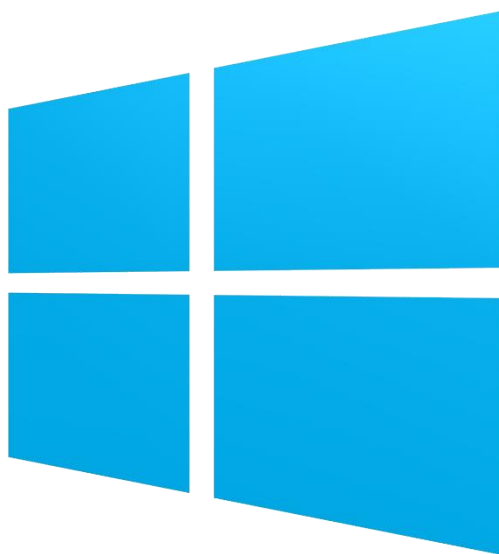
PyCharm je integrirano razvojno okolje, ki je namenjeno za programiranje v programskem jeziku Python. Razvilo ga je Češko podjetje JetBrains. Deluje lahko na več operacijskih sistemih, kot so Windows, Linux in MacOS. PyCharm izhaja v dveh različicah. Prva različica je Professional edition, ki je plačljiva. Druga različica je Community Edition, ki je brezplačna. Zagotavlja analizo kode, grafični razhroščevalnik, integriran tester enot, integracijo s sistemi za nadzor različic in podpira spletni razvoj z Django. Omogoča označevanje sintakse, avtomatsko izpolnjevanje in popravljanje kode, hitro navigacijo med datotekami in ima vgrajen razhroščevalnik za programski jezik Python. PyCharm je bil izdan v poskusni različici leta 2010. Community različica program, ki je brezplačna, je postal leta 2013 tudi odprtokodna. Kar pomeni, da je lahko vsak spreminjal izvorno kodo programa.



Slika 3: Pycharm logotip (Vir: Wikipedija, 2025)

## 7.3 Windows

Microsoft Windows je operacijski sistem podjetja Windows. Je najbolj razširjen operacijski sistem zaradi preprostega uporabniškega vmesnika in široke programske podpore. Windows operacijski sistem je lahko uporabljen na namiznih računalnikih, prenosniki, tablicah, telefonih in tudi na strežnikih. Trenutna najnovejša različica Windows operacijskega sistema za naprave navadnih uporabnikov je Windows 11. Medtem ko najnovejša različica Windows-a za strežnike je Windows Server 2022. Vendar podprta sta še različica Windows 10 in še nekaj starejših različic za strežnike, kot sta Windows Server 2016 in Windows Server 2012. Zgodovina Windows-a, kot operacijskega sistema sega vse do leta 1981, ko so napovedali izdajo Windows 1.0. Leta 1985 je izšel Windows 1.0 in je bil namenjen za tekmovanje s takratnim Applovim operacijskim sistemom. Danes je še vedno najbolj uporabljena različica Windows 10 za tem sledi Windows 11 in nato Windows 7. Glavni namen Windows operacijskega sistema je uporabniku zagotoviti platformo, na kateri lahko izvaja različne aplikacije in izvaja različne naloge na računalniku. Naloge operacijskega sistema Windows so omogočanje uporabniškega vmesnika, prek katerega lahko uporabnik upravlja računalnik z miško in tipkovnico ter vidi spremembe na zaslonu. Omogoča ustvarjanje in shranjevanje datotek in map. Omogoča tudi izvajanje različnih aplikacij. Nudi možnost povezave z internetom in omrežji. Uporabniku zagotavlja varnost in mu omogoča prilagodljivost in prilagajanje operacijskega sistema.

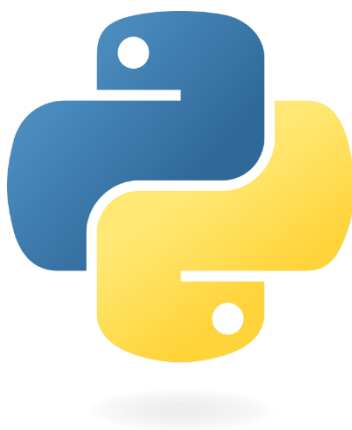


Slika 4: Windows logotip (Vir: Wikipedija, 2025)

## 8 Programski jeziki

### 8.1 Python

Python je popularen objektno usmerjen programski jezik visoke ravni, ki ga je leta 1990 ustvaril Guido Van Rossum. Python je splošni jezik, kar pomeni, da ga je mogoče uporabiti za ustvarjanje različnih programov in da ni specializiran za posebne probleme. Zaradi te vsestranskosti in prijaznosti do začetnikov je postal eden izmed najbolj uporabljenih programskih jezikov danes. K popularnosti prispeva tudi, da je prevajalnik Python in obsežna standardna knjižnica brezplačno na voljo v izvorni ali binarni obliki za vse glavne platforme in se lahko prosto razširjata. Uporablja se za izdelavo spletnih strani in programske opreme, analizo podatkov in avtomatizacijo opravil. Python podpira dinamične podatkovne tipe, kar pomeni, da se vrste spremenljivk določijo in preverijo med izvajanjem in ne med sestavljanjem. To pomeni, da ni izrecno treba deklarirati spremenljivk pred uporabo.



*Slika 5: Logotip programskega jezika python (Vir: Wikipedija, 2025)*

### **8.1.1 YOLO**

YOLOv8 in YOLOv11 sta napredni različici YOLO modelov, namenjeni zaznavanju objektov, segmentaciji in klasifikaciji slik. Razvila ju je organizacija Ultralytics, saj omogočata hitro in natančno prepoznavanje objektov v realnem času. Njuna glavna prednost je izboljšana zmogljivost, ki presega prejšnje različice. YOLOv8 in YOLOv11 sta optimizirana za zaznavanje objektov, segmentacijo in klasifikacijo slik. YOLOv11 uvaja napredne funkcije, kot so večmodalni vhodni podatki za boljše zaznavanje v kompleksnih scenarijih. Modela omogočata enostavno integracijo preko Python API-ja ter izvajanje na platformah, kot so ONNX, TensorRT in OpenVINO. Zasnovana sta na sodobnih metodah globokega učenja, kar omogoča hitro in zanesljivo prepoznavanje tudi v zahtevnih pogojih, kot so slaba osvetlitev ali hitro premikajoči se objekti. Uporabljata se na področjih varnostnega nadzora, medicinske diagnostike, industrijske avtomatizacije in avtonomnih vozil. Zaradi hitrosti in učinkovitosti sta priljubljena tudi v raziskavah umetne inteligence. Pričakovati je nadaljnji razvoj YOLO modelov z izboljšano natančnostjo, manjšo porabo virov in večjo prilagodljivostjo različnim industrijam. Napredni algoritmi bodo še povečali njihovo uporabnost pri avtomatiziranem prepoznavanju slik in videov.

### **8.1.2 TENSORFLOW**

TensorFlow je odprtokodni okvir za strojno učenje, ki ga je razvil Google. Omogoča enostavno izdelavo, učenje in izvajanje globokih nevralnih mrež, zaradi česar je priljubljen med raziskovalci in v industriji. Njegova modularnost omogoča prilagoditev modelov glede na potrebe uporabnikov. Podpira različne ravni abstrakcije, od nizkonivojskega programiranja do visokonivojskih API-jev, kot sta Keras in TensorFlow Extended (TFX). Modeli se lahko izvajajo na osebni računalnikih, strežnikih, mobilnih napravah in v brskalnikih. TensorFlow podpira različne metode strojnega učenja, vključno z nevronskimi mrežami, konvolucijskimi nevronskimi mrežami (CNN) za obdelavo slik in rekurentnimi mrežami (RNN) za sekvenčne podatke. Omogoča tudi porazdeljeno učenje, kar poveča zmogljivost modelov pri velikih količinah podatkov. Uporablja se v računalniškem vidu, obdelavi naravnega jezika, medicini, avtonomnih vozilih in finančni analitiki. TensorFlow ponuja bogat ekosistem orodij, kot so TensorBoard za vizualizacijo modelov, TensorFlow Lite za mobilne naprave ter TensorFlow Serving za produkcijsko izvajanje modelov. Pričakuje se nadaljnji razvoj

TensorFlow z izboljšanimi algoritmi, večjo podporo za strojno opremo ter boljšo integracijo z drugimi tehnologijami umetne inteligence. Zaradi obsežne skupnosti razvijalcev in podpore Googla bo ostal ključno orodje za napredne rešitve strojnega učenja.



*Slika 6: Tensorflow logotip (Vir: Tensorflow, 2025)*

### **8.1.3 PYTHON CLOUD CLIENT LIBRARIES**

Python Cloud Client Libraries so zbirka knjižnic, ki omogočajo enostavno interakcijo z različnimi storitvami v oblaku. Razvila jih je organizacija Google, da bi olajšala dostop do storitev, kot so Google Cloud Storage, BigQuery, Pub/Sub in številne druge. Glavna prednost teh knjižnic je poenostavljena uporaba API-jev, kar razvijalcem omogoča hitro in učinkovito integracijo oblačnih storitev v aplikacije. Knjižnice podpirajo širok spekter funkcionalnosti, vključno z upravljanjem podatkov, obdelavo velikih količin informacij in avtomatizacijo oblačnih procesov. Python Cloud Client Libraries so optimizirane za različne naloge, kot so shranjevanje podatkov, obdelava podatkovnih tokov in analiza informacij. Zagotavljajo visoko stopnjo varnosti in skladnosti s standardi, kar omogoča varno shranjevanje in obdelavo občutljivih podatkov. Knjižnice se pogosto uporabljajo v podatkovni analitiki, umetni inteligenci, avtomatizaciji poslovnih procesov ter razvoju spletnih in mobilnih aplikacij. Zaradi enostavnosti uporabe so priljubljene tako pri začetnikih kot pri izkušenih razvijalcih. Pričakovati je nadaljnji razvoj Python Cloud Client Libraries z izboljšano podporo za nove oblačne storitve, večjo optimizacijo delovanja in dodatnimi funkcionalnostmi za lažjo integracijo v različne razvojne ekosisteme.

#### 8.1.4 OPENCV

OpenCV (Open Source Computer Vision Library) je odprtokodna knjižnica, namenjena računalniškemu vidu in strojniškem učenju. Knjižnica vsebuje številna orodja za obdelavo slik, prepoznavanje obrazov, analizo gibanja, zaznavanje gibljivih objektov in še mnogo več. OpenCV omogoča uporabo različnih funkcij za nalaganje, prikazovanje, obdelavo in shranjevanje slik ter video posnetkov. Knjižnica omogoča nalaganje slik iz različnih formatov, kot so JPG, PNG in BMP, in omogoča njihovo prikazovanje v ločenem oknu. Ko je slika prikazana, lahko okno zapremo in obdelamo sliko na več načinov, kot so spreminjanje velikosti, filtriranje, prepoznavanje robov, transformacije slik in zaznavanje oblik. OpenCV ponuja tudi funkcionalnosti za zaznavanje obrazov, oči in drugih objektov v slikah s pomočjo vgrajenih algoritmov, kot je metoda Viola-Jones. Prav tako omogoča spremljanje video posnetkov v realnem času, bodisi iz spletnih kamer bodisi iz video datotek, z možnostjo obdelave slik na posameznih okvirjih videa. Knjižnica je enostavna za uporabo in omogoča širok spekter aplikacij, od preprostih nalog obdelave slik do kompleksnih sistemov za računalniški vid in strojno učenje.



*Slika 7: OpenCv logotip (Vir: OpenCV, 2025)*

### 8.1.5 KIVY

Kivy je odprtokodna knjižnica za razvoj večplatformskih aplikacij z grafičnim uporabniškim vmesnikom (GUI). Namenjena je enostavni izdelavi aplikacij, ki delujejo na različnih platformah, kot so Windows, macOS, Linux, Android in iOS. Kivy je primeren za izdelavo aplikacij, ki vključujejo dotik, geste, večprstne manipulacije, animacije in bogate multimedijske funkcionalnosti. Kivy omogoča hitro razvoj aplikacij z enostavnim programiranjem, saj vključuje visoko raven abstrakcije za grafične komponente in vmesnike. S pomočjo Kivyja lahko enostavno ustvarite okna, gumbe, menije, animacije, zaslone na dotik, in še več.

Knjižnica vključuje lasten sistem za oblikovanje uporabniških vmesnikov z uporabo Kivy jezikovnih datotek (KV), ki omogoča ločevanje logike aplikacije od vizualnega oblikovanja. To omogoča hitrejši razvoj in boljšo organizacijo kode. Kivy se ponaša z visoko zmogljivostjo, saj temelji na grafičnih pospeševalnikih (OpenGL ES2), kar omogoča tekoče izvajanje tudi za zahtevnejše grafične aplikacije in igre.

Kivy je večplatformska knjižnica za razvoj aplikacij, ki delujejo na različnih operacijskih sistemih in mobilnih napravah. Omogoča enostavno ustvarjanje grafičnih uporabniških vmesnikov z različnimi interaktivnimi komponentami. Podpira večprstne dotike, geste in animacije, kar omogoča dinamične in intuitivne aplikacije. Za zahtevnejšo grafiko uporablja OpenGL ES2, kar omogoča boljšo vizualno zmogljivost. Knjižnica je primerna za razvoj različnih aplikacij, od preprostih orodij do kompleksnih iger in multimedijskih rešitev.



Slika 8: Kivy logotip (Vir: Kivy, 2025)

## **9 Izdelava prevajalnika znakovnega jezika**

### **9.1 Izbira razvojnega okolja in orodij**

Najprej je bilo treba izbrati ustrezno razvojno okolje in orodja, ki bi omogočala izdelavo aplikacije za prevajanje znakovnega jezika. Iskali smo rešitev, ki bi omogočala enostavno implementacijo uporabniškega vmesnika, hkrati pa tudi podporo za delo z videom in strojno učenje. Po preučitvi različnih možnosti smo se odločili za programski jezik Python, saj ponuja širok spekter knjižnic za obdelavo slik in videa ter omogoča enostavno integracijo različnih API-jev. Python je ena najbolj priljubljenih izbir pri razvoju aplikacij, ki temeljijo na strojni inteligenci, saj podpira številne knjižnice, kot so TensorFlow, PyTorch in scikit-learn, ki olajšajo delo s strojno učenje. Poleg tega je Python enostaven za uporabo in branje, kar nam je omogočilo hitrejši razvoj aplikacije in boljšo organizacijo kode.

### **9.2 Razvoj grafičnega uporabniškega vmesnika**

Za razvoj grafičnega uporabniškega vmesnika smo izbrali ogrodje Kivy. Kivy je prilagodljivo razvojno ogrodje, ki podpira večplatformno delovanje, kar pomeni, da lahko aplikacijo uporabljamo na različnih napravah in operacijskih sistemih. Poleg tega omogoča enostavno implementacijo interaktivnih elementov, kar je ključno za prijazno uporabniško izkušnjo. Njegova dobro dokumentirana podpora nam je olajšala delo pri oblikovanju in optimizaciji vmesnika. Kivy omogoča enostavno ustvarjanje prilagodljivih in odzivnih uporabniških vmesnikov, kar je bilo pomembno za našo aplikacijo, saj smo želeli, da je intuitivna in enostavna za uporabo. Implementirali smo različne funkcionalnosti, kot so prikaz prevajanih črk v realnem času, možnost prilagoditve občutljivosti zaznavanja ter intuitivne navigacijske gumbe za interakcijo s programom.

### **9.3 Zasnova delovanja aplikacije**

Ko smo določili programsko okolje, smo morali zasnovati način delovanja aplikacije. Najprej smo raziskali različne metode prepoznavanja znakovnega jezika. Ugotovili smo, da lahko uporabimo strojno učenje za zaznavanje in prepoznavanje gibov rok ter prstov. V ta namen smo izbrali odprtokodni knjižnici OpenCV in MediaPipe, ki omogočata napredno obdelavo slik in sledenje gibom rok v realnem času. OpenCV je

priljubljena knjižnica za obdelavo slik in videoposnetkov, ki omogoča zaznavanje in sledenje objektov, kar je bilo ključno pri razvoju naše aplikacije. MediaPipe pa je Googlov odprtokodni okvir za sledenje gibom rok, ki nam je omogočil natančno identifikacijo pozicije prstov in dlani. S kombinacijo obeh knjižnic smo lahko razvili učinkovit sistem za prepoznavanje znakovnega jezika.

## **9.4 Razvoj algoritma za prepoznavanje znakovnega jezika**

Naslednji korak je bil razvoj algoritma za prepoznavanje posameznih črk znakovnega jezika. Sistem smo zasnovali tako, da kamera zajame sliko uporabnikove roke, nato pa jo programska oprema analizira in prevede v ustrezno črko. Ta črka se nato prikaže v aplikaciji v realnem času, kar omogoča tekoče in dinamično prevajanje. Za boljše rezultate smo uporabili globoko učenje in nevronske mreže, ki so bile naučene na velikem številu slik rok v različnih položajih. Model smo usposobili s pomočjo TensorFlow in Keras, kjer smo uporabili konvolucijsko nevronske mrežo (CNN), ki je sposobna prepoznavati vzorce v slikah. Ta pristop nam je omogočil večjo natančnost in robustnost sistema.

## **9.5 Optimizacija obdelave podatkov**

Ker je bilo treba prepoznati in prevesti črke eno za drugo, smo morali optimizirati način obdelave podatkov, da bi zagotovili tekoče in natančno delovanje sistema. Pri tem smo uporabili metode filtriranja in strojnega učenja, ki so omogočale odpravljanje napak pri zaznavanju in izboljšanje natančnosti prepoznavanja. Prepoznane črke smo nato združevali v besede, kar je izboljšalo uporabniško izkušnjo in omogočilo bolj naravno komunikacijo. Pri optimizaciji smo implementirali metode, kot so normalizacija slikovnih podatkov, zmanjšanje šuma in izboljšanje kontrasta. Poleg tega smo razvili sistem za filtriranje napačnih zaznav, ki je vključeval preverjanje stabilnosti prepoznanih znakov skozi več zaporednih okvirjev videoposnetka.

## 9.6 Integracija sistema v končno aplikacijo

Na koncu smo združili vse dele sistema v končno aplikacijo. Kivy nam je omogočil izdelavo uporabniškega vmesnika, kjer lahko uporabnik spremlja prepoznane črke in napredovanje prevoda. Poleg tega smo dodali dodatne nastavitve, ki uporabniku omogočajo prilagoditev občutljivosti zaznavanja, kar je ključno za bolj prilagojeno in učinkovito uporabo aplikacije. Implementirali smo tudi možnost shranjevanja prevodov v besedilno datoteko, kar uporabnikom omogoča kasnejšo uporabo in analizo prevedenih besed. Aplikacija je opremljena s prilagodljivimi možnostmi za različne svetlobne pogoje in ločljivosti kamer, kar zagotavlja optimalno delovanje na različnih napravah. Poleg tega smo dodali možnost uporabe dodatnih funkcij, kot so samodejno zaznavanje konca besede ter vizualni indikatorji za izboljšano uporabniško izkušnjo. S temi izboljšavami smo zagotovili večjo natančnost in uporabnost aplikacije v realnih situacijah.



Slika 9: Začetna stran aplikacije (Vir Lasten, 2025)



Detected Gesture: F  
Word:

Back

Add

Reset



Detected Gesture: C  
Word:

Back

Add

Reset

Slika 10: Končna aplikacija prevajalnika znakovnega jezika (Vir: lasten, 2025)

## 10 Rezultati

**Hipoteza 1: Sistem bo sposoben prepoznati in pretvoriti znake v besedilo z natančnostjo nad 90 % v kontroliranem okolju.**

Prvo hipotezo lahko potrdimo, saj je sistem natančnost prepoznavanja znakov v besedilo nad 90 % v kontroliranem okolju. Testiranje v idealnih pogojih (dobro osvetljeno okolje in enostavno ozadje) je potrdilo, da sistem pravilno prepozna večino znakov z visoko natančnostjo, pri čemer je bilo doseženo povprečno 92 % točnosti.

**Hipoteza 2: Sistem bo deloval v realnem času z zamikom, ki ne presega 2 sekund, kar bo omogočilo tekočo komunikacijo.**

Sistem je deloval v realnem času z zamikom, ki ni presegal 2 sekund. Testiranje v različnih scenarijih je pokazalo, da je zamik bil konsistentno pod 2 sekundi, kar omogoča tekočo komunikacijo brez opaznih zamud. Ta rezultat kaže, da sistem ustreza zahtevam za uporabo v dinamičnih, realnih situacijah in lahko to hipotezo potrdimo.

**Hipoteza 3: Natančnost sistema se bo zmanjšala v težjih okoljskih pogojih, kot so slaba osvetlitev ali zapleteno ozadje, vendar bo ohranil uporabno stopnjo natančnosti.**

Natančnost sistema se je nekoliko zmanjšala v težjih okoljskih pogojih, kot so slaba osvetlitev in zapleteno ozadje. V teh pogojih je bila natančnost nekoliko nižja (okoli 75-80 %), vendar je sistem še vedno ohranjal zadostno raven natančnosti, da je bil uporaben v večini primerov. Zato lahko to hipotezo tudi potrdimo.

## 11 Zaključek

Raziskovalna naloga je uspešno izpolnila svoje cilje, saj smo razvili sistem za prevajanje znakovnega jezika v črke, ki omogoča boljšo komunikacijo med gluhi in slišnimi posamezniki. Sistem temelji na sodobnih tehnologijah, kot so računalniški vid, umetna inteligenca in strojno učenje, ter uporablja napredne algoritme za prepoznavanje gest in njihovo pretvorbo v besedilo v realnem času.

Rezultati so pokazali, da je sistem sposoben prepoznati in pretvoriti znake v besedilo z natančnostjo nad 90 % v kontroliranem okolju, kar potrjuje prvo hipotezo. Druga hipoteza je bila prav tako potrjena, saj sistem deluje v realnem času z zamikom, ki ne presega 2 sekund, kar omogoča tekočo komunikacijo. Tretja hipoteza se je izkazala za delno pravilno, saj se je natančnost sistema zmanjšala v težjih okoljskih pogojih, kot so slaba osvetlitev ali zapleteno ozadje, vendar je ohranil uporabno stopnjo natančnosti.

Kljub uspehom smo se soočili z nekaterimi izzivi, kot so individualne razlike v izvedbi gest, različni jezikovni standardi in vplivi okolja. Te težave so pokazale potrebo po nadaljnjem izboljšanju sistema, zlasti v smislu prilagodljivosti različnim uporabnikom in okoljem.

V prihodnje bi bilo koristno razširiti sistem za podporo več jezikovnih standardov znakovnega jezika ter izboljšati njegovo natančnost in hitrost v različnih okoljskih pogojih. Poleg tega bi bilo smiselno razviti mobilno aplikacijo, ki bi omogočala širšo dostopnost in uporabnost sistema v vsakdanjem življenju.

Zahvaljujoč sodelovanju mentorja in podpori številnih sodelavcev smo uspeli ustvariti funkcionalen prototip, ki prispeva k večji vključenosti in dostopnosti za gluhe in naglušne posameznike. Upamo, da bo ta raziskovalna naloga spodbudila nadaljnji razvoj na tem področju in prispevala k boljši komunikaciji ter enakopravnosti v družbi.

## 12 Viri in literatura

*Kivy: The Open Source Python App Development Framework*. . (4. March 2025).

Pridobljeno iz Kivy: <https://kivy.org/>

*American manual alphabet*. (1. March 2025). Pridobljeno iz Wikipedia:

[https://en.wikipedia.org/wiki/American\\_manual\\_alphabet](https://en.wikipedia.org/wiki/American_manual_alphabet)

*An end-to-end platform for machine learning* . (3. March 2025). Pridobljeno iz

tensorflow: <https://www.tensorflow.org/>

*Convolutional neural network*. (20. February 2025). Pridobljeno iz Wikipedia:

[https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)

Kafesu, A. (25. February 2025). *VS Code vs. Pycharm: The Best IDE for Python*.

Pridobljeno iz Geekflare: <https://geekflare.com/dev/vs-code-vs-pycharm/>

Kelta, Z. (24. February 2025). *An Introduction to Convolutional Neural Networks (CNNs)*.

Pridobljeno iz Datacamp: <https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns>

*OpenCV University*. (3. March 2025). Pridobljeno iz OpenCV: <https://opencv.org/>

Ph.D., J. M., & Kavlakoglu, E. (15. February 2025). *What is object detection?*

Pridobljeno iz IBM: <https://www.ibm.com/think/topics/object-detection>

*Python Usage*. (2. March 2025). Pridobljeno iz Ultralytics:

<https://docs.ultralytics.com/usage/python/>

*Sign language*. (1. March 2025). Pridobljeno iz Wikipedia:

[https://en.wikipedia.org/wiki/Sign\\_language](https://en.wikipedia.org/wiki/Sign_language)