UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET



**Projekat:**

Face recognition, speech recognition, komande glasom

**Predmet:** Veštačka inteligencija

Studenti: Predmetni profesor:

Filip Trajković – 17503 Prof. dr Leonid V. Stoimenov

Veljko Mijović Ušković - 17224

Željko Miloradović - 17254

Niš, decembar 2021.

**Sadržaj**

[**1. Prepoznavanje lica/Face recognition 3**](#_Toc94014025)

[**1.1. O face-recognition modulu 3**](#_Toc94014026)

[**1.2. Zavisnosti I preduslovi korišćenja face-recognition modula 4**](#_Toc94014027)

**2. Zadavanje komandi glasom……………….………………………...…………5**

**2.1 Instaliranje…………………………......……………………………….5**

**2.2 Recognizer Klase…..…………………………………………………...5**

[**2.3. Speech Recognition API………………………………………………………………….6**](#_Toc94014028)

**3. Pretvaranje teksta u govor.................................................................................7**

* 1. **Šta je gTTS?...........................................................................................7**

**3.2 Glavne karakteristike............................................................................7**

**3.3 Instaliranje.............................................................................................7**

**3.4 Korišćenje..............................................................................................8**

**3.4.1 Osnovni poziv gTTS biblioteke....................................................8**

**3.4.2 Pamćenje prevedenog teksta........................................................8**

**3.4.3 Pokretanje mp3 datoteke.............................................................9**

**4. Rad sa fajlom.......................................................................................................9**

* 1. **Metode......................................................................................................9**
     1. **def unesiStanje(kome, writeIn, writePass)…………………9**
     2. **def vratiKljucem(ko, sta)……...……...……………………10**
     3. **def dekodiraj()………………………...……………………10**
     4. **def vratiDict(stringic)………………...……………………10**
     5. **def generisiText(dict).………………...……………………10**

# **Prepoznavanje lica/Face recognition**

Funkcija za prepoznavanje lica odradjena je pomoću face-recognition python biblioteke.  
Prilikom poziva funkcije VideoCapture funkcija iz openCv bilbioteke se poveže na prvi dostupan drajver za kameru koja je trenutno povezana na računar. I pokuša da uslika fotografiju osobe koja koristi računar. Ukoliko openCv ne uspe da se poveze sa drajverom ili nema povezanih kamera, funkcija vraća False, ukoliko je sve prošlo uspesno, slika trenutnog korisnika računara biće sačuvana u folder aplikacije.  
U folderu projekta treba da postoji known-faces folder gde ce se nalaziti frontalno slikane fotografije lica ljudi koji treba da imaju pristup aplikaciji. Funkcije iz face-recongition modula će ulitati slike lica potencijalnih korisnika u program i izvršiće analizu sliku koja uključuje lociranje lica na fotografijama i ekstrakciju biometrijskih informacija o propznatim licima, niz objekata sa ovim informacijama će ostati upmaćne u aplikaciji.   
Ista analiza će se izvršiti i za upravo fotografisanu fotografiju trenutnog korisnika računara. Funkcija compare\_faces vrši uporedjivanje podataka o licu trenutnog korisnika sa podacima lica poznatih korisnika, ukoliko se bar jedno lice poklapa, funkcija će vratiti True i time će signalizirati da trenutno korisnik ima pristup aplikaciji, u suprotnom će vratiti False.

# **O face-recognition modulu**

https://pypi.org/project/face-recognition/

Face-recognition je biblioteka koja sadrži definicije funkcija za prepoznavanje, lociranje, uporedjivanje i modifikaciju lica sa fotografija i video stream-ova. Ova biblioteka koristi dlib biblioteku i njene implementacije funkcionalnosti prepoznavanja lica korišćenjem dubokog učenja. Model za prepoznavanje lica koji se koristi u face-recognition modulu je treniran na veoma širokom skupu podataka za učenje koji uključuju lica ljudi razlilitih facijalnih struktura, polova, rasa i fenotipa. Model za prepoznavanje lica postiže tačnost od 99.38% za lica odraslih ljudi, za lica dece tačnost je nešto manja.

**Ugrađene funkcije koje se koriste u našem projektu:**

* face\_recognition.load\_image\_file(path) – funkcija koja učitava sliku iz file Sistema u program kao objekat.
* face\_recognition.face\_encodings(image) – funkcija koja analizira objekat koji predstavlja sliku sa licem na njoj, pronalazi sva lica na slici I vraća listu objekata koji predstavljaju univerzalni encoding biometrijskih informacija o tim licima. Svaki od tih encoding se može uporedjivati sa drugim licima dok god su u istom formatu, takodje se može modifikovati I tako postići efekat dubokog lažiranja (deep fake).

|  |
| --- |
| * face\_recognition.compare\_faces(known\_faces\_list, unknown\_face\_encoding) |
| - Funkcija koja vrši komparaciju encodiga informacija o različitim licima. Prvi parameter je lista encodinga poznatih lica, dok je drugi parameter encoding lica koje se uporedjuje, ukoliko se testirano lice  poklapa sa bar jednim licem iz liste funkcija će vratiti True, u suprotnom će vratiti False. Ova funkcija se može uz odredjene modifikacije koristiti I za prepoznavanje konkretnih osoba. **Zavisnosti I preduslovi korišćenja face-recognition modula** |
|  |
| Bilbioteka face-recognition je razvijena I testirana na linux sistemima I nije zvanično podržana u windows-u, mada je moguće konfigurisati Python virtual environment tako da ova bilbioteka funkcioniše I na tom operativnom sistemu. Zavisnosti face-recognition modula: Dlib (https://github.com/davisking/dlib)  Boost (https://www.boost.org/)  Cmake (https://cmake.org/)  Potencijalno: openCv (https://opencv.org/)  **Linux:**  Da bi biliboteka bila dostupna na linuxu potrebno je kreirati python virtual enviorment I preko pip package menadzera instalirati sledeće biblioteke.  Metoda 1:   * pip install cmake * pip install boost (neophodno samo za starije verzija dlib-a) * pip install dlib * pip install face-recognition   Metoda 2 (Za linux distribucije koje koriste apt package manager):   * sudo apt-get install build-essentials cmake * sudo apt-get install libgtk-3-dev * sudo apt-get install libboost-all-dev * aktviiraj virtual env * pip install numpy * pip install scipy * pip install scikit-image * pip install dlib * pip install face-recognition   **Windows:** Iako face-recognition modul nije zvanično podržan na windows operativnom sistemu, moguće je izvršiti sledecu konfiguraciju:   * Kreirati novu virtuelnu okolinu za Python projekat * pip install cmake * pip install boost * skinuti source kod sa github stranice face-recognition biblioteke * pokrenuti setup.py instalaciju face-recognition paketa koja se nalazi u repozitorijumu paketa |
| 1. **Zadavanje komandi glasom**   Biblioteka za izvođenje prepoznavanja govora, sa podrškom za nekoliko mašina i API-ja, onlajn i oflajn. |
| * 1. Instaliranje   Instaliranje se vrši pozivom komande:  Linux:   * pip install SpeechRecognition. |
| Windows:   * python -m pipwin install SpeechRecognition (u ovom slučaju morate prethodno da instalirate pipwin )   1. Recognizer Klase   SpeechRecognition biblioteka ima mnogo klasa, ali u našem projektu smo se fokusirali na klasu koja se zove Recognizer. Ovo je klasa koja nam omogućava da pretvorimo audio datoteke u tekst. Recognizer klasa se uključuje u projekat preko speech\_recognition biblioteke.  Instanciranje Recognizer klase se vrši pozivom:   * import speech\_recognition as sr |
| * recognizer = sr.Recognizer() |
| Recognizer klasi mozemo najpre postaviti vrednost jačine mikrofona (“threshold”) tačnije prag čujnosti mikrofona ispod koje se nece detektovati nikakav zvuk. Ova mogućnost pospešuje prepoznavanje samo jakih glasova bez uzimanja u obzir šumove I niske tonove.  Sledećom komandom postavljamo ovu vrednost na 300:   * recognizer.energy\_threshold = 300 |

# Speech Recognition API

Speech recognition biblioteka koristi različite API-je u prepoznavanju govora. Mogućnosti koje su dostupne koriščenjem ove biblioteke su.

* Bing Recognizer - koristi Microsoft-ove kognitivne usluge.
* Google Recognizer - koristi Google-ov besplatni API za web pretraživanje.
* Google Cloud Recognizer - koristi Google-ov API za govor u oblaku.
* Wit Recognizer - koristi platformu wit.ai.

U našem projektu smo koristili Google Recognizer API obzirom da smo aplikaciju primenili za korišćenje za srpsko I englesko govorno područje.

Sledeći deo koda predstavlja način na koji smo koristili biblioteku Speech Recognition:

with sr.Microphone() as source:

audio = recognizer.listen(source)

recognizer.recognize\_google(audio)

* sr.Microphone()
  + Ovaj poziv služi za pribavljanje dostupnog mikrofona preko aplikacije kao izvora za osluškivanje govora.
* recognizer.lister(source)
  + Poziv listen metode za osluškivanje zvuka sa mikrofona prosleđenog preko source-a što je u našem slučaju mikrofon; Povratna vrednost ove funkcije se pamti u privremenoj promenljivoj audio.
* recognizer.recognize\_google(audio, <language= ‘sr-SP’ | ‘en-US’>)
  + Ovom linijom se poziva Google Recognizer API. Kao parametar se prosleđuje audio promenljiva i kao mogući parametar se prosleđuje ‘language’ sa setovanim vrednostima ‘sr-SP’ ili ‘en-US’. Ova promenljiva se koristi radi boljeg

detektovanja unete komande. Ukoliko je na ulazu kao komanda doveden audio zapis sa srpskim jezikom, neophodno je da ‘language’ promenljiva obavezno bude postavljena za srpskom govorno područje. U suprotnom se neće jasno identifikovati srpske reči i neće biti prepoznata ulazna rečenica. Ukoliko ‘language’ promenljiva nije navedena, Google API će koristiti engleski jezik kao default parameter za prepoznavanje govora.

1. **Pretvaranje teksta u govor**

Za pretvaranje teksta u govor, koristili smo gTTS biblioteku. Ova biblioteka omogućava da bilo koji tekst na ulazu uz pomoć promenljiih poput tipa jezika sintetiše govor i zapamti ga u određeni audio fajl.

* 1. Šta je gTTS?

**gTTS (Google Text-to-Speech) je Python biblioteka i CLI alat za povezivanje sa Google Translate API-jem za pretvaranje teksta u govor. Tekstualne zapise je moguće konvertovati u mp3 podatke u datoteku ili objekat sličan fajlu (bytestring) za dalju audio manipulaciju ili stdout. Ili jednostavno unapred generišite Google Translate TTS URL-ove zahteva za prenos na drugim programima na upotrebu.**

**Dokumentacija za ovu biblioteku se nalazi na: http://gtts.readthedocs.org/**

* 1. Glavne karakteristike
* Prilagodljivi tokenizer rečenica specifičan za govor koji omogućava čitanje neograničenih dužina teksta, uz zadržavanje pravilne intonacije, skraćenica, decimala i još mnogo toga
* Prilagodljivi pretprocesori teksta koji mogu, na primer, da obezbede ispravke izgovora
  1. Instaliranje

Instaliranje se vrši komandom:

* Linux: pip install gTTS
* Windows: python –m pipwin install gTTS
  1. Korišćenje:
     1. Osnovni poziv gTTS biblioteke

myobj = gTTS(text=text, lang=language, slow=speed)

Parametri ovog poziva:

* text
  + Ulazna sekvenca teksta koji je potrebno konvertovati u govor
* language
  + Ulazni tip jezika na kojem je potrebno izvršiti konverziju teksta u govor
* speed
  + Brzina izgovaranja prilikom konverzije
    1. Pamćenje prevedenog teksta

myobj.save("<naziv datoteke>".mp3")

Jednostavnim pozivom save metode vršimo pamćenje konvertovanog teksta u određenu mp3 datoteku. Naziv datoteke je moguće modifikovati na različite načine kako bismo zapamtili veći broj sintetisanog teksta.

* + 1. Pokretanje mp3 datoteke

playsound.playsound('<nazivDatoteke>.mp3', True)

Za pokretanje mp3 datoteke koristimo playsound biblioteku koja otvara mp3 datoteku, reprodukuje je i zatvara nakon završetka reprodukcije. Zbog učestalih poziva ove metode, u našem projektu posedujemo više mp3 datoteka kako ne bi došlo do zakrčenja i „botlle neck“ prilikom učestalog otvaranje jedne iste datoteke.

1. **Rad sa fajlom**

Radsafajlom.py - je skripta koja sluzi za rad sa enkriptovanim tekstualnim fajlom. Tekstualni fajl sadrzi u prvom redu tzv. Fernet kljuc koji sluzi za dekripciju sifrovanog teksta, nakon toga nalaze se sifre koje su od znacaja korisnicima. Struktura dekriptovanog teksualnog fajla je sledeca:

1. Fernet kljuc
2. ime korisnika kome su namenjene sifre
3. naziv platforme kojoj je sifra namenjena +" " + sama sifra
4. koraci 2 i 3 se ponavljaju 3 puta obzirom da imamo 3 korisnika u ovom slucaju Zeljko, Veljko i Filip

Tri korisnika su u pitanju obzirom da ove funkcije mogu da se pozovu od strane korisnika cije se slike nalaze u folderu known-faces.

* 1. **Metode**
     1. **def unesiStanje(kome, writeIn, writePass)**

Funkcija za unos korisnikove sifre za odredjenu platformu radi tako sto se otvara tekstualni dokument, u prvom redu se nalazi Fernet kljuc za dekripciju tesktualnog fajla, nakon generisanja dekriptora pomocu kljuca radimo dekripciju ostatka fajla. U ostatku fajla se nalaze podaci od znacaja za korisnika. Nakon dekripcije ostatka, pronalazimo gde je potrebno da se unesu novi podaci za kog korisnika konkretno. Nakon unosa podatka, string se enkriptuje uz pomog istog prethodno generisanog dekriptora. Sve to zajedno sa kljucem ponovo unosimo u tekstualni dokument i zatvaramo ga. U slucaju uspesno izvrsene operacije vratice se True u drugom slucaju funkcija ne vraca nista.

Parametri:

* kome - filip, zeljko ili veljko jer su nase slike ubacene vec u folder known-faces
* writeIn - naziv platforme za koju se sifra unosi
* writePass – sifra
  + 1. **def vratiKljucem(ko, sta)**

Funkcija sluzi za citanje sifre za odredjenog korisnika za odredjenu platforumu koju je korisnik zatrazio pozivom ove funkcije. Dekripcija i enkripcija je potpuno ista kao u prethodnoj funkciji. Razlika od prethodne je ta sto se sav sadrzaj preslikava/mapira u dictionary, kako bi lakse pretrazili da li se zahtevani podaci nalaze u sifrovanom tekstualnom fajlu. U slucaju da fajl ne postoji, korisnik nije pronadjen ili platforma ne postoji vraca se touple sa False i praznim stringom, u drugom slucaju vraca se touple True i string koji predstavlja trazenu sifru.

Parametri:

* ko - filip, zeljko ili veljko jer su nase slike ubacene vec u folder known-faces
* sta - naziv platforme cija se sifra trazi
  + 1. **dekodiraj()**

Pomocna funkcija za dekodiranje tekstualnog fajla, kako ne bi doslo do dupliranja koda, poziva se u obe glavne funkcije. Kao povratne parametre ima touple koji vraca dekodirani string i drugi parametar touple-a je kljuc za generisanje Fernet dekodera.

* + 1. **def vratiDict(stringic)**

Pomocna funkcija koja na osnovu parametra string generise dictioanry, dictionary je takvog formata da su kljucevi nazivi korisnika, svaki value korisnika u sebi ima takodje dictionary, taj dictioanry kao kljuceve sadrzi platforme, dok su values sifre tih platformi.

* + 1. **def generisiText(dict)**

Pomocna funkcija koja na osnovu dictionary koji se prosledjuje kao parametar, vraca string koji je spreman za unos u tekstualni fajl nazvan passw.txt.