

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Nikola Čuljat, Emanuel Guberović, Tin Trčak, Filip Turčinović

**PERSONALIZIRANO PAMETNO ZRCALO S
IMPLEMENTIRANIM SUSTAVOM ZA DETEKCIJU I
PREPOZNAVANJE LICA**

Zagreb, 2018.

Ovaj rad izrađen je na Zavodu za radiokomunikacije pod vodstvom prof. dr. sc. Sonje Grgić i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2017./2018.

Popis kratica

RPi	<i>Raspberry Pi</i>	računalo Raspberry Pi
IoT	<i>Internet of Things</i>	Internet stvari
OSS	<i>Open Sound System</i>	otvoreni zvučni sustav
ALSA	<i>Advanced Linux Sound Architecture</i>	napredna zvučna arhitektura Linuxa
LBP	<i>Local Binary Pattern</i>	lokalni binarni uzorak
LDA	<i>Linear Discriminant Analysis</i>	linearna diskriminantna analiza
PCA	<i>Principal Component Analysis</i>	analiza glavnih komponenti

Sadržaj rada

1. Uvod.....	1
2. Opći i specifični ciljevi rada	2
3. Materijal i metode	4
3.1. Raspberry Pi 3 Model B i konfiguracija	4
3.2. Implementacija i konfiguriranje MagicMirror ²	6
3.3. Generiranje konfiguracijske datoteke	11
3.4. Implementacija potrebnih paketa za prepoznavanje govora	12
3.4.1. Implementacija i konfiguriranje PocketSphinx.....	13
3.4.2. Pytsx3 paket	15
3.5. Biblioteke za algoritam računalnog vida.....	15
3.5.1. Metode detekcije i prepoznavanja lica	17
3.5.2. Detekcija lica s Haarovom metodom.....	17
3.5.3. Prepoznavanje lica metodom svojstvenih lica.....	18
3.5.4. Prepoznavanje lica metodom Fisherovih lica.....	20
3.5.5. Prepoznavanje lica metodom lokalnih binarnih uzoraka	23
4. Rezultati	26
5. Rasprava.....	29
6. Zaključak.....	31
7. Zahvale	32
8. Popis literature	33
9. Sažetak	34
10. Summary.....	35

1. Uvod

Pojava kvalitetnijih i dostupnijih mikroračunala doprinosi razvoju Interneta stvari (*engl. Internet of Things*, skraćeno *IoT*) koji omogućava komunikaciju putem interneta između uređaja, između stvari (objekata) te komunikaciju stvari s ljudima s ciljem jednostavnijeg, preciznijeg i bržeg obavljanja rutinskih, ali i naprednijih svakodnevnih aktivnosti. Primjer takvog uređaja je pametno zrcalo koje na načelu prividne stvarnosti omogućava korisnicima prikaz određenih informacija na površini zrcala.

Pametno zrcalo sastoji se od jednosmjernog zrcala sa zaslonom i upravljačkim modulom putem kojih se mogu postaviti funkcije koje prikupljaju podatke s interneta i ispisuju ih na površinu zrcala. Ovisno o izradi pametnog zrcala, njime se može upravljati tipkama, glasom i/ili dodirrom.

Sustav pametnog zrcala čiju implementaciju prati ovaj rad koristi mikroračunalo Raspberry Pi 3 Model B kojeg pokreće četverojezgreni procesor s taktom od 1,2 GHz. Osim snage, ovaj model mikroračunala omogućio je lakše spajanje na mrežu putem ugrađenih modula za Wi-Fi i Bluetooth. Implementirano je personalizirano pametno zrcalo koje omogućava prijavu više osoba prepoznavanjem lica, a njegovo upravljanje ostvareno je glasovnim naredbama (prepoznavanjem glasa). Pametno zrcalo na svojoj površini prikazuje sat (vrijeme) i datum, vremensku prognozu za postavljeni grad, elektroničku poštu (*engl. e-mail*) te stanje u prometu za lokaciju prijavljene osobe.

2. Opći i specifični ciljevi rada

Ovaj projekt usmjeren je na konstrukciju i izvedbu pametnog zrcala koje će omogućiti autentifikaciju prepoznavanjem lica te upravljanje radom zrcala govornim naredbama. U ovom radu u izvedbi pametnog zrcala rabiće se računalo Raspberry Pi 3 Model B s odgovarajućim modulima. Izvedeno pametno zrcalo te njegova okolina treba omogućiti personaliziranu uporabu implementacijom modula za prepoznavanje lica i prepoznavanje govora. Projekt bi se mogao podijeliti u tri radne cjeline od kojih bi prva bila konstrukcija pametnog zrcala, implementacija korisničkog sučelja i vanjskih modula.

Fokus rada prve radne cjeline je odabir vanjskih modula koji će moći ispuniti definirane zahtjeve i implementacija korisničkog sučelja za realizaciju funkcionalnosti pametnog zrcala uporabom odgovarajuće programske podrške. U sklopu druge radne cjeline potrebno je izvesti logiku za pametno zrcalo pri čemu će interakcija s logikom biti izvedena govornim putem te pretvorbom govora u tekst. Korisnik će kontrolirati i zadavati naredbe pametnom zrcalu putem govora, tj. ključnim riječima. Tijekom izrade pametnog zrcala potrebno je proučiti načela rada algoritama za prepoznavanje govora te teorijsku pozadinu pretvorbe govora u tekst. Na temelju provedenog istraživanja potrebno je odabrati odgovarajući model i algoritam za prepoznavanje govora i pretvorbu u tekst. Uporabom javno dostupnih servisa provest će se implementacija prethodno definiranih funkcionalnosti koje će pametno zrcalo omogućiti korisniku.

Posljednja radna cjelina projekta je autentifikacija korisnika na spomenuti sustav prepoznavanjem lica. Za prijave korisnika sustav će koristiti sliku korisnika koja će se prethodno nalaziti u galeriji slika. Za realizaciju ovog zadatka potrebno je

proučiti metode za detekciju i prepoznavanje lica te provesti odabir metode koja će ispuniti specifične zahtjeve primjene u izvedbi pametnog zrcala. Bit će izveden modul za prepoznavanje lica te će se dati uvid u biblioteke korištene u implementaciji i integraciji sustava za prepoznavanje lica te moguća alternativna rješenja s navedenim prednostima i nedostacima.

Mikroračunalo Raspberry Pi 3 Model B predstavlja upravljački modul pomoću kojega zrcalo obavlja prethodno navedene funkcije. Uz mikroračunalo, za izradu pametnog zrcala potrebni su monitor (zaslon) i jednosmjerno zrcalo, a dodatno (za funkcije prepoznavanja lica i govora) kamera i mikrofon

Sustav ima četiri stanja:

- Inicijalno (početno) stanje
Personalizirano pametno zrcalo treba stalno biti u pripravnosti i "slušati" ključnu riječ.
- Prepoznavanje lica
Nakon što korisnik kaže postavljenu ključnu riječ, pametno zrcalo uključuje kameru i pokreće detektiranje i prepoznavanje lica. Sustav za prepoznavanje lica koristi prije postavljenu galeriju slika.
- Dohvaćanje podataka
Ukoliko je korisnik prepoznat, sustav iz baze podataka dohvaća njegove podatke.
- Prikaz podataka
Nakon što je dohvatio podatke, sustav generira datoteku koju program pametnog zrcala koristi za prikaz personaliziranih informacija (npr. elektronička pošta).

Glasovnom naredbom (određena ključna riječ) sustav prestaje prikazivati korisničke podatke i vraća se u inicijalno stanje.

3. Materijal i metode

3.1. Raspberry Pi 3 Model B i konfiguracija

Raspberry Pi 3 Model B idealan je kreiranje vlastitih rješenja Interneta stvari. Raspberry Pi 3 ima ugrađenu 802.11 b/g/n WLAN karticu i pruža Bluetooth povezivanje. Pokreće ga Broadcom BCM2837 procesor koji ima četiri 64-bitne ARM Cortex-A53 jezgre radnog takta od 1,2 GHz s 32 kB Level 1 i 512 kB Level 2 cachea. Radna memorija računala je 1 GB DDR2 memorije. Raspberry Pi računalo sastoji se od 4 USB 2.0 priključka, 10/00 Ethernet priključka, HDMI sučelja, 4-polniog 3.5mm utora, CSI kamera sučelja, 5V MicroUSB napajanja, MicroSD sučelja, DSI Display sučelja te 40-pinskog GPIO sučelja.^[1] Raspberry Pi 3 Model B, kamera te mikروفon koji su korišteni za izradu ovog rada prikazani su na slici 1.



Slika 1. Raspberry Pi 3 Model B

RPi računalo po svojoj veličini odgovara veličini kreditne kartice. Dimenzije Raspberry Pi 3 Modela B iznose 8,5 x 5,6 x 1,7 cm.

Kako bi pametno zrcalo izgledalo prirodnije i s obzirom na dimenzije stakla u uporabi, potrebno je korisničkim datotekama Raspberry Pi-a podesiti određene postavke. Prvo je potrebno otvoriti konfiguracijsku datoteku računala s naredbom `sudo nano /boot/config.txt`. U nju je tada potrebno dodati sljedeće linije teksta nakon čega je potrebno ponovno pokretanje računala:

```
display_rotate=1
avoid_warnings=1
```

Kako se na zrcalu ne bi prikazivao pokazivač miša, bila je potrebna instalacija aplikacije „*unclutter*“ koja nakon 3 sekunde nekorištenja miša uklanja pokazivač s ekrana. Naredba za opisanu proceduru je sljedeća:

```
sudo apt-get install unclutter
```

Na kraju datoteke `/etc/xdg/lxsession/LXDE-pi/autostart` upisuje se sljedeće:

```
$ sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
...
@unclutter -display :0 -idle 3 -root -noevents
```

Za dodatnu konfiguraciju računala, potrebno je onemogućiti rad *screensavera* uređivanjem datoteke `~/.config/lxsession/LXDE-pi/autostart` te dodavanjem sljedećih linija na njezin kraj:

```
@xset s noblank
@xset s off
@xset -dpms
```

Također, u istu svrhu je potrebno urediti `/etc/lightdm/lightdm.conf` te dodati sljedeću liniju:

```
xserver-command=X -s 0 -dpms
```

Osim *screensavera* potrebno je onesposobiti „*WiFi Power Save*“ funkciju koju svaku Raspberry Pi ima ugrađenu. Sljedeći koraci vrijede samo za Raspberry Pi 3

Model B koji je korišten u svrhu ovog rada. Potrebno je urediti `/etc/network/interfaces` te dodati sljedeću liniju teksta u „`wlan0`“ dijelu:

```
wireless-power off
```

Nakon ponovnog pokretanja računala više nema osnovnih upisa u datoteci `/etc/network/interfaces` te su umjesto njih umetnuti dodaci u skriptu koja se pokreće prilikom pokretanja. Sljedeće naredbe u terminalu potrebne su za izvedbu opisanih funkcija:

```
cat << EOF | sudo tee /etc/network/if-up.d/off-power-  
manager  
#!/bin/sh  
# off-power-manager - Disable the internal power manager of  
the (built-in) wlan0 device  
# Added by MagicMirrorSetup  
iw dev wlan0 set power_save off  
EOF
```

Podešavanjem dopuštenja pomoću sljedeće naredbe u terminalu završava se ovaj korak u konfiguraciji:

```
sudo chmod 755 /etc/network/if-up.d/off-power-  
manager  
sudo /etc/init.d/networking restart
```

3.2. Implementacija i konfiguriranje MagicMirror²

MagicMirror² je platforma otvorenog koda koja omogućava modularnu nadogradnju pametnog zrcala. S bazom u „*Electron*“ razvojnoj cjelini koja je također otvorenog koda te pomaže pri razvoju aplikacija za web. ^[2]

Koraci i lakoća instalacije platforme ovise o verziji Raspberry Pi-a koji se koristi te također o potrebama korisnika. Raspberry Pi verzija 2 i 3 podržavaju omotače prije

spomenute razvojne cjeline „*Electron*“ dok verzije 0 i 1 ne podržavaju punu verziju platforme već samo mogućnost uloge servera.

Automatska instalacija moguća je preko sljedeće naredbe:

```
bash -c "$(curl -sL
https://raw.githubusercontent.com/MichMich/MagicMirror/master/installers/raspberry.sh)"
```

U svrhu postizanja više razine kontrole prilikom instalacije platforme potrebno je učiniti sljedeće korake:

1. Preuzeti te podesiti odgovarajuću verziju „*Node.js*“¹.
2. Klonirati repozitorij te preuzeti glavnu granu na sljedećoj poveznici <https://github.com/MichMich/MagicMirror>
3. Pozicionirati se u repozitorij s naredbom `cd MagicMirror`
4. Naredbom `npm install && npm start` se pokreće instalacija te sama aplikacija. U slučaju da korisnik želi „*server-only*“ verziju koja se koristi na starijim verzijama Raspberry Pi-a, tada je potrebna naredba `npm install && node serveronly`.

Prilikom korištenja starijih verzija Raspberry Pi-a ili u slučaju da želimo pokrenuti platformu bez aplikacijskog prozora, moguće je pokrenuti pomoću naredbe `node serveronly`. Ovo će pokrenuti server nakon kojeg je moguće otvoriti preglednik po vlastitom odabiru. Ovdje je bitno napomenuti da je na Raspberry Pi-u potrebno dopustiti (*whitelist*) pristup sučelju „*interface/ip*“ u konfiguracijskog datoteci servera, inače neće server imati mogućnost pristupa internetu.^[3] Također je

¹ *Node.js* je runtime okruženje otvorenog koda koje služi za izvođenje JavaScript koda na strani servera preko različitih platformi

potrebno postaviti adresu lokalnog glavnog računala na IP adresu 0.0.0.0 kako bi Raspberry Pi mogao slušati promet na svim sučeljima, ne samo na „localhost“ (osnovno postavljena postavka). Sve spomenuto je moguće sa sljedećom linijom koda u konfiguracijskoj datoteci:

```
var config = {
  address: "0.0.0.0",
  port: 8080,
  ipWhitelist: ["127.0.0.1", "::ffff:127.0.0.1",
  "::1", ::ffff:172.17.0.1"]
};
```

Bitno je napomenuti da je moguće pristupiti serveru s vanjskog računala te da je onda moguće pokrenuti „client-only“ verziju programa na vlastitom, lokalnom računalu. Za ovo je potrebno upisati naredbu:

```
node clientonly --address 192.168.1.5 --port 8080
```

U naredbi je potrebno upisati vlastite podatke, IP adresu te pristupnu priključnicu (*port*) za server.

Nakon instalacije, unutar direktorija *MagicMirror/config* potrebno je umetnuti skriptu *config.js*.^[4] Svojstva unutar koda su koja su promjenjiva i moguća konfigurirati su sljedeća:

- *port* – označava *port* na kojemu se MagicMirror² server izvodi. Osnovno dodijeljena vrijednost je 8000.
- *address* – sučelje IP adrese koje prihvaća konekcije. Osnovno dodijeljena vrijednost je *localhost* kako bi onemogućila pristup ostalim računalima u mreži. Kao što je prethodno navedeno, kako bi se omogućio pristup ostalim računalima u mreži, potrebno je postaviti svojstvo n 0.0.0.0.

- *ipWhitelist* – lista IP adresa s kojih je moguće pristupiti uređaju. Opis dodavanja drugih IP adresa je prethodno detaljnije opisan, no nije spomenuto da u slučaju dodavanja maske pod mreže, tada se isto upisuje na sljedeći način u skriptu ["127.0.0.1", "127.0.0.1/24"] ili na način ["127.0.0.1", ["192.168.0.1", "192.168.0.100"]]. U slučaju upisa „[]“, bit će dozvoljen pristup svim IP adresama.
- *zoom* – omogućavanje skaliranja slike sa željenim zoom faktorom. Osnovno dodijeljena vrijednost je 1.0.
- *language* – svojstvo koje označava jezik sučelja. Moguće vrijednosti su en, nl, ru, fr. Originalna vrijednost je en.
- *timeFormat* – označava vremenski format u kojemu će se ispisivati. . Osnovno dodijeljena vrijednost je 24, a može biti i 12.
- *units* – označava sustavu kojemu će se prikazivati vrijednosti za vremenski modul. Moguće opcije su *metric* ili *imperial*. . Osnovno dodijeljena vrijednost je *metric* što označava metrički sustav.
- *modules* – označava promjenjivo polje s imenima, karakteristikama i uvjetima aktivnih modula. Polje sadrži objekte te će biti detaljnije obrađeno u kasnijim poglavljima.

Moduli su srž platforme te kao takva imaju veliku vrijednost u samom pametnom zrcalu. Prilikom instalacije platforme, dobivaju se osnovni moduli koje svaki korisnik može koristiti i prikazivati.

Svaki modul ima nekoliko opcija koje je moguće prilagoditi u opisu unutar polja „*modules*“ u datoteci *config.js* obrađenoj u prethodnom poglavlju. Opcije su sljedeće:

- *module* – ime samog modula. Također, svaki modul može sadržavati poddirektorije. Primjeri bi bili „*clock*“, „*default/calendar*“ ili „*custommodule/mymodule*“.
- *position* – lokacija na kojoj će se modul prikazivati. Moguće vrijednosti su *top_bar*, *top_left*, *top_center*, *top_right*, *upper_third*, *middle_center*, *lower_third*, *bottom_left*, *bottom_center*, *bottom_right*, *bottom_bar*, *fullscreen_above*, te *fullscreen_below*. iako je ovo područje opcionalno, većini modula je ono nužan uvjet kako bi se prikazala.
- *classes* – dodatne klase koje bi modul mogao obuhvaćati
- *header* – u slučaju prikaza teksta u obliku naslova iznad sadržaja samog modula
- *disabled* – opcija za onesposobljavanje modula. *true* vrijednost se postavlja u slučaju da se želi ukloniti prikaz modula dok on i dalje ostaje zapisan u skripti.
- *config* – opcionalno polje koje ovisi o samom modulu. Svaki modul ima vlastitu dokumentaciju te ukoliko zahtijeva dodatne konfiguracije one se postavljaju unutar ovog polja.

Sljedeći moduli su ugrađeni prilikom osnovnog postavljanja platforme: *Clock*, *Calendar*, *Current Weather*, *News Feed*, *Compliments*, *Hello World*, *Alert*.

Moduli koje su dodatno postavljeni na platformu su *email* i *MMM-GoogleMapsTraffic*.

3.3. Generiranje konfiguracijske datoteke

Prilikom prepoznavanja lica sustav pokreće Python skriptu koja između ostalog služi za generiranje konfiguracijske datoteke prilagođene osobi koja je prepoznata. Sam sustav prepoznavanje lica i njegov način rada bit će detaljnije obrađen u kasnijim poglavljima.

Nakon prepoznavanja lica, sustav dohvaća podatke o korisniku iz baze podataka. U bazi podataka nalazi se tablica *users* u koju su pohranjeni sljedeći podaci: jedinstveni broj korisnika, ime korisnika, prezime korisnika, grad, email, lozinka, lokacija. Na tablici 1. prikazane su atributi tablice *users*.

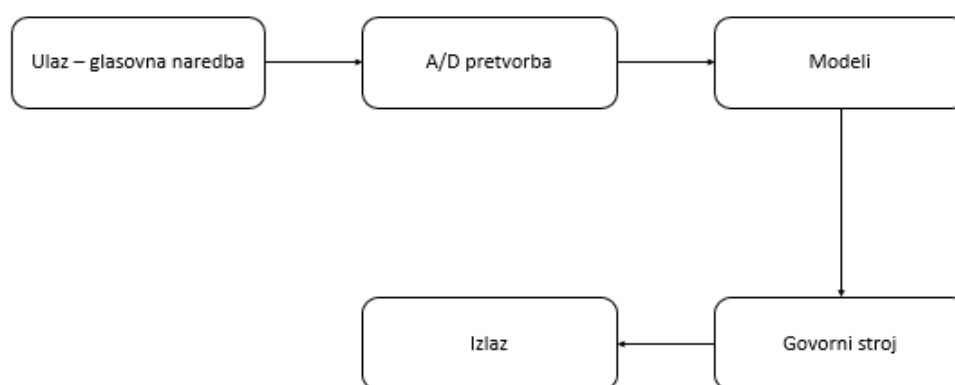
Tablica 1. Tablica *users*

users
id
first_name
last_name
city
email
email_password

Na temelju navedenih podataka, generira se personalizirana konfiguracijska datoteka koja će odgovarati podacima korisnika koji su upisani u bazu podataka. U ovom radu korištena je SQLite relacijska baza podataka. To je baza podataka otvorenog koda koja je temeljena na programskoj biblioteci pisanoj u programskome jeziku C. SQLite ne zahtjeva puno pažnje kod administracije te se često koristi u ugradbenim računalnim sustavima.

3.4. Implementacija potrebnih paketa za prepoznavanje govora

Govor je najbrži i najčešći način komunikacije između ljudi. Brži je od tipkanja te je uz mimiku mnogo učinkovitiji od pisanja. Znanstvenici širom svijeta već desetljećima razvijaju softvere koji bi omogućili da računala prepoznaju ljudski govor. Prepoznavanje govora (engl. *speech recognition*) je proces u kojem računalo transformira govor u tekst. Svoju primjenu prepoznavanje govora našlo je u zdravstvu, vojci, telekomunikacijama, prometu, i slično. Proces prepoznavanja govora prikazan je na slici 2.



Slika 2. Proces prepoznavanja govora

Dakle, nakon što korisnik izgovori glasovnu naredbu sustav za prepoznavanje govora dohvaća glas u analognom obliku. Nakon toga radi se analogno – digitalna pretvorba. Zatim se odabire jedan od modela, najčešće su to akustički ili leksički. Akustički model temelji se na odnosu između jezičnih jedinica govora i audiosignala. Jezični model se određuje na temelju razdiobe vjerojatnosti pojavljivanja riječi unutar rečenice.^[5] Govorni stroj (engl. *Speech Engine*) je softver koji omogućava računalu da vrati tekst u govor te ga prosljeđuje na izlaz – zvučnik.^[6] Također, umjesto toga može se prepoznata riječ ispisati na zaslonu.

U ovom radu prepoznavanje govora koristi se za uključivanje i isključivanje pametnog zrcala. Računalo treba prepoznati ključnu riječ te ovisno o njoj napraviti scenarij koji je unaprijed definiran. Dakle, potrebno je definirati ključne riječi za uključivanje i isključivanje. Prvo RPi računalo osluškuje ključnu riječ za uključivanje sustava. Nakon toga prolazi se proces usporedbe, odnosno prepoznavanja gdje se uspoređuje je li riječ koju je RPi računalo čulo jednako ključnoj riječi. Ukoliko nije ponovno se osluškuje ključna riječ za uključivanje sustava, a ako je RPi računalo čulo ključnu riječ uključuje se grafičko sučelje pametnog zrcala. Nakon uspješnog pokretanja pametnog zrcala, u pozadini se opet osluškuje hoće li korisnik zadati ključnu riječ za isključivanje pametnog zrcala. Kada korisnik izgovori glasovnu naredbu opet se radi proces prepoznavanja govora. Također, ako je bila izgovorena ključna riječ kojom se isključuje pametno zrcalo ono će se isključiti, inače će opet RPi računalo u pozadini osluškivati. U trenutku kada se pametno zrcalo ugasi, opet se u pozadini čeka hoće li korisnik izgovoriti ključnu naredbu za uključivanje.

3.4.1. Implementacija i konfiguriranje PocketSphinx

Na američkom sveučilištu Carnegie Mellon razvijeno je više modela za prepoznavanje govora koji se zovu zajedničkim imenom CMU Sphinx. ^[7] Sphinx je softver otvorenog koda koji služi za prepoznavanje govora. Za potrebe izrade ovog rada korištena je besplatna PocketSphinx programska biblioteka napisana u programskom jeziku C. PocketSphinx je sustav za prepoznavanje govora koji se može koristiti u ugradbenim sustavima i računalima, kao što je Raspberry Pi 3 Model B. PocketSphinx je povezan programskom bibliotekom SphinxBase. SphinxBase služi za zajedničku funkcionalnost u svim Sphinx sustavima te sadrži

sve Sphinx dekodere i različite alate za manipulaciju audiodatotekama i audioznačajkama.

Prvo se na RPI računalo instalira posljednja verzija SphinxBasea i PocketSphinx.

Naredba za preuzimanje SphinxBasea je:

```
wget
https://sourceforge.net/projects/cmusphinx/files/sphinxbase/5prealpha/sphinxbase-5prealpha.tar.gz/download -O sphinxbase.tar.gz .
```

Nakon toga se preuzima PocketSphinx:

```
wget
https://sourceforge.net/projects/cmusphinx/files/pocketsphinx/5prealpha/pocketsphinx-5prealpha.tar.gz/download -O pocketsphinx.tar.gz .
```

Zatim se datoteke raspakiraju u zasebne direktorije:

```
tar -xzvf sphinxbase.tar.gz
tar -xzvf pocketsphinx.tar.gz
```

Poslije toga, treba se ažurirati i dodati napredna zvučna arhitektura Linuxa (engl. *Advanced Linux Sound Architecture* skraćeno, ALSA). ALSA je paket za pružanje audio i MIDI funkcionalnost Linux operacijskom sustavu. Instalacijom ALSA paketa dobiva se: ^[8]

- podrška za sve vrste audio sučelja,
- modularni upravljački program zvuka,
- podrška za otvoreni zvučni sustav (engl. *Open Sound System*, skraćeno OSS),
- biblioteka za pojednostavljenje aplikacijskog programiranja i pružanje funkcionalnosti više razine.

Naredba za instalaciju biblioteke je:

```
sudo apt-get install bison libasound2-dev swig
```

Nakon toga, treba kompajlirati SphinxBasea i PocketSphinx:

```
cd sphinxbase-5prealpha
./configure --enable-fixed
make
sudo make install

cd ../pocketsphinx-5prealpha
./configure
make
sudo make install .
```

3.4.2. Pyttsx3 paket

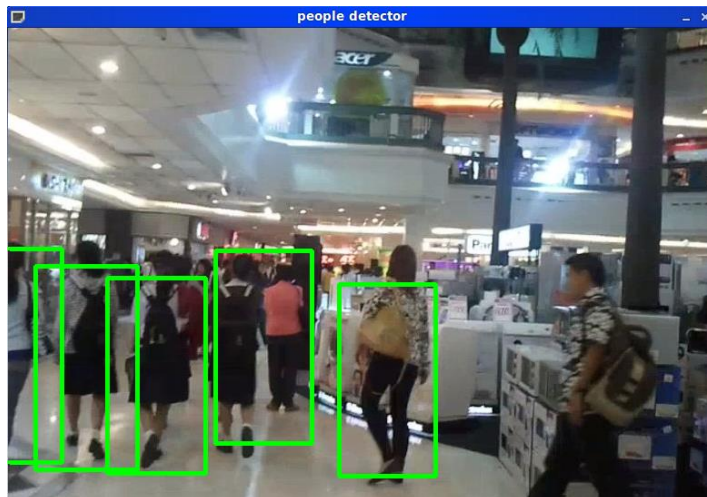
Sinteza govora ili postupak pretvaranja teksta u govor dostupan je u većini operacijskih sustava. Python omotava tekst u govorne funkcionalnosti operativnog sustava ili drugog modula za pretvorbu teksta u govor. Za pretvaranje teksta u glasovne naredbe potrebno je instalirati paket pyttsx3. Pyttsx3 je programska biblioteka koja služi za pretvaranje teksta u govor te je napisana u programskome jeziku Python 2. ^[9] Naredba za instalaciju navedenog paketa je:

```
pip install pyttsx3
```

3.5. Biblioteke za algoritam računalnog vida

Većina algoritama u računalnom vidu razvijena su za računala opće primjene i pisana u višim programskim jezicima. Jedan od izazova primjene na ugradbenim računalnim sustavima je i u optimizaciji tih algoritama za sklopovlje na kojem se primjenjuju.

OpenCV (*Open Source Computer Vision*, skraćeno *OpenCV*) je biblioteka gotovih rješenja za primjenu u računalnom vidu i strojnom učenju. Napisana je u C++-u i njeno sučelje je za C++ aplikacije. Izdana je nad Open Source BSD licencom. ^[10]



Slika 3. Primjena algoritma prepoznavanja ljudi koja se nalazi u OpenCV

Proizvođači uređaja su proizveli optimizirane verzije, tako npr. NVIDIA radi skupa s OpenCV zajednicom na algoritmima ubrzanim na njihovim GPU uređajima. MathWorks pruža MATLAB funkcije/objekte i Simulink blokove za mnoge algoritme te pruža mogućnost primjene istih u optimiziranim verzijama za određeno sklopovlje.

Xilinx je primjer proizvođača koji ima razvijene optimizirane biblioteke koje pruža korisnicima kao Plug and Play IP jezgre za generiranje hardverski ubrzanih algoritama računalnog vida u FPGA-a sklopovima. Primjeri nekih drugih biblioteka su Halcon, Matrox Imaging Library, Cognex VisionPRO, VXL, Cimg, Filters.

U ovom radu je opisana promjena OpenCV biblioteke na RaspberryPI 3 sklopu s web-kamerom i drugom periferijom. Aplikacija će koristiti algoritme za prepoznavanje lica koje su implementirane u FaceRecognizer razredu u OpenCV biblioteci.

3.5.1. Metode detekcije i prepoznavanja lica

Za primjenu računalnog vida u prepoznavanju oblika ljudskog lica potrebno je razdvojiti dva bitno različita pojma: detekcija lica i prepoznavanje lica. Detekcija lica u slici pronalazi gdje se lice nalazi, dok će prepoznavanje lica za detektirano lice pronaći uz određenu točnost najbližu odgovarajuću osobu iz skupa za učenje.

Detekcija lica se obavlja koristeći preporučenu Haarovu metodu nad istreniranim XML klasifikatorom detekcije lica. Prepoznavanje lica se obavlja koristeći implementirane metode poznatih algoritama: algoritam svojstvenih lica, algoritam Fisherovih lica i algoritam utemeljen na određivanju lokalnog binarnog uzorka.

3.5.2. Detekcija lica s Haarovom metodom

Efektivnu metodu detekcije objekata koristeći klasifikatore nad Haarovim značajkama predložili su Paul Viola i Michael Jones. ^[11] To je pristup prepoznavanju objekata strojnim učenjem nad puno pozitivnih i negativnih slika. Haarove značajke su značajke digitalnih slika korištene u prepoznavanju objekata, a naziv im dolazi iz sličnosti s Haarovim valićima. ^[12]



Slika 4. Harove značajke na ljudskom licu

Haarove značajke koriste susjedne pravokutne regije na specifičnim lokacijama u detekcijskom prozoru, izračunavajući sumu te razliku iznosa svjetline svih piksela u svakoj regiji. Kako čak i male slike imaju puno takvih karakteristika, algoritam se ubrzava odabirom najbitnijih karakteristike koje kvalificiraju sliku lica ili ne-lica. Za tu svrhu se koristi AdaBoost mehanizam strojnog učenja, iterativno odabirući najbitnije kvalifikatore.

Dodatno, kvalifikatori se slažu u sve veće i veće kaskade. Tako slike za koje se otkrije uz manje kaskade da ne sadrže lice ne moraju prolaziti kroz složenije kaskade.

3.5.3. Prepoznavanje lica metodom svojstvenih lica

Prepoznavanje lica metodom svojstvenih lica (Eigenfaces) nastoji uhvatiti varijacije u skupini slika i koristiti te informacije za kodiranje i uspoređivanje slika, slika 5.

Svojstvena lica predstavljaju glavne komponente distribucije lica, tj. svojstvene vektore matrica kovarijance nad skupom slika, slika 6. Jedna slika s N piksela je predstavljena kao točka u N -dimenzionalnom prostoru. ^[13]



Slika 5. Uzorci iz CMU PIE baze slika

Kako nam nisu sve dimenzije jednako korisne, koristiti će se one koje imaju najveću promjenjivost u podacima. Analiza glavnih komponenti (engl. *Principal Component Analysis*, skraćeno PCA) je statistička procedura koja koristi ortogonalne transformacije kako bi se skup moguće povezanih varijabli transformirao u manji skup nepovezanih varijabli. PCA metoda pronalazi smjerove najveće promjenjivosti podataka, zvane glavne komponente.



Slika 6. Prosječno lice i svojstvena lica dobivena iz CMU PIE baze slika

Prije generiranja svojstvenih lica, slike lica su normalizirane kako bi oči i usta bili na istim linijama. Zatim se ponovno uzrokuju radi postizanja jednake rezolucije piksela. Svojstvena lica su zatim dobivena iz skupa slika PCA analizom na sljedeći način:

- 1) Imamo M slika lica dimenzije $h \times w$, svaka slika se transformira u vektor veličine $\mathbf{D}(hw)$ i stavlja u skup:

$$\{\Gamma_1, \Gamma_2, \dots, \Gamma_M\}. \quad (1)$$

Svako lice se razlikuje od prosječnog za vektor razlike:

$$\Phi_i = \Gamma_i - \Psi \quad (2)$$

gdje Ψ predstavlja srednju vrijednost skupa $\{\Gamma_1, \Gamma_2, \dots, \Gamma_M\}$.

2) Definiramo matricu kovarijance $\mathbf{C} \in \mathbb{R}^{D \times D}$ kao:

$$\mathbf{C} = \mathbf{A}\mathbf{A}^T \quad (3)$$

gdje $\mathbf{A} = \{\Phi_1, \Phi_2, \dots, \Phi_M\} \in \mathbb{R}^{D \times M}$.

3) Izračun svojstvenih vektora matrice \mathbf{C} je složen zadatak za tipične veličine slike gdje je $D \gg M$. No, za efikasan izračun svojstvenih vektora matrice \mathbf{C} , može se prvo izračunati svojstvene vektore puno manje $M \times M$ matrice $\mathbf{A}^T\mathbf{A}$.

Svojstveni vektor i svojstvene vrijednosti matrice $\mathbf{A}^T\mathbf{A}$ su definirane kao:

$$\mathbf{V} = \{v_1, v_2, \dots, v_r\} \quad (4)$$

i $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_r\}$ gdje je r rang matrice \mathbf{A} .

4) Svojstvene vrijednosti i svojstveni vektori matrice \mathbf{C} su $\mathbf{\Lambda}$ i $\mathbf{U} = \mathbf{A}\mathbf{V}\mathbf{\Lambda}^{-1/2}$, gdje $\mathbf{U} = u_i$ je skupina svojstvenih lica.

3.5.4. Prepoznavanje lica metodom Fisherovih lica

Umjesto razmatranja kombinacija piksela kao promjenjivosti u podacima iz metode svojstvenih lica, metoda Fisherovih lica koristi linearne kombinacije piksela koje se vežu uz promjenjivosti u klasificiranim podacima, u ovom slučaju licima različitih osoba. Informacije vezane uz razlikovanje slika iste osobe se zanemaruju, a prednost se daje informacijama vezanim uz razlikovanje različitih osoba. ^[14]

Kako je bitnija bolja klasifikacija objekata, a ne njihova ponovna reprezentacija, bitno je napraviti što veću razliku između različito klasificiranih objekata, a što manju razliku između jednako klasificiranih objekata. To se postiže diskriminantnom analizom. Najpoznatija diskriminantna analiza je linearna

diskriminantna analiza (engl. *Linear Discriminant Analysis*, skraćeno LDA), koja je izrađena na temelju ideje R.A. Fishera (1936. godine). LDA nastoji pronaći potprostor reprezentacije slika lica, a njezini bazni vektori koji definiraju prostor su poznati kao Fisherova lica. ^[14]

Neka je \mathbf{X} slučajni vektor s uzorcima izvučenim iz c klasa:

$$\mathbf{X} = X_1, X_2 \dots X_c \quad (5)$$

$$\mathbf{X}_i = x_1, x_2 \dots x_n. \quad (6)$$

Matrice raspršenja S_B (između klasa) i S_W (unutar klase) se računaju:

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T \quad (7)$$

$$S_W = \sum_{i=1}^c \sum_{x_j \in X_i} N_i (x_j - \mu_i) (x_j - \mu_i)^T \quad (8)$$

gdje μ srednja vrijednost:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (9)$$

dok je μ_i srednja vrijednost klase :

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j \quad (10)$$

Fisherov klasični algoritam traži projekciju W koja maksimizira kriterij separacije:

$$W_{opt} = \arg \max_w \frac{|W^T S_B W|}{|W^T S_W W|} \quad (11)$$

Rješenje za optimizacijski problem je postignuto generaliziranim izračunom svojstvenih vrijednosti: ^[10]

$$S_B \vartheta_i = \lambda_i S_W \vartheta_i \quad (12)$$

$$S_W^{-1} S_B \vartheta_i = \lambda_i \vartheta_i \quad (13)$$

Postoji još jedan problem koji je potrebno riješiti. Rang matrice S_W je $(N - c)$, s N uzoraka i c klasa. U prepoznavanju uzoraka N je skoro uvijek manji od dimenzije ulaznih podataka (broj piksela) pa matrica raspršenja S_W postaje singularna. ^[15] Opisani problem riješen je primjenom PCA nad podacima i projekcijom uzoraka u $(N - c)$ - dimenzionalni prostor. LDA analiza je zatim korištena na smanjenom skupu podataka, jer S_W više nije singularan. Optimizacijski problem može biti napisan kao:

$$W_{pca} = \arg \max_w |W^T S_T W| \quad (14)$$

$$W_{fld} = \arg \max_w \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \quad (15)$$

Transformacijska matrica W koja projicira uzorak u $(c - 1)$ - dimenzionalan prostor dana je s:

$$W = W_{fld}^T W_{pca}^T \quad (16)$$

Metoda Fisherovih lica rezultira uz specifičnim transformacijskim matricama za pojedine klase objekata i ona nije osjetljiva na uvjete rasvjete kao metoda svojstvenih lica. Diskriminantna analiza umjesto toga određuje svojstva lica koja omogućavaju prepoznavanje između različitih osoba.

Važno je spomenuti da efikasnost algoritma Fisherovih lica također ovisi o skupu slika koji je korišten za treniranje, slika 7. Ukoliko su u skupu za treniranje samo slike s dobrom rasvjetom, a slika koja se rabi za prepoznavanje lica je načinjena u lošim uvjetima rasvjete, metoda će pogrešno odrediti komponente (jer iste komponente ne dolaze do izražaja na slikama nastalim u dobrim i lošim uvjetima rasvjete).

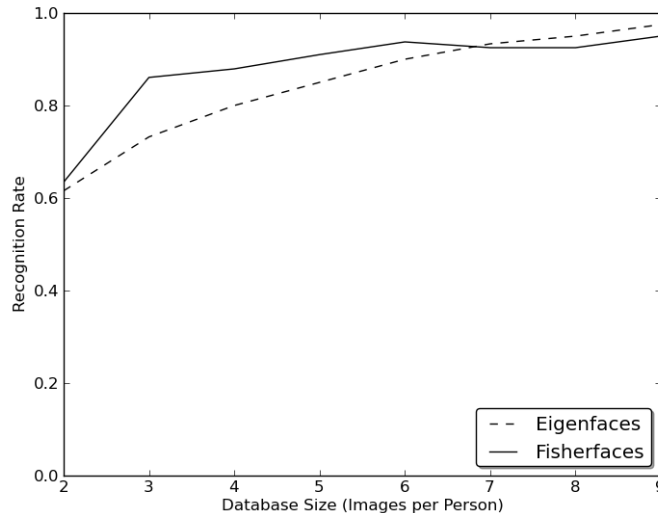


Slika 7. Prva četiri Fisherova lica iz skupa od 100 klasa

Fisherova lica omogućavaju rekonstrukciju slike, ali rekonstrukcija nije tako dobra kao u metodi svojstvenih lica.

3.5.5. Prepoznavanje lica metodom lokalnih binarnih uzoraka

Svojstvena i Fisherova lica koriste holistički pristup prepoznavanju lica. Podaci se tretiraju kao vektor u višedimenzionalnom prostoru. Svojstvena lica maksimiziraju ukupnu raspršenost, što dovodi do problema ako je promjenjivost generirana vanjskim izvorom poput promjene rasvjete. Kako bi se sačuvale razlikovne informacije primjenjuje se LDA analiza i optimizacija (metoda Fisherovih lica). Fisherova lica su bila dobra metoda, no prema grafu iz OpenCV dokumentacije prikazanom na slici 8. za dobro prepoznavanje lica trebamo bar oko osam slika po osobi. Ako imamo samo jednu sliku po osobi tada osjetljivost nije ovisna samo o rasvjeti, već ovisi i o skalabilnosti, translaciji i rotaciji.



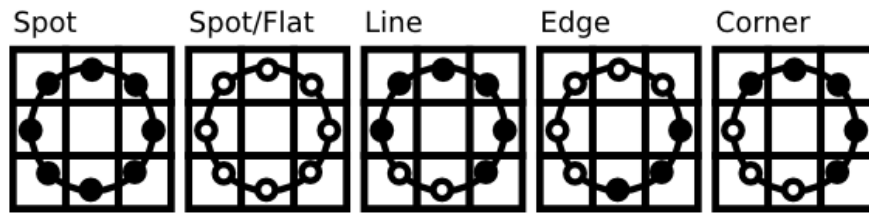
Slika 8. Omjer točnosti prepoznavanja i broja slika po osobi ^[16]

Lokalni binarni uzorak (engl. *Local Binary Pattern*, skraćeno LBP) ima svoje korijene u analizi 2D tekstura. Njena glavna ideja je sumiranje lokalnih struktura u slici uspoređujući svaki piksel sa susjednim. Uzima se piksel u sredini i uspoređuje se njegov intenzitet s intenzitetom susjednih piksela, označavajući rezultat usporedbe s 0 i 1. Rezultat koji se dobiva je binarni broj (npr. 11101101). S osam susjednih piksela bit će 2^8 različitih mogućih kombinacija nazvanih lokalni binarni uzorci ili LBP kodovi. ^[16]

Definicija LBP operatora je dana kao:

$$\text{LBP}(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c) \quad (17)$$

s (x_c, y_c) kao centralnim pikselom intenziteta i_c i i_p kao intenzitetima susjednih piksela, a s je signum funkcija vrijednosti 1 za $x \geq 0$ i vrijednosti 0 za $x < 0$. Na ovaj način omogućeno je prepoznavanje granuliranih detalja u slici. Ideja je poravnati proizvoljni broj susjeda na kružnici promjenjivog radijusa s čime se omogućava dohvaćanje idućih susjedstva, slika 9.



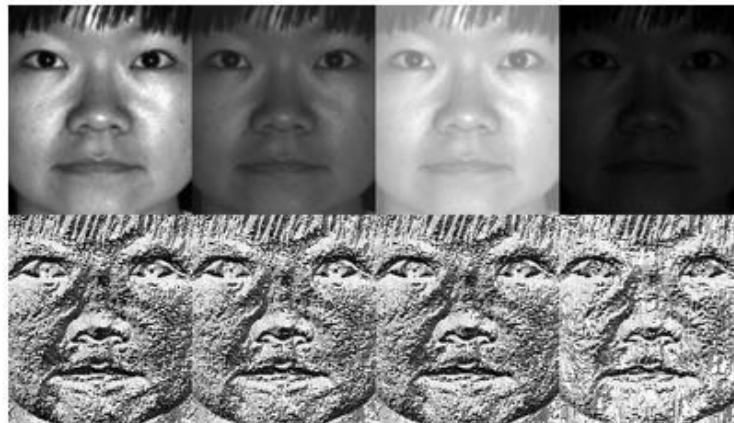
Slika 9. LBP susjedstva unutar radijusa ^[16]

Za danu točku (x_c, y_c) položaj susjeda (x_p, y_p) može se izračunati kao:

$$x_p = x_c + R \cos\left(\frac{2\pi p}{2\pi P}\right) \quad (18)$$

$$y_p = y_c - R \sin\left(\frac{2\pi p}{2\pi P}\right) \quad (19)$$

gdje je R radijus kružnice, a p broj točaka. Prema definiciji, LBP je robusan na jednoliku promjenu srednje razine sivog što je vidljivo na slici 10.

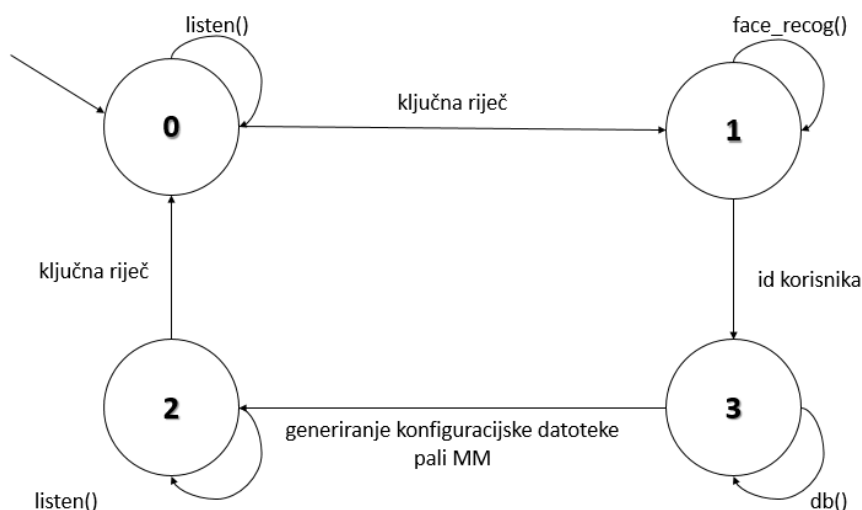


Slika 10. Robusnost LBP-a na promjenu razine sivog ^[16]

Za prostorni prikaz podataka prihvaćena je Ahonenova preporuka. ^[17] Ona obuhvaća raspodjelu LBP slike u lokalne regije i izračun histograma za svaku. Prostorni vektor je dobiven ulančavanjem lokalnih histograma. Ovi se histogrami zovu LBA histogrami te se zato i sama metoda zove LBPH metoda.

4. Rezultati

Rezultat rada je implementirano pametno zrcalo s funkcijama prepoznavanja govora i lica. Sustav omogućava prijavu više korisnika čije podatke drži u lokalnoj bazi podataka. Kao što je opisano u „Općim i specifičnim ciljevima rada“, sustavu se pristupa glasovnom naredbom koja uključuje kameru i započinje prvo detekcija, a potom i prepoznavanje lica. Nakon što je korisnik prepoznat, sustav iz baze podataka dohvaća njegove podatke i spaja ga na mail server kako bi prikazao dolaznu elektroničku poštu (kao primjer osobnih informacija). Uz to, na površini zrcala prikazane su funkcije sata, trenutne situacije u prometu i vremenske prognoze postavljenog grada. Sustav je jednostavan za korištenje i relativno brz s obzirom na složenost funkcija za pretvaranje govora u tekst i prepoznavanje lica. Na slici 11. prikazana su moguća stanja sustava i pripadajuće funkcije.



Slika 11. Moguća stanja sustava i funkcije

Cijeli projekt je pisan u programskom jeziku Python te se izvodi na operacijskom sustavu Raspbian. Za potrebe implementacije preuzete su i unesene razne biblioteke koje omogućavaju normalan rad pametnog zrcala. Projekt je zbog

reinicijaliziranja modula PiCamera kompletno pisan u jednoj skripti koja u beskonačnoj petlji mijenja stanje. Ovisno o stanju program ulazi u definirane funkcije:

listen() - osluškuje ključnu riječ koju kada registriira prelazi u sljedeće stanje (u ovom slučaju ili u stanje u kojem pokreće prepoznavanje lica ili u stanje u kojem gasi sučelje MagicMirror2)

face_recog() - funkcija koja uključuje Raspberry Pi kameru i detektira, a potom i prepoznaje lice korisnika koji se želi prijaviti u sustav, a njegov identifikacijski broj prosljeđuje sljedećoj funkciji

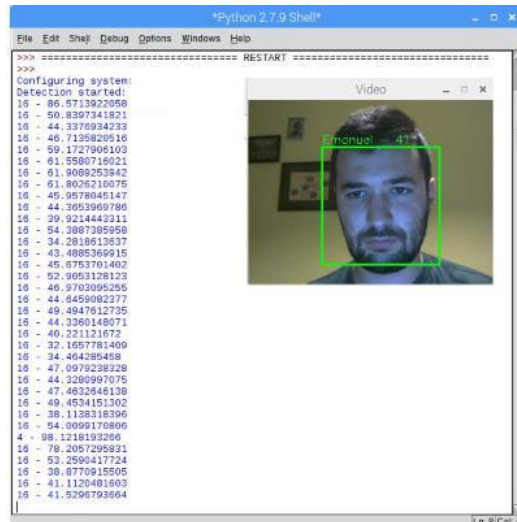
db() - funkcija prima argument (identifikacijski broj korisnika) na temelju kojeg iz baze podataka dohvaća podatke korisnika i generira konfiguracijsku datoteku prilagođenu MagicMirror2 sučelju.

Na slici 12. vidljiv je rezultat točno prepoznate ključne riječi te nakon toga prepoznavanje lica sa jedinstvenom oznakom korisnika 2.

```
Sound1
listening
Got it! Now to recognize it...
You said: magic
Starting face recognition
wait
Configuring system:
Detection started:
2 - 85.30747426215049
Palim Magic Mirror
2 - 85
```

Slika 12. Ispis rezultata

Na slici 13. prikazan je postupak detekcije i prepoznavanja lica. Prvo se detektira lice (zeleni kvadrat) zatim se računa koeficijent sličnosti temeljem prethodno opisanih algoritama.



Slika 13. Detekcija i prepoznavanje lica

Na slici 14. prikazan je krajnji rezultat cjelokupnog rada.



Slika 14. Grafičko sučelje pametnog zrcala

Sustav nakon prepoznavanja lica uključuje personalizirano grafičko sučelje pametnog zrcala za korisnika (u ovom primjeru korisnik je Filip). Također, prikazano je stanje prometa u blizini lokacije pametnog zrcala te ovisno o vremenu sustav ispisuje pripadajuću poruku.

5. Rasprava

Projekt je započeo izborom računala koje će se rabiti kao upravljački modul. Budući da je ideja bila imati i prepoznavanje govora i prepoznavanje lica što je procesorski zahtjevno, mikroračunalo RPi kojeg pokreće 64-bitni Cortex A53 procesor s taktom od 1.2 GHz uz 1 GB radne memorije se istaknulo kao logičan izbor. RPi dolazi s ugrađenim modulom za bežičnu mrežu (Wi-Fi) što je riješilo problem dohvaćanja podataka s interneta. Alternativa odabranom RPi-ju mogu biti sva mikroračunala Raspberry Pi obitelji sljedećih generacija (npr. Raspberry Pi 3 Model B+), a prethodno opisane zahtjevnije funkcionalnosti bi vjerojatno radile i na snažnijim modelima druge generacije Raspberry Pi-ja (Raspberry Pi 2 B i B+), ali bi njihovo izvršavanje bilo nešto sporije. Mikroračunala sličnih karakteristika (npr. ROCK64) bi uz manje preinake programskog koda bila jednako dobra (možda i bolja) u obavljanju definiranog zadatka.

Razvijanjem pametnih zrcala raste i broj otvorenih kodova za njih. Jedan takav, zvan MagicMirror2 za koji su razvijeni razni moduli korišten je u implementaciji ovog projekta. Najbolja alternativa tome je izrada vlastitog sučelja, ali su jednostavnost integracije, otvorenost platforme i unaprijed razvijeni razni moduli prevagnuli u odabiru.

Pri odabiru baze podataka, budući da je za razvijeni sustav bila dovoljna jednostavna tablica s čitanjem upisanih podataka korištena je SQLite koja radi dobro s malom količinom prometa.

Za prepoznavanje govora i pretvaranje zvuka u tekst korišten je PocketSphinx. Sphinx je otvoren kod namijenjen upravo prepoznavanju govora, a PocketSphinx je dio koji se koristi u ugradbenim računalnim sustavima. Alternativa odabranom

može biti Google Voice API koji bi vjerojatno radio bolje, ali se takva usluga naplaćuje dok je PocketSphinx kao otvoreni kod besplatan.

Kao što je prije spomenuto, većina algoritama u računalnom vidu razvijena su za računala opće primjene, no zadnjih godina uložen je velik trud u optimizaciju istih za korištenje na slabijim računalima. Za dio s prepoznavanjem lica korištena je OpenCV biblioteka kreirana za primjenu u računalnom vidu i strojnom učenju koja je zadnjih godina približila algoritme računalnog vida lakšom primjenom i pristupom otvorenog koda.

6. Zaključak

Svrha ovog rada bila je osmišljavanje, dizajniranje i izrada personaliziranog pametnog zrcala koja uključuje implementaciju i integraciju sustava za prepoznavanje lica, sustava za prepoznavanje govora i sučelja pametnog zrcala u funkcionalnu cjelinu. Konačni sustav izveden je na mikroračunalu Raspberry Pi 3 Model B koji omogućava bežično spajanje na internet i dovoljnu procesorsku snagu za obradu zahtjevnih funkcija prepoznavanja lica i govora.

Implementirano pametno zrcalo podržava prijavu više korisnika, a korisnike razlikuje prepoznavanjem lica čime je omogućen prikaz podataka prilagođenih prepoznatom korisniku. Pokretanje i zaustavljanje rada sustava provodi se govornim naredbama. Pametno zrcalo korisniku pruža jednostavan pristup željenim podacima do kojih može doći izgovaranjem ključne riječi koja će pokrenuti rad sustava. Pokreće se sustav za detekciju lica te algoritam za prepoznavanje korisnika. U slučaju uspješnog prepoznavanja, sustav generira personaliziranu konfiguracijsku datoteku temeljem podataka o korisniku u lokalnoj bazi podataka te se na površini zrcala prikazuje grafičko sučelje prilagođeno korisniku.

Implementacija je uspješno realizirana te svi podsustavi usklađeno funkcioniraju zadovoljavajućom brzinom. U usporedbi s postojećim rješenjima pametnih zrcala, implementirani sustav uvodi nova rješenja i tehnike kojima se postiže prepoznavanje lica i govora što rezultira u višem stupnju sigurnosti, automatizacije i personalizacije.

7. Zahvale

Zahvaljujemo se prof. dr. sc. Sonji Grgić koja je sa svojim znanstvenim i stručnim savjetima pomogla u izradi ovog rada.

8. Popis literature

- [1] <https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>, pristupano 12. ožujka 2018.
- [2] <https://github.com/MichMich/MagicMirror/wiki/3rd-Party-Modules>, pristupano 15. ožujka 2018.
- [3] <https://howchoo.com/g/ntcymzbimjv/how-to-install-magic-mirror-on-your-raspberry-pi>, pristupano 18. ožujka 2018.
- [4] <https://magicmirror.builders/>, pristupano 17. ožujka 2018.
- [5] Prachi Khilari, *A review on speech to text conversion methods*, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 4 Issue 7, July 2015, str. 3067 – 3072.
- [6] [https://msdn.microsoft.com/en-us/library/hh378337\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/hh378337(v=office.14).aspx), pristupano 21. ožujka 2018.
- [7] <https://github.com/cmuspinx/pocketsphinx>, pristupano 21. ožujka 2018.
- [8] https://www.alsa-project.org/main/index.php/Main_Page, pristupano 21. ožujka 2018.
- [9] <https://pyttsx.readthedocs.io/en/latest/>, pristupano 4. travnja 2018.
- [10] <https://opencv.org/about.html>, pristupano 10. travnja 2018.
- [11] <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>, pristupano 12. travnja 2018.
- [12] https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html, pristupano 12. travnja 2018.
- [13] <http://www.scholarpedia.org/article/Eigenfaces>, pristupano 14. travnja 2018.
- [14] <http://www.scholarpedia.org/article/Fisherfaces>, pristupano 16. travnja 2018.
- [15] http://www.scholarpedia.org/article/Local_Binary_Patterns, pristupano 17. travnja 2018.
- [16] https://docs.opencv.org/2.4/modules/contrib/doc/face_rec/face_rec_tutorial.html, pristupano 20. travnja 2018.
- [17] Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), str. 469 – 481.

9. Sažetak

Ime i prezime autora:

Nikola Čuljat, Emanuel Guberović, Tin Trčak, Filip Turčinović

Naslov rada:

Personalizirano pametno zrcalo s implementiranim sustavom za detekciju i prepoznavanje lica

Tekst sažetka:

Pametno zrcalo je uređaj Interneta stvari koji na načelu prividne stvarnosti proširuje mogućnosti običnog zrcala na način da određene informacije potrebne korisniku prikazuje na površini zrcala. Ovaj rad prati implementaciju personaliziranog pametnog zrcala s funkcionalnostima prepoznavanja lica koja omogućava prijavu više korisnika i prepoznavanja govora kojom se upravlja samim uređajem. Sustav je implementiran na mikroračunalu Raspberry Pi 3 Model B, a programski kod pisan je u programskom jeziku Python. Za neke funkcionalnosti korištene su vanjske biblioteke (OpenCV, PocketSphinx).

Ključne riječi:

pametno zrcalo, Internet stvari, Raspberry Pi, prepoznavanje lica, prepoznavanje govora

10. Summary

Authors:

Nikola Čuljat, Emanuel Guberović, Tin Trčak, Filip Turčinović

Title:

Personalized smart mirror with implemented system for face detection and recognition

Summary:

Magic Mirror is an Internet of Things device which, on the principle of virtual reality, extends the capabilities of ordinary mirror by providing certain information to the user on the mirror's surface. This work follows the implementation of a personalized magic mirror with face recognition functionality which enables multiple user login and voice recognition functionality used for device control. The system is implemented on the microcomputer Raspberry Pi 3 Model B and the code is written in the Python programming language. External libraries (such as OpenCV and PocketSphinx) were used for some functionalities.

Keywords:

magic mirror, Internet of Things, Raspberry Pi, face recognition, voice recognition