

**JEDNO REŠENJE IMPLEMENTACIJE PROGRAMSKE PODRŠKE ZA
PREPOZNAVANJE LICA PRILIKOM PRISTUPA ZAŠTIĆENOJ ZONI****SOFTWARE IMPLEMENTATION OF FACE RECOGNITION FOR SECURE ZONE
ACCESS**

Rade Vuković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratka sadržaj – U ovom radu predstavljena je upotreba mašinskog vida, na namenskim računarskim platformama korišćenjem neuronskih mreža za prepoznavanje lica. U radu su dati određeni teoretski uvodi iz oblasti koje su korišćene u izradi projekta, takođe je prikazana i implementacija i dobijeni rezultati na namenskim platformama Minnowboard [1], Raspberry Pi [2] i Movidius [3] uz upotrebu OpenVINO [4] platforme.

Ključne reči: Mašinska vizija, Neuronske mreže.

Abstract – This paper presents the use of machine vision on embedded platforms using neural networks for face recognition. The paper provides some theoretical insights into the areas used in project development, also demonstrates the implementation and results obtained on embedded platforms Minnowboard, Raspberry Pi and Movidius using the OpenVINO platform.

Keywords: Machine vision, Neural networks.

1. UVOD

U današnje vreme bezbednost predstavlja jedan od glavnih aspekata života. Pored potrebe da zaštitimo privatnu svojinu postoji i potreba za zaštitom podatka. Realizovani projekat integrisane kontrole pristupa obezbeđuje zaštitu privatnog prostora, na takav način da obezbeđuje dodelu i verifikaciju prava pristupa zaštićenom prostoru uz očuvanje privatnosti informacija korisnika. Pametna vrata se koriste u institucijama (objektima) kojima je potrebno da dodatno zaštite kako svoju imovinu tako i svoje korisnike. Pametna vrata se mogu koristiti u širokom spektru od privatnih i javnih do poslovnih objekata. Koncept pametnih vrata je zasnovan na dvostrukoj proveru korisnika:

- (1) Provera identifikacije korisnika - što ovom sistemu omogućava modul za prepoznavanje lica,
- (2) Provera šifre – što ovom sistemu omogućava modul za prepoznavanje govorne fraze.

U radu je detaljno opisan modul za prepoznavanje lica. Modul za prepoznavanje lica razvijen je u Python programskom okruženju, korišćenjem OpenCV (eng. *Open source computer vision*) [5] i FaceNet [6]

TensorFlow [7] modela neuronskih mreža za detektovanje i prepoznavanje lica.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Bojan Mrazovac, docent.

Od modula se očekuje da pouzdano može da detektuje korisnike čiji se podaci nalaze u bazi podataka, a da ne prihvati korisnike koji nisu pohranjeni u bazi podataka i da ih klasifikuje kao „nepoznate korisnike“.

Sam proces prepoznavanja lica predstavlja biometrijsku beskontaktnu proveru identiteta, dok šifra predstavlja govornu frazu koja se pretvara u niz karaktera jednoznačno dodeljenih korisniku. Za skladištenje podataka (ime, šifra) i proveru dobijenih podataka od modula za prepoznavanje lica, govorne fraze i izdavanje dozvola koristi se Blockchain [8]. Blockchain je rastuća lista zapisa, nazvanih blokovi, koji su povezani pomoću kriptografije. Svaki blok sadrži alfanumerički kod (eng. *hash*) prethodnog bloka, vremensku oznaku i podatke o transakciji.

Prednost sistema „pametnih vrata“ u odnosu na već postojeće sisteme je u tome što ovaj sistem nudi beskontaktnu proveru identiteta, za razliku od drugih načina utvrđivanja identiteta na osnovu biometrijskih podataka očne dužice, krvnih sudova dlana ili otiska prsta, kao i u sigurnosti podataka koje nam obezbeđuje upotreba Blockchain-a

**2. TEHNOLOGIJE KORIŠĆENE ZA
REALIZACIJU PROJEKTA**

Prepoznavanje lica predstavlja biometrijsko rešenje koje analizira jedinstvene karakteristike lica. Da bi se izvršilo prepoznavanje lica potrebno je da se svako lice predstavi na jedinstven način. Woodrow Bledsoe 1960. godine [9] predložio je tehniku koja analizira koordinate istaknutih karakteristika lica kao što su oči, nos i kosa. U ovom radu korišćena je tehnologija upotrebe neuronskih mreža, koja se sastoji od tri ključna elementa [9]:

• Detekcija lica

Glavni zadatak modula za detekciju lica je da odredi da li na ulaznoj slici postoji lice ili ne. Ukoliko lice postoji modul treba da odredi i mesto gde se ono nalazi. Izlaz modula za prepoznavanje lica predstavljaju koordinate svih detektovanih lica u ulaznoj slici

• Diskriminatorne analize i analize ekstrakcije

Slika koja se dobija na izlazu modula za detekciju lica ima određene nedostatke, slika obično sadrži preko hiljadu tačaka (eng. *pixel*). Jedan od nedostataka je i taj što slika može biti dobavljena iz različitih uglova kamere, sa neravnomernim osvetljenjem i različitim izrazima lica. Da bi se prevazišli nedostaci koristi se ekstrakcija pakovanja podataka (analiza slike postavljanjem mreže za

pronalaženje tačaka od značaja), smanjenje slike (vrši se smanjivanjem broja tačaka po vrstama i kolonama slike) i čišćenje od šuma (vrši se usrednjavanje tačaka (eng. *pixel*) korišćenjem matrice 3x3 za filtriranje slike). Posle ovako pripremljenih parametara lica, vrši se transformacija u vektor sa fiksnom dimenzijom.

- **Identifikacija i / ili verifikacija**

Da bi se izvršilo prepoznavanje lica mora postojati odgovarajuća baza podataka. Za svaku osobu snimljen je određen broj slika koje se čuvaju u bazi podataka, a koje služe za obuku modela. Kada se pojavi lice na ulaznoj slici prvo se izvrši detekcija lica i ekstrakcija podataka i tako dobijeni podaci se uporede sa podacima koji se nalaze u modelu (klasifikatoru).



Slika 1. Konfiguracija strukture sistema za prepoznavanje lica [9]

2.1. Duboke Neuronske Mreže (DNN) [10]

Duboke neuronske mreže (DNN) su neuronske mreže koje imaju određeni nivo složenosti, više od dva sloja. Izrađene su da simuliraju aktivnosti ljudskog mozga, tačnije prepoznavanje uzoraka i analize ulaza kroz različite slojeve simuliranih neuronskih mreža. Duboke neuronske mreže koriste matematičko modeliranje za obuku neurona.

Duboke neuronske mreže imaju ulazni i izlazni sloj i bar još jedan skriveni sloj između.

Za opisivanje dubokih neuronskih mreža koristi se izraz „duboko učenje”, jer duboko učenje predstavlja specifičan oblik mašinskog učenja gde tehnologije koriste aspekte veštačke inteligencije u svrhu korišćenja i klasifikacije informacija na načine koji nadilaze jednostavne protokole za ulaz/izlaz.

2.2. Konvolucione Neuronske Mreže (CNN) [11]

Konvolucione neuronske mreže (CNN ili ConvNet) pripadaju klasi neuronskih dubokih mreža i predstavlja regulisanu verziju višeslojnih perceptrona, koje se najčešće koriste za analizu slikovnih podataka. Višeslojni perceptron ili potpuno povezana mreža znači da svaki neuron iz jednog sloja ima vezu sa svim neuronima u narednom sloju.

Povezanost neurona u konvolucionim neuronskim mrežama podseća na organizaciju vizuelnog korteksa kod ljudi. Vizuelni korteks je deo moždane kore koji obezbeđuje slikovno-vizuelne informacije. Pojedini kortikalni neuroni reaguju na podražaje samo u ograničenom području vidnog polja koji je poznat pod nazivom receptivno polje. Receptivna polja različitih neurona se delimično preklapaju i na taj način se prekriva celokupno vizuelno polje.

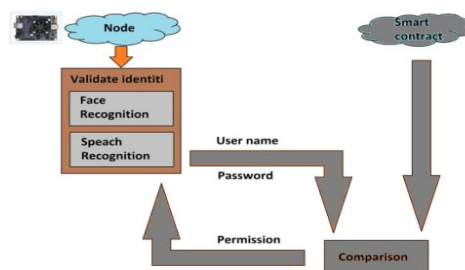
2.3. OpenCV [5]

OpenCV je napisan u programskom jeziku *C++* i njegov primarna sprega (eng. *interface*) je *C++*, ali i dalje je zadržana i starija *C* sprega. Takođe postoje verzije za *Python*, *Java* i *MATLAB / OCTAVE* [5]. *OpenCV* je biblioteka programskih funkcija uglavnom namenjenih za mašinsku obradu vizualnih podataka u realnom vremenu. Podržava okruženja dubokog učenja kao što su: *TensorFlow*, *Torch/PyTorch* i *Caffe*.

2.4. OpenVINO [4]

OpenVINO je programska platforma, koja je bazirana na setu operacija koje pružaju rešenja koja oponašaju ljudski vid. Na osnovu Konvolucione Neuronske Mreže (*CNN*), koja proširuje radno opterećenje mašinskog vida (*OpenCV*) i maksimizuje performanse na *Intel* fizigkoj arhitekturi.

3. REALIZACIJA UPRAVLJANJA



Slika 2. Pregled celog sistema

3.1. Implementacija FaceNet-a na Minnowboard računaru

Na namenskoj računarskoj platformi *Minnowboard* pokrenut je operativni sistem *Linux* 18.04. Na platformi je potom pokrenut isti kod koji je prethodno verifikovan na personalnom računaru. Dobijeni rezultati nisu zadovoljavajući, modul je koristio već unapred obučene modele i klasifikatore koji su korišćeni na računaru. Odziv koji je dobijen je oko 1 – 1.5 FPS (eng. *Frames per second*) što je nezadovoljavajuće jer u tim uslovima se ne može postići rad u trenutnom vremenu, zbog niskog FPS-a dolazi do kašnjenja i izuzetno lošeg kvaliteta slike. Tačnost koja je ostvarena korišćenjem modela se kreće od 60 do 95%, treba uzeti u obzir da su modeli obučeni na relativno malom skupu od 16 korisnika, dobijeni rezultati su prikazani na slici 3.

Slika 3. Tabelarno poređenje dva modela na *Minnowboard-u*

Model	Frames per second	Accuracy (%)
20180408-102900.pb	1,5 FPS	60-90 %
20170511-185253.pb	1 FPS	80-95%

Zbog niskog FPS-a potrebno je da protekne dosta dug vremenski period da dođe do detekcije lica i iz ovog razloga se prešlo na dalji razvoj koda koji bi mogao da radi u realnom vremenu i da da bolje rezultate na namenskoj računarskoj platformi što će biti objašnjeno u narednim tačkama.

3.2. Implementacija intermedialnih (.xml i .bin) modela formata FP16 na Minnowboard-u i Movidiusu korišćenjem OpenVINO-a

Zbog potrebe da se sistem optimizuje da bi mogao da radi na namenskoj računarskoj platformi, instalirana je programska platforma *OpenVINO*. Da je *OpenVINO* instaliran na platformu pokazuje ispis u terminalu “[*setup-vars.sh*] *OpenVINO environment initialized*”.

Pored namenske računarske platforme *Minnowboard* koristi se i *Movidius*, pa je potrebno razviti kod koji će moći da radi na ovim platformama. Prvi problem je taj što

Movidius ne podržava *FaceNet* arhitekturu za detekovanje lica, pored ovog problema modul ne može da koristi klasičan *TensorFlow* „zaleđeni” model (model.pb).

Da bi se rešili postojeći problemi potrebno je razviti kod koji koristi druge modele umesto *FaceNet*-a i *TensorFlow*-a. Umesto *FaceNet*-a za prepoznavanje lica koristi se model (*haarcascade_frontalface_alt2.xml*). *OpenVino* podržava *TensorFlow* model ali ne u ekstenziji .pb već je potrebno da se napravi *TensorFlow* model sa ekstenzijama .xml i .bin. Da bi se izvršila konverzija modela sa ekstenzijom .pb u .xml i .bin koristi se *OpenVINO*, koji ima ugrađene funkcije koje vrše prebacivanje modela u željeni format. Potrebno je voditi računa da se izabere odgovarajući format modela, u ovom slučaju potrebno je da format modela bude FP16 a ne FP32, zato što se format modela FP32 ne može koristiti za rad sa *Movidius*-om, a FP16 može i na CPU i *Movidius*-u.

Nakon razvoja programske platform na prethodno navedenim osnovama modul je uspešno pokrenut na namenskoj računarskoj platformi *Minnowboard* korišćenjem *Movidius*-a. Problem koji je nastao je da namenska računarska platforma nema dovoljno procesorske snage da izvrši novu obuku modela tako da je sistem potrebno trenirati na procesorski jačem okruženju (računaru).

Dobijeni odziv je oko 7 FPS, dobijeni rezultati su prikazani tabelarno na slici 4.

Slika 4. Tabelarno poređenje dva modela na *Minnowboard*-u i *Movidius*-a

Model (FP16)	Frames per second	Accuracy (%)
20180408-102900.xml 20180408-102900.bin	~6,8 FPS	60 – 90 %
20170511-185253.xml 20170511-185253.bin	~6,8 FPS	80 – 95 %

3.3. Implementacija intermedijalnih (.xml i .bin) modela formata FP16 na *Minnowboard*-u korišćenjem *OpenVINO*-a

Format modela FP16 je moguće koristiti na *Minnowboard*-u bez upotrebe *Movidius*-a. Format modela FP16 je prvobitno namenjen za rad sa GPU ili VPU ali je podržan i za rad sa CPU-om, ako se radi o *Intel*-ovoj familiji CPU, GPU, VPU.

Dobijeni odziv na *Minnowboard*-u je oko 5 FPS-a, tačnost koja je ostvarena kreće se od 60-95%. Dobijeni rezultati su prikazani tabelarno na slici 5.

Model (FP16)	Frames per second	Accuracy (%)
20180408-102900.xml 20180408-102900.bin	~5 FPS	60 – 90 %
20170511-185253.xml 20170511-185253.bin	~5 FPS	80 – 95 %

Slika 5. Tabelarno poređenje dva modela na *Minnowboard*-u

3.4. Implementacija intermedijalnih (.xml i .bin) modela formata FP16 na *Raspberry Pi* i *Movidius*-a korišćenjem *OpenVINO*-a

Za ovo testiranje korišćen je *Raspberry Pi* sa *Movidius*-om. Na *Raspberry Pi*-u je pokrenut *Raspbian* operativni sistem. *OpenVINO* se instalira slično kao na operativnom sistemu *Linux* razlika je u tome što za *Raspbian* operativni sistem ne postoji Gui (eng. *Graphical user*

interface) za instalaciju *OpenVINO*-a već se sve radi kroz komandnu liniju preko terminala. Kada je *OpenVINO* instalirano dobija se ista poruka kao na *Linux* operativnom sistemu.

Nakon potrebnih instalacija, pušten je u rad modul za prepoznavanje lica koji je korišćen na *Minnowboard*-u. Dobijeni su zadovoljavajući rezultati korišćenjem *Raspberry Pi* i *Movidius*-a ali koji su slabiji od rezultata dobijenih korišćenjem *Minnowboard*-a i *Movidius*-a. Razlozi za slabije rezultate na *Raspberry Pi* mogu se tražiti u karakteristikama kao što su procesorska snaga i radna memorija koje su slabije nego kod *Minnowboard*-a. Odziv koji je dobijen je oko 3,4 FPS-a, sa tačnošću od 60 – 95% u zavisnosti od korišćenog modela. Dobijeni rezultati su prikazani na slici 6.

Model (FP16)	Frames per second	Accuracy (%)
20180408-102900.xml 20180408-102900.bin	~3,4 FPS	60 – 90 %
20170511-185253.xml 20170511-185253.bin	~3,4 FPS	80 – 95 %

Slika 6. Tabelarno poređenje dva modela na *Raspberry Pi* i *Movidius*-a

Pokretanje koda na *Raspberry Pi* CPU nije moguće, jer *Raspberry Pi* nije zasnovan na *Intel*-ovoj fizičkoj arhitekturi samim tim nije pogodan za rad *OpenVINO*-a.

3.5. Implementacija intermedijalnih (.xml i .bin) modela formata FP32 na *Minnowboard*-u korišćenjem *OpenVINO*-a

Ovo testiranje se razlikuje od prethodnih po tome što se sada umesto formata modela FP16 koristi FP32 koji je namenjen za rad sa CPU-om. Format FP32 se dobija slično kao i FP16, u radnom okruženju *OpenVINO* gradimo i format FP32, tako što kroz komandnu liniju u terminalu zadamo traženi format modela.

Dobijeni rezultati korišćenjem FP32 se ne razlikuju puno od dobijenih rezultata korišćenjem formata FP16. Ovo se odnosi samo za rad sa *Minnowboard*-om pošto se jedino na ovoj platformi uspešno koristi FP32 format. Na *Raspberry Pi* nije moguće pokrenuti ovaj format modela razlog je što CPU koji je ugrađen na *Raspberry Pi* ne podržava FP16 format.

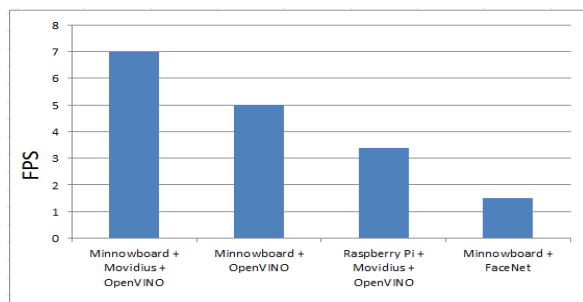
Dobijeni odziv je sličan kao kod FP16 formata modela i iznosi oko 5 FPS-a, tačnost iznosi 60-95%, u zavisnosti od modela što je prikazano na slici 7.

Model (FP32)	Frames per second	Accuracy (%)
20180408-102900.xml 20180408-102900.bin	~4,9 FPS	60 – 80 %
20170511-185253.xml 20170511-185253.bin	~4,9 FPS	80 – 95 %

Slika 7. Tabelarno poređenje dva modela na *Minnowboard*-u

3.6. Upoređivanje dobijenih rezultata korišćenjem format modela FP16 i originalnog modela .pb.

Upoređivanjem dobijenih rezultata na namenskim računarskim platformama *Minnowboard*-u i *Raspberry Pi* sa i bez korišćenja *Movidius*-a. Svi rezultati su dobijeni korišćenjem FP16 formata modela i originalnog modela .pb. Na slici 8 su prikazani dobijeni rezultati.



Slika 8. Grafičko poređenje dobijenih rezultata

4. ZAKLJUČAK

U prikazanom radu je izložen razvoj sistema za odobrenje pristupa zaštićenom prostoru pod nazivom „pametna vrata“, tačnije jednog njegovog dela koji služi za prepoznavanje lica korisnika, kao i mogućnosti i prednosti koje on omogućava. Ovakav vid kontrole zaštićenog prostora nam omogućava smanjen rizik, da se unutar zaštićenog prostora nađe osoba koja nema prava pristupa. Povećanu sigurnost nam garantuje dvostruka provera korisnika, koja se sastoji od prepoznavanja lica i prepoznavanja govorne fraze.

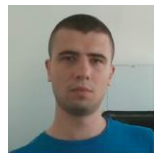
Prepoznavanje lica je realizovano korišćenjem neuronskih mreža i unapred obučenom modelu. Ovako definisan kod za prepoznavanje lica je pogodan za korišćenje na namenskim računarskim platformama, naročito na Intel *Movidius*-u koji je namenjen za rad sa neuronskim mrežama. Nakon prepoznavanja lica se aktivira modul za prepoznavanje govorne fraze i njenu konverziju u šifru. Prikupljanje podataka se vrši preko kamere i mikrofona, za koje je razvijena posebna programska platforma u *Python*-u. Dobijeni podaci (ime, šifra), se šalju na *Blockchain* na validaciju.

Upravljanje zaštićenim prostorom omogućava nam da u svakom trenutku imamo kontrolu nad svim korisnicima i njihovim odobrenjima unutar sistema „pametnih vrata“. Mogućnost upravljanja sistemom „pametnih vrata“ pruža nam sigurnost i zaštitu, kako nas, tako i naše imovine.

5. LITERATURA

- [1] Minnowboard (tehnička dokumentacija), pristup 04.11.2019. preko: (<https://minnowboard.org/minnowboard-turbot/technical-specs>)
- [2] Raspberry Pi (tehnička dokumentacija), pristup 12.07.2019. preko: (<https://www.raspberrypi.org/>)
- [3] Movidius (specifikacije), pristup 04.11.2019. preko: (<https://software.intel.com/en-us/neural-compute-stick>)
- [4] OpenVINO, pristup 14.11.2019. preko: (<https://docs.openvino toolkit.org/>)
- [5] OpenCV-Python Tutorials Documentation, pristup 04.11.2019. preko: (<https://buildmedia.readthedocs.org/media/pdf/opencv-python-tutroals/latest/opencv-python-tutroals.pdf>)
- [6] FaceNet: A Unified Embedding for Face Recognition and Clustering, pristup 04.11.2019. preko: (<https://arxiv.org/abs/1503.03832>)
- [7] TensorFlow, pristup 04.11.2019, preko: (<https://www.tensorflow.org/>)
- [8] Blockchain, pristup 14.11.2019. preko: (<https://en.wikipedia.org/wiki/Blockchain>)
- [9] Wei-Lun Chao GICE, National Taiwan University, Face Recognition
- [10] Deep Neural Network, pristup 14.11.2019. preko: (<https://www.techopedia.com/definition/32902/deep-neural-network>)
- [11] David Habrman, Face Recognition with Preprocessing and Neural Networks, LiTH-ISY-EX--16/4953—SE

Kratka biografija:



Rade Vuković rođen je u Zvorniku (BiH) 1995. god. Diplomirao je na Fakultetu tehničkih nauka 2018. god.

kontakt: rade.vuk95@gmail.com