Kombinovanje *AI* i *Full-Stack Engineering-a* za razvoj autentifikacije zasnovane na prepoznavanju lica

Studentski rad

Pavle Borčanin, Stefan Ristić, Nikola Milašinović
studenti prvog ciklusa studija
Elektrotehnički fakultet Univerziteta u Istočnom Sarajevu
Istočno Sarajevo, Bosna i Hercegovina
borcaninpavle10@gmail.com, stefanristic01@yahoo.com, nmilasinovic25@gmail.com

Sažetak—**Široka dostupnost interneta uslovljava implementaciju sigurne autentifikacije u veb aplikacijama. Zahtjev za većim stepenom sigurnosti može se realizovati primjenom multifaktorske autentifikacije. Multifaktorska autentifikacija zasnovana na lozinkama i prepoznavanju lica predstavlja kombinaciju dominantnog načina autentifikacije i jedinstvenih, stalno dostupnih resursa. Konvolucijske neuronske mreže omogućavaju kreiranje modela za prepoznavanje lica. Takvi modeli zahtjevaju veliki skup podataka, a u uslovima kada ih nije moguće osigurati koristi se *transfer learning*. Radom je obuhvaćen razvoj modela za prepoznavanje lica i njegova primjena u veb aplikaciji pri procesu autentifikacije.**

Ključne riječi - Autentifikacija; face recognition; CNN; VGG; veb tehnologije

I. Uvod

Razvoj interneta i internet servisa transformisao je mnoge oblasti ljudskog života i rada. Veb servis je postao platforma za obavljanje poslovnih aktivnosti, edukaciju, različite vidove komunikacije i mnoge druge aktivnosti [1]. Procenat svjetske populacije koji koristi internet kontinualno raste i tokom 2024. godine dosegao je 68% [2].

Zahvaljujući osobinama da su platformski nezavisne, da se automatski ažuriraju i da ih nije potrebno instalirati, veb aplikacije postaju sve popularnije. Veb aplikacije i sajtovi dostupni su svim korisnicima interneta, što predstavlja veliki izazov sa aspekta sigurnosti. Implementacija sigurne autentifikacije dobija sve više na značaju i postaje tema kojoj se posvećuje posebna pažnja.

Većina veb aplikacija koristi jednofaktorsku autentifikaciju zasnovanu na korisničkim imenima i lozinkama. Dakle, koristi se faktor *nešto što korisnik zna*. Ovaj vid autentifikacije odlikuje se brzom i jednostavnom implementacijom, što je uticalo na široku upotrebu. Glavni problem ovog pristupa ogleda se u sigurnosti korisničkih lozinki. Korisnici uglavnom biraju lozinke koje je lako zapamtiti, sa malim brojem karaktera i iz istog skupa karaktera. Za njihovo razbijanje koriste se *brute force* algoritmi [3]. Takođe, korisnici kreiraju lozinke na dosta predvidljiv način, pa je moguće koristit lozinke koje se nalaze u javno objavljenim skupovima kompromitovanih lozinki. Servis *Have I been pawned* omogućava uvid da li se lozinka nalazi u skupu javno objavljenih kompromitovanih lozinki. Korištenjem ovoga servisa dolazimo do informacije da je lozinka *12345* viđena preko 4 miliona puta u javno objavljenim skupovima podataka [4]. Autentifikacija zasnovana na lozinkama mora osigurati jaku enkripciju lozinki, što nije bio slučaj 2012. godine kada je *Linkedin* bio meta hakerskog napada, tokom koga su ukradene lozinke od preko 6,5 miliona korisničkih naloga [5]. Takođe, veliki broj korisnika koristi iste lozinke za različite korisničke naloge, što smanjuje njihovu bezbjednost [3].

Razvoj tehnologije omogućio je biometrijsku autentifikaciju. Biometrijska autentifikacija temelji se na provjeri fizičkih karakteristika osobe. Fizičke karakteristike, kao što su otisci prstiju, lice i šarenica oka jedinstvene su za svakog pojedinca i koriste se za identifikaciju i verifikaciju identiteta. Dakle, koristi se faktor *nešto što korisnik jeste*. Prednost ovog pristupa ogleda se u tome što korisnik u svakom trenutku ima resurse neophodne za autentifikaciju. Mana ovog vida autentifikacije ogleda se u potrebi za izuzetno sigurnim čuvanjem biometrijskih podataka. Biometrijski podaci se ne mogu promijeniti kao lozinka, pa je neophodno osigurati visok stepen njihove zaštite. Takođe, neophodno je koristiti naprednije softvere za njenu realizaciju [3].

Pored faktora *nešto što korisnik zna* i *nešto što korisnik jeste* postoji faktor *nešto što korisnik ima*. Autentifikacija zasnovana na tom faktoru koristi pametne kartice, *RFID* kartice, mobilne telefone i druge fizičke objekte za autentifikaciju. Mana ove autentifikacije leži u činjenici da je neophodno stalno imati taj fizički objekat, moguće ga je izgubiti ili napraviti njegovu kopiju [3].

Multifaktorska autentifikacija podrazumijeva kombinaciju više različitih faktora za autentifikaciju. Upotreba multifaktorske autentifikacije osigurava bolju zaštitu podataka nego jednofaktorska autentifikacija [3].

Realizacija multifaktorske autentifikacije mora biti ostvarena tako da se iskoriste najbolje osobine svakog tipa autentifikacija. U veb aplikacijama je dominantna autentifikacija zasnovana na lozinkama. To je tradicionalni način autentifikacije na koji su korisnici navikli. Veb aplikacije se koriste na uređajima koji uglavnom imaju kameru, što omogućava primjenu autetifikacije zasnovane na prepoznavanju lica. Kombinacija tradicionalnog načina autentifikacije i onoga što korisnik stalno posjeduje vodi ka realizaciji sigurnije autentifikacije bez narušavanja korisničkog iskustva pri logovanju.

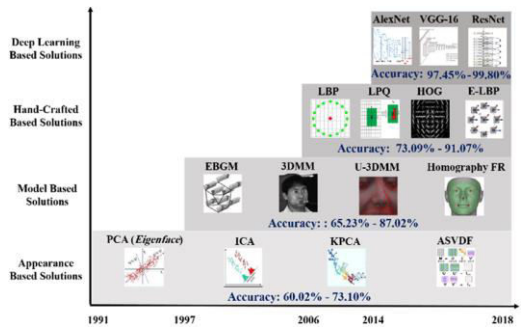
Svrha projekta, koji su autori ovog rada realizovali, je realizacija logike neophodne za autentifikaciju zasnovanu na prepoznavanju lica u kombinaciji sa lozinkama. Realizovana je logika koja je zasnovana na metodama vještačke inteligencije, a koja može komunicirati sa *backendom*, realizovanim u različitim tehnologijama. Ova logika integrisana je u veb aplikaciju koja simulira sistem za registraciju studenata i evidenciju osnovnih informacija o njima.

U nastavku su date metode za realizaciju autentifikacije zasnovane na prepoznavanju lica, a koje pronalazimo u drugim radovima.

II. POVEZANI RADOVI – DRUGE METODOLOGIJE

Neophodno je naglasiti da su za implementaciju prepoznavanja lica u okviru realizovanog projekta korištene konvolucijske neuronske mreže i *VGG Face Model*. U nastavku dajemo pregled drugih metodologija za prepoznavanje lica, kao i različitih načina za prepoznavanje lica primjenom konvolucijskih neuronskih mreža.

Na Sl. 1 su prikazane različite metodologije za realizaciju prepoznavanja lica, kao i različte implementacije unutar iste metodologije.



Slika 1. Evolucija rješenja za prepoznavanje lica [6]

Većina sistema za prepoznavanje lica danas koristi konvolucijske neuronske mreže jer se odlikuju visokim stepenom tačnosti. Pored *VGG Face* modela važno je spomenuti *FaceNet* i *DeepFace* modele za prepoznavanje lica [7]. Ovi modeli su pogodni za implementaciju autentifikacije jer omogućavaju precizno prepoznavanje i verifikaciju lica.

III. ARHITEKTURA SISTEMA

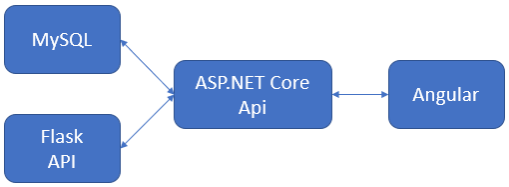
Kao što je istaknuto u uvodnom dijelu, projektom je obuhvaćen razvoj veb aplikacije koja koristi dvofaktorsku autentifikaciju zasnovanu na lozinkama i prepoznavanju lica. Za potrebe prepoznavanja lica razvijen je model korištenjem metoda vještačke inteligencije. U nastavku su opisane tehnologije koje su korištene pri implementaciji i principi implementacije.

Klijentski dio aplikacije kreiran je korištenjem *Angular frameworka*, a serverski dio aplikacije korištenjem *Asp.Net Core* i *Flask frameworka*. Za skladištenje podataka korištena je *MySql* baza podataka. Za komunikaciju između dijelova sistema koriste se *REST Api*-ji. Logika vezana za prepoznavanje lica i treniranje modela napisana je korištenjem *Python*-a.

Flask server sadrži *Api*-je koji služe za prepoznavanje lica i za treniranje modela. *Flask* se u odnosu na *Asp.Net Core* ponaša kao server, a *Asp.Net Core* u odnosu na *Flask* kao klijent. To znači da tokom autentifikacije *Asp.Net Core* poziva odgovarajuće *endpointe*, a *Flask* šalje odgovor. *Angular* se u odnosu na *Asp.Net Core* ponaša kao klijent.

Autentifikacija je veoma bitan dio svake aplikacije, ali je neophodno voditi računa i o drugim funkcionalnostima. Uvažavajući ovu činjenicu, logika vezana za autentifikaciju je realizovana u *Python*-u i smještena je na *Flask* serveru. Prednost ovog pristupa ogleda se u mogućnosti primjene različitih tehnologija za razvoj serverskog dijela aplikacije. Mana ovakvog pristupa ogleda se u produženju vremena nephodnog za obradu zahtjeva, uzrokovanog komunikacijom između izabrane tehnologije i *Flask* servera. Do kašnjenja ne bi dolazilo kada bi se za razvoj serverskog dijela aplikacije koristio *Python* ili neki njegov *framework*, ali *Python* nije adekvatan za razvoj svih vrsta aplikacija, jer korisnički zahtjevi i drugi faktori determinišu i izbor tehnologije. Uzimajući u obzir da je autentifikacija manje frekventna funkcionalnost u odnosu na same funkcionalnosti aplikacije, opravdano je obezbijediti brzo izvršavanje tih operacija primjenom drugih tehnologija, uz sigurnu, ali sporiju autentifikaciju.

Na Sl. 2 dat je prikaz komunikacije unutar realizovane veb aplikacije.



Slika 2. Izabrane tehnologije za razvoj aplikacije i njihova komunikacija

Kreirana aplikacija vezana je za fakultetski sistem, ali treba naglasiti da je primarna tema rada autentifikacija zasnovana na prepoznavanju lica. S tim u vezi, u okviru aplikacije su realizovane samo osnovne funkcionalnosti, koje omogućavaju evaluaciju kreiranog modela za prepoznavanje lica i evidenciju

studenata. Ipak, odabir tehnologija za razvoj aplikacije izvršen je pod pretpostavkom da se aplikacija može obogatiti dodatnim funkcionalnostima, koje su neophodne za fakultetski sistem. Kratak opis tehnologija za razvoj veb aplikacije dat je u nastavku, sa osvrtom na razloge zbog kojih su se autori odlučili za njih.

A. Angular

Angular je *framework* za dizajn aplikacija i razvojna platforma za kreiranje efikasnih i sofisticiranih *single-page* aplikacija. Angular aplikacije imaju arhitekturu zasnovanu na komponentama, koje predstavljaju osnovne gradivne blokove aplikacije. Svaka komponenta ima svoju logiku, šablon i stilizaciju. Arhitektura zasnovana na komponentama omogućava mogućnost višestruke upotrebe komponenti što omogućava bolju organizaciju koda i ubrzavanje procesa kreiranja aplikacije [8].

Angular kao *framework* predstavlja kolekciju dobro integrisanih biblioteka koje pokrivaju širok spektar funkcionalnosti, uključujući rutiranje, upravljanje formama, komunikaciju između klijenta i servera, i još mnogo toga. *Angular* koristi *TypeScript*, što olakšava održavanje aplikacija i bolje razumijevanje koda. Kao moderan *framework*, sa dobrom dokumentacijom, *Angular* je postao jedan od najpopularnijih alata za razvoj klijentskog dijela aplikacija [8].

S obzirom da je veb aplikacija vezana za fakultet kao veoma složen sistem, autori ovog rada su smatrali da je *Angular* dobar izbor za njenu realizaciju, jer kao *framework* jasno definiše okvire za realizaciju pojedinih dijelova sistema.

B. ASP.NET Core

ASP.NET Core je savremeni *framework* sa izuzetnim performansama koji se koristi za razoj veb aplikacija. *ASP.NET Core* podržava razvoj i pokretanje aplikacija na *Windows*, *Linux* i *macOS* operativnim sistemima. *ASP.NET Core* je dizajniran modularno, što omogućava lakše održavanje i skalabilnost, jer se koriste samo neophodne komponente, čime se smanjuje veličina aplikacije. *ASP.NET Core* omogućava moderne funkcionalnosti za veb razvoj kao što su *dependency injection*, *middelware* i asinhrono programiranje. *ASP.NET Core* pruža mnogo ugrađenih mehanizama za implementaciju sigurnosti, uključujući autentifikaciju, autorizaciju, zaštitu od napada i enkripciju [9].

ASP.NET Core podržava dva pristupa za kreiranje *API*-ja, a to su pristup zasnovan na kontrolerima i minimalni *API*-ji. Za realizaciju projekta korišten je pristup zasnovan na kontrolerima. Upotreba *ASP.NET Core Web Api framework*-a omogućava kreiranje *REST-ful* servisa koje može koristiti širok spektar klijenata, uključujući *browser*-e, mobilne telefone i teblete. *REST-ful* servisi su jednostavni, skalabilni i laki za integraciju [9].

Uzimajući u obzir prethodno navedene karakteristike, autori rada su se odlučili da koriste *ASP.NET Core Web API* za razvoj serverskog dijela aplikacije, jer korištenjem ove tehnologije moguće je ostvariti jednostavnu komunikaciju sa klijentskim dijelom aplikacije, proširivost sistema, izvršavanje na različitim sistemima, dobre performanse pri obradi zahtjeva i realizovati adekvatne sigurnosne mehanizme.

C. MySQL

Za izradu projekta korištena je *MySQL* baza podataka. *MySQL* je jedan od najpopularnijih *open-source* baza podataka. Koristi se za skladištenje, organizaciju i upravljanje podacima u različitim aplikacijama. *MySQL* je poznat po svojoj brzini, stabilnosti i jednostavnosti upotrebe [10].

MySQL omogućava rukovanje velikom količinom podataka, pri čemu su osigurane dobre performanse. MySQL radi na različitim operativnim sistemima i nudi različite mehanizme za sigurnost, uključujući kontrolu pristupa i enkripciju podataka. *MySQL* nudi različite alate i interfejse za upravljanje bazom podataka, uključujući *MySQL Workbench*, komandnu liniju i druge [10].

Za rad sa bazom podataka u okviru *ASP.NET Core Web API*-ja korišten je *Entity Framework* i *Code-First* pristup. Ovo značajno olakšava rad sa podacima jer omogućava fokus na logiku umjesto na rukovanje niskim nivoima baze podataka.

D. Flask

Logika za prepoznavanje lica i treniranje modela kreirana je u *Pythonu*, a da bi mogla komunicirati sa drugim dijelovima sistema kreirani su *API*-ji korištenjem *Flask* frameworka.

Flask je lagan i fleksibilan *Python* framework, koji omogućava jednostavnu i brzu izradu veb aplikacija. *Flask* pruža osnovne funkcionalnosti za razvoj aplikacija, a programer uključuje ostale biblioteke i ekstenzije. *Flask* omogućava brz razvoj malih aplikacija [11].

Uzimajući u obzir da je *Flask* lagan i brz *framework*, autori rada su smatrali da je adekvatan izbor za kreiranje *API*-ja koji sadrže logiku vezanu za vještačku inteligenciju. Iskorištena je *Flask*-ova mogućnost upravljanja rutama, a sama logika za prepoznavanje lica implementirana je uključivanjem odgovarajućih biblioteka.

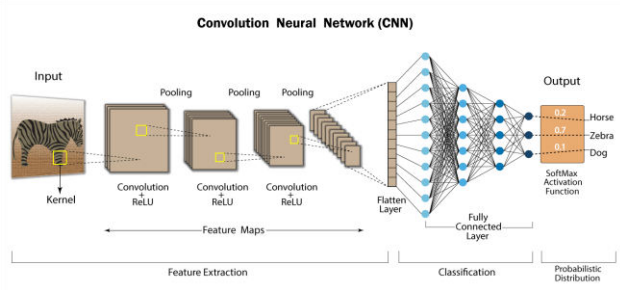
E. AI model

Za prepoznavanje lica u okviru projekta korišten je model zasnovan na konvolucijskim neuronskim mrežama. Jedna od mogućih primjena konvolucijskih neuronskih mreža je prepoznavanje objekata, lica i scena na slikama, što odgovara problemu projekta. Kroz višeslojne konvolucijske slojeve, konvolucijska neuronska mreža može da prepozna specifične osobine lica, kao što su oblik nosa, očiju, usana i druge karakteristike. U realizovnom projektu, model koristi slike lica kako bi prepoznao i klasifikovao različite osobe. Konvolucijske neuronske mreže imaju sposobnost procesuiranja velike količine slika i uočavanja detalja koji bi mogli da promaknu ljudskom oku.

Glavni dijelovi konvolucijske neuronske mreže su konvolucioni slojevi (eng. *convolution layers)*, slojevi za uzorkovanje (eng. *pooling layers)*, aktivacione funkcije (eng. *activation function)* i potpuno povezani slojevi (eng. *fully connected layers)*. Konvolucioni slojevi primjenjuju konvolucione operacije na ulazne slike, koristeći filtere

(poznate i kao jezgra) za detekciju karakteristika kao što su ivice, teksture i složeniji obrasci. Nakon konvolucionih slojeva slijede slojevi za uzorkovanje. Slojevi za uzorkovanje smanjuju prostorne dimenzije ulaza, smanjujući računarsku složenost i broj parametara u mreži. Nelinearne aktivacione funkcije, kao što je *ReLU*, uvode nelinearnost u model, omogućavajući mu da uči složenije odnose u podacima. Potpuno povezani slojevi su odgovorni za pravljenje predikcija na osnovu karakteristika koje su naučili prethodni slojevi. U potpuno povezanim slojevima svaki neuron je povezan sa svakim neuronom u sljedećem sloju [12].

Prikaz rada konvolucijske neuronske mreže dat je na Sl. 3.



Slika 3. Prikaz rada konvolucijske neuronske mreže [13]

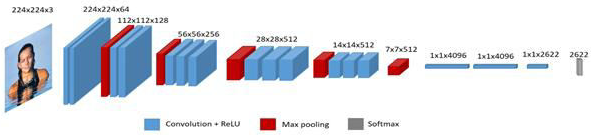
Za treniranje konvolucijskih neuronskih mreža koristi se veliki skup podataka sa označenim slikama. U slučajevima kada nije moguće obezbijediti takav skup, što je bio slučaj pri realizaciji ovog projekta, moguće je iskoristiti *transfer learning*.

Transfer learning je tehnika u mašinskom učenju koja se koristi kada želimo da iskoristimo model koji je već obučen na jednom zadatku i prilagodimo ga za novi, ali sličan zadatak. Ideja *transfer learning*-a je da se model koji je naučio opšte karakteristike iz velikih skupova podataka može prilagodi za specifičnu primjenu, što osigurava uštedu vremena i resursa. *Transfer learning* obuhvata nekoliko koraka. Prvi korak predstavlja izbor prethodno obučenog modela. Bira se model koji je obučen na velikim skupovima podataka, odnosno model koji je naučio da prepoznaje opšte karakteristike podataka. Nakon toga u izabranom modelu vrši se zamjena posljednjih slojeva slojevima koji su specifični za naš zadatak. Zatim se sprovodi treniranje dodatih slojeva novim podacima, dok ostale slojeve modela držimo zamrznutim. Ovo omogućava modelu da se brzo prilagodi na naš zadatak, bez potrebe za ponovnim učenjem osnovnih karakteristika koje je naučio. U nekim slučajevima izvodi se i fino podešavanje, to su slučajevi kada imamo dovoljno resursa i podataka. Fino podešavanje podrazumijeva upotrebu i slojeva koje smo prethodno zamrznuli, nakon što obučimo naše dodatne slojeve [14].

U kontekstu realizovanog projekta, koji se bavi prepoznavanjem lica, *transfer learning* omogućava da iskoristimo već obučene modele koji su naučili da prepoznaju osnovne karakteristike lica, kao što su oblik očiju, nosa, usta i konture lica. Te modele je moguće prilagoditi tako da prepoznaju specifična lica koja nas interesuju, čime se značajno ubrzava proces obuke i poboljšava tačnost modela.

Za realizaciju projekta izabran je *VGG Face Model*. *VGG Face* model koristi arhitekturu baziranu na popularnoj *VGG* (*Visual Geometry Group*) mreži, koja je postala poznata zbog svoje duboke i jednostavne strukture sa mnogim slojevima konvolucija, što omogućava modelu da nauči vrlo detaljne karakteristike slika. Ova mreža ima duboku arhitekturu sa velikim brojem konvolucionih slojeva, što joj omogućava da izuči vrlo složene obrasce. *VGG Face* model je obučen na 2,6 miliona slika lica koje predstavljaju 2622 identiteta (uglavnom poznatih ličnosti). Ovaj obim podataka omogućava modelu da prepozna širok spektar karakteristika lica, kao što su razlike u izrazu lica, starosti, etničkoj pripadnosti, i drugim specifičnostima koje mogu biti relevantne za prepoznavanje identiteta. *VGG Face* model se koristi za zadatke kao što suprepoznavanje identiteta (koja osoba se nalazi na slici) i verifikacija (da li dvoje slika prikazuju istu osobu). Ovaj model može da uporedi slike i da utvrdi sličnost, što ga čini korisnim za sisteme za autentifikaciju [15].

Prikaz strukture *VGG Face* modela dat je na Sl. 4.



Slika 4. Prikaz strukture VGG Face modela [16]

Kroz ovu poglavlje je dat pregled onoga što je korišteno za realizaciju projekta. Naredno poglavlje ima za cilj da pruži detaljan uvid u implementaciju najbitnijih dijelova sistema.

IV. IMPLEMENTACIJA

A. *Treniranje modela*

Dio koda za treniranje modela dat je u Kodnom listingu 1.

```
# Učitavanje VGGFace2 modela sa unaprijed naučenim
težinama (transfer learning)
base_model = VGGFace(model='vgg16',   include_top=False,
input_shape=(224, 224, 3))

# Dodavanje vlastitih slojeva na VGGFace2 model (fine-
tuning)
x = Flatten()(base_model.output) # Poravnanje izlaza iz
osnovnog modela
x = Dense(512, activation='relu')(x) # Dodavanje gusto
povezanog sloja
x = Dropout(0.5)(x) # Dodavanje sloja za isključivanje
50% veza radi regularizacije
```

```
predictions = Dense(active_student_count,
activation='softmax')(x) # Klasifikacija za broj osoba
koje želimo prepoznati

# Kreiranje novog modela sa slojevima temeljenim na
VGGFace2 modelu
model = Model(inputs=base_model.input,
outputs=predictions)
# Zamrzavanje svih slojeva VGGFace2 modela kako bi se
koristile samo unaprijed naučene težine
for layer in base_model.layers:
    layer.trainable = False

# Kompajliranje modela
model.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])

# Treniranje modela koristeći treniranu i validacionu
grupu podataka
model.fit(train_generator, epochs=4,
validation_data=validation_generator)

#Snimanje modela u datoteku
    model.save('face_recognition_model.h5')
```

Kodni listing 1. Dio koda za treniranje modela

U prethodnom kodu koristi se *transfer learning* pristup kako bi se iskoristile unaprijed naučene karakteristike modela *VGG16* iz *VGGFace* biblioteke, koja je trenirana na velikim skupovima slika lica. Model se učitava sa opcijom *include_top=False*, što znači da se isključuju završni slojevi modela koji su specifični za klasifikaciju, jer ćemo dodati vlastite slojeve. Nakon toga se dodaju slojevi za klasifikaciju, uključujući sloj za *flatteniranje* izlaza iz osnovnog modela, gusto povezane slojeve sa *ReLU* aktivacijom, sloj za regularizaciju (*dropout*) kako bi se smanjio rizik od prekomjernog učenja (*overfitting*), te završni sloj sa *softmax* funkcijom koji daje vjerovatnoće za svaku klasu (osobu). Slojevi osnovnog modela se zamrzavaju kako bi se koristile unaprijed naučene težine, dok će se samo naši dodati slojevi trenirati. Model se zatim trenira na specifičnom skupu podataka za prepoznavanje osoba, koristeći *Adam optimizator* i funkciju gubitka *categorical_crossentropy*, te pratimo metriku tačnosti. Nakon obuke, model se snima u fajl kako bi se mogao ponovo koristiti bez potrebe za ponovnim treniranjem.

B. *Predikcija modela*

Nakon što je lice detektovano i isječeno iz originalne slike, proces obrade ide kroz nekoliko koraka: prvo se slika lica skalira na dimenzije 224x224 piksela, što je zahtjev za model *VGG16* korišćen u *transfer learning*-u. Zatim se vrijednosti piksela normalizuju u opseg [0, 1] dijeljenjem sa 255.0, što je uobičajeni postupak za normalizaciju slika u dubokom učenju. Funkcija *np.expand_dims* dodaje dodatnu dimenziju na početak, kako bi podatak imao odgovarajući oblik za model (*batch* dimenzija). Model zatim predviđa izlaz za datu sliku lica, a rezultat je vektor vjerovatnoća za svaku od klasa. Na kraju, vrijednost najveće vjerovatnoće se uzima pomoću *np.max*, dok se indeks klase sa najvećom vjerovatnoćom uzima korišćenjem *np.argmax*, što se koristi za određivanje identiteta osobe.

C. *Heširanje lozinke*

Heširanje lozinki vršeno je pomoću *Bcrypt* algoritma. *Bcrypt* koristi nasumičan salt i prilagodljive iteracije kako bi otežao *brute-force* napade.

V. ZAKLJUČAK

Široka dostupnost interneta osim benificija donosi i brojne izazove. Uzimajući u obzir da se sve više poslova obavlja *online*, neophodno je realizovati odgovarajuće sigurnosne mehanizme. Multifaktorska autentifikacija je jedan od tih mehanizama. Povećanje broja različitih faktora autentifikacije vodi ka većoj sigurnosti, ali se pri razvoju veb aplikacija mora voditi računa i o korisničkom iskustvu pri autentifikaciji. Dvofaktorska autentifikacija zasnovana na lozinkama i prepoznavanju lica nudi sigurnost podataka i relativno jednostavan proces logovanja, jer predstavlja kombinaciju tradicionalnog načina autentifikacije sa onim što je jedinstveno za svakog korisnika. Autentifikacija zasnovana na prepoznavanju lica obavlja se lako, ali zahtjeva naprednije tehnologije kako bi se realizovala i veliku dozu sigurnosti u pogledu čuvanja biometrijskih podataka. Za realizaciju ovog vida autentifikacije mogu se koristiti konvolucijske neuronske mreže. Za realizaciju adekvatnog modela, neophodno je koristiti skup podataka širokog opsega. Takav skup mora da sadrži što veći broj osoba i njihovih izraza lica kako bi se tokom treniranja razvio model koji će sa visokim stepenom tačnosti vršiti klasifikaciju osoba. U slučajevima kada su resursi ograničeni, moguće je modifikacijom prethodno istreniranih modela razviti model koji odgovara trenutnom problemu. Opravdano je smatrati da je savršen model teško realizovati, zbog toga je pri realizaciji projekta kao dodatna sigurnosna mjera upotrebljena autentifikacija zasnovana na lozinkama.

Autentifikacija je vršena prepoznavanjem osoba, ali se nije uzelo u obzir da se možda radi o slici osobe, a ne o samoj osobi koja pristupa procesu autentifikacije. Realizovani rad može se poboljšati realizacijom prepoznavanja lica pri čemu se vodi računa da li se radi o živoj osobi. Osim toga, autori rada planiraju istestirati ponašanje i drugih tehnologija za razvoj veb aplikacije kako bi analizirali performanse u pogledu brzine autentifikacije. Takođe, zavisno od resursa, autori rada planiraju realizovati model koji se ne temelji na *transfer learning*-u.

ZAHVALNICA

Autori rada se zahvaljuju višem asistentu Zorani Štaki i prof. dr Dragoljubu Krneti za pomoć pri realizaciji rada. Rad

je nastao u okviru predmeta *Vještačka inteligencija* na prvom ciklusu studija.

LITERATURA

[1] Увод у веб програмирање, HTML, CSS и JavaScript, Д. Мијић, Електротехнички факултет, Источно Сарајево, Академска мисао, Београд, 2019

[2] Number of internet users worldwide from 2005 to 2024, dostupno na: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide , posljednji pristup: januar, 2025. godine

[3] Višefaktorska autentifikacija, dostupno na: https://www.cert.hr/wp-content/uploads/2018/12/visefaktorska_autentifikacija.pdf, posljednji pristup: januar, 2025. godine

[4] https://haveibeenpwned.com/Passwords, posljednji pristup: januar, 2025. godine

[5] 6.5 Million LinkedIn Hacked Passwords, dostupno na: 6.5 Million LinkedIn Hacked Passwords | Acunetix, posljednji pristup: januar, 2025.

[6] Slika preuzeta sa sajta https://www.researchgate.net/figure/Evolution-of-face-recognition-solutions-over-time-grouped-based-on-the-feature_fig5_330132409, posljednji pristup januar 2025. godine

[7] Face Recognition Medium, dostupno na https://medium.com/@khwabkalra1/face-recognition-e45aff329fba, posljednji pristup januar 2025. godine

[8] Angular.io documentation, dostupno na: https://v17.angular.io/docs, posljednji pristup: januar, 2025. godine

[9] Overview of ASP.NET Core, dostupno na: https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0, posljednji pristup: januar, 2025. godine

[10] MySql documentation, dostupno na: https://dev.mysql.com/doc, posljednji pristup: januar, 2025. godine

[11] Flask documentation, dostupno na: https://flask.palletsprojects.com/en/stable, posljednji pristup: januar, 2025. godine

[12] Convolutional Neural Network (CNN) in Machine Learning, dostupno na: https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/, posljednji pristup, januar 2025. godine

[13] Slika preuzeta sa sajta https://ingoampt.com/cnn-convolutional-neural-networks-day-53, posljednji pristup januar 2025. godine

[14] What is Transfer Learning?, dostupno na: https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning, posljednji pristup: januar, 2025. godine

[15] VGG Face, dostupno na: https://exposing.ai/vgg_face, posljednji pristup januar, 2025. godine

[16] Slika preuzeta sa sajta https://sefiks.com/2018/08/06/deep-face-recognition-with-keras, posljednji pristup januar 2025. godine

ABSTRACT

The wide availability of the Internet requires the implementation of secure authentication in web applications. Multi-factor authentication can be used to meet the need for a higher level of security. Password and facial recognition-based multi-factor authentication combines a popular authentication technique with special, always-available resources. Face recognition models are made possible by convolutional neural networks. Such models require a large set of data, and in conditions when it is not possible to provide them, transfer learning is used. The creation of a facial recognition model and its integration into a web application for authentication are included in the work.

Combining AI and Full-Stack Engineering for User Authentication: A Facial Recognition Approach

Pavle Borčanin, Stefan Ristić, Nikola Milašinović