

# Praktična primjena neuralnih mreža za napredno prepoznavanje lica

Ćemer Alma, Omeragić Nedim

Prirodno-matematički fakultet, Odsjek za matematičke i kompjuterske nauke

**Abstract** - Biometrija je skup metoda za jedinstveno raspoznavanje ljudi na osnovu jedne ili više fizičkih osobina. Jedna od najzastupljenijih biometrija je prepoznavanje lica. Ima ogromne sposobnosti i može obavljati nekoliko zadataka osim detekcije i prepoznavanja lica. Istraživanje u području prepoznavanja lica počelo je već u 1960-ima, mjerenjem udaljenosti različitih dijelova lica, poput očiju, usta i nosa, te zatim izračunavajući različite udaljenosti kako bi se odredio identitet osobe. Tek krajem 1980-ih prepoznat je potencijal prepoznavanja lica kao poslovne potrebe. Danas, zahvaljujući tehnološkom napretku u kompjutacionoj moći i umjetnoj inteligenciji, prepoznavanje lica postaje popularan način identifikacije osoba. Jedan od najraširenijih slučajeva upotrebe prepoznavanja lica su mobilni uređaji. Mnogi korisnici svakodnevno primjenjuju ovu biometrijsku tehnologiju kako bi osigurali i otključali svoje telefone. Međutim, mogućnosti se protežu daleko od mobilnih uređaja i pružaju značajne prednosti u sigurnosti i efikasnosti u različitim industrijama.

## I. UVOD

U ovom radu je opisana primjena različitih metoda *Machine Learning-a* i *Deep Learning-a* za detekciju lica i njegovu klasifikaciju. Skup podataka je sačinjen od 1000 slika, odnosno po 100 slika za 10 različitih osoba. U nastavku će biti objašnjena prvenstveno priprema slika, odnosno njihovo pretprocesiranje, za ovaj problem sa dva algoritma: Haar Cascade i YOLO (You only look once), te će biti izvršeno njihovo poređenje. Zatim ćemo provesti analizu modela za prepoznavanja lica: obična neuralna mreža, konvolucijska neuralna mreža, VGG16, VGGFace i YOLO; i vršimo njihovu komparaciju. Slijed radnji za svaki model je isti: kada se dobije slika na kojoj je potrebno prepoznati osobu, prvo se koristeći YOLO algoritam detektuje lice (analizom u poglavlju II ćemo vidjeti zašto smo se odlučili za ovaj algoritam); zatim se isječak slike sa licem šalje određenom modelu. Za kraj će biti prikazani rezultati svih modela i kako je izvršeno njihovo testiranje u real-time pomoću web kamere.

## II. PRIPREMA PODATAKA

Skup podataka prvobitno može imati slike različitih rezolucija, te ih je potrebno procesirati da bi sve slike skupa podataka bile iste rezolucije. Pored toga, na jednoj slici se može nalaziti nešto što nije relevantno problemu prepoznavanja lica, odnosno neki drugi detalji na slici bi mogli navesti modele da ne uče dobro svoje težine za rješavanje problema prepoznavanja lica. Zbog toga je potrebno na slikama izvršiti detekciju lica prije prepoznavanja osobe. Jedna od glavnih biblioteka koju smo koristili pri procesiranju slika je OpenCV i njen glavni modul cv2. OpenCV koristi BGR format slike, te smo nakon čitanja slike vršili njeno pretvaranje u RGB format.

Za problem detekcije lica smo razmatrali dva algoritma: Haar Cascade i YOLO. Na Figure 1 je dat primjer kako algoritam YOLO detektuje jedno lice i više lica na jednoj slici. U oba slučaja, nakon što algoritmi detektuju lice na slici, lice se isječe i sprema u novu sliku dimenzije  $224 \times 224$ , što je prikazano na Figure 2. Slike na kojima algoritmi nisu našli lice, ili je na slikama više lica, su izbrisane iz skupa podataka. Na ovaj način smo izvršili pretprocesiranje podataka i spremanje novih slika u naš skup podataka.

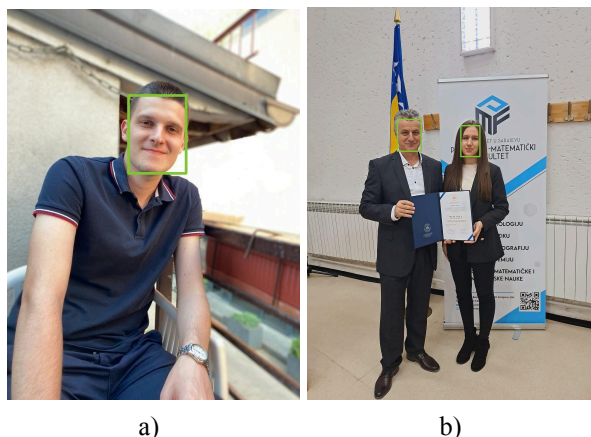


Fig.1 Primjer detekcije lica sa YOLO algoritmom: a) Jedna osoba  
b) Više osoba na slici



Fig.2 Primjeri slika nakon preprocesiranja

U nastavku su objašnjeni Haar Cascade i YOLO koji su korišteni za detekciju lica na našem problemu.

### A. Haar Cascade

Haar Cascade je metod za detekciju objekata na slikama baziran na mašinskom učenju, koji se koristi još od 2001. godine. Konkretno, u radu je korišten model za detekciju lica (`haarcascade_frontalface_default`) koji je prethodno istreniran. Algoritam funkcioniše tako što pokušava pronaći regiju u kojoj se može nalaziti lice, te tako dobija na brzini nekoristeći kompletnu sliku odjednom. Algoritam u pozadini koristi Adaboost metodu, te se smatra jednim od najbržih za detekciju objekata. U nastavku je opisano razvijanje algoritama kroz četiri faze: računanje Haar karakteristika (*eng. features*), kreiranje integralnih slika, korištenje Adaboost metode i implementiranje kaskadnih klasifikatora.

Dakle, prvi korak je sakupljanje Haar karakteristika koje su u suštini proračuni koji se izvode na susjednim pravouglim regijama. Proračun uključuje sabiranje intenziteta piksela u svakoj regiji i računanje razlika između suma. Na Figure 3 su prikazani primjeri Haar karakteristika. Ove karakteristike mogu biti teške za pronaći na velikim slikama. Zbog toga dolazi do primjene integralnih slika jer se u tom slučaju broj operacija smanjuje.

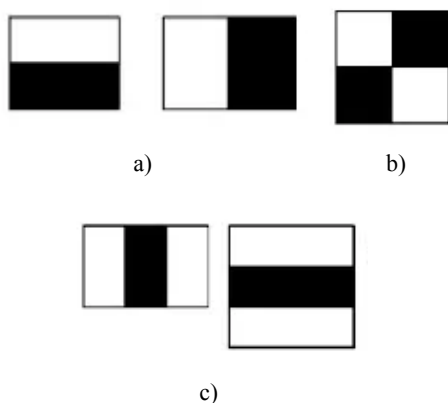


Fig.3 Primjeri Haar karakteristika (features) a) Ivica feature b) Pravougaonik feature c) Linija feature

Integralne slike su ubrzale računanje Haar karakteristika. Umjesto računanja na svakom pikselu, na ovaj način se kreiraju podpravougaonici i nizovi za svakog od podpravougaonika. Primjer dobivanja integralne slike je prikazan na Figure 4.

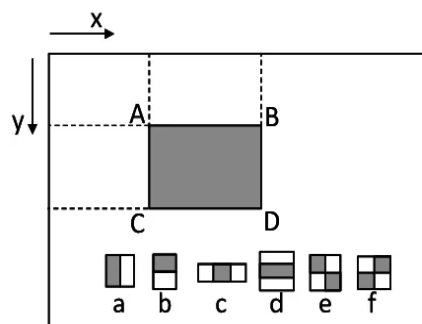


Fig.4 Dobivanje integralne slike

Važno je napomenuti da će skoro sve Haar karakteristike biti relevantne prilikom detekcije objekta, jer su te karakteristike važne za dati objekat. Postavlja se pitanje kako odrediti najbolje karakteristike koje reprezentiraju objekat od svih mogućih Haar karakteristika. Ovdje dolazi Adaboost metoda. Adaboost u suštini bira najbolje karakteristike tj. feature i trenira klasifikatore da ih koriste. Koristi kombinaciju slabih klasifikatora (modeli klasifikacije čije su performanse nešto bolje od random pogađanja) za kreiranje jakog klasifikatora koji algoritam može koristiti za otkrivanje objekata. Slabi klasifikatori se dobijaju pomjeranjem prozora preko ulazne slike i računanjem Haar karakteristika za svaki dio slike. Razlika se računa pomoću naučenog odstupanja koje razdvaja objekat i nešto što nije objekat. Zbog toga što se koriste slabi klasifikatori, potrebno je naučiti što više Haar karakteristika. Zadnji korak je kombinacija slabih klasifikatora u jaki klasifikator koristeći *cascading classifiers*. Cascade klasifikator se sastoji od niza faza, gdje je svaka faza skup slabih klasifikatora. Slabi klasifikatori se treniraju pomoću *boosting-a*, što omogućava veoma precizan klasifikator iz srednjeg predviđanja od svih slabih klasifikatora. Na osnovu ovog predviđanja, klasifikator ili odlučuje da naznači da je objekat pronađen (pozitivno) ili da pređe na sljedeću regiju (negativno).

Primjenom Haar Cascade algoritma, naš izvorni skup podataka se brojčano znatno smanjio. Razlog tome je što algoritam dosta griješi, ne pronalazi lica na slikama ili pronađe lice gdje se ono ni ne nalazi.

### B. YOLO za detekciju

YOLO je jedan od najpoznatijih modela za detekciju objekata u stvarnom vremenu zbog svoje brzine i tačnosti.

Prvo ga je uveo Joseph Redmon 2016. godine i od tada je prošao kroz nekoliko iteracija, a zadnja je YOLO v8. Konkretno, u radu je korišten model za detekciju lica (yolov8n-face.pt) koji je prethodno istreniran. Uzima sliku kao ulaz, a zatim koristi jednostavnu konvolucijsku neuronsku mrežu za detekciju objekata na slici. Algoritam funkcionira tako što sliku posmatra kao mrežu pravougaonika, te svaki pravougaonik posjeduje vjerovatnoću da se u njemu nalazi objekat. Algoritam dalje uklanja pravougaonike sa malom vjerovatnoćom, te preostale pomjera i modifikuje da bi pronašao odgovarajući objekat. Arhitektura CNN modela koji predstavlja osnovu za YOLO je prikazana na Figure 5.

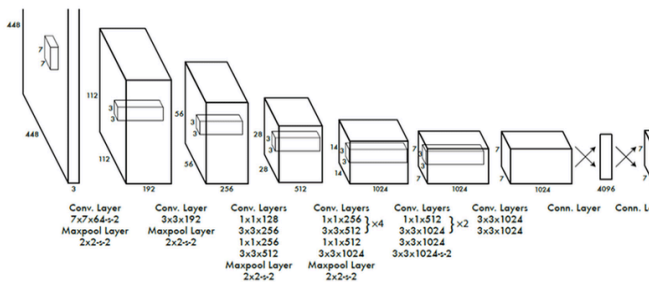


Fig.5 YOLO osnovna arhitektura

Prvih 20 konvolucijskih slojeva modela je istrenirano pomoću ImageNet (prva verzija YOLO) dodavanjem privremenog *average pooling* i potpuno povezanog sloja (*fully connected layer*). Zatim se ovaj prethodno istreniran model konvertuje za obavljanje detekcije jer su istraživanja pokazala da dodavanje konvolucije i povezanih slojeva u već istreniranom modelu poboljšavaju performanse. Zadnji YOLO povezani sloj predviđa vjerovatnoće klasa i koordinate pravougaonika mreže.

YOLO dijeli ulaznu sliku u  $m \times m$  mrežu. Ako se centar objekta nađe u jednoj ćeliji mreže, ta ćelija je odgovorna za detekciju objekta. Svaka ćelija predviđa koordinate pravougaonika i pouzdanost za te pravougaonike. Pouzdanost reflektuje koliko je siguran model da taj pravougaonik sadrži objekat i koliko tačno misli da je predviđen pravougaonik.

YOLO je mnogo napredovao kroz svojih 8 verzija, gdje su u prvim verzijama svi pravougaonici bili iste veličine, kasnije u predefinisanim veličinama i u posljednjoj verziji pravougaonici se prilagođavaju objektima. U svakoj verziji je postignuta veća brzina i tačnost u odnosu na prethodnu verziju. Također, u posljednjoj verziji je korištena EfficientNet arhitektura koja dodatno doprinosi na brzini i tačnosti modela. Model YOLO posljednje verzije nano veličine, koji je korišten u ovom radu može da obradi do 155 slika u sekundi, što ga stavlja daleko ispred svoje konkurencije.

Algoritam YOLO je pokazao bolje rezultate u pretprocesiranju podataka u odnosu na Haar Cascade. Slike koje su izbrisane u ovom procesiranju su uglavnom one na kojima se nalazi više lica. Zbog dobrog pretprocesiranja slika sa YOLO algoritmom za svaku osobu je ostalo približno 100 slika, te je novi skup podataka dobiven na ovaj način korišten u daljem radu.

### III. MODELI

Kako je spomenuto ranije, u našem istraživanju smo implementirali dva modela: neuralna mreža i konvolucijska neuralna mreža; zatim tehnikom *transfer learning-a* modele VGG16, VGGFace i YOLO smo prilagodili našem problemu i podacima.

Cijeli skup podataka se nalazi na Google Drive-u zbog jednostavnosti i lakšeg pristupa podacima. Učitavanjem podataka, dijele se na skup za trening (80% od ukupnog skupa) i skup za validaciju (20% od ukupnog skupa). Skup slika za testiranje nije pravljen iz ukupnog skupa podataka. Testiranje modela je rađeno povezivanjem web kamere sa Google Colab-om što je objašnjeno u poglavlju V. Također, postoji mogućnost proširivanja ukupnog skupa podataka pohranjujući slike dobijene tokom praktične primjene modela.

#### A. Neuralna mreža ANN

Naše rješavanje problema prepoznavanja lica smo započeli sa običnom neuralnom mrežom. Formirali smo mrežu od 5 slojeva sa 1024, 512, 256, 128 i 10 neurona, respektivno. Za prva 4 sloja aktivacijska funkcija je ReLu, te za posljednji sloj korišten je softmax. Također, korišten je Adam optimizator i categorical cross entropy za izračunavanje funkcije gubitka, te je batch veličine 16. Naravno, prije samog treniranja ovog modela bilo je potrebno slike pretvoriti u nizove brojeva. Ovaj model je pokazao znakove *overfitting-a* pa smo pokušavali dodati Dropout i L2 regularizaciju, međutim i uz korištenje Early Stopping i Model Checkpoint, model je bio jako nestabilan i nepraktičan za korištenje. Iz navedenih razloga je i izmišljena konvolucijska neuralna mreža koju koristimo u nastavku.

#### B. Konvolucijska neuralna mreža CNN

Naš CNN model se sastoji od 4 konvolucijska i 4 potpuno povezana sloja (*eng. fully connected layers*). Na

Figure 6 je prikazana konstrukcija konvolucijske neuralne mreže. Konvolucijski slojevi imaju 32, 64, 128 i 256 filtera, respektivno, te su filteri 3x3 dimenzije. Potpuno povezani slojevi imaju 1024, 512, 256 i 10 neurona, respektivno, te je za svaki sloj aktivacijska funkcija ReLu, osim za posljednji gdje se koristi softmax zbog klasifikacije. Na posljednji konvolucijski i na prva dva sloja dodan je Dropout od 0.2, kako model ne bi postigao overfitting. Korišten je Adam optimizer sa 0.00002 stopom učenja (*eng. learning rate*) i categorical cross entropy za izračunavanje funkcije gubitka, te je batch veličine 16. Kako bi model mogao bolje trenirati korištena je augmentacija podataka pomoću funkcije ImageDataGenerator iz biblioteke Keras. Funkcija će kroz svaku epohu, svaku sliku sa određenom vjerovatnoćom i određenom veličinom, zumirati, smicati (*eng. shear*) i horizontalno zaokrenuti. Također, korištene su callback funkcije EarlyStopping i ModelCheckpoint, koje zaustavljaju treniranje modela kada prestane napredovati, te spremaju težine modela.

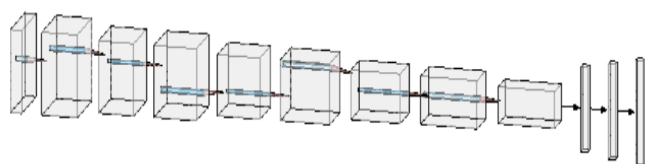


Fig.6 Arhitektura CNN modela

### C. VGG16

VGG16 je konvolucijska neuralna mreža sa 16 slojeva od kojih je 13 konvolucijskih i 3 potpuno povezana sloja (*eng. fully connected layers*). U konvolucijskim slojevima se koriste filteri dimenzije 3x3, a aktivacijska funkcija je ReLu. Prva dva potpuno povezana sloja imaju 4096 neurona, te zadnji 1000 neurona. Ulaz u mrežu zahtijeva sliku dimenzije 224x224. Arhitektura VGG16 modela je data na Figure 7. Ova mreža je istrenirana na više od milion slika iz ImageNet baze podataka i može da vrši klasifikaciju slika u 1000 različitih kategorija. Kao rezultat, mreža je naučila bogate reprezentacije. Preuzeli smo istreniranu mrežu koja je dostupna u biblioteci tensorflow. Od istreniranog VGG16 modela su uzeti konvolucijski slojevi, te smo dodali dva naša potpuno povezana sloja. Prvi potpuno povezani sloj sa aktivacijskom funkcijom ReLu ima 1024 neurona, a drugi sloj namijenjen je za predikciju te koristi aktivacijsku funkciju softmax i ima 10 neurona. Za treniranje mreže je korištena ImageDataGenerator funkcija iz biblioteke Keras i callback funkcije EarlyStopping i ModelCheckpoint.

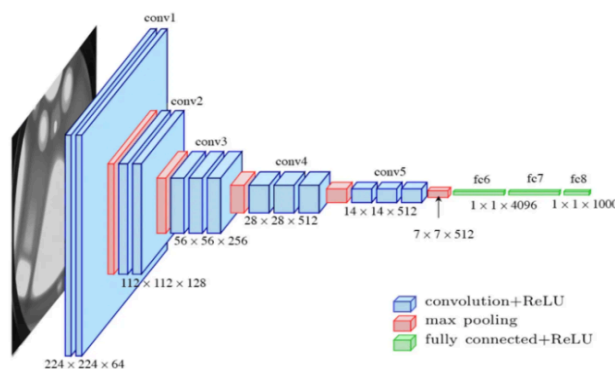


Fig 7. Arhitektura VGG16 modela

### D. VGGFace

Popularan i često korišten model za prepoznavanje lica je VGGFace. Ovaj model je predstavljen od strane Visual Geometry Group (VGG) na Univerzitetu u Oxfordu. Baziran je na modelu VGG16. Prepoznavanje lica radi izračunavanjem veza (*eng. embeddings*) slike lica i uspoređuje ih sa vezama izračunatih za drugu sliku lica tako što računa mjeru sličnosti sa Euklidovoj udaljenosti. Kao kod VGG16, ulaz u mrežu zahtijeva sliku dimenzije 224x224. Određivanje podudaranja lica se vrši uspoređivanjem udaljenosti. Sastoji se od 16 slojeva, 13 konvolucijskih slojeva i 3 potpuno povezana sloja, gdje posljednji sloj ima 2622 neurona. Arhitektura VGGFace modela je prikazana na Figure 8. Model je istreniran na skupu podataka od 2.6 miliona slika lica od 2622 osobe, gdje je za svaku osobu jednak broj slika lica. Težine ovog modela su open-source. Za model VGGFace smo uzeli konvolucijske slojeve i dodali dva potpuno povezana sloja kao u modelu VGG16. Također, kao u prethodna dva modela, korištene su callback funkcije EarlyStopping i ModelCheckpoint, i augmentacija prilikom treniranja.

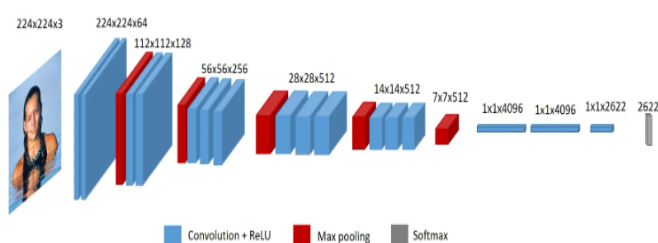


Fig 8. Arhitektura VGGFace modela



#### E. YOLO za klasifikaciju

Iako je primarno osmišljen za detekciju, YOLO model se može koristiti i za klasifikaciju na slikama. Naravno, bazirajući se na osnovnom modelu za detekciju objekata, uklanjajući korištenje mreže i pravougaonika, fokusirajući se samo na klasifikaciju objekta. U radu je korišten prethodno istrenirani (yolov8n-cls.pt) model, koji je treniran na ImageNet skupu podataka. Model izvorno prima slike dimenzija 224x224, što se podudara sa dimenzijama slika iz našeg skupa podataka i daje mu dodatnu prednost. Također, postoji još 4 robusnija modela koji izvorno daju nešto veću tačnost, međutim gube na brzini izvršavanja. Arhitektura YOLO modela je prikazana na Figure 5. YOLO model zahtjeva da se slike osoba nalaze u zasebnim folderima nazvanim po imenu osobe, te da se svi ti folderi nalaze u jednom folderu, kako bi se ispravno izvršilo labeliranje slika za treniranje, kao i za validaciju. Tokom treniranja korištene su callback funkcije EarlyStopping kao i ModelCheckpoint koje dolaze ugrađeni sa YOLO modelom iz biblioteke Ultralytics, te se šalju kao parametri. Također, model ima mogućnost da predviđa 5 osoba sa najvećom vjerovatnoćom da su baš one na slici, što model čini još korisnijim u praksi.

#### IV. REZULTATI

U Figure 9 su prikazane trening i validacijske tačnosti nakon treniranja svakog od modela navedenih u poglavlju III.

Model	Trening tačnost	Validacijska tačnost
ANN	68.16%	66.50%
CNN	89.26%	83.01%
VGG16	84.39%	74.76%
VGGFace	98.38%	96.12%
YOLO	98.08%	98.10%

Fig.9 Rezultati nakon treniranja modela

Rezultati modela ANN su pokazatelj da obična neuralna mreža nije pogodna za slike kao ulaz u mrežu. Model je nestabilan, nepraktičan za korištenje i nema veliku tačnost.

CNN model je pokazao znatno bolje rezultate u odnosu na običnu ANN mrežu. Tačnost CNN modela je zadovoljavajuća, uzimajući u obzir da naš skup podataka ima 1000 instanci slika što ne predstavlja velik broj

podataka.

VGG16 model daje znake za *overfitting*. Razlog tome je što je ovaj model učen na klasifikaciji različitih objekata, ne striktno lica osoba, pa nije prilagodljiv našem problemu. VGGFace je pokazao jako dobre rezultate i ima odličnu tačnost, zbog toga što je učen istom problemu i prilagodljiv je našim podacima.

YOLO model je postigao najbolje rezultate. U odnosu na ostale modele ima najveću validacijsku tačnost.

#### V. TESTIRANJE

Na osnovu rezultata, odlučili smo da za praktičnu primjenu koristimo naš CNN model, VGGFace model i YOLO model, jer se analizom rezultata pokazalo da najbolje vrše klasifikaciju. Kao što smo napomenuli ranije, testiranje modela izvršeno je real-time preko web kamere. Dakle, konstantno se procesiraju slike iz videa.

Prvi korak za testiranje modela je detekcija lica pomoću YOLO modela isključivo za detekciju lica, kao što je rađeno kod pretprocesiranja podataka. Primjenom YOLO modela za detekciju lica se dobiju koordinate okvira, odnosno pravougaonika koji uokviruje lice. Nakon što lice bude pronađeno i izrezano, šalje se modelu na predviđanje u formatu 224x224. Nakon što slika bude u potpunosti obrađena, pronađeno lice na videu biva uokvireno, te se iznad okvira ispiše ime osobe i vjerovatnoća koju je dao određeni model da je baš ta osoba ispred kamere.

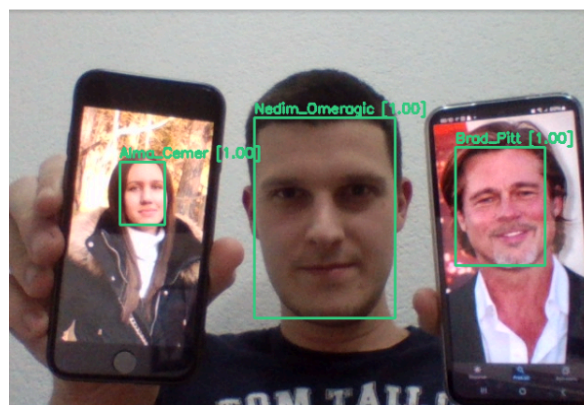


Fig.10 Real-time testiranje modela

Pošto YOLO model za detekciju lica pronalazi sva lica na slici, svako izrezano lice se šalje modelu i ovo omogućava prepoznavanje više lica na jednoj slici. Primjer ovakvog testiranja je dat na Figure 10. Ukoliko modeli prilikom testiranja daju vjerovatnoće manje od 0.7, smatramo da se detektovano lice ne nalazi u našem skupu podataka. Ta osoba će biti nepoznata, te se iznad okvira ispisuje *Unknown person* (prikazano na Figure 11). Važno je napomenuti da veliki uticaj na tačnost predviđanja ima

kvaliteta web kamere, svjetlost, udaljenost od kamere i slično.

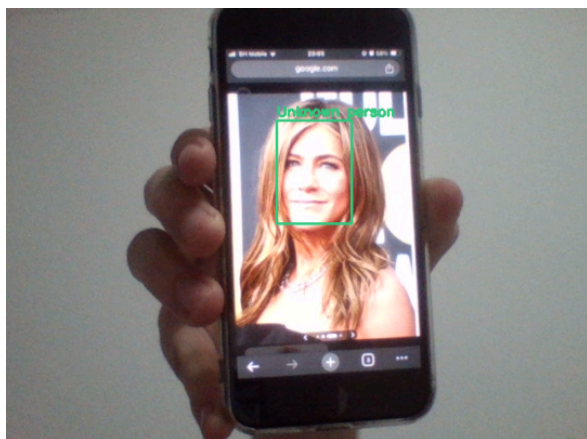


Fig.11 Real-time testiranje modela - nepoznata osoba

## VI. ZAKLJUČAK

Tokom našeg istraživanja razvili smo pet modela za rješavanje problema prepoznavanja lica. Naš cilj je bio napraviti model koji brzo radi prepoznavanje lica i da bude primjenjiv u real-time. Model VGGFace ima najbolju trening tačnost i nešto slabiju validacijsku tačnost ako se poredi sa YOLO modelom. Međutim, VGGFace zbog svoje kompleksnosti usporava proces prepoznavanja lica u real-time na prosječnim računarima. Drugi model sa najboljom trening tačnosti je YOLO za prepoznavanje lica. Pokazao se jako bržim od drugih modela pri real-time testiranju i daje realnije vjerovatnoće pri klasifikaciji. Također se kao jako brzim modelom pokazao naš CNN. Bilo bi moguće razviti bolji CNN model prikupljanjem većeg broja podataka.

YOLO model za prepoznavanje lica u kombinaciji sa YOLO modelom za detekciju lica daje ubjedljivo najbolje rezultate u odnosu na preostala četiri modela, ima veliku tačnost i brzinu. Ovakva kombinacija modela je sigurna za korištenju u nekom industrijskom sistemu za verifikacije osoba.