

SVEUČILIŠTE JURJA DOBRILE U PULI
FAKULTET INFORMATIKE

Antonio Labinjan

**Sustav za prepoznavanje lica koristeći CLIP i FAISS za skalabilnu
real-time identifikaciju**

ZAVRŠNI RAD

Pula, rujan 2025.

SVEUČILIŠTE JURJA DOBRILE U PULI
FAKULTET INFORMATIKE

Antonio Labinjan

**Sustav za prepoznavanje lica koristeći CLIP i FAISS za skalabilnu
real-time identifikaciju**

ZAVRŠNI RAD

JMBAG: 0303106891, redoviti student
Studijski smjer: Informatika
Kolegij: Web-aplikacije
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Nikola Tanković
Komentor: doc. dr. sc. Ivan Lorencin

Pula, rujan, 2025. godine

Sažetak

Prepoznavanje lica sve se češće primjenjuje u industrijama poput obrazovanja, sigurnosti i personaliziranih usluga [1]. Ovaj rad predstavlja sustav za prepoznavanje lica koji koristi mogućnosti modela CLIP za generiranje vektorskih reprezentacija. CLIP, model treniran na multimodalnim [2] podacima poput slika i videozapisa, stvara visokodimenzionalne značajke koje se pohranjuju u vektorsku bazu podataka [3] za daljnje upite. Sustav je dizajniran za preciznu identifikaciju u stvarnom vremenu, s potencijalnim primjenama u praćenju prisutnosti i sigurnosnim provjerama. Konkretni primjeri uporabe uključuju evidenciju dolazaka na događaje, implementaciju naprednih sigurnosnih sustava i druge slične scenarije.

Proces uključuje kodiranje poznatih lica u visokodimenzionalne vektore, njihovo indeksiranje pomoću FAISS vektorskog indeksa te usporedbu s nepoznatim slikama na temelju kosinusne sličnosti [4]. Eksperimentalni rezultati pokazuju visoku točnost [5, 6] veću od 90% te učinkovitu izvedbu čak i na skupovima podataka s velikim brojem zapisa.

Ključne riječi: prepoznavanje lica, CLIP, FAISS, identifikacija u stvarnom vremenu, praćenje prisutnosti, sigurnosne provjere, vektorska pretraga, ugradnje značajki

Abstract

Face recognition is increasingly being adopted in industries such as education, security, and personalized services [1]. This research introduces a face recognition system that leverages the embedding capabilities of the CLIP model. CLIP, a model trained on multimodal [2] data, such as images and videos, generates high-dimensional features, which are then stored in a vector store [3] for further queries. The system is designed to facilitate accurate real-time identification, with potential applications in areas such as attendance tracking and security screening. Specific use cases include event check-ins, implementation of advanced security systems, and more. The process involves encoding known faces into high-dimensional vectors, indexing them using a vector index FAISS, and comparing them to unknown images based on cosine similarity [4]. Experimental results demonstrate a high accuracy [5, 6] that exceeds 90% and a performance efficiency even in datasets with a high volume of entries.

Keywords: Face recognition, CLIP, FAISS, real-time identification, attendance tracking, security screening, vector search, embeddings

Uvod

U današnjem digitalnom dobu, tehnologije prepoznavanja lica imaju ključnu ulogu u transformaciji interakcije s računalnim sustavima, osiguravanju prostora te optimizaciji svakodnevnih procesa. Prepoznavanje lica jedno je od najbrže rastućih područja biometrijske autentikacije [7, 8], koje omogućuje brzu i pouzdanu identifikaciju osoba bez potrebe za fizičkim kontaktom ili dodatnom interakcijom [9]. Uz sve veću dostupnost naprednih računalnih resursa [10, 11] i sofisticiranih algoritama, implementacija sustava za prepoznavanje lica postala je nezaobilazan alat u raznim industrijama [9, 12].

Primjene ove tehnologije vrlo su raznolike i obuhvaćaju više sektora. U obrazovanju, prepoznavanje lica može se koristiti za automatizirano bilježenje prisutnosti studenata, čime se osigurava učinkovitije praćenje i vođenje evidencije. U sigurnosnom sektoru, ova tehnologija omogućuje identifikaciju sumnjivih osoba i unapređuje nadzor u javnim i privatnim prostorima, značajno pridonoseći prevenciji kriminala i povećanju sigurnosti. Nadalje, u personaliziranim uslugama, prepoznavanje lica pomaže u prilagodbi korisničkog iskustva omogućujući pristup relevantnim informacijama i uslugama temeljenim na identitetu korisnika [13].

Cilj ovog istraživanja je razviti i evaluirati učinkovit sustav za prepoznavanje lica koji koristi mogućnosti modela CLIP (Contrastive Language-Image Pre-Training) za ekstrakciju visokodimenzijskih značajki iz slika i videozapisa. Sustav, koji se temelji na tehnikama vektorske pretrage implementiranim u FAISS-u (Facebook AI Similarity Search), omogućuje brzo i precizno prepoznavanje lica u stvarnom vremenu. Kroz eksperimentalnu evaluaciju, istraživanje nastoji pokazati kako kombinacija naprednih metoda strojnog učenja i optimiziranih algoritama pretrage može unaprijediti točnost i performanse postojećih sustava za prepoznavanje lica, s potencijalnim primjenama u sigurnosnim sustavima, praćenju prisutnosti i drugim relevantnim područjima.

Povezani radovi

FAISS (Facebook AI Similarity Search)[14, 15] i CLIP (Contrastive Language-Image Pretraining)[16, 17, 18] predstavljaju dva napredna pristupa u područjima računalnog vida [19] i obrade prirodnog jezika [20] za obradu i dohvat podataka. FAISS je visoko učinkovit sustav za pretragu vektorske sličnosti [21], optimiziran za rad s velikim bazama podataka. Njegove primjene uključuju dohvat slika temeljen na vizualnoj sličnosti, biometrijske sustave za prepoznavanje lica, sustave preporuka te semantičku pretragu tekstualnih podataka i genetskih sekvenci. Posebno je koristan u analizi IoT podataka i detekciji anomalija [22], omogućujući brze usporedbe vremenskih serija [23] i obrazaca senzora. Nadalje, FAISS se koristi u ubrzanim sustavima za pretragu dokumenata, analizi korisničkog ponašanja i klasifikaciji podataka u stvarnom vremenu, značajno poboljšavajući skalabilnost i učinkovitost modela strojnog učenja.

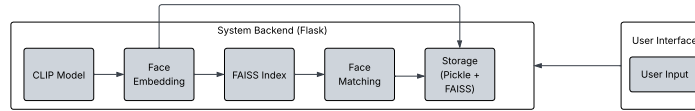
S druge strane, CLIP je model temeljen na transformatorima, treniran na velikom skupu podataka koji kombinira slike i deskriptivne tekstove, što mu omogućuje razumijevanje vizualnog sadržaja u kontekstu prirodnog jezika. Njegove primjene uključuju dohvat slika temeljen na tekstualnim upitima, automatsko generiranje opisa slika, zero-shot klasifikaciju [24, 25, 26], kao i unapređenje generativnih modela slika poput DALL·E [27] i Stable Diffusion [28]. CLIP se također koristi za moderiranje sadržaja, analizu videa, interakciju autonomnih sustava s okolinom te u kreativnim industrijama za generiranje umjetničkih vizuala temeljenih na semantičkim opisima.

Iako su FAISS i CLIP pojedinačno primijenjeni u različitim domenama, do sada nije zabilježena implementacija koja ih kombinira za skalabilno prepoznavanje lica u stvarnom vremenu. Ovaj rad uvodi prvi takav pristup, iskorištavajući mogućnosti CLIP-a za generiranje robusnih reprezentacija lica [29] te učinkovitost FAISS-a za iznimno brzu pretragu unutar velikih baza podataka [30]. Integracija ova dva modela omogućuje razvoj naprednih sustava za multi-modalnu obradu podataka, pružajući učinkovitije metode pretrage i klasifikacije u širokom spektru primjena - od sigurnosnih sustava do kreativnih industrija. Naša implementacija ne samo da poboljšava brzinu i točnost identifikacije lica, već i otvara nove mogućnosti u području računalnog vida koje dosad nisu istražene.

Metodologija i implementacija

Opisani sustav implementiran je u Pythonu koristeći CLIP za ekstrakciju embeddinga iz slika te FAISS za brzo pretraživanje. Videozapisi ljudskih lica obrađuju se u pojedinačne frameove. Skup podataka sastoji se od 50 klasa, pri čemu svaka klasa predstavlja jednu osobu, s dva videa po klasi - jednim za „trening“ i drugim za „validaciju“.[31]

Za prepoznavanje lica sustav najprije učitava poznate slike lica pomoću funkcije `load_known_faces()`, koja iterira kroz direktorije označene imenima klasa (osoba). Svaka klasa sadrži poddirektorij nazvan `train`, u kojem se nalaze slike korištene za „učenje“. Za svaku sliku u direktoriju za treniranje poziva se funkcija `add_known_face()`, koja izvlači embedding slike koristeći CLIP model i pridružuje ga odgovarajućoj klasi. Ovim postupkom gradi se baza poznatih lica koja se kasnije koristi za prepoznavanje.



Slika 1:

Shematski prikaz sustava.

Nakon ekstrakcije embeddinga, izračunati vektori organiziraju se u FAISS indeks pomoću funkcije `build_index`. FAISS omogućuje brzo pretraživanje vektora u visoko-dimenzijskom prostoru koristeći L2 udaljenost, poznatu i kao euklidska udaljenost.[32, 33]

L2 udaljenost između dva vektora $A = (a_1, a_2, \dots, a_n)$ i $B = (b_1, b_2, \dots, b_n)$ definira se kao:

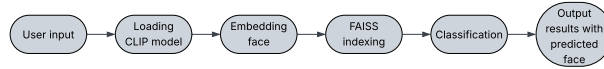
$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Ova metrika mjeri fizičku udaljenost između dviju točaka u n -dimenzionalnom prostoru i koristi se za usporedbu embeddinga lica (elemenata svakog lica) - što je udaljenost manja, to su embeddingi sličniji. FAISS omogućuje učinkovito pretraživanje najbližih susjeda [34, 35, 36] koristeći ovu metriku, čime se ubrzava proces prepoznavanja i izbjegava linearno pretraživanje.

U prikazanom sustavu FAISS se koristi tako da pohranjuje sve poznate embeddinge lica u `IndexFlatL2` indeks, koji omogućuje brzo pretraživanje temeljeno na L2 udaljenosti. Kada se dodaje novi embedding, on se jednostavno umeće u indeks zajedno s pripadajućom oznakom klase. Prilikom prepoznavanja novog lica, embedding ulazne slike uspoređuje se s embeddingima u FAISS indeksu, a sustav pronalazi $k2$ najbližih susjeda (embeddinga s najmanjom L2 udaljenošću). Ti se rezultati zatim koriste za klasifikaciju metodom glasanja pomoću funkcije `classify_face()`; među prvih $k1$ rezultata odabire se klasa koja se najčešće pojavljuje, brojeći sve pojave svake klase u tom skupu.[37, 38, 39]

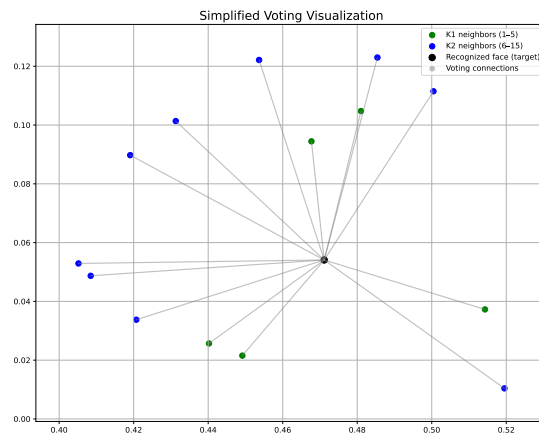
Ako threshold sličnosti [40] nije zadovoljen, u odluku se uključuju dodatni rezultati (od $k1$ do $k2$) kako bi se poboljšala točnost klasifikacije.

Klasa s najviše glasova postaje konačna predikcija, osim u slučaju kada nijedna klasa ne zadovoljava kriterije - tada se lice klasificira kao **Unknown**.



Slika 2: System

flowchart.



Slika 3: Pojednostavljena

vizualizacija votinga.

Eksperimenti, testiranja i rezultati

Sustav je prvo testiran na skupu podataka od približno 500 klasa, gdje je svaka osoba imala 1 videozapis, pri čemu su neki frameovi korišteni za učenje, a ostatak za validaciju (80-20 podjela). U ovom slučaju rezultati su pokazali 100% točnost jer su podaci za učenje i validaciju bili gotovo identični. Nadalje, kako bi se postigli realniji rezultati, korišten je *ytfaces*[41] skup podataka, koji se sastoji od videozapisa ljudi u različitim okruženjima. Ovaj pristup odabran je jer lica ljudi u videozapisima mogu uspješno simulirati ulaz koji bi sustav primao prilikom korištenja video kamere za prepoznavanje uživo. Iz skupa podataka izdvojena su 2 videozapisa za svaku klasu kako bi se simuliralo skeniranje lica u različitim uvjetima tijekom evaluacije.



Slika 4: Primjer slike
za učenje.



Slika 5: Primjer slike
za validaciju.

Proces evaluacije

Nakon definiranja skupa podataka, provedena je opsežna procedura testiranja različitih kombinacija parametara $k1$, $k2$ i $threshold$. Za učinkovito pretraživanje parametara korišten je `grid search`[42, 43, 44]. Definirani su sljedeći rasponi parametara: $k1$: 1-9, $k2$: 2-10, $threshold$: 0.5 - 1.0 s korakom od 0.01.

Dodatno je uvedeno pravilo da se preskaču kombinacije u kojima su $k1$ i $k2$ jednaki te gdje je $k2$ manji od $k1$, kako bi se izbjegli besmisleni koraci u pretraživanju.

Proces testiranja i evaluacije provodio se tako da je svaka slika iz validacijskih direktorija proslijeđena CLIP modelu radi kreiranja pripadajućih embeddinga. Zatim su embeddingi normalizirani pomoću `linalg.norm`[45], kako bi usporedba bila jednostavnija.

Za svaku ugradnju iz validacijskog direktorija izvršena je klasifikacija, tj. prepoznavanje lica. Za svaku sliku bilježila se stvarna oznaka klase i predviđena oznaka, gdje stvarna oznaka označava kojoj klasi slika pripada, a predviđena oznaka predstavlja rezultat modela. Ako se poklapaju, riječ je o točnoj klasifikaciji; ako se razlikuju, znači da je došlo do pogreške u predikciji. Predikcija se temelji na uzimanju svake validacijske ugradnje i usporedbi s poznatim ugradnjama iz skupa za treniranje, pri čemu se pronalazi najsličnija ugradnja (s najmanjom udaljenošću). Nakon što sustav odredi kojoj klasi pripadaju najsličnije ugradnje, validacijska ugradnja se pridružuje toj klasi i bilježi kao predikcija, uz automatsku provjeru točnosti.

Minimalni $threshold$ je namjerno postavljen na 0.5 jer bi predikcije s nižim $threshold$ om bile besmislene zbog pretjerane strogoće. Što je $threshold$ viši, to se uzima u obzir više slika, a broj predikcija raste. (Niži $threshold$ znači strožu odluku i obrnuto.)

Tijekom evaluacije rezultata, vrijednosti dobivene za svaki korak pretraživanja automatski su se spremale u datoteku `grid_search_results`. Konkretno, ona sadrži sljedeće vrijednosti: $k1$, $k2$, $threshold$, $precision$, $recall$, $f1$, tp , fp , tn , fn . [39]

Table 1: Objašnjenje metrika klasifikacije

tp	Točno pozitivne predikcije, tj. ispravno predviđeni pozitivni uzorci
fp	Netočno pozitivne predikcije, tj. pogrešno predviđeni pozitivni uzorci
tn	Točno negativne predikcije, tj. ispravno predviđeni negativni uzorci
fn	Netočno negativne predikcije, tj. pogrešno predviđeni negativni uzorci
Precision	Koliko od svih predviđenih pozitivnih uzoraka je zaista pozitivno
Recall	Koliko je pozitivnih uzoraka ispravno predviđeno kao pozitivno
F1-score	Harmonijska sredina precisiona i recalla

Kao izlaz svakog koraka grid searcha, generirao se redak podataka koji je sadržavao parametre **k1**, **k2** i **threshold**, zajedno s rezultatima dobivenim kombinacijom tih parametara. Na primjer:

k1, **k2**, **threshold**, **precision**, **recall**, **f1**, **tp**, **fp**, **tn**, **fn** (1, 2, 1.0, 0.7305, 0.7305, 0.677, 2702, 991, 46577, 991)

Svaka pojedinačna kombinacija parametara generirala je evaluacijske metrike koje su omogućile analizu performansi sustava. Rezultati su prikazani u obliku tablica, uključujući korištene vrijednosti parametara te dobivene metrike: precision, recall, F1-score, kao i apsolutne vrijednosti za true positive (TP), false positive (FP), true negative (TN) i false negative (FN) primjere. Ovakav izlaz pruža detaljan uvid u to kako promjene hiperparametara utječu na performanse klasifikacije, omogućujući odabir optimalne kombinacije koja postiže najbolju ravnotežu između izlaznih metrika modela.

Na primjer, konfiguracija **k1=1**, **k2=2** i **threshold=1.0** rezultirala je vrijednostima **precision=0.7305** i **recall=0.7305**, uz pripadajući broj točnih i netočnih klasifikacija (TP=2702, FP=991, TN=46577, FN=991). Analizom većeg broja ovakvih rezultata moguće je procijeniti idealnu ravnotežu između različitih metrika te prilagoditi model kako bi se postigla optimalna učinkovitost klasifikacije u prepoznavanju lica.

Tijekom evaluacije također se pratila izvedba sustava na različitim veličinama skupova podataka. Kao što se logično može zaključiti, vrijednosti metrika uglavnom opadaju kako raste broj klasa i slika. Kod skupa s manje od 25 klasa, performanse su bile savršene jer je broj embeddinga bio dovoljno malen da se izbjegnju pogrešne klasifikacije, zahvaljujući velikom raspoloživom prostoru u vektorskom prostoru. S povećanjem broja slika, pojavile su se poneke pogrešne klasifikacije.

Table 2: Evaluacijske metrike po klasi za 25 klasa

Klasa	Precision	Recall	F1-Score	Support
1	1.0000	1.0000	1.0000	57
2	1.0000	1.0000	1.0000	67
3	0.9662	1.0000	0.9828	257
4	1.0000	1.0000	1.0000	249
5	0.7037	1.0000	0.8261	57
6	1.0000	1.0000	1.0000	102
7	1.0000	1.0000	1.0000	41
8	1.0000	0.6712	0.8033	73
9	0.7273	1.0000	0.8421	56
10	1.0000	1.0000	1.0000	16
11	1.0000	1.0000	1.0000	38
12	0.9494	0.9740	0.9615	77
13	1.0000	1.0000	1.0000	7
14	0.9608	1.0000	0.9800	49
15	0.8333	1.0000	0.9091	25
16	1.0000	1.0000	1.0000	13
17	1.0000	1.0000	1.0000	18
18	0.8788	1.0000	0.9355	58
19	0.8621	1.0000	0.9259	25
20	1.0000	1.0000	1.0000	98
21	1.0000	1.0000	1.0000	199
22	1.0000	1.0000	1.0000	17
23	0.0000	0.0000	0.0000	49
24	1.0000	0.9649	0.9821	114
25	1.0000	0.9592	0.9792	49
Unknown	0.0000	0.0000	0.0000	0

Učinci augmentacije

Tijekom eksperimentalne faze na skup za treniranje primijenjena je augmentacija podataka[46] koristeći tehnike poput rotacije, zrcaljenja, izrezivanja i zamučivanja. Međutim, te metode su značajno narušile performanse modela uvođenjem neprirodnih distorzija koje su čak i ljudima bile teško prepoznatljive. Takve augmentacije ne odražavaju uvjete stvarnog svijeta, gdje se javljaju samo blage varijacije u osvjetljenju ili kutu - promjene koje sustav pouzdano može podnijeti. Realističnije varijacije, poput promjene frizure, brade ili šminke, imaju minimalan utjecaj na embeddinge i ne narušavaju prepoznavanje.

Ograničenja testiranja

Kod 25 klasa, pojavljuju se manja kriva klasificiranja na pojedinim sličnim slikama, ali ona su gotovo zanemariva, a svaka klasa je u većini slučajeva ispravno predviđena. Kod 50 klasa, točnost klasifikacije pada na 73%. Način za poboljšanje rezultata svakako bi bio ručno snimanje 50 ili više osoba u 2 videa pod različitim uvjetima, što bi omogućilo kvalitetnije testne podatke i bolje embeddinge. Ovaj dio bit će dodatno istražen tijekom evaluacije skalabilnosti.

Testiranje i validacija provodili su se usporedbom predikcija na validacijskom skupu sa slikama iz skupa za treniranje, umjesto stvarnog ulaza u stvarnom vremenu (live input). Iako je ova metoda evaluacije nužna, ona ne odražava u potpunosti izazove prepoznavanja lica u stvarnom vremenu s live feeda, gdje faktori poput promjena osvjetljenja, različitih kutova lica, izraza lica i varijacija u rezoluciji mogu dodatno utjecati na performanse modela. S druge

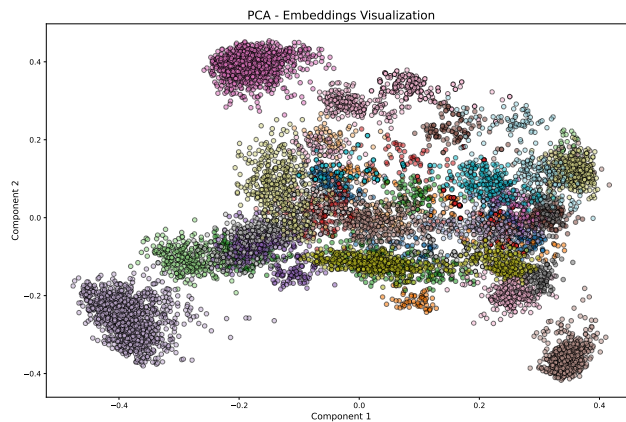
strane, isključuje i određene prednosti koje pruža testiranje u stvarnom vremenu. Nažalost, zbog ograničenja standardnih eksperimentalnih protokola, simulacija ovih stvarnih uvjeta kroz validacijski skup trenutno predstavlja najbližu moguću metodu testiranja prije implementacije u sustave prepoznavanja lica u stvarnom vremenu. Korištenje live feeda omogućuje analizu mnogo više uzastopnih frejmova u kraćem vremenu, što omogućuje agregaciju predikcija i smanjuje utjecaj pojedinih anomalija koje se mogu pojaviti na pojedinim slikama zbog lošeg osvjetljenja, kutova snimanja, zamućenja itd. Umjesto oslanjanja na statične slike, sustav može tijekom vremena prikupiti više uzoraka lica od korisnika, poboljšavajući pouzdanost prepoznavanja. Ovakav pristup je posebno koristan u dinamičnim sustavima gdje je cilj postići stabilno i precizno prepoznavanje, poput sigurnosnih sustava ili bezkontaktne autentifikacije [47, 48].

Dodatno, live input omogućuje primjenu adaptivnih tehnika poput praćenja lica (face tracking), gdje model kontinuirano može korigirati i poboljšavati detekciju analizom mikro-pokreta i različitih izraza lica. Ova fleksibilnost često vodi do bolje ukupne točnosti u stvarnim uvjetima nego što bi se očekivalo na temelju rezultata validacije dobivenih tradicionalnim metodama testiranja sa statičnim slikama.

Implementacija FAISS-a

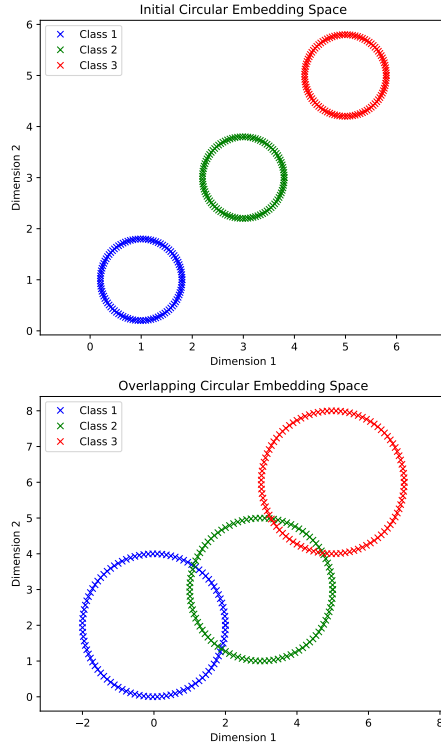
Embeddinzi lica vizualizirani su u dvodimenzionalnom prostoru korištenjem analize glavnih komponenti (PCA)[49, 50], koja smanjuje dimenzionalnost transformiranjem višedimenzionalnih podataka u manji broj glavnih komponenti, pritom zadržavajući što je moguće više varijabilnosti iz originalnih podataka i njihovog rasporeda unutar CLIP embedding prostora. Ovakav pristup omogućuje uvid u njihovu distribuciju i međusobne odnose. Primijećeno je određeno preklapanje između embeddinga različitih klasa[51, 52], što ukazuje na izazove u razlikovanju vrlo sličnih lica. Preklapanje je bilo osobito izraženo kod osoba sa sličnim crtama lica, osvjetljenjem ili izrazima lica, što otežava jednostavne metode prepoznavanja.

Što je više sličnih klasa u sustavu i što je veći broj njihovih instanci, to je preklapanje veće, što otežava jasno odvajanje klasa i povećava rizik od pogrešne klasifikacije ili zabune među klasama koje su blizu u prostoru. Dva moguća jednostavna rješenja bila bi smanjenje ukupnog broja klasa ili smanjenje broja instanci po klasi, no nijedna od opcija nije zadovoljavajuća. Ova situacija naglašava potrebu za naprednim metodama pretraživanja, poput FAISS-a, koji omogućuje brzo i precizno prepoznavanje unatoč preklapanju embeddinga.



Slika 7: Realističan prikaz klasa i njihovog preklapanja

Kako bi se riješio problem povećanog preklapanja, uveden je FAISS (Facebook AI Similarity Search), optimizirani sustav za pretraživanje sličnosti u velikim embedding prostorima. FAISS koristi učinkovite strukture podataka, poput ANN (Approximate Nearest Neighbors) pretraživanja i kvantizacije, kako bi ubrzao proces prepoznavanja i smanjio negativne učinke preklapanja. To omogućuje točniju i bržu klasifikaciju instanci unatoč povećanoj gustoći podataka u prostoru. Prethodno spomenuti parametri k_1 i k_2 posebno se odnose na FAISS i predstavljaju broj podudaranja koje traži u svakom koraku klasifikacije embeddinga.



Slika 8: Pojednostavljeni prikaz preklapanja

Postizanje maksimalnih performansi

Nakon postizanja određene vrijednosti thresholda, model doseže svoj maksimalni potencijal klasifikacije, gdje daljnje povećavanje thresholda više ne utječe na metrike. To ukazuje na visoku sigurnost u njegove odluke i sposobnost jasnog razlikovanja pozitivnih i negativnih primjera. Sve pozitivne klasifikacije su iznad određenog thresholda, dok su negativne ispod njega, što znači da model dosljedno i točno donosi odluke bez neizvjesnih predikcija u srednjem rasponu vjerojatnosti. Ovaj efekt potvrđuje robusnost modela i njegovu otpornost na promjene thresholda, postižući optimalnu točnost u stvarnim scenarijima. To je posebno korisno u sustavima gdje je pouzdanost klasifikacije ključna[53, 54], a daljnja podešavanja mogu se koristiti samo za fino podešavanje modela u specifičnim slučajevima.

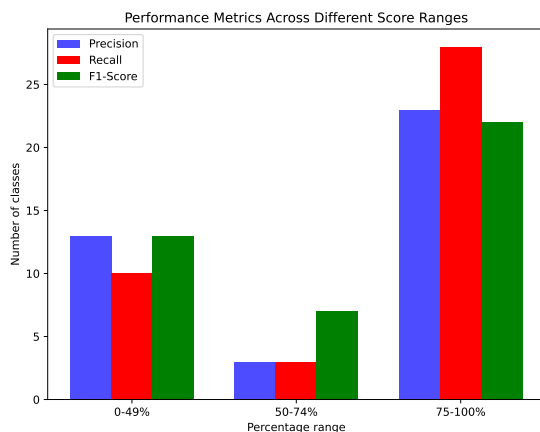
Precision i recall ostaju iste jer su FP i FN uvijek jednaki unutar svake kombinacije parametara. To znači da model dosljedno miješa određene klase, bez obzira na threshold ili k-vrijednosti koje se uzimaju u obzir.

Rezultati po klasama i rukovanje problematičnim klasama

Tijekom procesa grid search-a, najbolje metrike postignute su pri određenoj konfiguraciji. Detaljna evaluacija klasa za najbolju točku grid search-a s 50 klasa dala je sljedeće rezultate: (1, 3, 0.82, 0.7305, 0.7305, 0.6770, 2702, 991, 46577, 991) - što ukazuje na ukupno 991 pogrešno klasificiranu sliku.

Detaljnija analiza ovih rezultata pokazuje da neke klase značajno lošije rade od drugih, što doprinosi ukupnoj stopi pogrešne klasifikacije.

Evaluacija pokazuje da neke klase postižu izvrsne rezultate, dok su druge značajno slabije, čime utječu na ukupnu učinkovitost sustava. Konkretno, tablica prikazuje raspodjelu klasa u tri grupe prema uspješnosti (0-49%, 50-74%, 75-100%).



Slika 10: Distribucija metrika po rasponima uspješnosti.

Identifikacijom problematičnih klasa koje je čak i vizualno teško klasificirati zbog njihove niske kvalitete, došli smo na ideju evaluirati model bez tih klasa. Uklanjanjem 17 izrazito nekvalitetnih klasa testirana je najbolja kombinacija parametara za prethodni skup podataka te su postignuti sljedeći rezultati:

Accuracy: 0.8786

Recall: 0.8786

F1 Score: 0.8628

TP: 2295, FP: 317, TN: 9827, FN: 317

Detaljnija analiza confusion matrixa [55] otkriva da postoje određene problematične klase koje se konstantno miješaju jedna s drugom. Konkretno, to su: 11-34, 39-50, 36-50, 29-49, 48-47, 35-23, 28-30. Ako se neke od njih uklone, metrike se automatski popravljaju, iako ne savršeno. Vidljivo je da su sada miješanja 36-50, 39-50, 47-6, 34-13, 34-3. Dakle, teoretski bi se sustav mogao dovesti do 100% performansi iterativnom evaluacijom i eliminacijom klasa koje se miješaju, ali takva evaluacija ne bi odražavala realne uvjete implementacije u stvarnom svijetu. Međutim, uklanjanje klasa s niskokvalitetnim ili nekonzistentnim podacima ima izrazito pozitivan utjecaj na ukupnu izvedbu sustava.

Table 3: Metrike za svaku klasu





Class	Precision	Recall	F1-Score	Support
1	0.98	1.00	0.99	57
2	0.92	0.99	0.95	67
3	1.00	1.00	1.00	257
4	1.00	0.98	0.99	249
5	0.65	0.96	0.78	57
6	0.92	1.00	0.96	102
7	1.00	1.00	1.00	41
8	0.91	0.67	0.77	73
9	0.98	1.00	0.99	56
10	0.31	1.00	0.48	16
11	0.20	1.00	0.33	38
12	0.99	0.97	0.98	77
13	0.88	1.00	0.93	7
14	0.21	0.14	0.17	49
15	0.52	1.00	0.68	25
16	1.00	1.00	1.00	12
17	1.00	1.00	1.00	18
18	0.94	1.00	0.97	58
19	1.00	1.00	1.00	25
20	1.00	1.00	1.00	98
21	1.00	0.91	0.95	199
22	1.00	1.00	1.00	17
23	0.00	0.00	0.00	49
24	0.40	0.69	0.51	114
25	1.00	0.96	0.98	49
26	0.87	1.00	0.93	13
27	0.95	1.00	0.98	61
28	1.00	0.05	0.10	57
29	0.00	0.00	0.00	57
30	0.12	0.05	0.07	185
31	0.48	0.69	0.56	58
32	0.99	1.00	0.99	87
33	1.00	1.00	1.00	71
34	1.00	0.09	0.17	173
35	0.51	1.00	0.68	53
36	0.75	1.00	0.85	156
37	0.61	1.00	0.76	35
38	0.62	1.00	0.76	73
39	0.53	0.98	0.69	61
40	1.00	1.00	1.00	13
41	0.00	0.00	0.00	0
42	1.00	1.00	1.00	45
43	0.00	0.00	0.00	53
44	0.00	0.00	0.00	57
45	0.42	0.93	0.58	83
46	0.62	1.00	0.76	69
47	1.00	0.10	0.18	153
48	0.42	0.95	0.58	77
49	1.00	0.91	0.95	56
50	1.00	0.04	0.08	142
Unknown	0.00	0.00	0.00	0

Table 4: Tablica distribucije performansi

	Precision	Recall	F1-Score
0-49%	10, 11, 14, 23, 24, 29, 30, 31, 41, 43, 45, 48	14, 23, 28, 29, 30, 34, 41, 43, 44, 47, 50	10, 11, 14, 23, 28, 29, 30, 34, 41, 43, 44, 47, 50
50-74%	5, 15, 35, 37, 38, 39, 46	8, 24, 31	15, 24, 31, 35, 39, 45, 48
75-100%	1, 2, 3, 4, 6, 7, 8, 9, 12, 13, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 32, 33, 34, 36, 40, 42, 47, 49, 50	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 32, 33, 35, 36, 38, 39, 40, 42, 45, 46, 48, 49	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 32, 33, 34, 36, 38, 42, 46, 49

Uklanjanjem problematičnih klasa iz skupa podataka dolazi do znatno manjeg stupnja zabune tijekom klasifikacije, a izvedba sustava se poboljšava, što je još jedan pokazatelj da većina pogrešaka nije proizašla iz same implementacije sustava, već iz klasa čije slike za treniranje i validaciju imaju određene nedostatke koji onemogućuju njihovu ispravnu klasifikaciju. Naravno, krivulje poput ROC-a (stisnute prema gornjem lijevom kutu) ili P-R krivulje (dijagonalne) i dalje će izgledati slično zbog prirode multiklasifikacije i velikog broja TN, tj. jednakosti lažno pozitivnih i lažno negativnih.

Slijedi nekoliko primjera parova train-val:

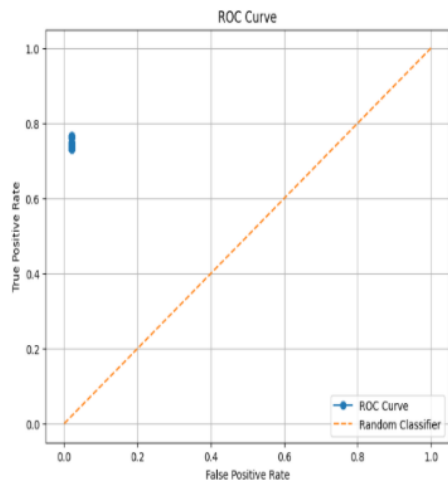
Class	Images	Description
20		Class 20 achieves perfect metrics with a precision of 1.00, meaning that all samples predicted as class 20 are indeed correct.
17		Class 17 also records excellent results with a perfect precision of 100%, making it flawless in predictions.
46		Class 46 achieves a precision of 0.62, which is decent, but there is room for improvement.
39		Class 39 has a precision of 0.53, which means there is a significant number of false positive predictions.

The "Unknown" class achieves a precision of 0.00, which is actually a good sign because the "Unknown" class is defined for cases where the model doesn't know how to classify a particular image. This indicates that the system will provide an output for each input that, with sufficient data quality, will be accurate.

Slika 13: Nekoliko

primjera klasa.

ROC krivulja



Slika 14: Globalna ROC krivulja stisnuta u gornji lijevi kut zbog multiklasifikacije.

ROC krivulja [37, 39, 56] - standardni alat za evaluaciju u binarnoj klasifikaciji - nudi ograničenu interpretabilnost u našem višeklasnom okruženju zbog značajne neravnoteže između true negative (TN) i true positive (TP) primjera. Kao što je prikazano na Slici 14, krivulja ostaje stisnuta u gornjem lijevom kutu grafa, pokazujući minimalna pomicanja kroz različite pragove.

U višeklasnoj klasifikaciji, svaka točna predikcija implicitno generira negativne predikcije za sve ostale klase, što dovodi do iznimno niskih False Positive Rate (FPR) vrijednosti i minimalne varijacije u True Positive Rate (TPR) [40]. Nadalje, false positives i false negatives su intrinzično povezani - pogrešna klasifikacija uzorka u jednu klasu inherentno znači pogrešnu klasifikaciju za drugu - što rezultira zrcalnim odnosom između FPR i FNR vrijednosti.

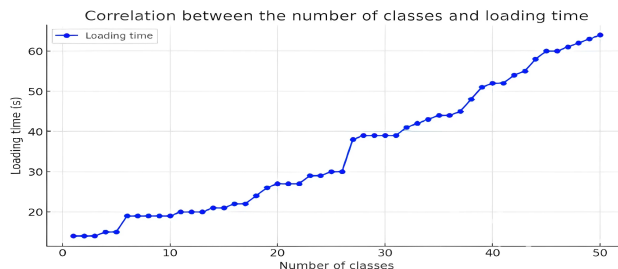
Iako ROC krivulja ne otkriva progresivne TPR trendove, njezin položaj potvrđuje sposobnost modela da dosljedno prepozna true positives uz mali broj false positives. Ovo ponašanje odražava samu strukturu zadatka: za svaku ulaznu sliku, model mora odbaciti 49 klasa i prihvatiti samo jednu, čime se stvara neuravnotežen omjer TN i TP vrijednosti.

Vremenski test skalabilnosti

Prilikom testiranja performansi sustava promatran je utjecaj povećanja broja klasa na vrijeme učitavanja, s posebnim naglaskom na faze pretvaranja slika u embeddings i njihovo učitavanje u FAISS [44, 52]. Kao što se i očekivalo, vrijeme učitavanja nije raslo proporcionalno s brojem klasa. Umjesto linearnog odnosa, sustav je pokazao neproporcionalan rast u vremenu (povećanje broja klasa bilo je veće od povećanja vremena izvršavanja), što ukazuje na optimizirane procese učitavanja i dobru skalabilnost. Ovaj rezultat sugerira da sustav može

učinkovito rukovati većim brojem klasa bez značajnog povećanja vremena učitavanja. Očekivano povećanje pratilo bi sljedeću formulu:

$$\text{Total loading time} = \text{Number of classes} \times \text{Loading time per class}$$



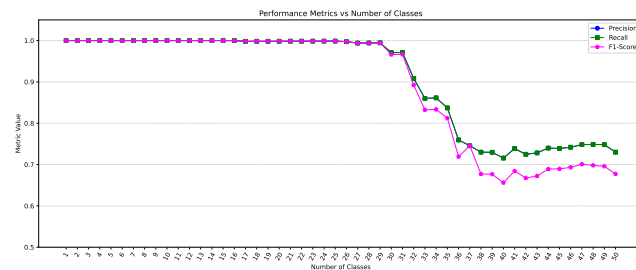
Slika 15: Vremenski

scalability test.

Međutim, povećanje je bilo primjetno manje. Za učitavanje sustava s 1 klasom bilo je potrebno 14 sekundi, dok je za 50 klasa trebalo 64 sekunde. Dakle, uz povećanje broja klasa od 4900%, vrijeme učitavanja povećalo se za samo 357.14%.

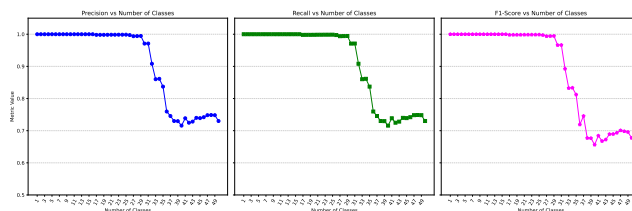
Test skalabilnosti temeljen na metrikama

Graf prikazuje kako izvedba modela ovisi o broju klasa, pri čemu Precision, Recall i F1-score inicijalno ostaju na visokoj razini, a značajan pad javlja se tek nakon otprilike 30 klasa. Međutim, pad u izvedbi nije nužno vezan samo uz povećanje broja klasa, već i uz kvalitetu i raznolikost slika u skupu podataka. Također, budući da vrijednosti preciznosti i odziva ostaju konzistentno usklađene, krivulja preciznosti-odziva (precision-recall) bi bila gotovo dijagonalna linija od (0,0) do (1,1), pružajući minimalne dodatne informacije i stoga ne bi bila osobito informativna u ovom kontekstu. Iako sve tri metrike opadaju s povećanjem broja klasa, kvaliteta podataka igra ključnu ulogu u održavanju dobrih rezultata. S dobro pripremljenim skupom podataka, model može održati visoku izvedbu čak i uz veći broj klasa. Štoviše, dodavanjem klasa visoke kvalitete moguće je poboljšati metrike, što sugerira da struktura, kvaliteta i informativnost [57] podataka imaju veći utjecaj na izvedbu sustava nego sam broj klasa.



Slika 16: Metrički

scalability test.



Slika 17: razdvojeni

grafikoni.

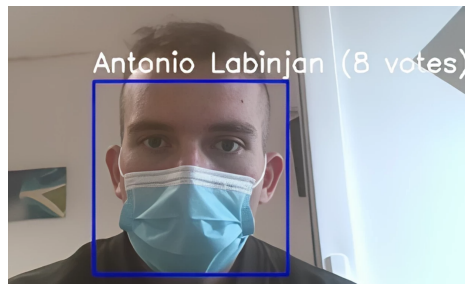
Usporedba sa sličnim rješenjima

Model	Architecture	Key metric	Accuracy	Pros	Cons
FaceNet (Google)	CNN + triplet loss	Cosine similarity	99.63%	Small embeddings (128 dim), robust	High training resource usage
DeepFace (Facebook)	CNN (9 layers)	Euclidean + cosine	97.35%	First human-level model	Hard to deploy and optimise
VGG-Face	CNN (37 layers)	Euclidean	98.95%	Deep architecture improves accuracy	Many parameters, slow inference
ArcFace	CNN + Arc loss	Cosine similarity	99.40%	Better class separation	Needs fine-tuned models
YOLO-Face	YOLO CNN	IoU	99.8%	Fast all-in-one recognition	Weaker on complex faces
Our system (CLIP + FAISS)	CLIP + FAISS search	L2/Cosine	~87-100% depending on dataset	Fast, scalable, no training required	Embedding quality affects performance

Table 6: Usporedba sustava za prepoznavanje lica

U usporedbi sa sličnim rješenjima [12], naš sustav ima određene prednosti. Prvo, nije potrebno tradicionalno treniranje; nove klase/lica mogu se dodati vrlo brzo i jednostavno. Nadalje, sustav je iznimno jednostavan za implementaciju u aplikacijama koje uključuju različite oblike detekcije lica, bez potrebe za značajnim izmjenama, jer princip rada ostaje nepromijenjen. Također, zahvaljujući snažnim mogućnostima CLIP-a u ekstrakciji embeddinga, uz skup podataka visoke kvalitete prikupljen u kontroliranim uvjetima, sustav pokazuje značajno

poboljšanje izvedbe. Omogućuje uspješnu identifikaciju iznimno velikog broja različitih lica, posebno ako su uvjeti za snimanje referentnih slika koje se dodaju u sustav unaprijed definirani. Posebno je zanimljivo da uspješna identifikacija ne zahtijeva da live input obuhvati cijelo lice. Sustav nije ograničen na pokrivala za glavu, maske i slično. Ako je lice i dalje dovoljno prepoznatljivo unatoč takvim preprekama, sustav će ga pravilno klasificirati. Ključno je da sustav uspješno prepozna specifične dijelove lica koji najviše doprinose razlikovanju pojedinca od drugih.



Slika 18: Primjer live detekcije prekrivenog lica.

Primjena u stvarnom životu

Opisani sustav može se implementirati za detekciju i prepoznavanje lica u stvarnom vremenu koristeći računalnu kameru, tako da se frameovi kamere proslijede kao ulaz, oko svake detektirane osobe kreira bounding box [58, 59], izvuku embeddingi i proslijede u opisani sustav. Predviđena najvjerojatnija klasa (s najviše glasova) vraća se kao izlaz funkcije i dodaje kao oznaka na frame. Sustav također može istovremeno prepoznavati više poznatih lica unutar istog framea, što dodaje dodatnu dimenziju sustavu i omogućuje naprednije prepoznavanje u složenim situacijama. Također je važno napomenuti da se lica mogu klasificirati i u zahtjevnim uvjetima, poput lošeg osvjetljenja. Ova implementacija pruža brojne mogućnosti, kao što su sustavi za evidenciju dolazaka studenata, obračun plaća prema odrađenim satima, sigurnosni sustavi, razni interaktivni sučelja i slično. Kašnjenje sustava [60] je manje od 50 ms, a sustav može obraditi oko 200 frejmova u sekundi, što je pokazatelj prilično dobre izvedbe.



Slika 19: Primjer live detekcije u zatamnjenom okruženju.



Slika 20: Primjer live detekcije s više lica.

Ograničenja i zaključci

Analiza prepoznavanja lica pokazala je da kvaliteta slike i pozadina značajno utječu na točnost modela. Kada su pozadine previše raznolike, embeddingi postaju manje precizni, što otežava ispravno prepoznavanje pojedinaca. Dodatni izazovi nastaju zbog teksta, objekata u frejmu i velikih promjena osvjetljenja, jer oni uvode varijacije koje nisu povezane s identitetom osobe, već s okolinom.

Ovaj problem je posebno izražen kada sustav radi s velikim brojem različitih lica - dok model postiže visoku točnost u kontroliranim uvjetima, njegova izvedba može opasti ako podaci variraju u osvjetljenju, kutu snimanja ili kvaliteti slike. Faktori poput zamućenja, niske rezolucije i kontrasta dodatno mogu smanjiti pouzdanost prepoznavanja.

Za poboljšanje rezultata, korisno bi bilo standardizirati uvjete za snimanje referentnih slika ili primijeniti metode koje osiguravaju veću konzistentnost ulaznih podataka. Na primjer, u sustavu za automatsko prepoznavanje studenata na sveučilištu, preporučljivo je da sve referentne slike budu snimljene pod istim uvjetima i s uniformnom pozadinom, što bi neutraliziralo pogreške u prepoznavanju.

Osim tehničkih ograničenja, etički aspekti igraju važnu ulogu u odgovornom korištenju opisanog sustava za prepoznavanje lica [61, 62].

Pristup predstavljen u ovom radu osigurava očuvanje privatnosti po dizajnu: korisnici dobrovoljno daju ulazne slike, koje se odmah transformiraju u nereverzibilne embedinge koji se koriste isključivo za prepoznavanje. Originalne slike se niti ne pohranjuju niti su dostupne izvan sustava, čime se eliminira rizik od zloupotrebe ili vanjskog curenja. Nadalje, budući da sustav radi u potpunosti na lokalnoj infrastrukturi bez slanja biometrijskih podataka izvana, on je u skladu sa suvremenim principima zaštite podataka. Ove zaštite čine predloženo rješenje prikladnim za aplikacije osjetljive na privatnost, poput obrazovanja i kontrole pristupa, gdje su učinkovitost i etički standardi ključni.

Zaključno, rezultati pokazuju da sustav nema inherentne probleme s prepoznavanjem lica, ali je njegova točnost najviše pogođena vanjskim čimbenicima koji mogu degradirati kvalitetu embeddinga.

Za razliku od tradicionalnih metoda [63] koje se oslanjaju na duboke konvolucijske mreže [64, 65, 66] specijalizirane za lica, naš sustav koristi multimodalne embeddinge, nudeći veću fleksibilnost i širu primjenjivost u različitim scenarijima prepoznavanja. Nadalje, integracija FAISS indeksiranja omogućuje izuzetno brza pretraživanja, značajno poboljšavajući učinkovitost i responzivnost modela u stvarnom vremenu. Tako, ovaj rad doprinosi području računalnog vida predstavljajući novu metodologiju koja kombinira snagu transformer embeddinga i optimizirano pretraživanje vektora, otvarajući mogućnosti za buduća istraživanja i unapređenja u brzom i skalabilnom prepoznavanju lica.

References

- [1] Wenyi Zhao, Rama Chellappa, P. Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458, 2003.
- [2] Sharon Manmothe and Jyoti Jadhav. Integrating multimodal data for enhanced analysis and understanding: Techniques for sentiment analysis and cross-modal retrieval. *Journal of Advanced Zoology*, 45:22–28, 03 2024.
- [3] Yikun Han, Chunjiang Liu, and Pengfei Wang. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *arXiv preprint arXiv:2310.11703*, 2023.
- [4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Accessed 2025-05-20.
- [5] Google Machine Learning Crash Course. Classification: Accuracy, recall, precision, and related metrics.
- [6] Scikit-learn developers. Scikit-learn: Metrics and scoring: quantifying the quality of predictions.
- [7] Lixiang Li, Xiaohui Mu, Siying Li, and Haipeng Peng. A review of face recognition technology. *IEEE Access*, 8:139110–139120, 2020.
- [8] Cong Wang, Yulong Wang, Jin Gao, and Tongliang Liu. A survey on deep learning for face recognition. *Journal of Artificial Intelligence Research*, 72:215–283, 2021. Accessed: 2025-04-19.
- [9] Songze Zhu. Enhancing facial recognition: A comprehensive review of deep learning approaches and future perspectives. *Applied and Computational Engineering*, 110:137–145, 11 2024. Accessed: 2025-04-19.
- [10] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*, 2015. Accessed: 2025-04-19.
- [11] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deep-face: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.
- [12] Thinking Neuron. Face recognition using deep learning (cnn) in python. <https://thinkingneuron.com/face-recognition-using-deep-learning-cnn-in-python/>, 2020. Accessed: 2025-04-25.
- [13] Divyarajsinh N. Parmar and Brijesh B. Mehta. Face recognition methods & applications, 2014. Accessed: 2025-04-19.

- [14] Facebook AI Research. Faiss: A library for efficient similarity search and clustering of dense vectors. <https://faiss.ai/>, 2025. Accessed: 2025-04-16.
- [15] Asna Shafiq. How to use faiss to build your first similarity search. <https://medium.com/loopio-tech/how-to-use-faiss-to-build-your-first-similarity-search-bf0f708aa772/>, 2022. Accessed: 2025-04-17.
- [16] Nhan T. Luu. Clip unreasonable potential in single-shot face recognition. *arXiv preprint arXiv:2411.12319*, 2024. Accessed: 2025-04-19.
- [17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. Accessed: 2025-04-19.
- [18] Aaditya Bhat and Shrey Jain. Face recognition in the age of clip & billion image datasets. *arXiv preprint arXiv:2301.07315*, 2023. Accessed: 2025-04-19.
- [19] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [20] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (3rd ed. draft)*. Prentice Hall, 2023. Available at <https://web.stanford.edu/~jurafsky/slp3/>.
- [21] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. In *IEEE Transactions on Big Data*, 2017.
- [22] Ganta Sai Reddy, Bandaru Sri Hari Kumar, Kondeti Ravi Kumar, Hari Krishna M, and G. A. Rama Krishna. Predictive maintenance for industrial machinery using llm. *International Research Journal on Advanced Engineering Hub (IRJAEH)*, 2024.
- [23] Eamonn Keogh and Shvetha Kasetty. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(1):55–71, 2001.
- [24] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.
- [25] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418, 2009.

- [26] Andrea Frome, Gregory S Corrado, Jon Shlens, Samy Bengio, Jeffrey Dean, Marc’Aurelio Ranzato, and Tomáš Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.
- [27] Aditya Ramesh, Prafulla Dhariwal, Yan Leike, Mark Chen, Krishna Murthy, Alec Radford, Ilya Sutskever, and Wojciech Zaremba. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2021.
- [29] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.
- [30] DataCamp Community. What is faiss (facebook ai similarity search)? <https://www.datacamp.com/blog/faiss-facebook-ai-similarity-search>, mar 2024. Accessed on 22 May 2025.
- [31] Jimin Tan, Jianan Yang, Sai Wu, Gang Chen, and Jake (Junbo) Zhao. A critical look at the current train/test split in machine learning. *arXiv preprint arXiv:2106.04525*, 2021.
- [32] Jason Brownlee. Distance measures for machine learning. <https://machinelearningmastery.com/distance-measures-for-machine-learning/>, 2019. Accessed: 2025-04-24.
- [33] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006.
- [34] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
- [35] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.
- [36] Yuval Malkov and Dmitry Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–834, 2020.

- [37] Rohit Kundu. Precision vs. recall: Differences, use cases & evaluation, 2022. Accessed: 2025-04-24.
- [38] Max Berman, Amal Rannen Triki, and Matthew B Blaschko. Multi-class classification evaluation with precision-based metrics. In *International Conference on Machine Learning (ICML)*, pages 759–768, 2020.
- [39] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011. Archived from the original on 2019-11-14, Accessed: 2025-04-24.
- [40] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009. Accessed 2025-05-20.
- [41] Lior Wolf, Tal Hassner, and Itay Maoz. Youtube faces database. <https://www.cs.tau.ac.il/~wolf/ytfaces/>, 2011. Accessed: 2025-04-24.
- [42] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [43] GeeksforGeeks. Precision-recall curve in machine learning. <https://www.geeksforgeeks.org/precision-recall-curve-ml/>. Accessed: 2025-04-22.
- [44] Erik Scepanski and Sonja Zillner. Ai systems and their scalability – a systematic literature review. In *Proceedings of the 35th Australasian Conference on Information Systems (ACIS 2024)*, 2024.
- [45] NumPy Developers. numpy.linalg.norm — numpy v1.26 manual, 2024. Accessed: 2025-04-24.
- [46] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*, 2022.
- [47] Mohammad Hasan and Abhishek Mishra. Design and implementation of deep learning based contactless authentication system using hand gestures. *Sensors*, 10(2):182, 2021.
- [48] CIE-Group. Access control contactless authentication methods. <https://cie-group.com/how-to-av/videos-and-blogs/contactless-authentication>, Accessed on May 21, 2025.

- [49] Jake Lever, Martin Krzywinski, and Naomi Altman. Principal component analysis. *Nature Methods*, 14(7):641–642, 2017. Accessed: 2025-04-19.
- [50] Ian T. Jolliffe. Principal component analysis. *Springer Series in Statistics*, 2002.
- [51] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, 2006.
- [52] Xiaoqin Lin, Yufei Zhang, and Xu Han. Evaluation metrics for deep learning-based image recognition: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):456–474, 2023.
- [53] CSIS. How accurate are facial recognition systems – and why does it matter? *Strategic Technologies Blog, Center for Strategic and International Studies*, 2020.
- [54] Md. . Face recognition and identification using deep learning. *ResearchGate*, 2024.
- [55] GeeksforGeeks. Understanding the confusion matrix in machine learning. <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>. Accessed: 2025-04-22.
- [56] Zhe Hui Hoo, Jane Candlish, and Dawn Teare. What is an roc curve? *Emergency Medicine Journal*, 34(6):357–359, 2017. Accessed: 2025-04-24.
- [57] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [58] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.
- [59] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.
- [60] David Chu, Nicholas D. Lane, Ted Tsung-Te Lai, Cong Pang, Xiangying Meng, Qing Guo, Fan Li, and Feng Zhao. Balancing energy, latency and accuracy for mobile sensor data classification. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, SenSys ’11, page 54–67, New York, NY, USA, 2011. Association for Computing Machinery.
- [61] Eglė Kavoliūnaitė-Ragauskienė. Right to privacy and data protection concerns raised by the development and usage of face recognition technologies in the european union. *Journal of Human Rights Practice*, 16(2):658–674, 2024.

- [62] Evan Selinger and Brenda Leong. The ethics of facial recognition technology. *SSRN Electronic Journal*, 2018.
- [63] Ena Barcic, Petra Grd, and Igor Tomicic. Convolutional neural networks for face recognition: A systematic literature review, 07 2023.
- [64] Kunihiro Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.
- [65] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [66] Yann LeCun, Bernard Boser, John S. Denker, Dong Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.