

Dokumentacija projekta:

Find your CelebA Doppelganger

Sastavili i razvili:

Fran Glušac

Gabriel Janđel

Sven Matić

Sven Matković

Prepoznavanje lica koristeći strojno učenje

Pregled

Ovaj projekt primjenjuje strojno učenje kako bi analizirao slike lica, prepoznao ključne obilježja i izveo zaključke o nekoliko aspekata prikazane osobe. Koristeći konvolucijske neuronske mreže, sustav je sposoban otkriti lice na slici, klasificirati dob, spol i rasu osobe, procijeniti neprimjerenost slike i identificirati ključne točke (landmarks) na licu. Sve to postiže kroz proces obrade i analize slika iz skupa podataka.

Cilj projekta

Ovaj projekt je izrađen s ciljem istraživanja i implementacije naprednih tehnika strojnog učenja, posebice konvolucijskih neuronskih mreža, u domeni prepoznavanja lica. Suočeni s izazovima u analizi slika lica, razvili smo ovaj sustav kako bismo automatizirali i poboljšali proces prepoznavanja ključnih obilježja lica.

Naš sustav rješava problem identifikacije različitih aspekata lica - kao što su dob, spol, rasa i ključne točke lica - na osnovu slika. Također, sustav je sposoban procijeniti neprimjerenost slike, što može biti korisno u kontekstu moderiranja sadržaja.

U širem kontekstu, ovaj projekt se uklapa u rastuće područje tehnologije prepoznavanja lica i strojnog učenja. Takva tehnologija prepoznavanja lica postaje sve važnija u mnogim sektorima, uključujući sigurnost, zabava, marketing... Istovremeno, strojno učenje nastavlja napredovati i omogućiti nove aplikacije. Kroz ovaj projekt, nastojimo doprinijeti ovim razvojem i demonstrirati moćne mogućnosti koje strojno učenje može pružiti u obradi slika i analizi lica.

Struktura projekta

Projekt je organiziran u više dijelova koji se bave različitim aspektima procesa:

Moduli za formatiranje i procesiranje podataka

[ReFormater.py](#) i [more_reformatting.py](#) - preuzimaju originalne koordinate ključnih točaka lica, prilagođavaju ih prema promijenjenim dimenzijama slike i spremaju reformirane koordinate za daljnje korake.

Moduli za učitavanje i procesiranje slika

[general_loader.py](#) i [dataset_processor.py](#) - koriste prethodno istrenirane modele za izvlačenje raznih informacija sa slike, kao što su spol, dob, rasa i neprimjerenost slike. Također identificiraju lice na slici i stvaraju bounding box oko njega.

Moduli za treniranje modela

[trainers](#) - ovaj direktorij sadrži skripte za treniranje različitih modela za prepoznavanje lica, određivanje rase, spola, ključnih točaka lica, itd

Moduli za skladištenje modela

[trained_models](#) - ovaj direktorij sadrži prethodno istrenirane modele koji se koriste za prepoznavanje različitih obilježja lica.

Alati i pomoćne funkcije

[tools](#) - ovaj direktorij sadrži različite alate koji se koriste kroz cijeli projekt, uključujući klase za kreiranje dataseta, metode za računanje točnosti i metode za treniranje i testiranje modela.

Definicije modela

[testing](#) – ova mapa sadrži definicije različitih arhitektura konvolucijskih neuronskih mreža koje se koriste u projektu.

Kontrola kvalitete klasifikacije

[quality_control](#) - ovaj direktorij sadrži kod za interaktivnu kontrolu kvalitete klasifikacija.

Evaluacija modela

[plotting](#) - ovaj direktorij sadrži alate za grafičku evaluaciju i optimizaciju modela.

Podaci za anotaciju

[annotation_datasets](#) - ovaj direktorij sadrži algoritme za pronalazak najsličnijih lica temeljenih na obilježjima lica, rasi, dobi i spolu, te podatke za anotaciju.

Moduli za baze podataka i formatiranje

[feature_matching_preformatting.py](#), [feature_matching_algorithm.py](#), [Modular_DB_creator.py](#), i [modular_DB_opener](#). Ovi moduli su odgovorni za stvaranje baze podataka, formatiranje podataka i algoritme za uparivanje značajki.

Tehničke specifikacije

Pregled funkcija te sučelja svih javnih funkcija projekta

****Ako neka funkcija ili datoteka nije navedena, nije namijenjena za samostalnu uporabu!**

[zeleno](#) – funkcije

[crveno](#) – imena modela

[plavo](#) - klase

ReFormater.py

- Formatiranje obilježja lica
- Procesiranje koordinata obilježja lica tako da su točne na slici s promijenjenim dimenzijama
- Upisivanje reformiranih koordinata u novi file
- [format_landmarks](#) – uzima DataFrame i sprema njegove podatke u rječnik tako da ključ bude ID slike, a vrijednost koordinate obilježja lica
- [process_landmarks](#) – uzima nove dimenzije i koordinate obilježja lica te ih prilagođava novim dimenzijama

more_reformatting.py

- Stvaranje standardiziranog formata za spremanje koordinata obilježja lica za svaku sliku

general_loader.py

- Vraća procesiranu sliku, novi bbox, dob osobe, neprimjerenost slike, spol osobe, rasu osobe
- `process_image` – uzima put do slike, ime slike i model te primjenjuje model na slici. Vraća rezultat i sliku.
- `process_image_bbox` – uzima put, ime file-a i model, pomoću modela određuje bbox, potom izreže sliku i vraća nazad bbox-ove promijenjenih dimenzija te izrezanu sliku.
- `process` – put i ime slike, iskorištava gore definirane funkcije i modele dob, rasu i spol. Vraća izrezanu sliku, bbox promijenjenih dimenzija, obilježja po novim koordinatama, primjerenost slike, dob, spol i rasu osobe.

dataset_processor.py

- Procesiranje datasea slika i vađenje informacija o neprimjerenosti i rasi koristeći modele
- Spremanje navedenih informacija u CSV file koji se kasnije koristi za kontrolu kvalitete
- `process` – uzima ime slike i ID slike. Vraća ID slike, ime slike, neprimjerenost i rasu u obliku rječnika. Koristi funkcije `process_image` i `process_image_bbox` iz `general_loader-a`

trainers

- Sadrži file-ove koji služe za treniranje različitih CNN modela
- `bbox_trainer.py` – program koji služi za treniranje modela za prepoznavanje lica na slici te crtanje okvira oko lica osobe. Koristi **CNN** arhitekturu
- `custom_attributes_trainer.py` – program koji služi za treniranje modela za prepoznavanje rase osobe. Koristi **MircoCNN** arhitekturu.
- `existing_attributes_trainer.py` – program koji služi za treniranje modela za prepoznavanje spola osobe. Koristi **MicroCNN** arhitekturu.
- `landmark_trainer.py` – program koji služi za treniranje modela za prepoznavanje položaja očiju, nosa i rubova usta. Koristi **CNN_small** arhitekturu.

trained_models

- Sadrži .pt file-ove unutar kojih su spremljeni istrenirani modeli
- `age_model.pt` – procjenjuje je li osoba stara ili ne
- `bbox_model.pt` – vraća koordinate okvira na slici unutar kojeg se nalazi lice
- `gender_model.pt` – vraća spol osobe na slici
- `inapprop_model.pt` – procjenjuje je li slika neprimjerena ili ne
- `landmark_model.pt` – vraća položaj očiju, vrh nosa te rubove očiju

- `race_model_bbox.pt` – vraća kakve je osoba puti (crne ili svijetle)

tools

- Sadrži file-ove s funkcijama koje se koriste na više mjesta u projektu
- Također sadrži klase za lakše kreiranje dataseta
- `datasets.py`
 - `CelebADatasetParent` – roditeljska klasa `CelebADataset` klasa. U konstruktoru preuzima folder slike, file slike, podatke o transformaciji slike, podatke o transformaciji oznaka (label-a) slike te podatke o tome kako pročitati informacije iz CSV file-a
 - `SubsetFactory` – u konstruktoru uzima veličinu podatka za treniranje, validaciju i testiranje. Na poziv dijeli dataset na dijelove danih dimenzija
- `training_utilities.py`
 - `acc_calc` – uzima izlaz modela i vraća omjer točnih odgovora i ukupnog broja odgovora, odnosno točnost
 - `train` – uzima model, podatke za treniranje, funkciju za računanje pogreške modela, funkciju za optimizaciju i mjesto na kojem se model trenira (CPU/GPU). Vraća prosječan gubitak i točnost
 - `test` – uzima model, loader za test podatke, funkciju za računanje pogreške i mjesto na kojem se model trenira. Vraća prosječan gubitak i točnost. Koristi se prilikom treniranja modela

testing

- Sadrži file `cnn_class.py` koji unutar sebe ima različite arhitekture CNN-ova koje smo koristili u projektu
- `cnn_class.py` –
 - `CNN` – model najveće arhitekture, sadrži pet blokova, koristi se za određivanje okvira lica
 - `CNN_small` – model srednje veličine, sadrži dva bloka, koristi se za određivanje položaja očiju, nosa i rubova usta
 - `MicroCNN` – najmanji model, koristi se za određivanje rase i spola osobe

RNN

****Ovo nije dio CelebA CNN-a, već je dodatan samostalni sadržaj koji nema veze s projektom!**

- Sadrži kod potreban za kreiranje pjesama pomoću RNN modela
- Uključuje dva RNN modela
- Sadrži server na kojemu se vrši generacija tekstova

quality_control

`rapid_classifier.py`

- Stvara GUI koji na sebi ima dvije slike, korisnik određuje je li osoba na slici tamnoputa ili nije
- Sakupljene informacije se koriste za treniranje modela

plotting

- Grafička evaluacija istreniranih modela

- Pomoću analize grafičkih podataka lakše optimiziramo model
- `Metric_evaluator.py`
 - Za različite etape treniranje nekog modela vraća prosječnu i apsolutnu točnost i sprema ih u .json file-ove

anotation_datasets

- CSV file-ovi različitih podataka
- `feature_matching_preformatting.py` -> algoritam za traženje najbližnjeg lica s obzirom na obilježja lica, rase, dobi i spola osoba na slikama
- `dataset_fixing_uilities.py`
 - funkcije za micanje sufiksa ispred ID-ova slika, zamjenu separatora u CSV file-ova i miješanje datasetova

Ukupna funkcionalnost

Projekt koristi konvolucijske neuronske mreže (CNN) da nauči kako prepoznati različite aspekte lica na slikama. Skup podataka CelebA, koji sadrži slike slavnih osoba, koristi se za treniranje modela. Slike su obrađene i transformirane u tenzore, a oznake (labels) su obrađene na odgovarajući način za svaki zadatak.

Podaci su zatim podijeljeni na skup za treniranje i testiranje. Ovisno o skripti za treniranje, koristi se odgovarajući model za instanciranje i treniranje na odgovarajućim podacima. Modeli se periodički spremaju tijekom treniranja, a na kraju svake epohe ispisuju se gubitak (loss) i točnost (accuracy) za treniranje..

Kako koristiti

Za korištenje ovog projekta, trebate imati pristup CelebA skupu podataka i odgovarajućim oznakama za svaki zadatak. Za svaki zadatak trebate pokrenuti odgovarajuću skriptu za treniranje iz trainers direktorija.

Zahtjevi

Pisano I testirano za: Python 3.9,3.11

MINIMALNA verzija pythona: 3.9

Potrebni moduli (samo testiranje): sqlite3, numpy, torch, torchvision, pillow, kivy, kivymd

Dodatni moduli (treniranje te alati): pandas, sklearn, imblearn, scipy, matplotlib, CelebA Dataset

Preporučeno je korištenje GPU-a pri treniranju te CPU-a pri testiranju i korištenju.

Za kompatibilnost te osposobljavanje GPU podrške pogledati web stranicu [pytorch-a](https://pytorch.org)

Zaključak

Ovaj projekt koristi metode dubokog učenja kako bi izveo kompleksne analize na slikama lica. Iako se koristi za određene zadatke, arhitektura i metode koje se koriste mogu se lako prilagoditi za druge slične zadatke u domeni prepoznavanja slika i analize slika.

Za sva pitanja te nedoumice, greške kontaktirajte nas na teamsu!

(korisnička imena: Gabriel Jandžel ili Fran Glušac)

V. Gimnazija, Zagreb, MMXXIII.