Mundarija

I-bo’lim Suniy Intelekt dasturi talablar

Yuzni aniqlovchi dasturning talablarini ishlab chiqish……………………………2

Suniy intellekt (SI) dasturi yaratish uchun bosqichlar…………………………….3

Tizim imkoniyatlari……………………………………………………………….4

Tizim strukturasi....………………………………………………………………..4

Tizim modular bilan bog’lanishi………………………………………………….5

Tizim versiyalari.....……………………………………………………………….5

Version 1.0.0 versiyasi haqida….…………………………………………………5

Version 2.0.0 versiyasi….……………………………………………………….11Yuzni aniqlovchi dasturning talablarini ishlab chiqish

Yuzni aniqlovchi suniy intellekt (AI) dasturlariga talablar, yuzni tanib chiqish va tasdiqlash imkonini beruvchi, yuz qismi aniqlash va tarkibini tushunishga qodir bo'lgan algoritmalarni o'z ichiga oladi. Quyidagi talablar yuzni aniqlovchi AI dasturlari uchun muhim hisoblanadi:

* Yuz ma'lumotlarini to'plash: Yuzni aniqlovchi dasturlar yuz ma'lumotlariga ihtisoslashgan datasetlardan foydalanish uchun yuz suratlarining to'plab qilinishini talab qiladi. Bu datasetlarda turli shaxslar, turli kundalik holatlarda yuzlar, turli kamera o'rtasida kutilgan qisqa ma'lumotlar mavjud bo'ladi.
* Algoritmalarni tashkil etish: Yuzni aniqlovchi dasturlar uchun algoritmalarni tuzish kerak. Bu algoritmalarga yuzning belgilash xususiyatlari, yuz qismi tushunish, yuzni aniqlash uchun sinov va tarkibini tasdiqlash kabi vazifalar tayinlanishi kerak.
* Yuzni belgilash:Yuzni belgilash uchun yuz mavjudligini tekshirish , agar topilsa yuzni belgilash uchun yuzning chegaralarini aniqlash talab qilinadi.
* Xususiy parametrlarni aniqlash: Yuzni aniqlovchi dasturlar yuzning xususiy parametrlarini, masalan : yuz ifodasi (ho’mraygan , jilmaygan), yuz rangi (jigarrang, oq ,qora), yuz nigohi(o’ngga ,chapga,to’g’riga qaragan), katta yoki kichikligi kabi xususiyatlarni aniqlashga qodir bo'lishi kerak.
* Hatolarni identifikatsiya qilish: Yuzni aniqlovchi dasturlar hatolarni aniqlash va tekshirishga yordam beradigan mehanizmalarga ega bo'lishi talab qilinadi. Yuzni aniqlashda xatolar , tushunmovchilik moammolar yuzaga kelganda ularni qayd qilib borish.

Suniy intellekt (SI) dasturi yaratish uchun quyidagi bosqichlar mavjud:

* Maqsad va talablar: Aniqlovchi yuzning yaratilishi bo'yicha maqsad va talablar belgilanishi kerak. Misol uchun, yuzni aniqlash, tanib olish, yoshi aniqlash yoki emotsiyalarni tushunishga yo'l qo'yish bo'lishi mumkin.
* Ma'lumotlarni to'plash: Suniy intellekt dasturi yaratishda ma'lumotlarni to'plash shart. Ushbu ma'lumotlar odatda malumotlar bazasi yoki axborotlarni to'playdigan sensorlar orqali olingan ma'lumotlar bo'lishi mumkin.Yuz ifodalari haqida malumotlar (yuzning parametrlari).
* Ma'lumotlarni tahlil qilish: Tahlil qilish bosqichida ma'lumotlar tahlil qilinadi va dastur uchun kerakli formatga o'tkaziladi. Bu tahlil jarayoni dasturning talablari va maqsadlari asosida belgilanadi.Masalan dastur yuzni aniqlashi kerak , tana, qo’l,oyoq yoki boshqa obyektlarni aniqlashi kerakmi ? .
* Modelni yaratish: Suniy intellekt dasturi yaratish uchun, dastur uchun model yaratish kerak.
* Dastur o'qish va o'rganish: Model yaratilgandan so'ng, dastur ma'lumotlarni o'qish va o'rganish bo'yicha o'rnatiladi. Ushbu bosqichda modelni o'qitish uchun ma'lumotlar to'plami bilan dastur o'rgatiladi.
* Dastur sinovdan o'tkazish: Suniy intellekt dasturini yaratishdan so'ng, uni sinovdan o'tkazish kerak. Sinovdan o'tkazish orqali dasturning aniqlovchi yuzni aniqlovchi ishonchdarligi, aniqlash darajasi va xatoliklar soni haqida ma'lumot olish mumkin.
* Dasturni optimallashtirish: Sinovdan so'ng, dastur o'zgartirishlar va optimallashtirishlar bilan yangilanadi. Bu maqsadga erishish uchun algoritmalarni takomillashtirish, modellarni to'g'ri qilish va natijalarni boshqarish mumkin.
* Ishga tushirish: Dastur to'g'rilangan va optimallashtirilganidan so'ng, uni ishga tushirish mumkin. Bu dasturda o'zgarishlarni kuzatish, natijalar uchun kelajak proyektsiyalarni taqdim etish va kerakli muammolarni hal qilishdan iborat bo'ladi.

Bu asosiy bosqichlar

1. Ma'lumotlar yig'ish: AIni o'rgatishda foydalanish uchun ijobiy va salbiy misollar bo'lgan katta hajmdagi yuz tasviri ma'lumotlarini to'plash.

2. Oldindan ishlov berish: yuz xususiyatlarini ajratib olish va tasvirlarni tekislash uchun maʼlumotlarni oldindan qayta ishlash.

3. Xususiyatlar ajratib olish: Yuzdagi aniq nuqtalarni aniqlash kabi yuz tasvirlaridan xususiyatlarni ajratib olish uchun algoritmlarni loyihalash.

4. Oʻrgatish: AIni olingan funksiyalar va teglar boʻyicha oʻrgatish uchun mashinani oʻrganish algoritmidan foydalanish.

5. Sinov va tekshirish: uning aniqligini tekshirish uchun tashqi tasvir maʼlumotlar toʻplamidan foydalangan holda AI unumdorligini baholash.

6. Qayta sozlash: Sinov natijasida olingan natijalar asosida AI ish faoliyatini y axshilash.

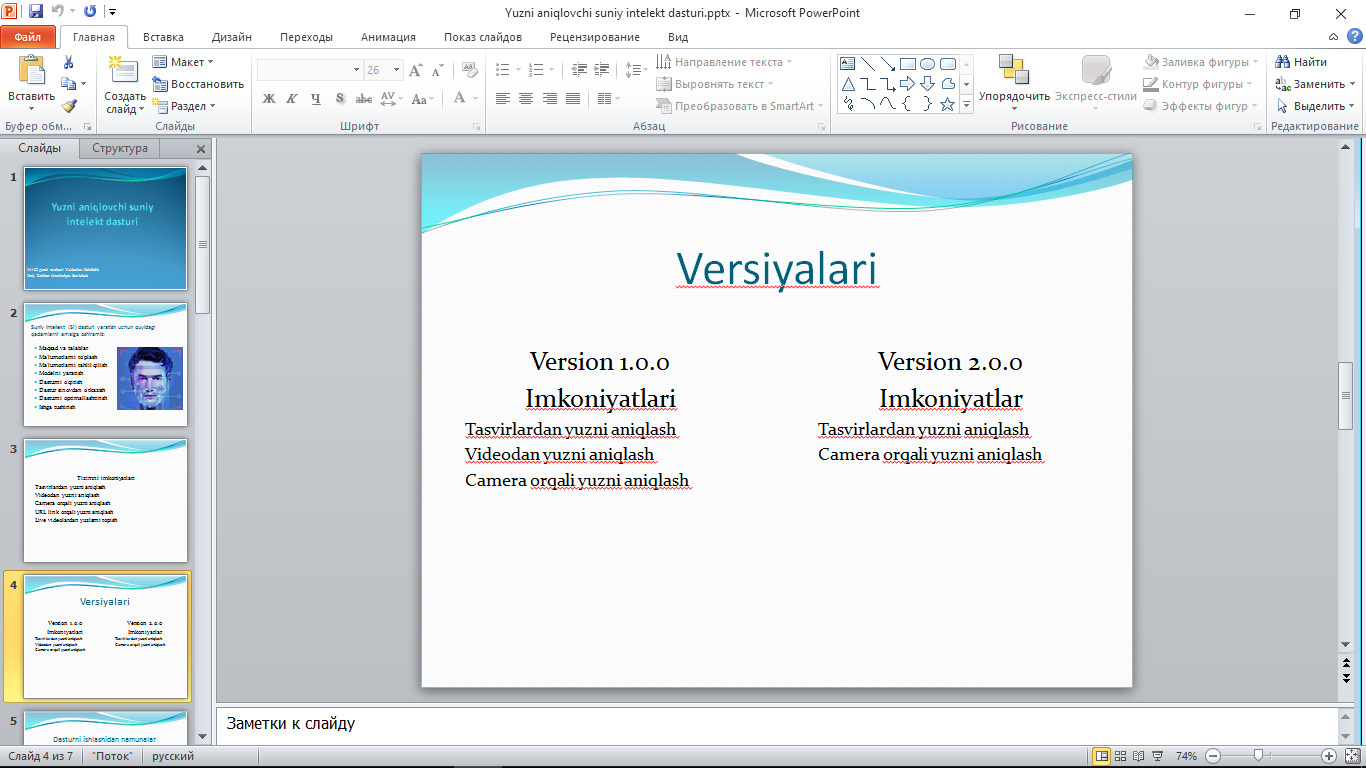
7. O'rnatish: AI tizimini real dunyo ilovalarida o'rnatish, uning ishlashi va qulayligini optimallashtirish.

Tizimni imkoniyatlari

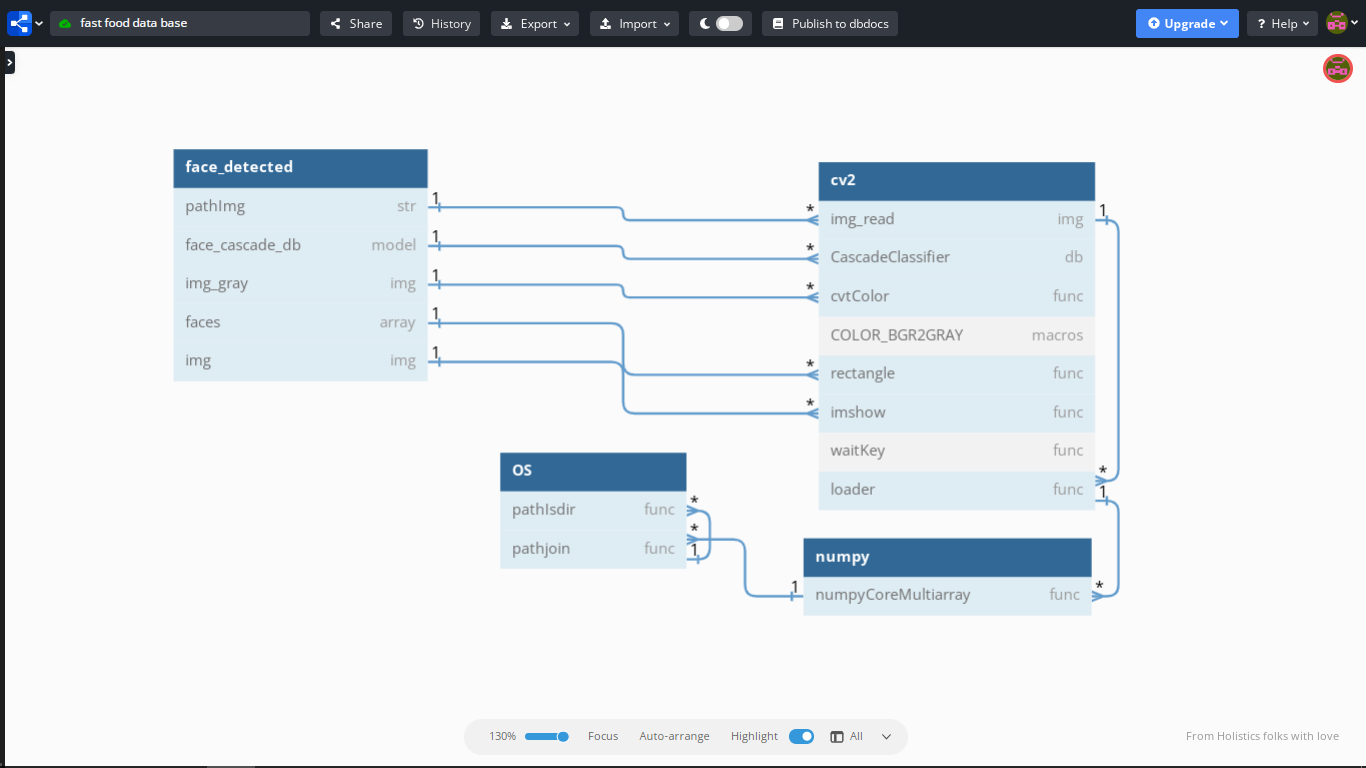
* Tasvirlardan yuzni aniqlash
* Videodan yuzni aniqlash
* Camera orqali yuzni aniqlash
* URL link orqali yuzni aniqlash
* Live videolardan yuzlarni topish



1-rasm. face\_detected moduli strukturasi

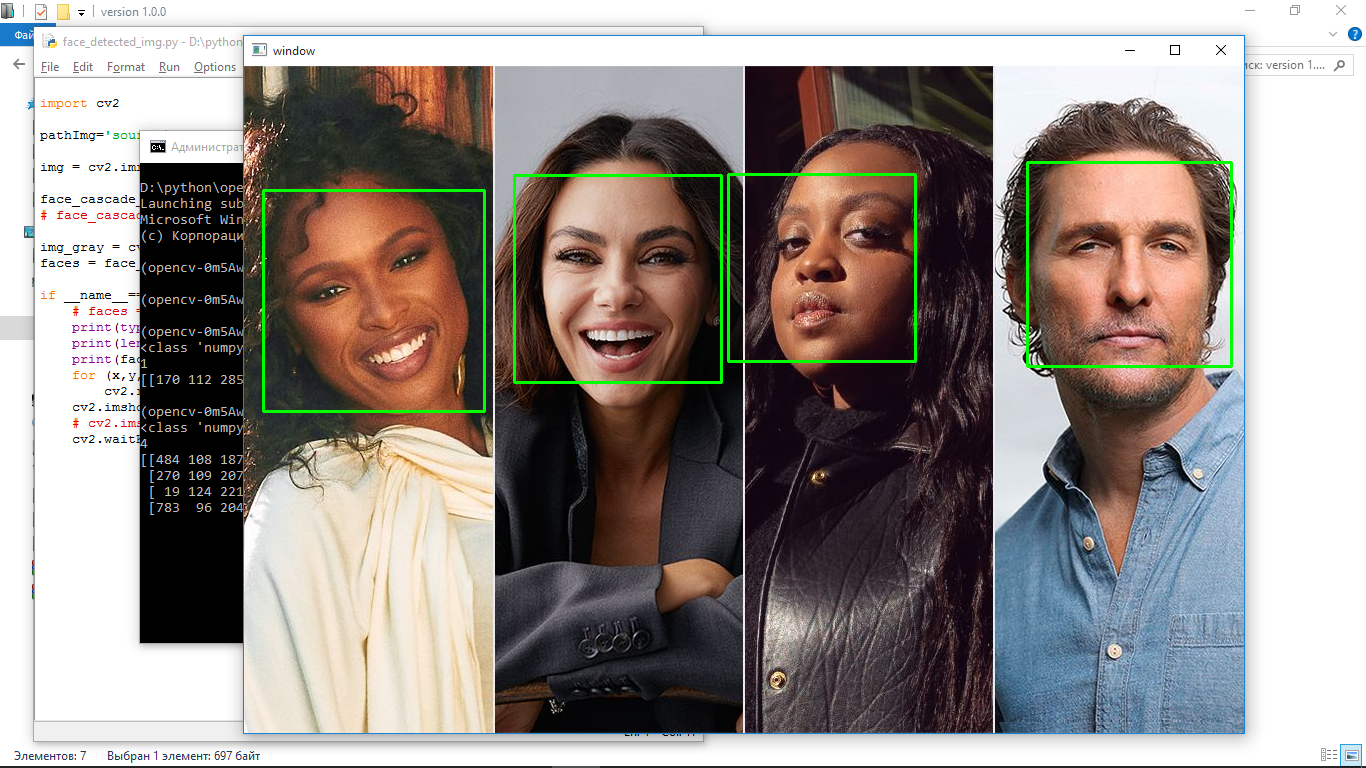


2-rasm.Tizim versiyalari va imkoniyatlari



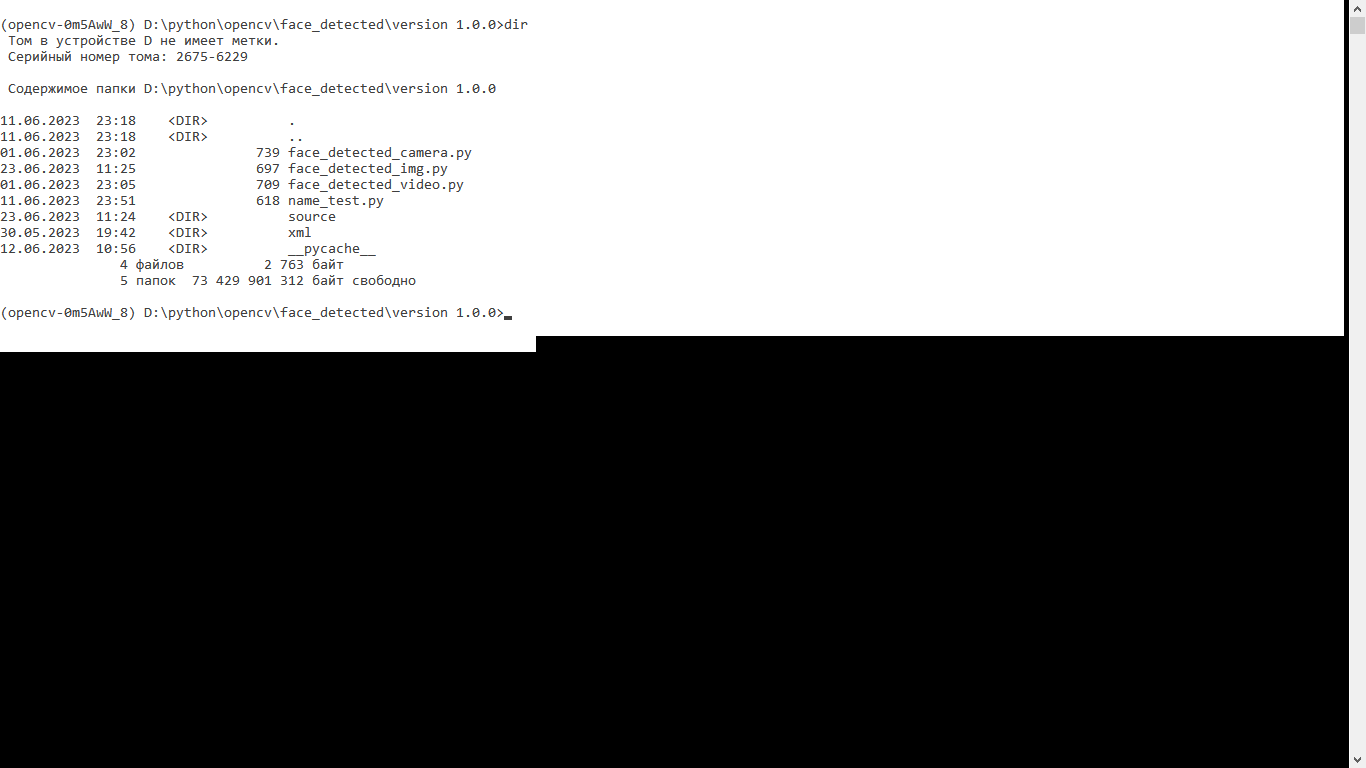
3-rasm. version 1.0.0 modullar orasidagi bog’liqlik

Ushbu rasmda birinchi versiya modullar orasidagi bog’liqliklar , bog’lanishlar qisqacha ko’rsatib o’tilgan.



4-rasm.version 1.0.0 ishlash jarayoni

Version 1.0.0 ya’ni birinchi versiyamiz tasvirlardan shu jumladan video fayllardan , kamera tasvirlaridan yuzlarni aniqlaydi. Ushbu versiya faqat tasvirdan yuzlarni aniqlaydi , bu versiyani takomillashtirgan holda tasvirdagi shaxsni kimligini aniqlash masalasi turadi , ushbu masala version 2.0.0 yani ikkinchi versiyamizda mavjud.



5-rasm. version 1.0.0 fayllar strukturasi

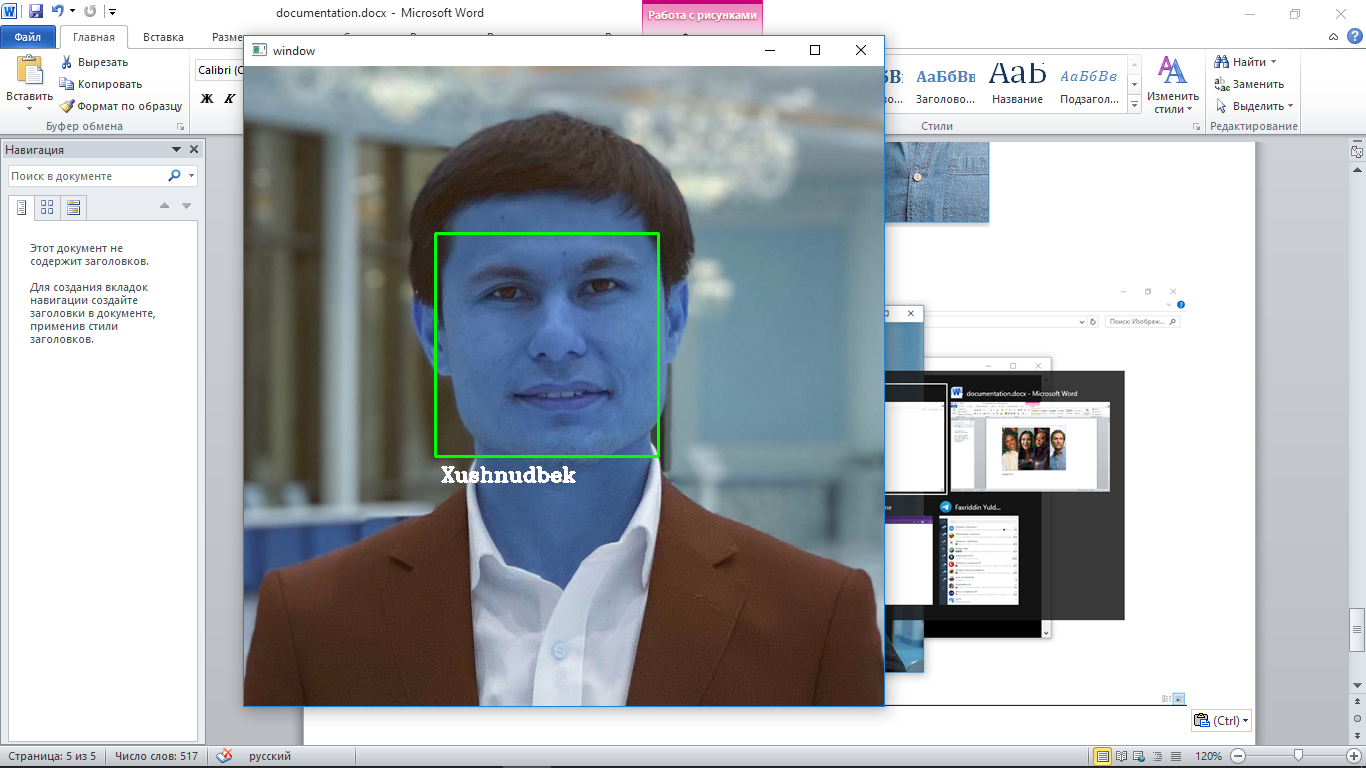
face\_detected\_camera.py – bu dastur hisoblanib ,ushbu dastur python dasturlash tilida yozilgan. Ushbu modul vazifasi qurilma kamerasi yordamida tasvirlardan inson yuzini aniqlashdir.

face\_detected\_img.py – bu dastur hisoblanib ,ushbu dastur python dasturlash tilida yozilgan. Ushbu modul vazifasi qurilmaga kiritilgan tasvirlardan inson yuzini aniqlashdir.

face\_detected\_video.py – bu dastur hisoblanib ,ushbu dastur python dasturlash tilida yozilgan. Ushbu modul vazifasi qurilmaga kiritilgan video fayllardan inson yuzini aniqlashdir.

name\_test.py - ushbu dastur tizimni test qilish dasturi hisoblanib , eng avvalo ushbu dastur ishga tushishi kerak. Bu dasturni ishga tushirish orqali bbizni modulimiz uchun kerakli modullar o’rnatilganmi , muhit sozlanganmi(virtualenv), ushbu ishlarni dastur avtomatik aniqlaydi, xatolikni aniqlaydi(xatolik – deganda modullar o’rnatilganligi nazarda tutilmoqda) ular aniqlangan xolda modullar qayta tekshirilishi kerak.

source – katalogida loyihani video va tasvirli fayllari saqlanadi. Dastur ishlashi davomida ushbu katalogdagi video va tasvirli fayllardan foydalanadi.



6-rasm.version 2.0.0 ishlash jarayoni

version 2.0.0 ikkinchi versiyamiz ishlash jarayoni yuqorida ko’rasatilgan.Bu versiyani avzalliklari inson yuznini topadi va ushbu yuz egasini tanib olish imkoniyatiga ega , buning uchun ma’lumotlar bazasiga ushubu shaxs haqida ma’lumotlar joylangan bo’lishi kerak.



7-rasm.version 2.0.0 katalogi strukturasi

persons – katalogida shaxs ma’lumotlari joylanadi, ya’ni bu bizga ma’lumotlar bazasi sifatida hizmat qiladi,Ushbu katalogga rasm nomi bilan yozilishi yetarli (Namuna: "Steeve Jobs.png").

source – katalogida esa tekshirilishi kerak bo’lgan tasvirlar joylanadi, bu katalogdagi ma’lumotlarni person katalogidagi ma’lumotlar asosida tekshiradi.

Version 1.0.0 dasturiy maxsulotimiz kodlari

face\_detected\_img.py fayli

import cv2

pathImg='source/8.png'

img = cv2.imread(pathImg)

face\_cascade\_db = cv2.CascadeClassifier('xml/haarcascade\_frontalface\_default.xml')

# face\_cascade\_db = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

img\_gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_cascade\_db.detectMultiScale(img,1.1,5)

if \_\_name\_\_=='\_\_main\_\_':

# faces = face\_cascade\_db.detectMultiScale(img\_gray,1.1,10)

print(type(faces))

print(len(faces))

print(faces)

for (x,y,w,h) in faces:

cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

cv2.imshow('window',img)

# cv2.imshow('window2',img\_gray)

cv2.waitKey()

face\_detected\_video.py fayli

# video fayldagi yuzlarni aniqlash

import cv2

cap = cv2.VideoCapture('source/test.mp4')

cap.set(4,480)

face\_cascade\_db = cv2.CascadeClassifier('xml/haarcascade\_frontalface\_default.xml')

while True:

succes ,img = cap.read()

# img\_gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_cascade\_db.detectMultiScale(img,1.1,10)

print(type(faces))

print(len(faces))

print(type(img))

print(len(img))

for (x,y,w,h) in faces:

cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

cv2.imshow('window',img)

# cv2.imshow('window2',img\_gray)

if cv2.waitKey(1) & 0xff == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

face\_detected\_camera.py fayli

# video fayldagi yuzlarni aniqlash

import cv2

cap = cv2.VideoCapture(0)

cap.set(3,1080)

cap.set(4,720)

face\_cascade\_db = cv2.CascadeClassifier('xml/haarcascade\_frontalface\_default.xml')

while True:

succes ,img = cap.read()

img = cv2.flip(img,2)

# img\_gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

faces = face\_cascade\_db.detectMultiScale(img,1.1,10)

# print(type(faces))

# print(len(faces))

print(type(img))

print(len(img))

for (x,y,w,h) in faces:

cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

cv2.imshow('window',img)

# cv2.imshow('window2',img\_gray)

if cv2.waitKey(1) & 0xff == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

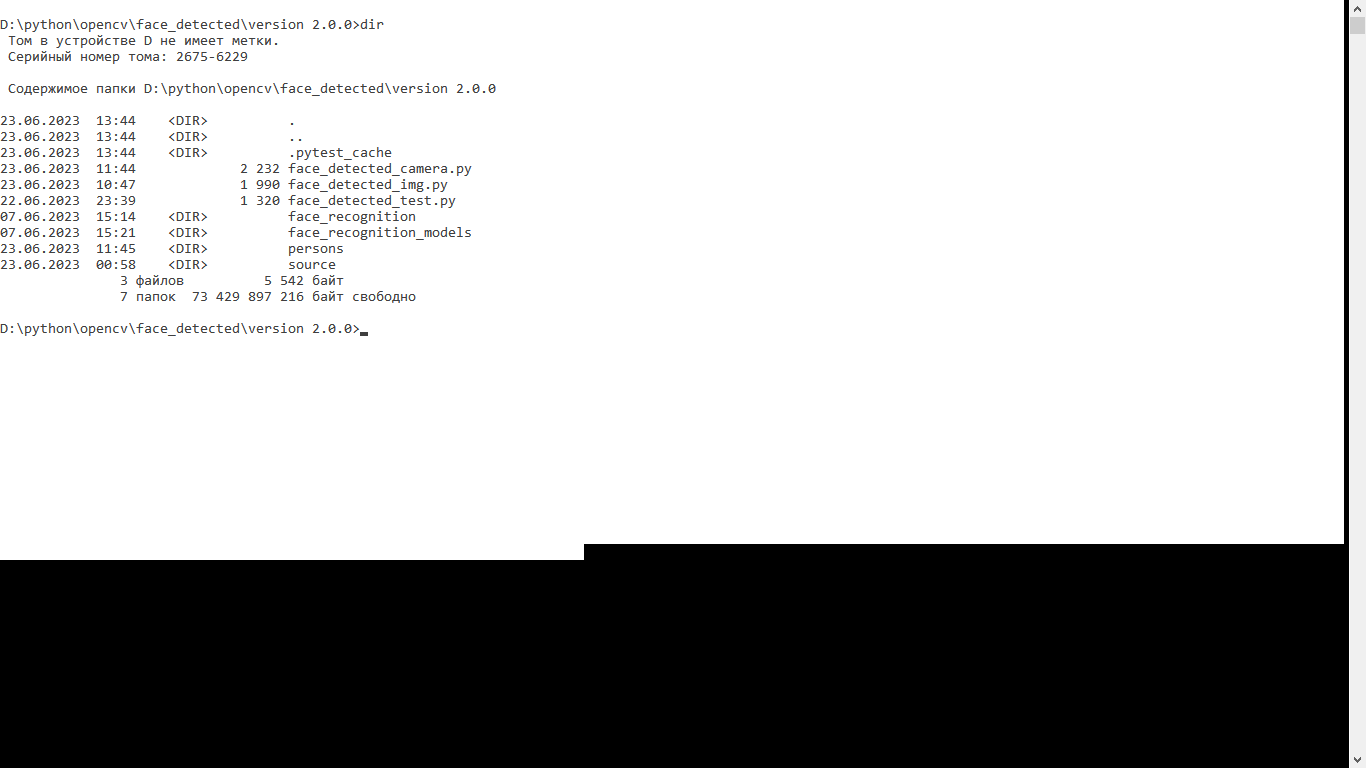
Yuqoridagi fayllarning barchasi version 1.0.0 ishlatilgan kodlardir.Bu kodlar tasvirdan , video tasvirdan va qurilma kamerasidan foydalangan holada inson yuzini topa oladi.

Version 2.0.0

Version 2.0.0 bu tasvirlardan, kameradan inson yuzini topish uni kimligini aniqlash vazifasini hal qilish muammosi asosida yuzaga keldi, va ushbu dastur yuzni tanib olish texnologiyasiga asoslangan holda ishlab chiqildi.Dastur oldidagi vazifalar tasvirda inson yuzi bormi, agar bor bo’lsa ushbu yuz kimniki ekanligini aniqlash edi va ushbu oldiga qo’ygan vazifani muoffaqiyatli amalga oshdi.



8-rasm.version 2.0.0 strukturasi



9-rasm.version 2.0.0 fayllar strukrurasi

face\_detected\_camera.py – bu dastur hisoblanib ,ushbu dastur python dasturlash tilida yozilgan. Ushbu modul vazifasi qurilma kamerasi yordamida tasvirlardan inson yuzini aniqlashdir.

face\_detected\_img.py – bu dastur hisoblanib ,ushbu dastur python dasturlash tilida yozilgan. Ushbu modul vazifasi qurilmaga kiritilgan tasvirlardan inson yuzini aniqlashdir.

persons – katalogida shaxs ma’lumotlari joylanadi, ya’ni bu bizga ma’lumotlar bazasi sifatida hizmat qiladi,Ushbu katalogga rasm nomi bilan yozilishi yetarli (Namuna: "Steeve Jobs.png").

source – katalogida esa tekshirilishi kerak bo’lgan tasvirlar joylanadi, bu katalogdagi ma’lumotlarni person katalogidagi ma’lumotlar asosida tekshiradi.

face\_detected\_camera.py faylidagi yozilgan dastur kodi

print('dastur ishga tushdi')

import os

import cv2

import face\_recognition

import numpy as np

BLUE = (255,0,0)

GREEN = (0,255,0)

RED = (0,0,255)

WHITE = (255,255,255)

path = 'persons'

images = []

classNames = []

myList = os.listdir(path)

#print(myList)

for cl in myList:

tempImage = cv2.imread(f'{path}/{cl}')

images.append(tempImage)

classNames.append(os.path.splitext(cl)[0])

print(classNames)

#print(images)

def findEncodings(images):

encodingList = []

for img in images:

img = cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)

encod = face\_recognition.face\_encodings(img)[0]

encodingList.append(encod)

return encodingList

encodeListKnow = findEncodings(images)

print('Encoding complete '+str(len(encodeListKnow)))

cap = cv2.VideoCapture(0)

while True:

succ,frame = cap.read()

#img = cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)

if succ:

faceCurrentFrame = face\_recognition.face\_locations(frame)

encodeCurrenrFrame = face\_recognition.face\_encodings(frame,faceCurrentFrame)

#faceCurrentFrame = face\_recognition.face\_locations(img)

#encodeCurrenrFrame = face\_recognition.face\_encodings(img,faceCurrentFrame)

for encodeFace,faceLoc in zip(encodeCurrenrFrame,faceCurrentFrame):

matches = face\_recognition.compare\_faces(encodeListKnow,encodeFace)

faceDis = face\_recognition.face\_distance(encodeListKnow,encodeFace)

#print(faceDis)

matchIndex = np.argmin(faceDis)

name = 'Not Found'

color = RED

if matches[matchIndex]:

name = classNames[matchIndex].title()

#print(name,matches[matchIndex])

color = GREEN

(y1,x1,y2,x2) = faceLoc

cv2.rectangle(frame,(x1,y1),(x2,y2),color,2)

cv2.putText(frame,name,(x2+20,y2+6),cv2.FONT\_HERSHEY\_COMPLEX,1,WHITE,2)

cv2.imshow('window',frame)

if cv2.waitKey(1) & 0xff == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

face\_detected\_img.py faylidagi dastur kodlari

print('dastur ishga tushdi')

import os

import cv2

import face\_recognition

import numpy as np

BLUE = (255,0,0)

GREEN = (0,255,0)

RED = (0,0,255)

WHITE = (255,255,255)

path = 'persons'

images = []

classNames = []

myList = os.listdir(path)

#print(myList)

for cl in myList:

tempImage = cv2.imread(f'{path}/{cl}')

images.append(tempImage)

classNames.append(os.path.splitext(cl)[0])

print(classNames)

#print(images)

def findEncodings(images):

encodingList = []

for img in images:

img = cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)

encod = face\_recognition.face\_encodings(img)[0]

encodingList.append(encod)

return encodingList

encodeListKnow = findEncodings(images)

print('Encoding complete '+str(len(encodeListKnow)))

for i in range(len(os.listdir('source'))):

known\_image = face\_recognition.load\_image\_file(f"source/{i}.jpg")

faceCurrentFrame = face\_recognition.face\_locations(known\_image)

encodeCurrenrFrame = face\_recognition.face\_encodings(known\_image,faceCurrentFrame)

#faceCurrentFrame = face\_recognition.face\_locations(img)

#encodeCurrenrFrame = face\_recognition.face\_encodings(img,faceCurrentFrame)

for encodeFace,faceLoc in zip(encodeCurrenrFrame,faceCurrentFrame):

matches = face\_recognition.compare\_faces(encodeListKnow,encodeFace)

faceDis = face\_recognition.face\_distance(encodeListKnow,encodeFace)

print(faceDis)

(y1,x1,y2,x2) = faceLoc

name = 'Not Found'

matchIndex = np.argmin(faceDis)

color = RED

if matches[matchIndex]:

name = classNames[matchIndex].title()

print(name,matches[matchIndex])

color = GREEN

cv2.rectangle(known\_image,(x1,y1),(x2,y2),color,2)

cv2.putText(known\_image,name,(x2+6,y2+25),cv2.FONT\_HERSHEY\_COMPLEX,0.650,WHITE,2)

cv2.imshow('window',known\_image)

cv2.waitKey(0)

Yuzni tanib olish dasturi biometrik dasturiy ta'minotning bir turi bo'lib, u tasvirlar yoki videolardagi yuzlarni mavjud identifikatsiya ma'lumotlar bazasiga moslashtirish uchun sun'iy intellekt va mashina o'rganishdan foydalanadi. Ushbu texnologiya turli xil ilovalar, jumladan, xavfsizlik, huquqni muhofaza qilish va mijozlarga xizmat ko'rsatishda qo'llaniladi.

Yuzni tanib olish uchun ishlatiladigan mashinani o'rganish algoritmlarining eng keng tarqalgan turlaridan biri bu chuqur o'rganish konvolyutsion neyron tarmog'i (CNN). CNN tasvirlarni tasniflash vazifalari uchun juda mos keladi, chunki ular tasvirlardan xususiyatlarni ajratib olishni o'rganishlari va tasvirlarni turli toifalarga ajratish uchun ushbu xususiyatlardan foydalanishlari mumkin. CNN ning chuqurligi yuzni tanib olish uchun muhim, chunki u CNN ga yanada murakkab yuz xususiyatlarini o'rganish imkonini beradi.

Eng mashhur yuzni tanish dasturiy platformalaridan ba'zilari Amazon Rekognition, Deep Vision AI, Face++ va FaceFirstni o'z ichiga oladi. Bu platformalar yuzni aniqlash, yuzni moslashtirish va yuzni tekshirish kabi turli funksiyalarni taklif etadi. Ulardan suratlar yoki videolardagi odamlarni aniqlash, odamlarning shaxsini tekshirish va odamlarning harakatlarini kuzatish uchun foydalanish mumkin.

Yuzni aniqlash dasturi turli maqsadlarda foydalanish mumkin bo'lgan kuchli vositadir. Biroq, ushbu texnologiya bilan bog'liq bo'lishi mumkin bo'lgan xavf-xatarlardan xabardor bo'lish muhim, masalan, maxfiylik muammolari va tarafkashlik.

Yuzni tanish AI dasturidan foydalanishning ba'zi afzalliklari:

* Haqiqiy vaqtda identifikatsiya
* Spoofingga qarshi choralar
* Irqiy yoki gender tarafkashligining kamayishi
* Bir nechta kameralarda foydalanish mumkin

Yuzni tanish AI dasturidan foydalanishning ba'zi kamchiliklari:

* Maxfiylik muammolari
* Tarafsizlik
* Noto'g'ri foydalanish ehtimoli

Umuman olganda, yuzni tanish AI dasturi turli maqsadlarda ishlatilishi mumkin bo'lgan kuchli vositadir. Biroq, uni ishlatishdan oldin ushbu texnologiya bilan bog'liq bo'lishi mumkin bo'lgan xavflardan xabardor bo'lish muhimdir.