

Міністерство освіти і науки України
Департамент науки і освіти Харківської облдержадміністрації
Харківське територіальне відділення МАН України

Секція: Інформаційні системи, бази даних та системи штучного інтелекту

Голосовой ассистент

Роботу виконав:

Шандибін Анатолій Богданович,
учень 9-А класу Харківського Навчально Виховного
Комплексу № 45
Харківської міської ради Харківської області

Науковий керівник:

Руккас Кирилл Маркович,
професор Харківського національного університету
імені В.Н. Каразіна.
м. Харків

Харків 2021

Анотація

Вступ

Сучасні темпи розвитку інформаційних технологій супроводжуються появою великої кількості нових додатків та можливостей, також потребує від людини оброблення безлічі інформації, іноді не дуже корисної. Часом зорієнтуватися в цьому просторі дуже складно, особливо людям похилого віку та з особливими можливостями. Метою моєї роботи є створення якомога функціонального та швидкого асистента, який максимально простий для виконання найпоширеніших запитів найбільш пристосованих до сучасних технологій та роботи зі звичайним програмним забезпеченням верств населення, зробити життя цих людей більш сучасним, пристосованим до суспільства та комфортним.

Результати

В результаті проведеної роботи був створений голосовий асистент для допомоги не пристосованим верствам населення з кожно-денними запитами, пов'язаними з користуванням комп'ютером та пошуком інформації. Була реалізована достатньо проста в користуванні програма із захистом даних за допомогою розпізнавання обличчя

Зміст

Вступ.....	4
1. Мета та	
актуальність.....	5
2. Етапи дослідження	
.....	7
2.1 Чому пайтон.....	7
2.2 Алгоритми розпізнавання	
голосу.....	9
2.3 Функціонал програми.....	10
3. Приклади роботи асистенту.....	11
Висновок.....	13
Джерела.....	14

ВСТУП

Сучасний етап розвитку інформаційних технологій намагається як можна більше полегшити життя людям. Вигадана велику кількість додатків, які допомагають та полегшують життя людини.

Програми постійно вдосконалюються та надають нові можливості. Разом з тим для їх освоєння потрібно більше навичок та часу. Але всі ці зміни/новації не кожний може освоїти. Найбільш складно це для людей похилого віку чи з обмеженими можливостями.

Щоб спростити доступ, використання різних програм та пошук інформації - вигадали голосовий асистент. Голосовий асистент - це програмний агент, який може виконувати завдання (або сервіси) для користувача на основі інформації, введеної користувачем, даних про його місцезнаходження, а також інформації, отриманої з різних інтернет-ресурсів (погода, час, значення слова, відправка e-mail, курси валют).

РОЗДІЛ 1

Мета та актуальність

Мета моєї роботи - зробити якомога функціональний та швидкий асистент, який максимально простий для виконання найпоширеніших запитів найбільш непристосованих до сучасних технологій та роботи зі звичайним програмним забезпеченням верств населення.

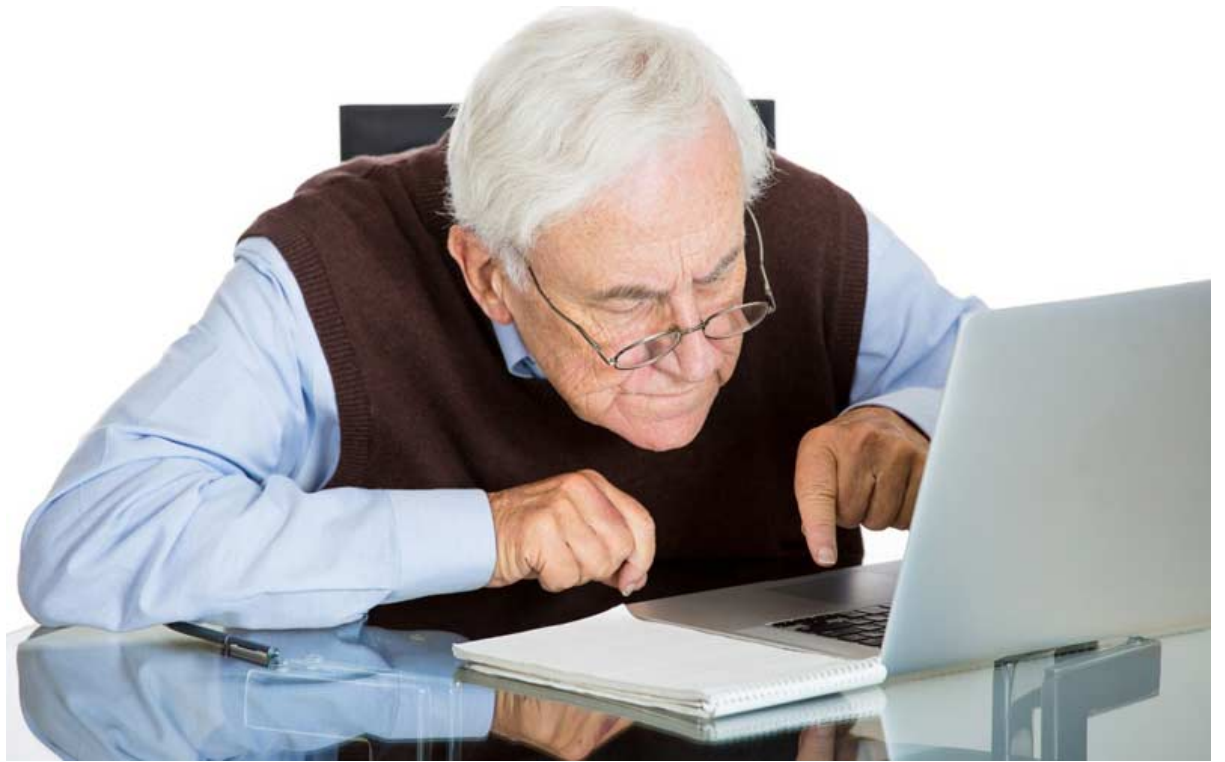
Кому він може допомогти?

Людям похилого віку, яким складно освоїти безліч програм, людям з обмеженими можливостями (наприклад, проблеми зору, координації руху) для покриття своїх потреб. Асистент також направлений для швидкої реалізації задачі за допомогою голосового керування в ситуаціях, коли мануально керувати пристроєм немає можливості (знаходишся у складному становищі, брудні руки - а треба зв'язатися з кимось або перевірити інформацію, наприклад рецепт).

Як я вивчав проблему?

Зараз я допомагаю у технічних питаннях двом моїм бабусям та похилій сусідці. І помітив, що в них схожі питання, вони виникають в різний час доби, а спроби користування звичайними додатками потребують у них часу, терпіння, окулярів та мати записану інструкцію перед очима. З подібними запитами також стикаються і мої товариші. Я зайнявся вивченням проблеми. Найбільш поширене рішення - це різноманітні освітні програми, такі як "Освіта для всіх", "Базові цифрові навички", "Смартфон для батьків" і тому подібне. Навчання - це добре, але є багато випадків, коли наявність простого способу отримання інформації чи спілкування може значно покращити життя. Також важливо брати до уваги психологію покоління. Для більш старших людей дуже важливим є стабільність/незмінність звичних речей, простота в користуванні і безпека. Існуючі голосові помічники, наприклад Сірі, направлені на більш молоде покоління, при використанні не прості і не завжди видають передбачувані відповіді. Велика кількість додаткових опцій і можливостей часто лякає пенсіонерів і заводять в тупик. Amazon планує запуснути в кінці 2021 платний сервіс на базі голосового помічника Alexa для похилих людей - Alexa Together, головним призначенням якого буде піклування близьких за родичами, що старіють. Але багато пенсіонерів ще хочуть самостійно контролювати своє життя та не потребують багатьох із запропонованих Alexa функцій.

Далі, базуючись на спілкуванні та найчастіших питаннях/проханнях допомогти чи пояснити, я склав карту інтересів людей похилого віку, яка полягла за основу написання асистенту. Це такі найчастіші питання про погоду, час, значення слова, відправка e-mail. Додав можливість голосового вводу питань і захист інформації за допомогою розпізнавання обличчя користувача.



РОЗДІЛ 2

Етапи дослідження

2.1 Чому пайтон

Чому вибрана мова Python для розробки асистенту:

Python дуже простий але універсальний завдяки багатому наборові бібліотек (набори функцій, які доступні зі швидким завантаженням), тому його застосовують в самих різних областях:

- веб-розробка;

- machine Learning і AI (TensorFlow, scikit-learn, NLTK);

- Big Data;

- веб-скануванні(Google);

- розробці ігор;

- виробництві фільмів(Pixar);

- комп'ютерна безпека;

- шпигунстві(ЦРУ);

- наукові дослідження (Математика (NumPy, SciPy, Pandas, SymPy), біологія – секвенування ДНК);

- обчислювальна хімія, біоінформатика, генетика, обчислювальна математика і статистика, астрономія, економіка.

Найбільші ІТ-гіганти послугуються саме цією мовою як основною в компанії. До їхнього списку входять Facebook, Instagram, NASA, IBM, Pinterest

На Python є відкрита бібліотека машинного навчання на основі бібліотеки Torch, що використовують для таких застосувань, як комп'ютерне бачення та обробка природної мови, та спеціальні бібліотеки для будь-яких цілей, наприклад:

Для багатовимірних масивів та високорівневих матриць використовуйте NumPy.

- Для розрахунків у інженерній справі - SciPy.

- Для дослідження, аналізу та маніпулювання даними - Pandas.

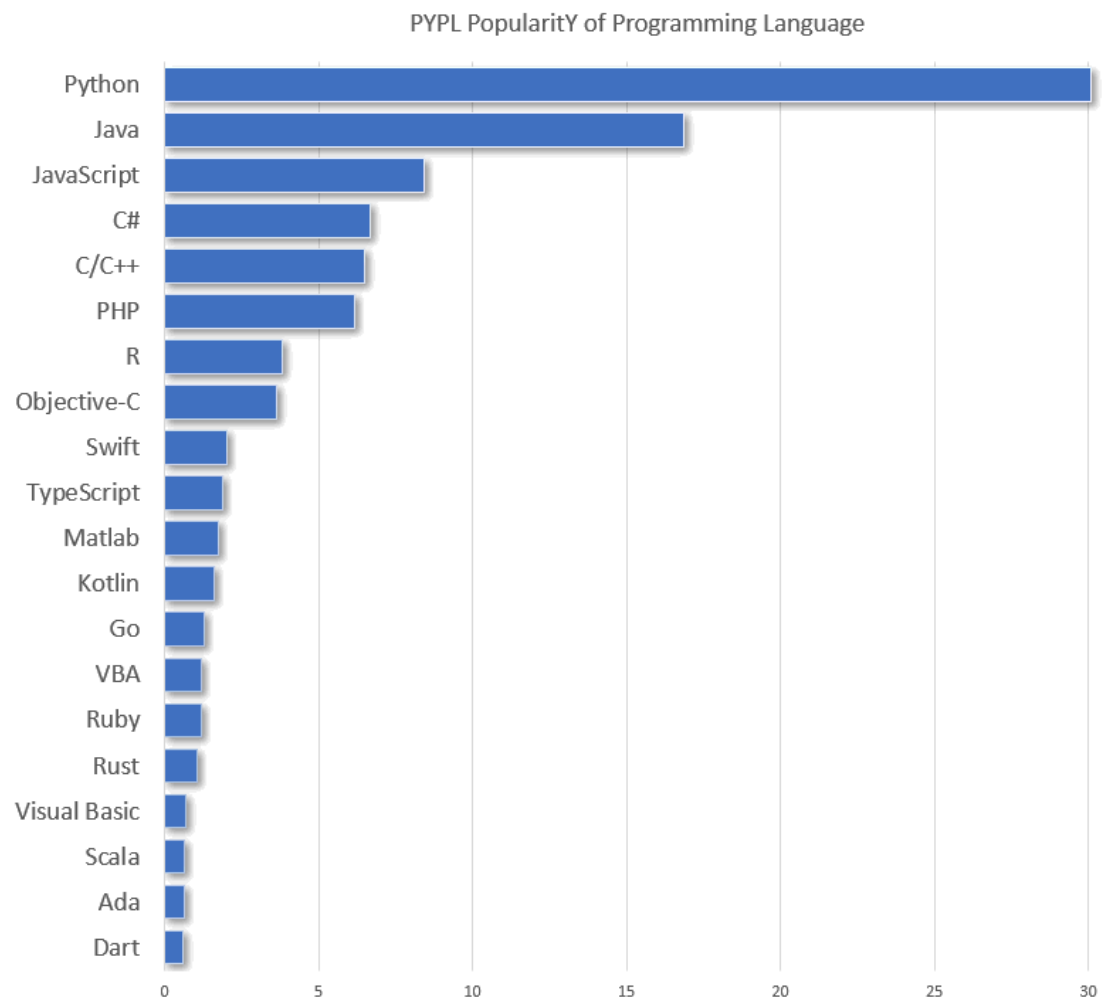
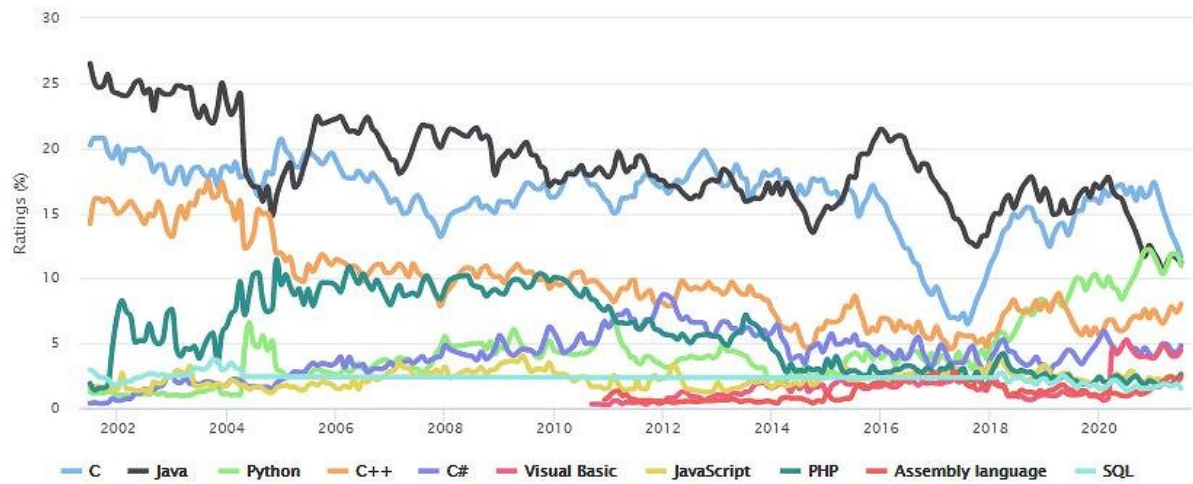
- Для роботи зі штучним інтелектом - Scikit-Learn.

- Для розробки мобільних додатків - пакет під назвою Kivy.

- Для обробки усної та письмової мови - PyText.

Вони всі були написані людьми, які програмували на ньому багато років і розуміли - як буде легше і що найпотрібніше. Завдяки простоті та великому суспільству розробників кожен може швидко знайти відповіді на велику кількість питань.

Нижче наведена статистика зростання популярності цієї мови, яка в 2021 році посідає 1 місце в рейтингах. Python вчетверте поспіль заслужив титул "Мова року" ("TIOBE's Programming Language of 2020") та показав найбільший приріст популярності за один рік за інформацією TIOBE.



2.2 Алгоритми розпізнавання голосу

Одним із головних функціоналів асистенту є розпізнавання мовлення - процес перетворення мовного сигналу в цифрову інформацію.

Завданням розпізнавання є зіставлення набору акустичних ознак мовного сигналу чи спостережень $X(x_1, \dots, x_n)$ послідовності слів $W(w_1, \dots, w_k)$, мають найбільшу ймовірність правдоподібності серед усіх кандидатів. Для цього використовується

формула Байєса:
$$W = \operatorname{argmax} \left[\frac{P(W) * P(X|W)}{P(X)} \right]$$
 Причому, у процесі розпізнавання ймовірність вже одержаних ознак $P(X)$ не підлягає оптимізації і знаменник у формулі не використовується.

$$W = \operatorname{argmax}[P(W) * P(X|W)]$$

Систем розпізнавання мови багато, вони класифікуються за різними ознаками: залежність від диктора, за розміром словника, за призначенням (системи диктування, командні системи), за алгоритмом (нейронні мережі, приховані Марківські моделі, динамічне програмування), за типом і принципом виділення структурної одиниці.

Методи розпізнавання мови з урахуванням порівняння з еталоном діляться на динамічне програмування (Dynamic Time Warping), контекстно залежну класифікацію(виділення окремих лексичних елементів - фонем та алофонів, які потім об'єднуються в склади та морфеми), дискримінантний аналіз (Bayesian discrimination), приховані Марківські моделі (Hidden Markov Model) та нейронні мережі (Neural networks).

2.3 Функціонал програми

Опишемо головні функції та їх методи реалізації.

Реалізація команд у мене поділялася на три типи:

- Робота з ноутбуком(включення, вимкнення, т.д)
- Запити у пошукову систему google через програму
- За допомогою open source бібліотек

Перший метод полягав у налаштуванні прав доступу до управління системи. Після чого можна було за допомогою додаткових бібліотек(os,sys,subprocess) робити різні запити на керування ноутбуком.

Другий метод це коли я за допомогою сторонніх бібліотек(requests,google api client) моєї мови програмування роблю запити у пошукову систему google і отримую від неї все те, що користувач бачить, коли робить пошук. Далі обробляю ці дані за допомогою бібліотек(bs4,lxml) і отримую те, що можу перетворити на відповідь програми.

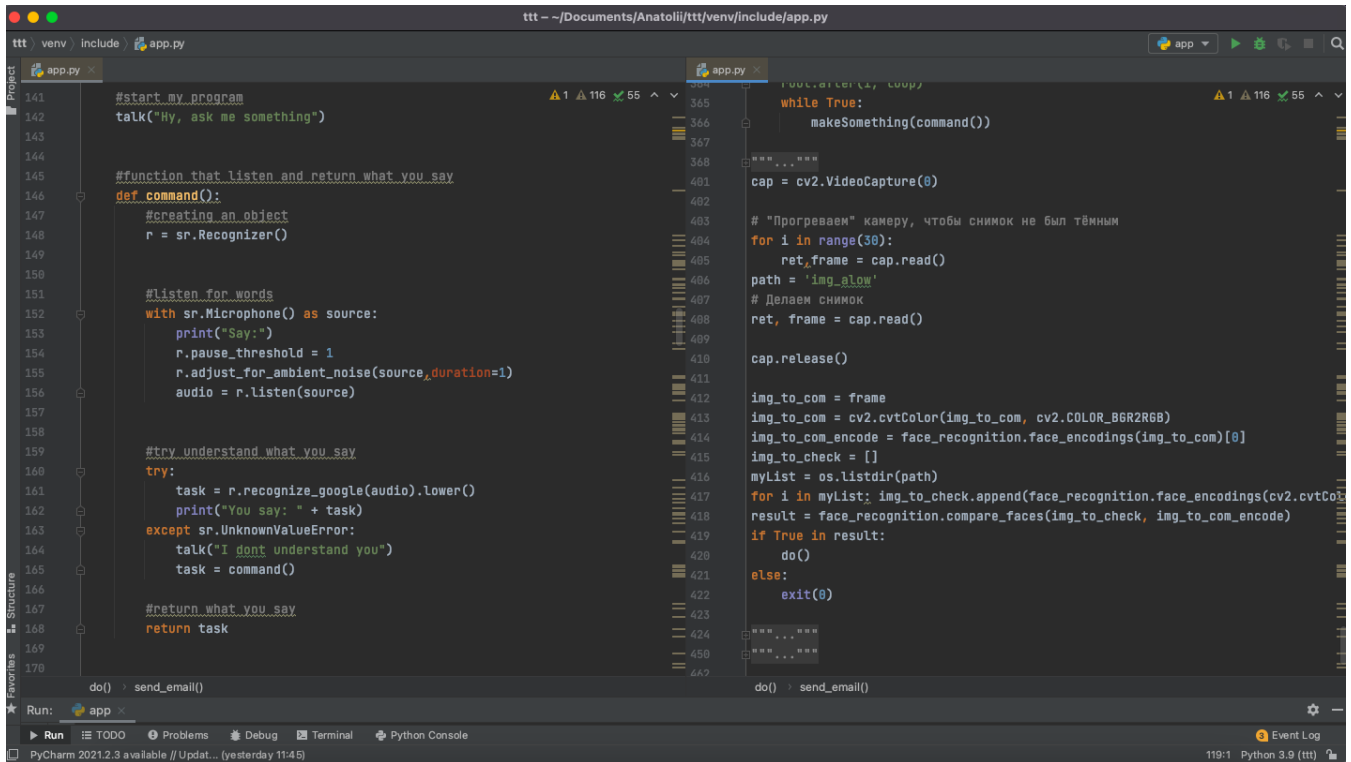
Третій метод це коли я шукаю бібліотеку, яка мені може допомогти вирішити моє завдання. Насправді майже всі завдання, які у тебе є вже хтось вирішив і оформив це як бібліотеку. Що таке бібліотека? Бібліотека - це збірка підпрограм або об'єктів, що використовуються для розробки програмного забезпечення (ПЗ). У деяких мовах програмування (наприклад, у Python) те саме, що й модуль, у деяких — кілька модулів.

Наприклад, за допомогою бібліотеки(opencv) я можу перетворювати зображення у цифрові данні, а потім з бібліотекою(face_recognition) виділяти на ній обличчя та порівнювати його з іншими обличчями. За допомогою використання декількох бібліотек для однієї задачі я роблю рішення краще ніж існуючі.

РОЗДІЛ 3

Приклади роботи асистенту

1) Розпізнавання голосу та face id



```
#start my program
talk("Hy, ask me something")

#function that listen and return what you say
def command():
    #creating an object
    r = sr.Recognizer()

    #listen for words
    with sr.Microphone() as source:
        print("Say:")
        r.pause_threshold = 1
        r.adjust_for_ambient_noise(source,duration=1)
        audio = r.listen(source)

    #try understand what you say
    try:
        task = r.recognize_google(audio).lower()
        print("You say: " + task)
    except sr.UnknownValueError:
        talk("I dont understand you")
        task = command()

    #return what you say
    return task

do() -> send_email()

while True:
    makeSomething(command())

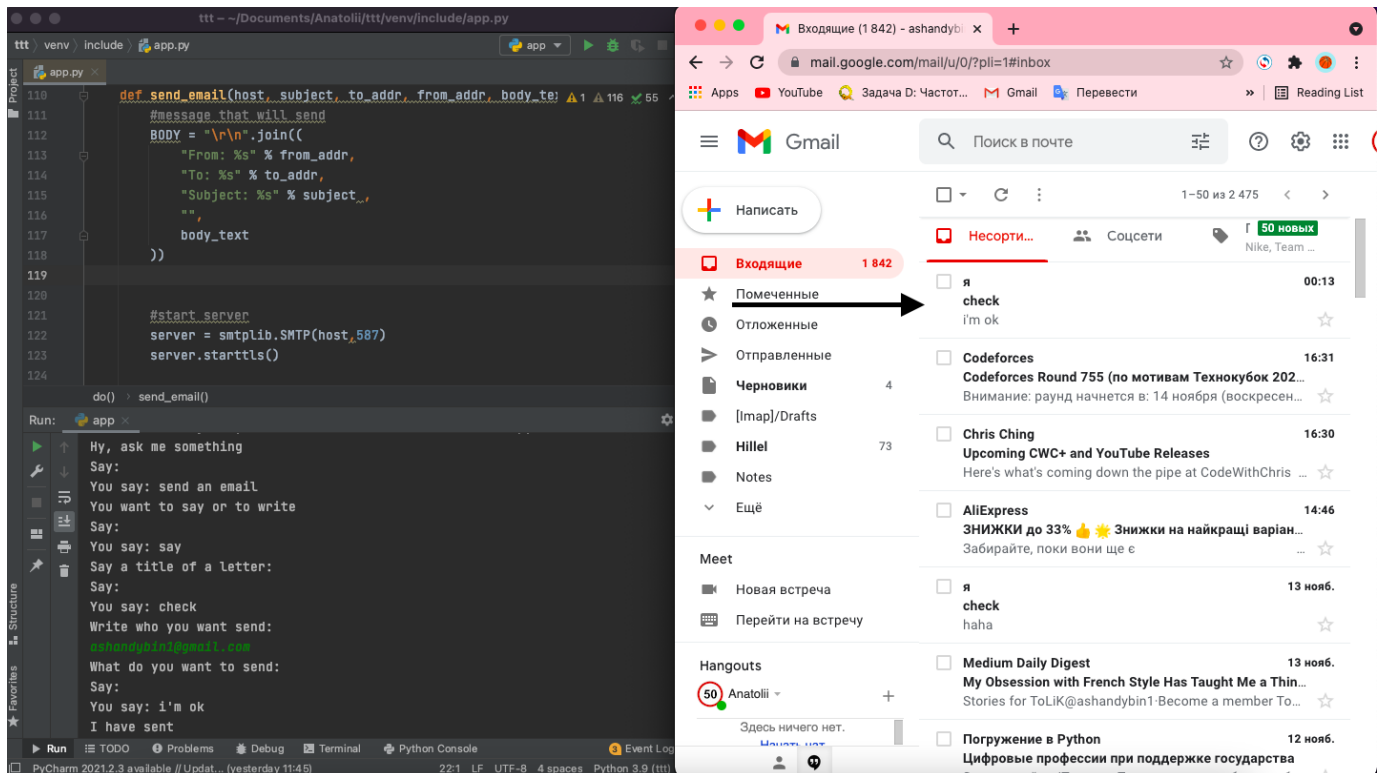
cap = cv2.VideoCapture(0)

# "Прогреваем" камеру, чтобы снимок не был тёмным
for i in range(30):
    ret, frame = cap.read()
    path = 'img_low'
    # Делаем снимок
    ret, frame = cap.read()

cap.release()

img_to_com = frame
img_to_com = cv2.cvtColor(img_to_com, cv2.COLOR_BGR2RGB)
img_to_com_encode = face_recognition.face_encodings(img_to_com)[0]
img_to_check = []
myList = os.listdir(path)
for i in myList: img_to_check.append(face_recognition.face_encodings(cv2.cvtColor(img_to_check[i], cv2.COLOR_BGR2RGB)))
result = face_recognition.compare_faces(img_to_check, img_to_com_encode)
if True in result:
    do()
else:
    exit(0)
```

2) Відправка e-mail



```
def send_email(host, subject, to_addr, from_addr, body_text):
    #message that will send
    BODY = "\r\n".join((
        "From: %s" % from_addr,
        "To: %s" % to_addr,
        "Subject: %s" % subject,
        ""
    ))
    body_text

    #start server
    server = smtplib.SMTP(host,587)
    server.starttls()

do() -> send_email()
```

Входящие (1842) - ashandybi

mail.google.com/mail/u/0/?pli=1#inbox

Написать

Несорти...

Входящие 1842

Помеченные

Отложенные

Отправленные

Черновики 4

[imap]/Drafts

Hillel 73

Notes

Ещё

Meet

Новая встреча

Перейти на встречу

Hangouts

50 Anatolii +

Здесь ничего нет.

50 новых

я check i'm ok 00:13

Codeforces Codeforces Round 755 (по мотивам Технокубок 202... 16:31

Chris Ching Upcoming CWC+ and YouTube Releases Here's what's coming down the pipe at CodeWithChris ... 16:30

AliExpress ЗНИЖКИ до 33% 🙌🔥 Знижки на найкращі варіан... 14:46

я check haha 13 нояб.

Medium Daily Digest My Obsession with French Style Has Taught Me a Thin... 13 нояб.

Погружение в Python Цифровые профессии при поддержке государства 12 нояб.

3) Погода у будь-якому місті

The screenshot shows a Python application running in PyCharm on the left and a Google search result for "weather in tokyo" on the right.

Python Application (Left):

```
221 #if you want get weather
222 elif 'weather' in task:
223     #make a request
224     html = requests.get(f"https://www.google.com/search?q={task}", headers=
225
226
227     #creating a soup object
228     soup = BeautifulSoup(html, 'lxml')
229
230     #print weather with parsing a first website
231     print(f"soup.find('span', class='wob_t q8U8x').text) degrees\n"
232           f"Probability of precipitation:{soup.find('span', id='wob_pp').tex
233           f"Humidity:{soup.find('span', id='wob_hm').text}\n"
234           f"Winter:{soup.find('span', id='wob_ws').text}\n")
235     talk(f"soup.find('span', class='wob_t q8U8x').text) degrees, Probabili
236
237 do()
238 send_email()
```

Google Search Result (Right):

weather in tokyo

Результатов: примерно 92 400 000 (0,51 сек.)

10 °C | °F

Токио, Япония
воскресенье 07:00
Облачно

Вероятность осадков: 40%
Влажность: 70%
Ветер: 3 км/ч

Температура | Вероятность осадков | Ветер

10 14 16 16 13 11 11 10

08:00 11:00 14:00 17:00 20:00 23:00 02:00 05:00

ВС ПН ВТ СР ЧТ ПТ СБ ВС

17° 9° 20° 9° 18° 11° 18° 9° 18° 10° 17° 9° 17° 9° 17° 8°

4) Значення слова

The screenshot shows a Python application running in PyCharm on the left and a Wikipedia page for the word "school" on the right.

Python Application (Left):

```
338 search = command()
339
340 def google_search(search_term, api_key, cse_id, **kwargs):
341     service = build("customsearch", "v1", developerKey=api_key)
342     res = service.cse().list(q=search_term, cx=cse_id, **kwargs).execute()
343     if "items" in res.keys():
344         return res["items"]
345     else:
346         return None
347
348 results = google_search(
349     search, my_api_key, my_cse_id, num=1)
350 arr = list(BeautifulSoup(str(results[0]['htmlSnippet']), 'lxml').text)
351 arr = arr[0:-4]
352 if "." in arr:
353     for i in arr[::-1]:
354         if i == ".":
355             break
356         else:
357             arr = arr[0:-1]
358     talk("".join(arr))
359 #the program not stop
360 do()
361 makeSomething() elif 'time' in task
```

Wikipedia Page (Right):

School - Wikipedia

en.wikipedia.org/wiki/School

A school is an educational institution designed to provide learning spaces and learning environments for the teaching of students under the direction of teachers. Most countries have systems of formal education, which is sometimes compulsory.^[2] In these systems, students progress through a series of schools. The names for these schools vary by country (discussed in the Regional section below) but generally include primary school for young

Larkmead School in Abingdon, Oxfordshire, England

Kallavesi High School, a secondary school building in Kuopio, Finland

First primary school building in Badagry, Nigeria, built in 1845^[1]

Та інші...

ВИСНОВОК

В результаті проведеної роботи був створений голосовий асистент для допомоги не пристосованим верствам населення з кожно-денними запитами, пов'язаними з користуванням комп'ютером та пошуком інформації. Була реалізована достатньо проста в користуванні програма із захистом даних за допомогою розпізнавання обличчя. Було проаналізовано багато даних та зроблено функції якими буде зручно користуватися. Я порівняв свою роботу з іншими аналогами. Та протестував її на своїй бабусі.

ДЖЕРЕЛА

https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D0%B5%D1%87%D0%B8

<https://stackoverflow.com/>

<https://overcoder.net/>

<https://www.python.org/>

<https://github.com/>